

Міністерство освіти і науки України
Міжнародний гуманітарний університет
Кафедра комп'ютерних наук

Розенвассер Д.М.

ТЕОРІЯ ІНФОРМАЦІЇ ТА КОДУВАННЯ

Конспект лекцій

Одеса 2025

Затверджено Вченою Радою Міжнародного гуманітарного університету.
Протокол № 5 від 20 січня 2025 р.

Розенвассер Д.М. Теорія інформації та кодування. Конспект лекцій [Електронне видання]. Кафедра комп'ютерних наук Міжнародного гуманітарного університету. Одеса, 2025. 61 с.

Навчальна дисципліна «Теорія інформації та кодування» знайомить здобувачів із принципами кількісного опису інформації, методами ефективного та завадостійкого кодування, оцінки пропускну здатності каналів зв'язку, а також сучасними алгоритмами стиснення даних і корекції помилок у цифрових системах передавання та зберігання інформації.

ЗМІСТ

Тема 1. Вступ – Сучасні системи збереження та передавання інформації	4
Тема 2. Інформаційні характеристики джерел дискретних повідомлень	8
Тема 3. Методи ефективного кодування джерел дискретних повідомлень	10
Тема 4. Інформаційні характеристики джерел неперервних повідомлень та методи кодування джерел	22
Тема 5. Інформаційні характеристики каналів зв'язку	32
Тема 6. Передавання інформації каналами зв'язку з завадами	35
Тема 7. Блокові коректувальні коди	38
Тема 8. Згорткові коректувальні коди	45
Тема 9. Ефективність систем передавання	51

Тема 1. Вступ – Сучасні системи збереження та передавання інформації

Інформація – це відомості (знання) про навколишній світ, які зменшують невизначеність у свідомості отримувача. У теорії інформації (за К. Шенноном) кількість інформації вимірюється в бітах і залежить від імовірності події: чим рідкісніша подія — тим більше інформації несе повідомлення про неї.

Повідомлення – це форма подання інформації, впорядкована послідовність символів (знаків), призначена для передавання або зберігання. Повідомлення є носієм інформації.

Сигнал – фізичний процес (електричний, електромагнітний, акустичний, оптичний тощо), який використовується для перенесення повідомлення в просторі або в часі.

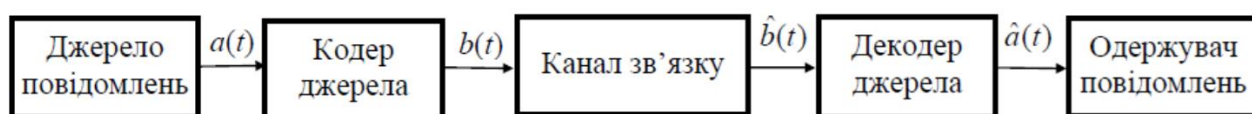


Рисунок 1 – Структурна схема системи передавання

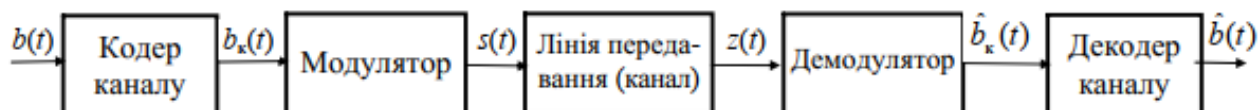


Рисунок 2 – Структурна схема каналу зв'язку

Повідомлення від джерела (текст, мова, дані тощо) спочатку потрапляє до кодера джерела, який виконує стиснення (зменшення надлишковості) і перетворює його в компактну послідовність $b(t)$. Ця послідовність передається через канал зв'язку, де на неї діють завади (шум, спотворення, перешкоди). На приймальному кінці з зашумленої версії $\hat{b}(t)$ декодер джерела відновлює максимально близьке до оригіналу повідомлення, яке надходить отримувачу.

Спочатку кодер джерела видає цифрову послідовність $b(t)$. Далі **кодер каналу** додає спеціальну надлишковість (завадостійке кодування), щоб можна було виявляти та виправляти помилки – виходить послідовність $b_k(t)$. **Модулятор** перетворює її в фізичний сигнал $s(t)$, придатний для передачі (наприклад, радіохвилі, оптичний імпульс, електричний струм).

Сигнал проходить через реальний канал і приходиться як спотворений $z(t)$. **Демодулятор** витягує з нього цифрову послідовність $\hat{b}_k(t)$, а **декодер каналу** виправляє помилки, спричинені завадами, і видає відновлену послідовність $\hat{b}(t)$, яку потім обробляє декодер джерела.

Коди – це правила (алгоритми) перетворення повідомлень або сигналів з однієї форми в іншу для ефективної передачі, зберігання, захисту від помилок чи адаптації до фізичного каналу.

Коди в системах передачі та збереження інформації поділяються на кілька основних груп залежно від їхнього призначення. Перша велика група – це коди джерел, які застосовуються для стиснення даних і зменшення надлишковості повідомлень, щоб ефективніше використовувати пропускну здатність каналу. Серед них виділяються коди аналого-цифрового перетворення, що перетворюють неперервні аналогові сигнали в цифрову форму, наприклад за допомогою імпульсно-кодової модуляції чи дельта-модуляції. Окремо стоять коди джерел дискретних повідомлень, які працюють уже з цифровими даними, такими як текст чи числа, і зазвичай належать до неперервних кодів змінної довжини, де коротші кодові комбінації призначаються більш імовірним символам.

Друга велика група – коректуючі коди, або завадостійкі коди, основне завдання яких полягає у виявленні та виправленні помилок, що виникають через шум і спотворення в каналі зв'язку. Вони бувають блоковими, коли дані кодуються фіксованими блоками, і неперервними, або згортковими, коли кодування відбувається потоком з урахуванням попередніх символів.

Третя група – модуляційні коди, які забезпечують перетворення цифрової послідовності в сигнал, придатний для передачі через фізичний канал. До них належать бінарні коди, що використовують прості рівні сигналу, а також код Грея – рефлексний код, де сусідні значення відрізняються лише одним бітом, що значно зменшує ймовірність помилок при переходах між рівнями.

Четверта група – коди лінійного тракту, або лінійні коди, які адаптують цифровий сигнал до конкретних фізичних характеристик лінії передавання, таких як кабель чи радіоканал. Вони усувають постійну складову, забезпечують синхронізацію тактових імпульсів і покращують спектральні властивості сигналу. Серед них розрізняють однополярні коди, де сигнал має лише одну полярність, і двополярні, що використовують обидві полярності для кращого балансу та синхронізації.

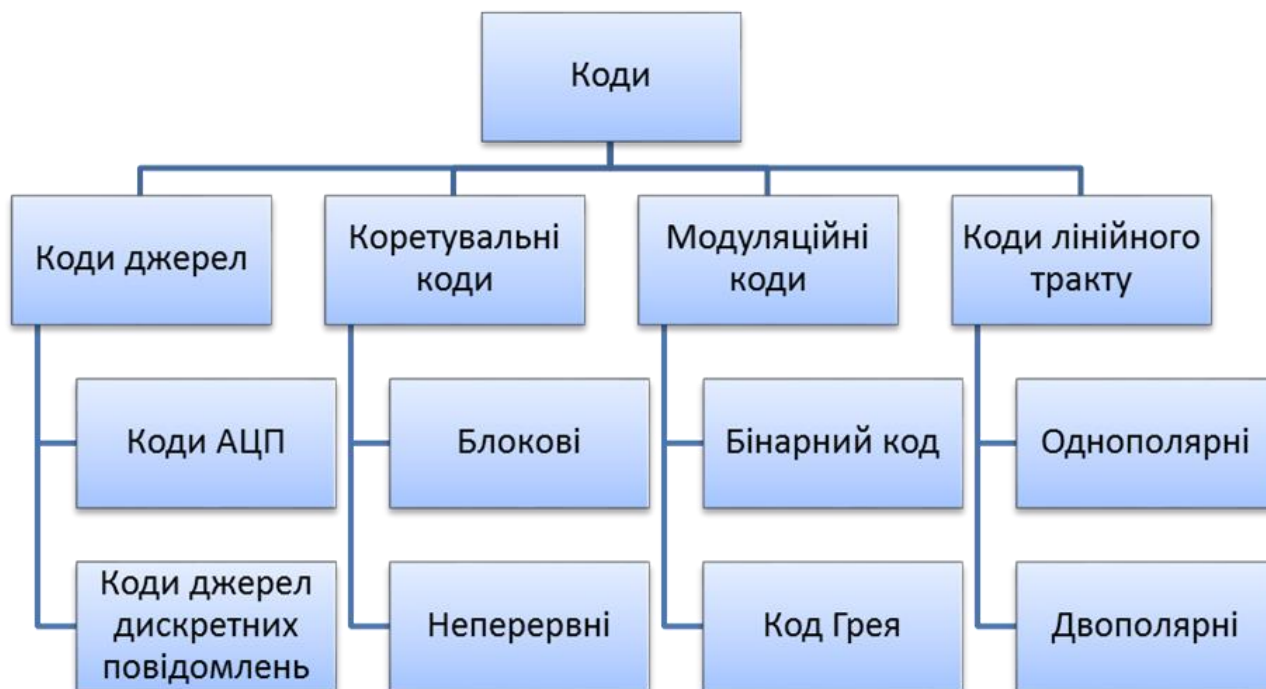


Рисунок 3 – Класифікація кодів

Коди джерел дискретних повідомлень призначені для ефективного представлення дискретних (цифрових) символів або послідовностей з урахуванням їхньої статистики, щоб мінімізувати середню довжину кодового слова без втрати інформації.

Спочатку вони поділяються на дві основні категорії залежно від імовірнісних властивостей символів: рівномірні та нерівномірні. Рівномірні, або примітивні, коди призначають кожному символу однакову кількість бітів незалежно від частоти його появи, тому вони прості, але неефективні для джерел з нерівномірним розподілом імовірностей. До них належать такі широко відомі коди, як МТК-2 (міжнародний телеграфний код, що використовувався в телетайпах), код Морзе (змінної довжини, але історично вважається примітивним через ручне призначення), ASCII (стандартне 7- або 8-бітне фіксоване кодування для латинських символів) та Unicode (сучасне багаторівневе кодування, що підтримує тисячі символів різних мов і зазвичай використовує фіксовану або змінну довжину в UTF-8/16/32).

Нерівномірні коди, навпаки, враховують різні ймовірності символів і призначають коротші кодові комбінації частішим символам, а довші – рідкісним, що дозволяє досягти кращої компресії. Вони поділяються на кілька підгруп. Статистичні коди базуються безпосередньо на імовірностях окремих символів і належать до префіксних кодів, які гарантують однозначне декодування без роздільників. Найвідоміші з них – код Шеннона-Фано, що будується шляхом послідовного поділу множини символів на дві приблизно

рівноймовірні частини зверху вниз, та код Хаффмана, який вважається оптимальним серед префіксних кодів змінної довжини і будується знизу вгору через об'єднання двох найменш імовірних символів у вузли дерева.

Окрему підгрупу становлять словникові коди, які працюють не з окремими символами, а з послідовностями (фразами) і динамічно будують словник повторюваних підрядків під час кодування. До них належать сімейство алгоритмів Лемпеля-Зіва, зокрема LZ77 (що використовує ковзаюче вікно для пошуку попередніх збігів і видає посилання на них плюс новий символ) та LZ78 (який будує словник фраз поступово і видає номери словникових записів).

Нарешті, комбіновані коди поєднують елементи статистичних і словникових методів або застосовують кілька етапів стиснення для ще кращої ефективності, наприклад, у сучасних архіваторах, де LZ77 часто комбінується з Хаффманом (як у deflate, що лежить в основі ZIP, gzip, PNG).

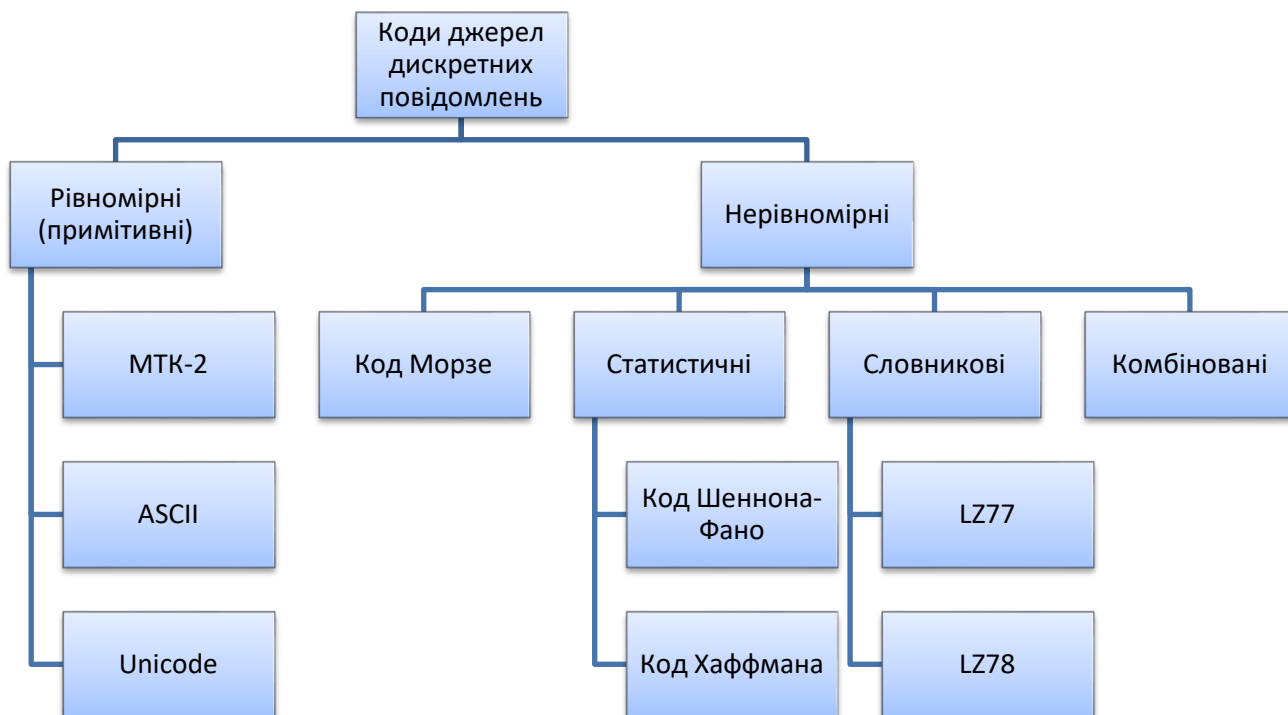


Рисунок 4 – Класифікація кодів джерел дискретних повідомлень

Тема 2. Інформаційні характеристики джерел дискретних повідомлень

В основу визначення кількості інформації в повідомленнях в теорії й техніці зв'язку покладені ймовірнісні характеристики повідомлень, які показують їхній ступінь невизначеності. Кількість інформації $I(a)$ у повідомленні a , ймовірність появи якого $P(a)$, визначається як

$$I(a) = \log_2 \frac{1}{P(a)} = -\log_2 P(a). \quad (1)$$

Логарифмічна міра має властивість адитивності (кількість інформації, що міститься в декількох незалежних повідомленнях, дорівнює сумі кількостей інформації в кожному повідомленні). Оскільки $0 < P(a) \leq 1$, то величина $I(a)$ є невід'ємною й скінченою. Якщо $P(a) = 1$, то кількість інформації дорівнює нулю (повідомлення про відому подію ніякої інформації не несе).

Одиницею виміру кількості інформації є двійкова одиниця (дв. од.) або біт (1 дв. ед. визначається як кількість інформації в повідомленні, ймовірність якого $P(a) = 0,5$).

Джерело повідомлень A використовує M_A знаків (M_A називають обсягом алфавіту). Знаки a_k звичайно мають різні ймовірності $P(a_k)$, і кількості інформації $I(a_k)$ у знаках відповідно до формули (1) різні. Для опису джерела в середньому уведене поняття ентропія. Ентропія джерела $H(A)$ – це середня кількість інформації в одному знаку (вимірюється у дв. од.). Якщо знаки незалежні, то середня кількість інформації визначається як математичне сподівання значень $I(a_k)$:

$$H(A) = \overline{I(a_k)} = \sum_{k=1}^{M_A} P(a_k) I(a_k) = -\sum_{k=1}^{M_A} P(a_k) \log_2 P(a_k), \quad (2)$$

Фізично ентропія є мірою невизначеності стану джерела повідомлень, вона є об'єктивною інформаційною характеристикою джерела.

З виразу (2) випливає, що ентропія невід'ємна. При цьому ентропія дорівнює нулю тільки тоді, коли ймовірність одного із знаків дорівнює 1, а ймовірності інших знаків дорівнює нулю. Ентропія досягає максимального значення у випадку, коли всі знаки на виході джерела повідомлень рівноймовірні й незалежні:

$$H_{\max}(A) = \log_2 M_A. \quad (3)$$

У загальному випадку для ентропії справедливий вираз $H(A) \leq \log_2 M_A$.

Дуже часто $M_A = 2$. Тоді $H_{\max}(A) = 1$ дв. од. Отже, при передаванні інформації двійковими символами, кожний символ не може переносити більше ніж 1 дв. од. інформації.

Іншою важливою характеристикою джерела є надмірність. Це властивість джерела повідомлень видавати інформацію більшою кількістю знаків, ніж можна було б. Надмірність джерела зменшує його ентропію в порівнянні з максимальною ентропією. Основні причини надмірності:

- різні ймовірності окремих знаків;
- є статистична залежність між знаками джерела.

Кількісно надмірність джерела оцінюється коефіцієнтом надмірності

$$K_{\text{над}} = \frac{H_{\text{max}}(A) - H(A)}{H_{\text{max}}(A)} = 1 - \frac{H(A)}{H_{\text{max}}(A)}. \quad (4)$$

Продуктивність джерела R_d , дв. од./с – це кількість інформації, видаваної джерелом у середньому за одиницю часу

$$R_d = \frac{H(A)}{T_{\text{сер}}}, \quad (5)$$

де $T_{\text{сер}}$ – середня тривалість одного знака й визначається як

$$T_{\text{сер}} = \sum_{k=1}^{M_A} P(a_k) T_k, \quad (6)$$

де T_k – тривалість k -го знаку.

Відповідно до теореми кодування Шеннона для каналу без завад дискретне повідомлень із алфавітом $\{a_k\}$ і ймовірностями знаків $\{P(a_k)\}$ може бути закодоване ефективним префіксним кодом таким чином, що

$$H(A) \leq \bar{n} \leq H(A) + 1, \quad (7)$$

де \bar{n} – середня довжина кодових комбінацій, визначена як

$$\bar{n} = \sum_{i=1}^{M_A} n_k P(a_k), \quad (8)$$

де n_k – довжина кодової комбінації, що відповідає знаку a_k .

Теорема кодування Шеннона для каналу без завад указує на можливість створення алгоритмів ефективного кодування дискретних повідомлень, при яких середня довжина кодової комбінації може наближатися до ентропії джерела як завгодно близько, але не може бути меншою ентропії.

Оскільки в більшості випадків знаки мають різні ймовірності, то кодові комбінації повинні мати різні довжини з метою зменшення надмірності повідомлень, що передаються. Таке кодування називається ефективним або стисненням інформації.

Для оцінки ефективності обраного алгоритму кодування вводиться коефіцієнт ефективності кодування

$$\mu = \frac{H(A)}{\bar{n}}. \quad (9)$$

Також можна розрахувати коефіцієнт стиснення повідомлення (в порівнянні з рівномірним кодом)

$$\eta = \frac{n}{\bar{n}}, \quad (10)$$

де $n \geq \log_2 M_A$ – мінімальне ціле число, при якому виконується рівність-нерівність.

Основний принцип ефективного кодування полягає в тому, що більш ймовірним знакам повинні відповідати більш короткі кодові комбінації, а знакам з малою ймовірністю – більш довгі.

Префіксним називається кодування, у результаті якого жодна з коротких кодових комбінацій не є початком більш довгих комбінацій. Тим самим префіксне кодування забезпечує однозначний поділ послідовності символів на кодові комбінації при декодуванні.

Тема 3. Методи ефективного кодування джерел дискретних повідомлень

Для кодування незалежних знаків використовується алгоритм Хаффмана. Для його реалізації необхідна таблиця частот (імовірностей) знаків у повідомленні. На підставі цієї таблиці будується дерево кодування Хаффмана за наступним правилом.

1. Всі знаки розташовуються в порядку убутання їхніх імовірностей зверху вниз (кожному знаку відповідає свій вихідний вузол дерева).

2. Від двох знаків з найменшими ймовірностями виходять вітки, які сходяться у вузол, що відповідає складеному знаку із сумарною ймовірністю поєднаних знаків (далі ці два знаки не розглядаються, а складений знак розглядається нарівні з іншими).

3. Одній вітці, наприклад верхній, привласнюється символ “1”, іншій вітці – “0” (або навпаки).

4. Кроки, починаючи із другого, повторюються доти, поки в списку знаків не залишиться тільки один. Він і буде вважатися коренем дерева.

Для визначення кодової комбінації кожного зі знаків, що входить в алфавіт, необхідно пройти шлях від кореня дерева до вихідного вузла (знаку), накопичуючи символи 0 або 1 при переміщенні по вітках дерева.

Приклад 1. Задано джерело незалежних дискретних повідомлень із обсягом алфавіту $M_A = 6$ з імовірностями знаків: $P(A) = 0,3$; $P(E) = 0,25$; $P(B) = 0,22$; $P(\Gamma) = 0,1$; $P(D) = 0,08$; $P(B) = 0,05$. Побудувати дерево кодування Хаффмана й визначити кодові комбінації знаків повідомлення.

Побудуємо дерево кодування Хаффмана (рис. 1).

На підставі кодового дерева запишемо відповідним знакам кодові комбінації: А – 11; Е – 10; В – 00; Г – 010; Д – 0111; Б – 0110.

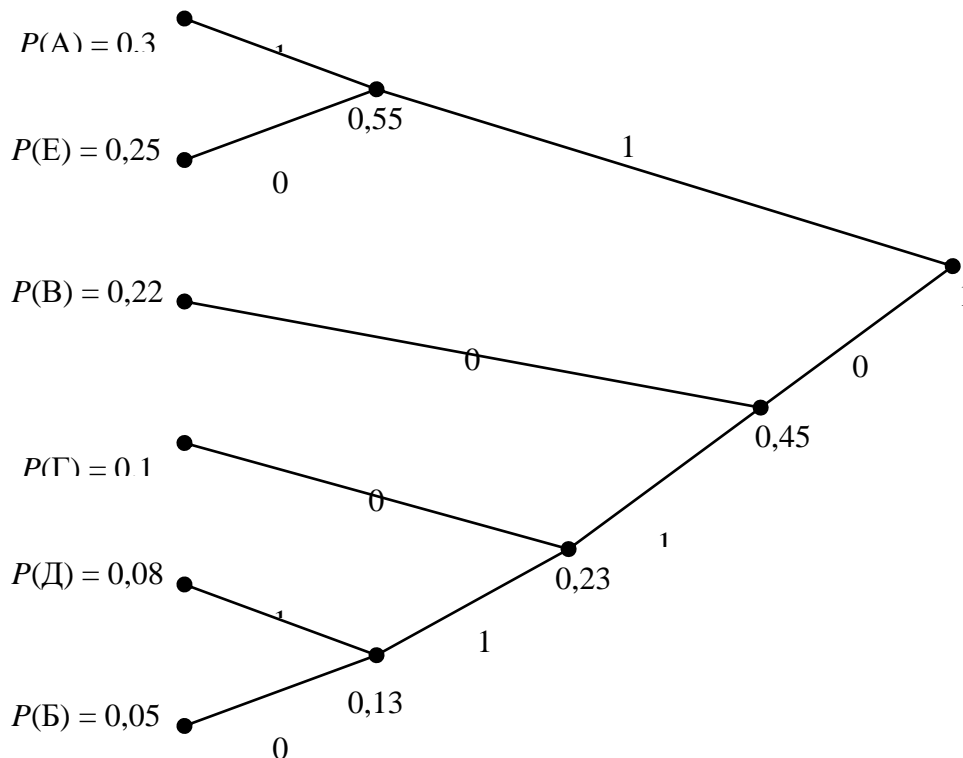


Рисунок 1 – Дерево кодування Хаффмана

Для кодування незалежних знаків використовується також алгоритм Шеннона-Фано. Для його реалізації необхідна таблиця частот (імовірностей) знаків у повідомленні. Алгоритм Шеннона-Фано наступний.

1. Усі знаки розташовуються в порядку убутання їхніх імовірностей зверху вниз.

2. Усі знаки діляться на дві підгрупи, у яких сумарні ймовірності знаків приблизно рівні.

3. Усім знакам верхньої підгрупи привласнюється "0", а знакам нижньої підгрупи – "1".

4. Кроки, починаючи із другого (стосовно до підгруп, що утворилися), повторюються доти, поки в підгрупах залишиться по одному знаку.

Кодова комбінація створюється, як послідовність двійкових символів, що відповідають групам, у яких брав участь при розбивці даний знак, і випикується зліва направо.

Приклад 2. Задано джерело незалежних дискретних повідомлень із обсягом алфавіту $M_A = 6$ з імовірностями знаків: $P(A) = 0,3$; $P(E) = 0,25$; $P(B) = 0,22$; $P(\Gamma) = 0,1$; $P(D) = 0,08$; $P(\text{Б}) = 0,05$. Побудувати таблицю розбивок на підгрупи за алгоритмом Шеннона-Фано й визначити кодові комбінації знаків повідомлення. Табл. 1 ілюструє алгоритм Шеннона-Фано.

Таблиця 1 – Алгоритм Шеннона-Фано

Знаки a_k	Імовірності $P(a_k)$	Розбивка на підгрупи				Кодові комбінації	
		I	II	III	IV		
A	0,3	0	0			00	
E	0,25		1			01	
B	0,22		0			10	
Г	0,1	1		0		110	
Д	0,08			1		0	1110
Б	0,05				1		1111
					1		

Особливості алгоритмів Хаффмана й Шеннона-Фано:

- якщо до початку кодування повідомлень не відомі ймовірності знаків, то потрібно два проходження по повідомленню, що кодується: одне для складання таблиці ймовірностей знаків і коду, а друге для кодування;

- необхідність передавання таблиці кодових комбінацій (коду) разом зі стислим повідомленням приводить до зменшення сумарного ефекту від стиснення;

- для двійкового джерела повідомлень безпосереднє застосування коду Хаффмана або Шеннона-Фано не дає ефекту;

- надмірність закодованого повідомлення дорівнює нулю тільки у випадку, коли ймовірності знаків алфавіту є цілими негативними степенями двійки ($1/2$; $1/4$; $1/8$ і т.д.);

- стиснення за алгоритмом Хаффмана є оптимальним і в більшості випадків середня довжина кодової комбінації при стисненні за алгоритмом Шеннона-Фано така ж, як і при стисненні за алгоритмом Хаффмана; при деяких

розподілах імовірностей знаків кодування за алгоритмом Шеннона-Фано не буде оптимальним.

Джерело дискретних повідомлень – це джерело, яке видає послідовність знаків кінцевого алфавіту обсягом M_A . Дискретні джерела класифікуються на:

– джерела без пам'яті – поява будь-якого знаку послідовності не залежить від попередніх знаків. Таке джерело задається безумовними ймовірностями знаків алфавіту $P(a_k)$, для $k = 1, 2, \dots, M_A$.

– джерела з пам'яттю – поява будь-якого знаку послідовності залежить від попередніх знаків. Таке джерело задається умовними ймовірностями знаків алфавіту. Для математичного опису джерел дискретних повідомлень з пам'яттю використовують ланцюги Маркова, а такі джерела називаються марківськими K -го порядку. У цьому випадку поява будь-якого знаку послідовності залежить тільки від K попередніх знаків. Тоді марківське джерело 1-го порядку (коли ймовірність появи знаку в послідовності залежить тільки від попереднього знаку) буде задаватися умовними ймовірностями $P(a_k/a_j)$, для $k, j = 1, 2, \dots, M_A$.

Двійкове джерело повідомлень ($M_A = 2$) без пам'яті задається безумовними ймовірностями $P(a_1) = p$ и $P(a_2) = (1 - p)$. Залежність ентропії такого джерела від p показана на рис. 1 і записується як:

$$H(A) = -p \log_2 p - (1 - p) \log_2 (1 - p). \quad (1)$$

Ентропія максимальна, коли знаки рівноймовірні ($P(a_1) = P(a_2) = 0,5$) і дорівнює 1 дв. од. Таким чином, кожен символ в середньому містить не більше 1 дв. од. інформації. У випадку, коли $p \neq 0,5$ – ентропія $H(A) < H_{\max}(A)$ і таке джерело має надлишковість. При $p = 0$ або $p = 1$ ентропія дорівнює 0, оскільки невизначеність відсутня.

Для марківського двійкового джерела повідомлень з пам'яттю 1-го порядку необхідно задати умовні ймовірності $P(a_1/a_1)$, $P(a_1/a_2)$, $P(a_2/a_1)$ та $P(a_2/a_2)$. Ентропія такого джерела визначається:

$$H(A) = - \sum_{i=1}^2 P(a_i) \sum_{j=1}^2 P(a_i/a_j) \log_2 P(a_i/a_j). \quad (2)$$

Приклад 1. Задане марківське двійкове джерело повідомлень з пам'яттю 1-го порядку з імовірностями $P(a_1) = P(a_2) = 0,5$; $P(a_1/a_1) = P(a_2/a_2) = 0,3$; $P(a_1/a_2) = P(a_2/a_1) = 0,7$. Знайти ентропію джерела.

За формулою (2) знаходимо

$$H(A) = - 0,5(0,3 \log_2 0,3 + 0,7 \log_2 0,7) - 0,5(0,3 \log_2 0,3 + 0,7 \log_2 0,7) = 0,881 \text{ дв. од.}$$

Ентропія джерела з пам'яттю менша, ніж ентропія джерела без пам'яті з рівними ймовірностями ($H_{\max}(A) = 1$ дв. од.). У цьому випадку надлишковість обумовлена статистичним зв'язком між знаками.

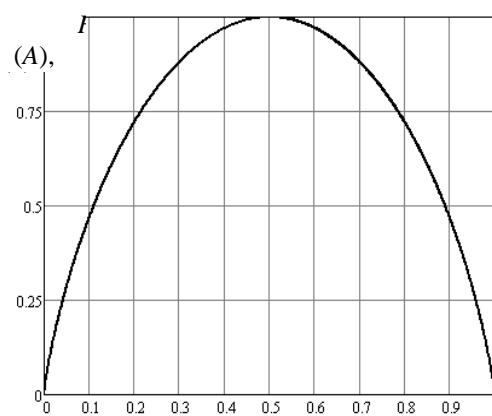


Рисунок 1 – Залежність ентропії $H(A)$ від імовірності p

У випадку двійкового джерела без пам'яті, коли знаки різноймовірні, то ентропія менша 1 дв.од. (рис. 1). Наприклад, $P(a_1) = 0,8$, $P(a_2) = 0,2$. Тоді ентропія дорівнює $H(A) = 0,722$ дв.од. Застосування ефективного кодування (наприклад, коду Хаффмана) до такого двійкового джерела не дасть ніякого ефекту (кожен із знаків буде кодуватися одним двійковим символом, незалежно від ймовірності його появи). Для ефективного кодування потрібне попереднє укрупнення алфавіту.

Під укрупненням алфавіту будемо розуміти формування нового алфавіту укрупнених знаків (укрупнений знак є сполученням з m знаків первинного алфавіту). Обсяг нового (вторинного) алфавіту M_B визначається як

$$M_B = M_A^m. \quad (3)$$

Кодування укрупненого (вторинного) алфавіту буде більш ефективним.

Приклад 2. Розглянемо укрупнення по 3 знаки ($m = 3$) для двійкового наведеного вище джерела без пам'яті з ентропією $H(A) = 0,722$ дв.од.

Оскільки двійкове джерело без пам'яті (тобто таке, що видає знаки незалежно один від одного), то ймовірність появи укрупненого знаку буде дорівнювати добутку ймовірностей вхідних в нього знаків первинного алфавіту ($P(a_1a_1a_1) = P(a_1)P(a_1)P(a_1)$). Обсяг укрупненого алфавіту з формули (3) дорівнює $M_B = M_A^m = 2^3 = 8$. Розрахуємо ймовірності укрупнених знаків і застосуємо до них ефективно кодування (наприклад, кодом Шеннона-Фано) (табл. 1).

Таблиця 1 – Кодування укрупненого алфавіту

Укрупнені знаки b_i	Ймовірності $P(b_i)$	Підгрупи					Кодові комбінації
		I	II	III	IV	V	
$a_1a_1a_1$	0,512	0					0
$a_1a_1a_2$	0,128	1	0	0			100
$a_1a_2a_1$	0,128			1			101
$a_2a_1a_1$	0,128			0			110
$a_1a_2a_2$	0,032		1	1	0	0	11100
$a_2a_1a_2$	0,032						1
$a_2a_2a_1$	0,032				1	0	11110
$a_2a_2a_2$	0,008					1	11111

Середня довжина кодів комбінацій для укрупнених знаків:

$$\bar{n}_m = \sum_{i=1}^{M_B} P(b_i) n_{mi} = 2,184 \text{ символів,}$$

а середнє число кодів символів на знак визначається як

$$\bar{n} = \frac{\bar{n}_m}{m}. \quad (4)$$

Для розглянутого випадку отримаємо $\bar{n} = 0,728$ символів. З порівняння з ентропією $H(A) = 0,722$ дв.од. видно, що попереднє укрупнення алфавіту значно підвищило ефективність стиснення (кодування первинного алфавіту без укрупнення давало $\bar{n} = 1$).

Зі збільшенням кількості знаків при укрупненні m можна отримувати більш ефективний код, тобто середнє число кодових символів на знак буде більше наближатися до ентропії.

Укрупнення алфавіту також доцільне у випадку джерела дискретних повідомлень з пам'яттю. Відомо, що статистична залежність між знаками первинного алфавіту, розташованих поруч, більша, ніж розташованих на відстані декількох знаків. Відповідно, укрупнені знаки будуть мати значно меншу залежність (очевидно, що чим більше m , тим менша залежність між укрупненими знаками). У цьому випадку (для джерела з пам'яттю) для визначення ймовірностей укрупнених знаків необхідно враховувати умовні ймовірності ($P(a_k a_j) = P(a_k) P(a_j / a_k)$).

Таким чином, надлишковість, яка обумовлена статистичними зв'язками між знаками первинного алфавіту, перетвориться в надлишковість, обумовлену різними ймовірностями укрупнених знаків. А таку надлишковість можна зменшити, використовуючи ефективне кодування (наприклад, код Хаффмана).

Дискретним повідомленням є таке, що складається з символів скінченного алфавіту $A = \{a_1, \dots, a_{M_A}\}$, наприклад, текстове повідомлення. Для передавання дискретних повідомлень каналами зв'язку кожному символу повідомлення a_i ставиться у відповідність двійкова комбінація b_i , яка складається з n розрядів:

$$n = \log_2 M_A, \quad (1)$$

де M_A – обсяг алфавіту або кількість знаків алфавіту A ; тут і далі вважаємо, число M_A є ціла степінь числа 2.

Перетворення $a_i \rightarrow b_i$ називається примітивним або рівномірним кодуванням. Прикладами примітивних кодів є міжнародний телеграфний код МТК-2, американський стандартний код ASCII та інші. Найчастіше таке кодування є надлишковим, тобто знаки повідомлення можна передавати ощадливіше, комбінаціями меншої довжини, ніж n .

За теоремою Шеннона для каналу без завад можна закодувати повідомлення так, що середня довжина кодової комбінації \bar{n} буде як завгодно близькою до ентропії джерела повідомлень:

$$\log_2 M_A \geq \bar{n} = H(A) + \varepsilon, \quad (2)$$

де: $H(A)$ – ентропія джерела повідомлень, дв.од.;

ε – як завгодно мала величина.

Таке кодування називають ефективним кодуванням або стисненням повідомлення без втрат інформації.

З виразу (2) випливає, що стиснення повідомлення без втрат інформації буде мати місце, лише коли $\log_2 M_A \geq H(A)$. Таким чином, значення $\log_2 M_A$ є максимальним значенням ентропії джерела повідомлень та досягається воно, коли символи повідомлення є статистично незалежні та рівноймовірні. Однак у більшості випадків повідомлення мають надлишковість, тобто символи повідомлення є статистично залежними та мають різні ймовірності, наприклад, як у текстових повідомленнях.

Для того, щоб стиснути повідомлення, кодер (алгоритм стиснення) повинен усунути причини надлишковості повідомлення. Алгоритми стиснення поділяються на три класи:

- алгоритми, що усувають різномовірність символів;
- алгоритми, що усувають статистичну залежність символів;
- універсальні алгоритми, що усувають як різномовірність, так і статистичну залежність символів.

До алгоритмів першого класу відносяться класичні коди Хаффмана та Шеннона-Фано. Алгоритмами другого класу є словникові, найбільш поширеними з яких є алгоритми Лемпеля-Зіва: LZ77, LZ78, LZW та їх модифікації. Універсальними алгоритмами, що дозволяють як завгодно наблизити середню довжину комбінації до ентропії джерела, тобто $\bar{n} \rightarrow H(A)$, є арифметичні коди.

Найбільше поширення алгоритми стиснення без втрат інформації знайшли у комп'ютерних програмах-архіваторах: RAR, ZIP та інших. Усі сучасні архіватори є комбінованими, тобто будуються на основі двох алгоритмів. Першим алгоритмом усіх архіваторів є LZ77 або його модифікації. Другим алгоритмом є адаптивний код Хаффмана. У підсумку усуваються обидві причини надлишковості.

Найбільш відомим прикладом використання алгоритмів стиснення без втрат інформації у системах зв'язку є протокол V.42bis, який використовується у модемах V.42 для передавання по телефонних каналах. Ядром цього протоколу є алгоритм стиснення LZW, який є модифікацією класичного алгоритму LZ78 і був запропонований Т. Уелчем. Отже, у модемах V.42 усувається лише статистична залежність між символами, оскільки адаптивний код Хаффмана, що усуває різномовірність символів, є неефективним з точки зору швидкодії, оскільки дані повинні передаватися модемом без значної затримки.

Особливої уваги заслуговують словникові алгоритми стиснення LZ77, LZ78 та LZW. Беззаперечною перевагою цих алгоритмів є їх адаптивність, тобто, ці алгоритми однаково стискають повідомлення різних алфавітів, наприклад, текстові повідомлення різних мов. Арифметичний код та код Хаффмана (які є доволі простими) призначені для стиснення, наприклад, англійських текстів не зможуть ефективно стиснути, наприклад, українській текст. Для того, щоб арифметичний код або код Хаффмана однаково ефективно стискали повідомлення різних алфавітів вони повинні бути адаптивними, що значно ускладнює ці алгоритми та зменшує їх швидкодію.

Найбільш досконалим словниковим алгоритмом стиснення вважається LZW. Принцип роботи алгоритму LZW можна викласти трьома пунктами:

- у процесі кодування повідомлення розбивається на послідовності знаків, що не повторюються – на рядки;
- рядки нумеруються – формується словник;
- по каналу передається номер рядка, за яким можна відновити відповідний рядок.

Для того, щоб алгоритм був однозначним, словник необхідно ініціалізувати, тобто сформувати його початкові значення. Для ініціалізації словника вибирають будь-який примітивний код, наприклад, ASCII. Цей код є восьмирозрядним, $n = 8$, та дозволяє закодувати 256 знаків, які є елементарними рядками. Процедура ініціалізації словника полягає у тому, що цим знакам приписуються номери від 0 до 255. У результаті до початку стиснення словник має вигляд табл. 1.

Таблиця 1 – Ініціалізований словник

Рядок	Номер рядка у десятковій системі числення	Номер рядка у двійковій системі числення
"NULL"	0	00 0000 0000
...
"A"	65	00 0100 0001
"B"	66	00 0100 0010
"C"	67	00 0100 0011
"D"	68	00 0100 0100
...
"Ĵ"	255	00 1111 1111
...

З табл. 1 видно, що номери рядків десятирозрядні, а не восьмирозрядні. Це необхідно для того, щоб у процесі стиснення можна було розширювати словник, тобто доповнювати його новими рядками. Всього у наведений в табл. 1 словник може бути записано 1024 рядків. Отже, обсяг словника визначається:

$$M_A = 2^m, \quad (3)$$

де $m > n$ – кількість розрядів для подання номера рядка.

Після ініціалізації словника починається процес стиснення:

- 1) у вхідний буфер заносяться символи до тих пір, доки рядок, що утворюється цими символами, вже є у словнику;
- 2) якщо при занесенні у вхідний буфер наступного символу утворюється відсутній у словнику рядок, то такий рядок заноситься до словника, як новий;
- 3) на вихід кодера (до каналу зв'язку) подається номер останнього знайденого у словнику рядка, а у буфері залишається останній із занесених символів.

Робота алгоритму представлена у табл. 2 на прикладі повідомлення: "АВАВСАВСАВСА".

Кодер виконує за кроками 1...12 наступні дії:

1. Вхідний символ А, у вхідний буфер заносимо символ "А", який є у словнику. У словник нічого не записується, нічого не передається. У вхідний буфер заносимо "А".

2. Вхідний символ В, у вхідний буфер заносимо символ "AB", якого немає у словнику. У словник записується рядок "AB" за номером 256, передається вміст попереднього стану вхідного буфера, тобто код літери "А" – 65. У вхідний буфер заносимо "В".
3. Вхідний символ А, у вхідний буфер заносимо символ "BA", якого немає у словнику. У словник записується рядок "BA" за номером 257, передається вміст попереднього стану вхідного буфера, тобто код літери "В" – 66. У вхідний буфер заносимо "А".
4. Вхідний символ В, у вхідний буфер заносимо символ "AB", який є у словнику. У словник нічого не записується, нічого не передається. У вхідний буфер заносимо "AB".
5. Вхідний символ С, у вхідний буфер заносимо символ "ABC", якого немає у словнику. У словник записується рядок "ABC" за номером 258, передається вміст попереднього стану вхідного буфера, тобто код літер "AB" – 256. У вхідний буфер заносимо "С".

Таблиця 2 – Приклад стиснення повідомлення за алгоритмом LZW

№	Символ	Вхідний буфер	Є у словнику?	Занести у словник		Передається	Вхідний буфер
				рядок	за номером		
1	А	"А"	так	-	-	-	"А"
2	В	"AB"	ні	"AB"	256	65	"В"
3	А	"BA"	ні	"BA"	257	66	"А"
4	В	"AB"	так	-	-	-	"AB"
5	С	"ABC"	ні	"ABC"	258	256	"С"
6	А	"CA"	ні	"CA"	259	67	"А"
7	В	"AB"	так	-	-	-	"AB"
8	С	"ABC"	так	-	-	-	"ABC"
9	А	"ABCA"	ні	"ABCA"	260	258	"А"
10	В	"AB"	так	-	-	-	"AB"
11	С	"ABC"	так	-	-	-	"ABC"
12	А	"ABCA"	так	-	-	260	""

6. Вхідний символ А, у вхідний буфер заносимо символ "CA", якого немає у словнику. У словник записується рядок "CA" за номером 259, передається вміст попереднього стану вхідного буфера, тобто код літери "С" – 67. У вхідний буфер заносимо "А".
7. Вхідний символ В, у вхідний буфер заносимо символ "AB", який є у словнику. У словник нічого не записується, нічого не передається. У вхідний буфер заносимо "AB".
8. Вхідний символ С, у вхідний буфер заносимо символ "ABC", який є у словнику. У словник нічого не записується, нічого не передається. У вхідний буфер заносимо "ABC".

9. Вхідний символ А, у вхідний буфер заносимо символ "АВСА", якого немає у словнику. У словник записується рядок "АВСА" за номером 260, передається вміст попереднього стану вхідного буфера, тобто код літери "АВС" – 258. У вхідний буфер заносимо "А".
10. Вхідний символ В, у вхідний буфер заносимо символ "АВ", який є у словнику. У словник нічого не записується, нічого не передається. У вхідний буфер заносимо "АВ".
11. Вхідний символ С, у вхідний буфер заносимо символ "АВС", який є у словнику. У словник нічого не записується, нічого не передається. У вхідний буфер заносимо "АВС".
12. Вхідний символ А, у вхідний буфер заносимо символ "АВСА", який є у словнику. У словник нічого не записується. Оскільки на вході кодера символів більше немає, передається вміст вхідного буфера, тобто "АВСА" – 260.

Отже, у результаті стиснення повідомлення "АВАВСАВСАВСА" каналом зв'язку передається послідовність з шести чисел: 65 66 256 67 258 260.

Усі алгоритми стиснення, у тому числі і алгоритм LZW, характеризуються трьома параметрами:

- коефіцієнтом стиснення η ;
- швидкістю стиснення та декодування повідомлення;
- обсягом пам'яті, необхідної для роботи кодера та декодера.

Швидкість стиснення, декодування та обсяг пам'яті на сьогодні вже не є визначальними при виборі того чи іншого алгоритму стиснення – визначальним параметром є коефіцієнт стиснення, який забезпечується заданим алгоритмом стиснення.

Коефіцієнтом стиснення називається відношення:

$$\eta = \frac{n}{\bar{n}} = \frac{N_{\text{вх}}}{N_{\text{вих}}}, \quad (4)$$

де: $N_{\text{вх}}$ – кількість двійкових символів, використовуваних для подання повідомлення на вході кодера;

$N_{\text{вих}}$ – кількість двійкових символів, використовуваних для подання повідомлення на виході кодера.

Кількість двійкових символів на вході кодера, необхідних для подання повідомлення "АВАВСАВСАВСА", дорівнює добутку кількості символів повідомлення та кількості розрядів: $N_{\text{вх}} = 12 \times 8 = 96$. Відповідно, кількість двійкових символів на виході кодера: $N_{\text{вих}} = 6 \times 10 = 60$. Таким чином, повідомлення "АВАВСАВСАВСА" стиснуто у 1,6 раз.

Нехай тепер повідомлення складається лише з трьох символів "АВА". У цьому випадку на виході кодера буде спостерігатися наступна послідовність чисел: 65 66 65. У результаті, коефіцієнт стиснення менший за одиницю: $(3 \times 8) / (3 \times 10) = 0,8$. Таким чином, кодер не стискає повідомлення, а навпаки, збільшує надлишковість.

Останній приклад показує основний недолік усіх словникових алгоритмів – погане стиснення коротких повідомлень. Цей факт пояснюється тим, що у процесі стиснення таких повідомлень словник практично не заповнюється, і, в основному, кодуються одиночні символи, а не послідовності символів.

Значною мірою визначальним моментом процесу стиснення повідомлення за алгоритмом LZW є момент повного заповнення словника. У класичному алгоритмі LZW, що вивчається, після заповнення словник не змінюється, а на виході кодера з'являються номери рядків, що присутні у цьому словнику.

Процес заповнення словника – це процес описаної адаптації алгоритму до повідомлення, що кодується, і доки заповнюється словник постійно зростає коефіцієнт стиснення. Як тільки словник заповнюється, алгоритм перестає адаптуватися до повідомлення, а коефіцієнт стиснення перестає збільшуватися.

Класичне рішення цієї проблеми в алгоритмі LZ78 – скид словника після його повного заповнення. Після скиду словника процес його заповнення повторюється. Таким чином алгоритм адаптується до окремих частин повідомлення, що дозволяє дещо збільшити коефіцієнт стиснення. Недолік цього рішення – малий коефіцієнт стиснення на початку заповнення словника.

Також класичне рішення проблеми – збільшення обсягу словника. Теоретично необмежений за обсягом словник дозволяє досягти максимально можливого коефіцієнту стиснення $\log_2 M_A / H(A)$, але при значному збільшенні M_A значно зростає час, що необхідний на пошук рядка у словнику, тобто зменшується швидкість стиснення. Так, протокол V.42bis не обмежує обсяг словника, вказується лише мінімально можливе значення – $M_A \geq 512$.

Найбільш вдале рішення – не припиняти заповнення словника. У цьому випадку номери рядків у словнику постійно змінюються. Номери рядків, що зустрічаються дуже часто, зменшуються, тобто ці рядки розташовуються на початку словника. Рядки, що зустрічаються рідко, розташовуються у кінці словника. Коли словник заповнений, то новий рядок заноситься у словник замість останнього рядка, який найбільш рідко зустрічається у повідомленні.

Пошук рядків у словнику є окремою задачею, а його ефективність визначає швидкість стиснення. Важливо, що при збільшенні обсягу словника для збільшення коефіцієнту стиснення, зменшується швидкість пошуку рядків у словнику. Щоб збільшити швидкість стиснення, одночасно із збільшенням обсягу словника обмежується довжина рядка у словнику N_C . Тобто, у словник не заносяться рядки, довжина яких перевищує N_C , але при цьому, знов таки, дещо зменшується коефіцієнт стиснення. У протоколі V.42bis вказується, що максимальна довжина рядка може змінюватися у діапазоні від 6 до 250 символів.

Перевагою алгоритму LZW є те, що декодування повідомлення не потребує додаткових даних і здійснюється з високою швидкістю. Принцип декодування аналогічний стисненню:

- 1) словник декодера ініціалізується таким же чином, як і словник кодера;
- 2) за прийнятим номером у словнику відшукується відповідний рядок;
- 3) знайдений рядок подається на вихід декодера;

4) у словник заноситься рядок, що складається з попереднього рядка та першого символу поточного рядка.

Приклад декодування стиснутого за алгоритмом LZW повідомлення "65 66 256 67 258 260" наведено у табл. 3. Необхідно звернути увагу на виключну ситуацію, коли приймається номер рядка, якого ще немає у словнику. У цьому випадку рядок заноситься у словник у процесі декодування: новий рядок складається з попереднього рядка та його першого символу.

Таблиця 3 – Приклад декодування повідомлення за алгоритмом LZW

Номер	Є у словнику?	Декодований рядок	Буфер	Занести у словник		Буфер
				рядок	за номером	
65	так	"А"	"А"	-	-	"А"
66	так	"В"	"АВ"	"АВ"	256	"В"
256	так	"АВ"	"ВА"	"ВА"	257	"АВ"
67	так	"С"	"АВС"	"АВС"	258	"С"
258	так	"АВС"	"СА"	"СА"	259	"АВС"
260	ні	"АВСА"	"АВСА"	"АВСА"	260	"АВСА"

Відомий факт, що повідомлення з малою надлишковістю дуже чутливі до помилок, які виникають у каналах зв'язку. Наприклад, якщо при передаванні першого числа стиснутого повідомлення відбудеться помилка і буде прийнято число "193" замість "65", то наступить колізія, яку декодер не здатний вирішити. У такому випадку, відповідно до протоколу V.42bis, здійснюється запит на повторне передавання. Але більш небезпечними є помилки, що не призводять до колізій. Наприклад, помилка відбувається у четвертому числі й приймається послідовність "65 66 256 66 258 260". На виході декодера отримаємо "АВАВВАВВАВВА". Таким чином, одна помилка призвела до появи ще двох, тобто спостерігається явище розмноження помилок. Отже, у разі стиснення повідомлень до каналів зв'язку пред'являються жорсткі вимоги.

Тема 4. Інформаційні характеристики джерел неперервних повідомлень та методи кодування джерел

Дискретизація неперервних за часом сигналів. Під дискретизацією неперервного за часом сигналу $s(t)$ розуміють подання сигналу його миттєвими значеннями (відліками) $s(kT_d)$, де $k = \dots, -1, 0, 1, 2, \dots$; T_d – інтервал дискретизації. Послідовність відліків на графіках зображують вертикальними лініями висотою $s(kT_d)$ кожна (рис. 1) і називають її дискретним сигналом $s_d(t)$.

У реальних пристроях відлік сигналу $s(kT_d)$ – це імпульс з амплітудою $s(kT_d)$ і тривалістю $\tau \leq T_d$, що починається в момент часу kT_d (рис. 2). Але, за звичай, $\tau \ll T_d$. Пристрій, який формує відліки, називається дискретизатором. У разі $\tau \ll T_d$ дискретизатор – це ключ, який замикає коло від джерела до навантаження на час τ (рис. 3).

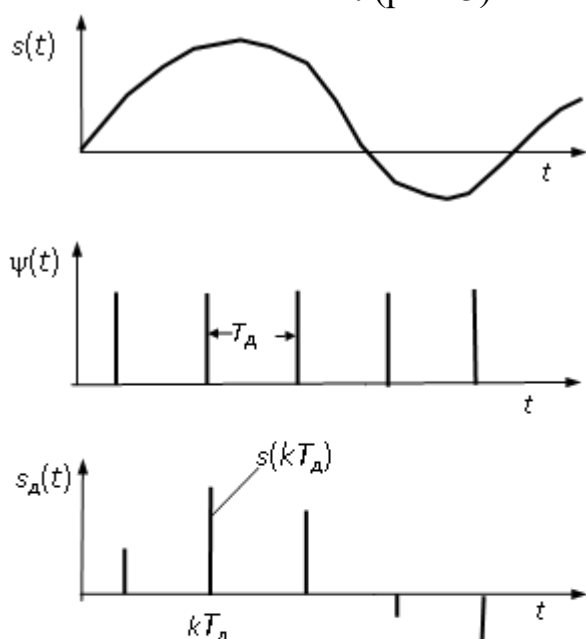


Рисунок 1 – Процес отримання дискретного сигналу

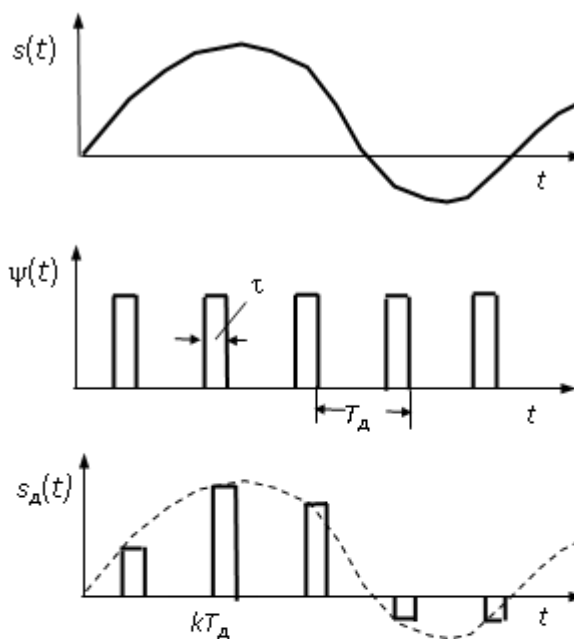


Рисунок 2 – Процес дискретизації в реальних пристроях

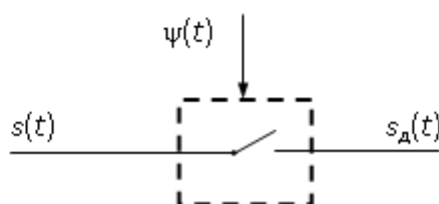


Рисунок 3 – Дискретизатор

Аналітичний вираз дискретного сигналу $s_d(t)$:

$$s_d(t) = s(t) \psi(t) = s(t) \sum_{k=-\infty}^{\infty} p(t - kT_d), \quad (1)$$

де $\psi(t)$ – послідовність відлікових імпульсів, що визначає моменти часу, в які беруться відліки сигналу, й їх тривалість;

$p(t)$ – відліковий імпульс:

$$p(t) = \begin{cases} 1 & \text{для } 0 \leq t < \tau, \\ 0 & \text{для } t < 0, t \geq \tau. \end{cases} \quad (2)$$

Спектр дискретного сигналу. Перетворення Фур'є правої частини виразу (1) визначає спектральну густину $S_d(j2\pi f)$ дискретного сигналу (відповідні математичні викладки можна знайти в [1, с. 33...37]):

$$S_d(j2\pi f) = \sum_{n=-\infty}^{\infty} a_n S(j2\pi(f - nf_d)), \quad -\infty < f < \infty, \quad (3)$$

де $f_d = 1/T_d$ – частота дискретизації;

$$a_n = \frac{\tau}{T_d} \cdot \frac{\sin n\pi f_d \tau}{n\pi f_d \tau} \quad (4)$$

коефіцієнти розкладу імпульсу $p(t)$ у ряд Фур'є; оскільки $\tau \ll T_d$, то для малих значень n коефіцієнти практично не залежать від n , тобто $a_n = \tau/T_d$;

$S(j2\pi f)$ – спектральна густина неперервного сигналу $s(t)$.

Із виразу (3) випливає, що спектр дискретного сигналу – це сума спектрів $S(j2\pi f)$ неперервного сигналу $s(t)$, зміщених на величину f_d і спадаючих зі збільшенням n згідно з виразом (4).

Для первинних сигналів електрозв'язку характерно, що їх спектри прилягають до нульової частоти. На рис. 4, а наведений амплітудний спектр довільної форми $S(f)$ первинного сигналу, що простирається до максимальної частоти F_{\max} . Далі на рис. 4 зображені амплітудні спектри сигналів, які можуть мати

місце під час дискретизації сигналу зі спектром, наведеним на рис. 4, а:

рис. 4, б – спектр $S_\psi(f)$ послідовності відлікових імпульсів $\psi(t)$, побудований на основі представлення $\psi(t)$ рядом Фур'є:

$$\psi(t) = \sum_{n=0}^{\infty} a_n \cdot \cos 2\pi n f_d t;$$

рис. 4, в – спектр $S_d(f)$ дискретного сигналу, коли $f_d > 2F_{\max}$;

рис. 4, г – спектр $S_d(f)$, коли $f_d = 2F_{\max}$;

рис. 4, д – спектр $S_d(f)$, коли $f_d < 2F_{\max}$.

Відновлення сигналів за їх відліками. Згідно з **теоремою відліків** (Котельникова) будь-який сигнал з обмеженим спектром можна точно відновити (інтерполювати) за його відліками, взятими через інтервал $T_d \leq 1/(2F_{\max})$, де F_{\max} – максимальна частота спектра сигналу.

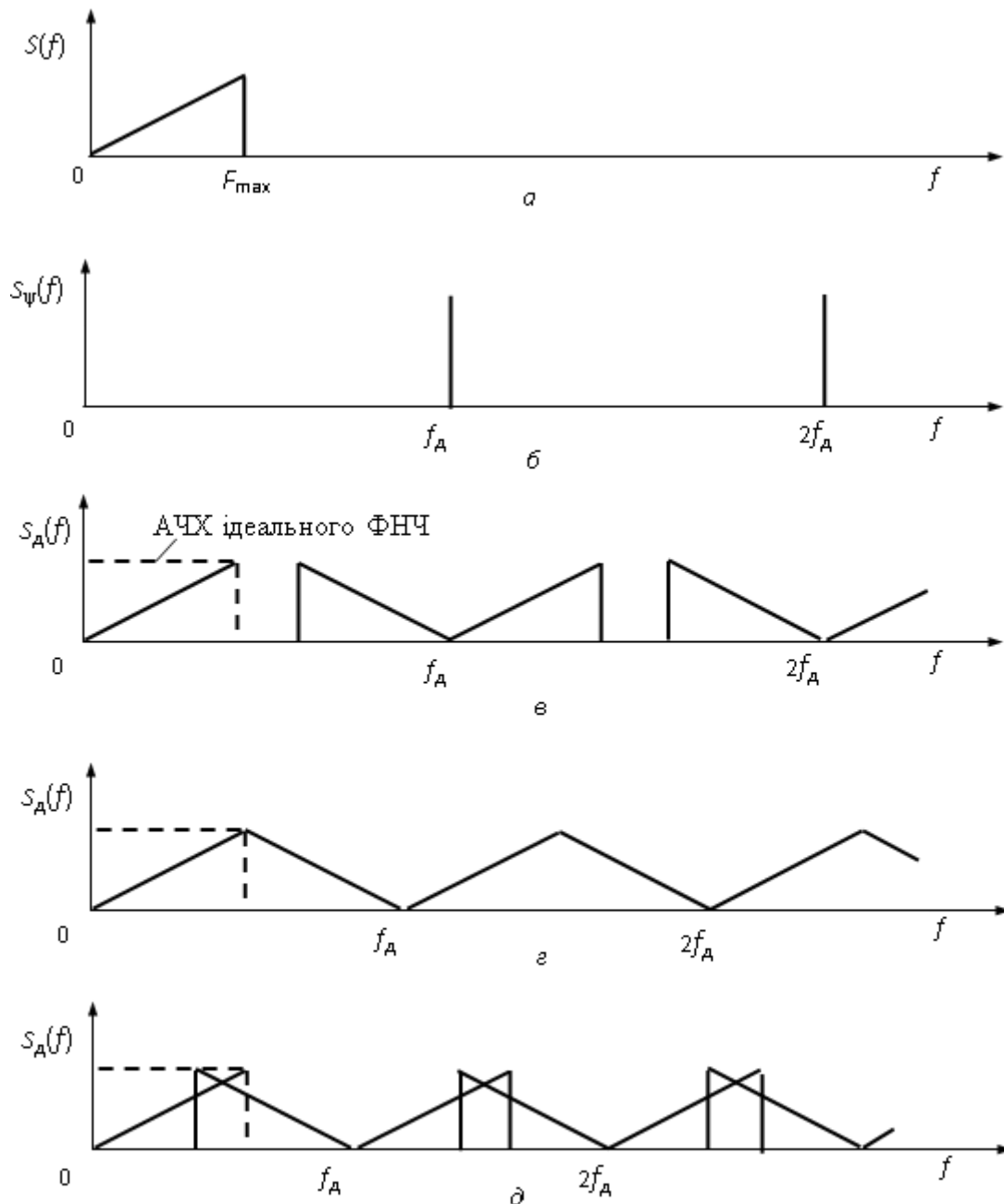


Рисунок 4 – Спектральні діаграми, що ілюструють процеси дискретизації та відновлення неперервних сигналів

У справедливості теореми Котельникова легко переконатися, розглянувши рис. 4, в, г, д. Якщо $f_d \geq 2 F_{\max}$ (рис. 4, в, г), то після подачі дискретного сигналу до входу ідеального ФНЧ з частотою зрізу $F_{\max} \leq F_{зр} \leq f_d - F_{\max}$ на виході одержимо сигнал зі спектром $S(f)$ (рис. 4, в, г), тобто відновлений неперервний сигнал. На рисунках штриховими лініями показана АЧХ ідеального ФНЧ з частотою зрізу $F_{зр} = F_{\max}$. Якщо ж $f_d < 2F_{\max}$, то, як видно з рис. 4, д, неможливо виділити спектр $S(f)$, оскільки має місце перекриття спектрів.

Процес **відновлення** неперервного сигналу за його відліками можна трактувати і в часовій області. Якщо для відновлення сигналу використовується

ідеальний ФНЧ з частотою зрізу $F_{зр}$, то його імпульсний відгук (без урахування затримки у фільтрі):

$$g(t) = \frac{\sin 2\pi F_{зр} t}{2\pi F_{зр} t}. \quad (5)$$

Оскільки відлікові імпульси короткі ($\tau \ll T_d$) (наближаються до δ -функції), то можна вважати, що відгук ФНЧ на імпульс з амплітудою $s(kT_d)$, який поданий у момент $t = k T_d$, має вигляд

$$s(k T_d) = \frac{\sin(2\pi F_{зр} (t - kT_d))}{2\pi F_{зр} (t - kT_d)}. \quad (6)$$

Якщо подати до входу ФНЧ сигнал $s_d(t)$, на його виході одержимо суму відгуків

$$\hat{s}(t) = \sum_{k=-\infty}^{\infty} s(kT_d) \cdot \frac{\sin(2\pi F_{зр} (t - kT_d))}{2\pi F_{зр} (t - kT_d)}. \quad (7)$$

Порівняємо цей вираз з **рядом Котельникова**, який є математичним виразом теореми Котельникова,

$$s(t) = \sum_{k=-\infty}^{\infty} s(kT_d) \cdot \frac{\sin(2\pi F_{\max} (t - kT_d))}{2\pi F_{\max} (t - kT_d)}. \quad (8)$$

Якщо $F_{зр} = F_{\max}$, то $s(t) = \hat{s}(t)$, тобто має місце точне відновлення неперервного сигналу.

Похибки, що виникають під час відновлення сигналів

Необмеженість спектрів реальних сигналів. Реальних сигналів зі строго обмеженим спектром не існує, оскільки сигнали зі скінченною тривалістю мають необмежені спектри – коли $f \rightarrow \infty$, спектри спадають з кінцевою швидкістю. Для реальних сигналів максимальна частота спектра F_{\max} визначається за умови, що складові з частотами $f > F_{\max}$ малі (у певному сенсі). У спектрах реальних дискретних сигналів виникає перекриття спектрів принаймні складових суми (3) з індексами $n = 0$ і $n = 1$ (рис. 5). Припустимо, що для відновлення неперервного сигналу використовується ідеальний ФНЧ з частотою зрізу $F_{зр} = F_{\max}$, його АЧХ показана пунктирною лінією на рис. 5. Відновлений сигнал буде мати дві складові похибки відновлення:

– лінійні спотворення за рахунок відсікання складових сигналу $s(t)$ з частотами $f > F_{\max}$;

– накладення складових спектра $S(f - f_d)$ із частотами $f < F_{\max}$ на спектр сигналу $s(t)$ (похибка накладення спектрів).

З урахуванням сказаного значення F_{\max} та f_d визначають за умови, щоб похибка відновлення була достатньо малою.

Відхилення характеристик реального ФНЧ від ідеального. У ідеального ФНЧ АЧХ має прямокутну форму, а ФЧХ – лінійну. Тобто, ідеальний ФНЧ без спотворень пропускає всі складові спектра сигналу в межах смуги пропускання,

коли $f < F_{зр}$, і повністю ослаблює складові з частотами $f > F_{зр}$. Реальні ФНЧ описуються граничною частотою смуги пропускання $F_{сп}$ і граничною частотою смуги затримання $F_{сз}$ (рис. 6, б).

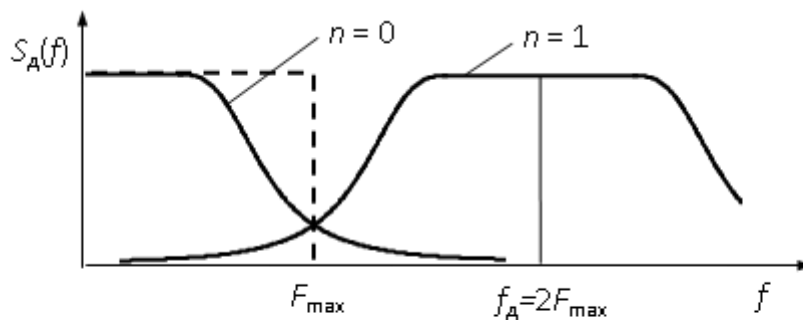


Рисунок 5 – Ілюстрація виникнення похибки відновлення неперервного сигналу через необмеженість спектра неперервного сигналу

Якщо ФНЧ призначений для відновлення неперервного сигналу з максимальною частотою F_{max} з дискретного сигналу з частотою дискретизації f_d , то необхідно, щоб $F_{сп} \geq F_{max}$ і $F_{сз} \leq f_d - F_{max}$. У разі реальних ФНЧ можуть виникати дві складові похибки відновлення:

- через непостійність АЧХ і нелінійність ФЧХ у смугі пропускання фільтр вносить лінійні спотворення у відновлений сигнал;
- через недостатнє ослаблення у смугі затримання ФНЧ пропускає складові сигналу $s_d(t)$ з частотами $f > f_d - F_{max}$, які утворюють похибку накладення спектрів.

Реальні ФНЧ для відновлення неперервних сигналів проектують так, щоб похибка відновлення була достатньо малою.

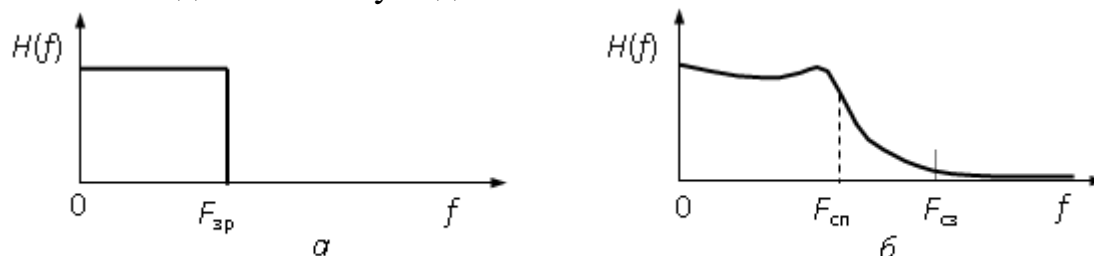


Рисунок 6 – АЧХ: а – ідеального ФНЧ; б – реального ФНЧ

Цифрові методи передавання. У сучасних телекомунікаціях широко використовуються методи цифрового передавання аналогових сигналів: аналоговий сигнал перетворюється в цифровий, тобто в послідовність двійкових символів, цифровий сигнал передається цифровим каналом зв'язку, після чого відновлюється аналоговий сигнал. Будь-який метод цифрового передавання характеризується швидкістю цифрового сигналу R (біт/с) і точністю передавання – відношенням сигнал/шум квантування $\rho_{кв}$. Звичайно ставиться задача – задовольнити вимозі за відношенням $\rho_{кв}$ при мінімальному значенні R . Це призвело до розробки великої кількості цифрових методів передавання. Найпростішими серед них є: імпульсно-кодова модуляція (ІКМ),

диференціальна імпульсно-кодова модуляція (ДІКМ) і дельта-модуляція (ДМ)¹. Перелічені назви поєднують собою цілі сімейства методів.

Дискретизація аналогових сигналів. При будь-якому методі цифрового передавання аналоговий сигнал $b(t)$, передусім, перетворюється в дискретний сигнал, що представляє собою послідовність відліків $b(kT_d)$, узятих через інтервал дискретизації $T_d \leq 1/(2F_{\max})$, де F_{\max} – максимальна частота спектра сигналу $b(t)$. Це перетворення називається *дискретизацією сигналу за часом*, а пристрій для її здійснення – *дискретизатором*. Частота дискретизації має бути не менше подвоєної частоти F_{\max} :

$$f_d = 1/T_d \geq 2F_{\max}. \quad (1)$$

Згідно з теоремою Котельникова, виконання цього співвідношення гарантує можливість точного відновлення аналогового сигналу за відліками. Таке відновлення здійснюється ФНЧ із частотою зрізу F_{\max} (рис. 1).

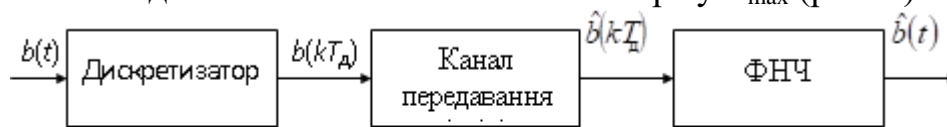


Рисунок 1 – Передавання аналогового сигналу відліками

Усі методи цифрового передавання аналогових сигналів відрізняються способами представлення дискретних сигналів цифровими. Перетворювач відліків у цифровий сигнал називається кодером цифрової системи передавання (ЦСП), а перетворювач цифрового сигналу у відліки – декодером ЦСП.

Методи ІКМ. Особливістю цих методів є те, що кожний відлік представляється цифровим сигналом незалежно від інших відліків. Схема, що відображає передавання відліків при ІКМ, наведена на рис. 2.

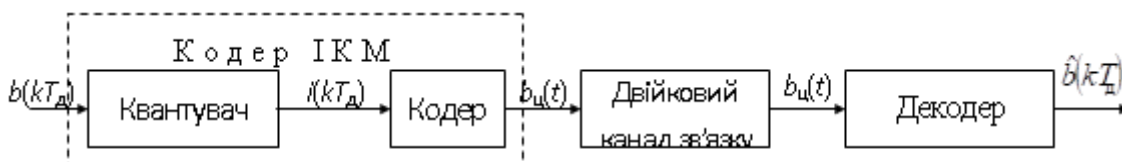


Рисунок 2 – Кодер та декодер ІКМ

Основний параметр квантувача – число рівнів квантування L . При рівномірному квантуванні діапазон значень b від $-b_{\max}$ до b_{\max} розбивається на $L - 1$ інтервалів величиною

$$\Delta b = 2b_{\max} / (L - 1), \quad (2)$$

яка називається кроком квантування. На рис. 3 показана розбивка при $L = 8$. Дискретні значення b_i відповідають серединам інтервалів. Індекс i приймає значення $0, \pm 1, \pm 2, \dots, \pm 0,5L - 1$. Дискретні значення визначаються $b_i = i \cdot \Delta b$. При квантуванні кожен відлік $b(kT_d)$ округляється до найближчого дискретного

¹ Незважаючи на наявність слова «модуляція», ці методи передавання не мають відношення ні до аналогової, ні до цифрової модуляції.

значення b_i , а на вихід квантувача надходить ціле число $i(kT_d)$. Представлення відліку $b(kT_d)$ дискретним значенням b_i вносить похибку

$$\varepsilon_{\text{кв}}(kT_d) = i(kT_d) \cdot \Delta b - b(kT_d), \quad (3)$$

яка називається шумом квантування.

У кодері, що входить до складу кодера ІКМ (рис. 2), числа $i(kT_d)$ представляються заданим двійковим кодом. Довжина коду

$$n = \log_2 L. \quad (4)$$

Цифровий сигнал на виході кодера $b_{\text{ц}}(t)$ має швидкість

$$R = n \cdot f_d. \quad (5)$$

Декодер з цифрового сигналу $b_{\text{ц}}(t)$ формує числа $i(kT_d)$, за якими відновлюються квантовані відліки: $i(kT_d) \Delta b = b_{\text{кв}}(kT_d) = \hat{b}(kT_d)$. З формули (3) випливає, що відліки відновлюються з похибками $\varepsilon_{\text{кв}}(kT_d)$. Середній квадрат похибки квантування (*середня потужність шуму квантування*) визначається кроком квантування

$$\overline{\varepsilon_{\text{кв}}^2} = \frac{(\Delta b)^2}{12},$$

а відношення сигнал/шум квантування

$$\rho_{\text{кв}} = \frac{P_b}{\overline{\varepsilon_{\text{кв}}^2}} = \frac{3(L-1)^2}{K_A^2}, \quad (6)$$

де K_A – коефіцієнт амплітуди аналогового сигналу.

Широко використовуються методи ІКМ з нерівномірним квантуванням: в області великих значень $|b|$ кроки квантування більші і навпаки. Це еквівалентно нелінійному перетворенню відліків з наступним рівномірним квантуванням. За рахунок такого перетворення зменшується K_A і зростає $\rho_{\text{кв}}$ при незмінному числі L – формула (6), або, зберігаючи значення $\rho_{\text{кв}}$, можна зменшити число L , довжину коду n і швидкість цифрового сигналу R – формули (4) і (5).

Методи ДІКМ. Вони відносяться до методів передавання із завбаченням. При цифрових методах передавання частота дискретизації (1) вибирається з умови відсутності накладення складових спектра дискретного сигналу. При цьому відліки реальних аналогових сигналів є корельованими. Це дозволяє з тією чи іншою точністю завбачувати значення чергового відліку сигналу за його попередніми відліками. Ця можливість покладена в основу схем кодера і декодера ДІКМ, що наведені на рис. 4. Завбачники у кодері і декодері цілком ідентичні. Вони обчислюють похибку завбачення

$$d(kT_d) = b(kT_d) - \tilde{b}(kT_d), \quad (7)$$

де $\tilde{b}(kT_d)$ – завбачений відлік, сформований завбачником на основі N попередніх відліків $\hat{b}((k-1)T_d)$, $\hat{b}((k-2)T_d)$, Λ $\hat{b}((k-N)T_d)$. У різних варіантах

використання методу ДІКМ число N знаходиться в межах від 1 до 6. Завбачник при $N \geq 2$ виконується за схемою нерекурсивного фільтра. У випадку $N = 1$ завбаченим відліком $\tilde{b}(kT_d)$ є попередній відлік $\hat{b}((k-1)T_d)$.

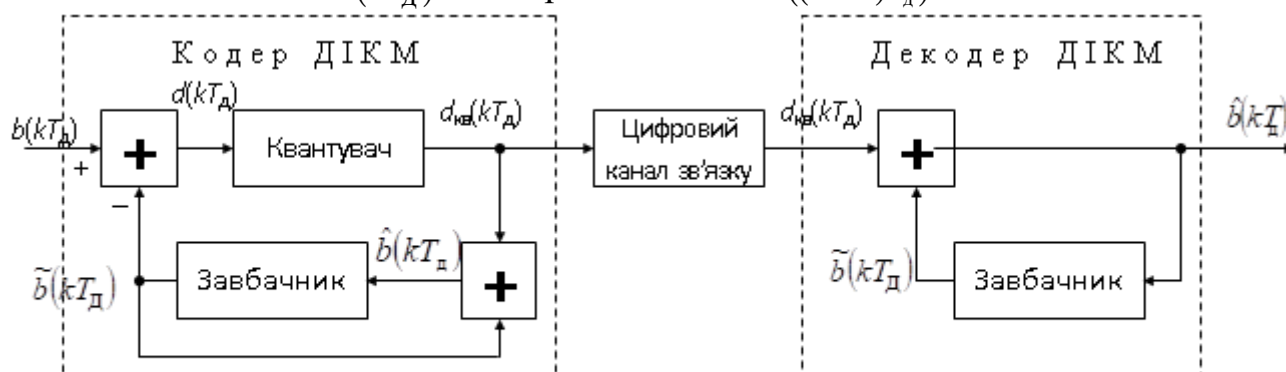


Рисунок 4 – Кодер і декодер ДІКМ

У кодері похибка завбачення надходить на квантувач, аналогічний квантувачу ІКМ, а потім квантована похибка $d_{\text{кв}}(kT_d)$ передається цифровим сигналом каналом зв'язку (на рис. 4 не показані кодер для представлення $d_{\text{кв}}(kT_d)$ двійковим кодом і декодер для відновлення $d_{\text{кв}}(kT_d)$ – вони включені до складу каналу зв'язку).

Похибка квантування при ДІКМ

$$\begin{aligned} \varepsilon_{\text{кв}}(kT_d) &= \hat{b}(kT_d) - b(kT_d) = [\tilde{b}(kT_d) + d_{\text{кв}}(kT_d)] - [\tilde{b}(kT_d) + d(kT_d)] = \\ &= d_{\text{кв}}(kT_d) - d(kT_d), \end{aligned} \quad (8)$$

визначається тільки параметрами квантувача.

Розмах дискретного сигналу $d(kT_d)$ менший, ніж розмах сигналу $b(kT_d)$, тому число рівнів квантування L при незмінному кроці квантування буде меншим, ніж при ІКМ, а, отже, менше довжина коду n і швидкість цифрового сигналу R .

Методи ДМ. Вони, так само як і ДІКМ, відносяться до методів передачі з завбаченням. Методи ДМ відрізняються тим, що використовуються дворівневі квантувачі ($L = 2$). Це стає можливим, коли частота дискретизації вибирається в декілька разів більшою за $2F_{\text{max}}$, і сусідні відліки з дискретизатора мало відрізняються. На рис. 5 наведені схеми кодера і декодера, що пояснюють один з методів ДМ.

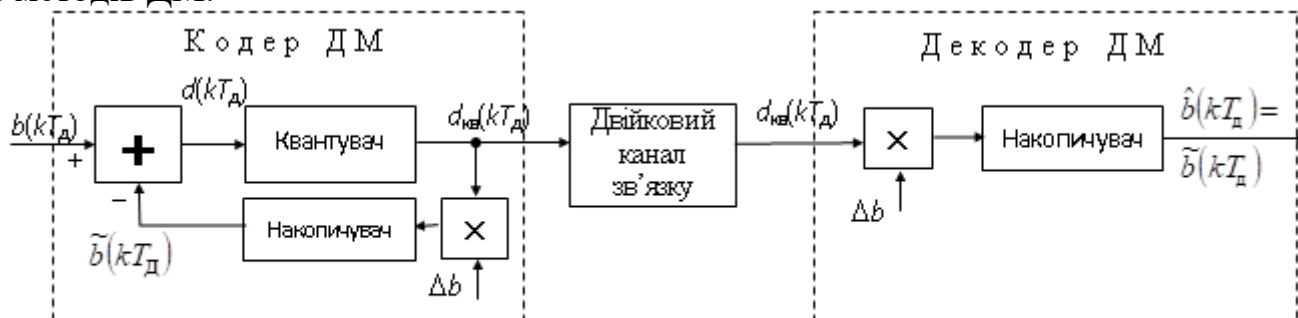


Рисунок 5 – Кодер і декодер ДМ

Похибка завбачення обчислюється так само, як і при ДІКМ, – співвідношення (10.7), а завбачений відлік – результат роботи накопичувача

$$\tilde{b}(kT_D) = \sum_{i=0}^{k-1} d_{\text{кв}}(iT_D) \cdot \Delta b, \quad (9)$$

де Δb – коефіцієнт;

$$d_{\text{кв}}(kT_D) = \begin{cases} +1, & \text{якщо } d(kT_D) \geq 0, \\ -1, & \text{якщо } d(kT_D) < 0 \end{cases} \quad (10)$$

квантована на два рівні похибка завбачення, що передається двійковим каналом зв'язку.

Описаний метод кодування ілюструється часовими діаграмами на рис. 6. Тут завбачений сигнал і сигнал квантованої похибки завбачення представлені сигналами неперервного часу. Видно, що завбачений сигнал $\tilde{b}(t)$ “відслідковує” зміни вхідного сигналу. З рисунку впливає зміст коефіцієнта Δb – це крок квантування, тому що з цим кроком квантується сигнал $\tilde{b}(t)$. На рисунку видно дві області:

1) область, де спостерігаються спотворення перевантаження за нахилом – завбачений сигнал $\tilde{b}(t)$ не встигає відслідковувати зміни вхідного сигналу;

2) область, де спостерігається шум дроблення – при незмінному вхідному сигналі завбачений сигнал змінюється з розмахом Δb .

Ясно, що для зменшення першого ефекту необхідно збільшувати крок квантування, а для зменшення другого ефекту – зменшувати крок квантування. Очевидно, що існує оптимальний крок квантування, за якого мінімізується сумарний ефект проявлення перевантаження за нахилом і шуму дроблення на реалізаціях сигналу $b(t)$ великої тривалості.

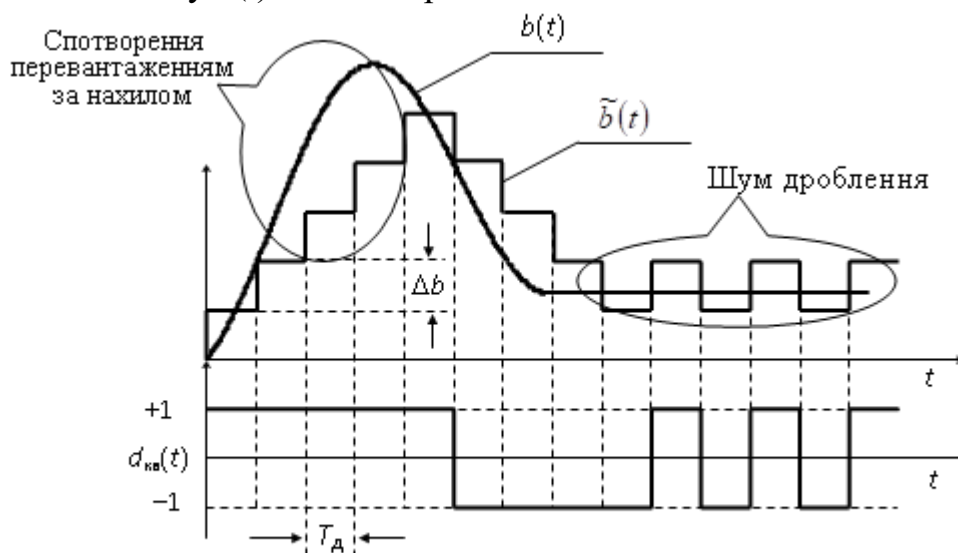


Рисунок 6 – Ілюстрація роботи кодера ДМ

Робота декодера ДМ (рис. 5) зводиться до обчислення відліків завбаченого сигналу за формулою (9).

На основі опису роботи кодера і декодера ДМ можна сформулювати особливості методів передачі з ДМ:

- частота дискретизації дискретизатора f_d (рис. 1) у декілька разів більша за $2F_{\max}$;
- оскільки квантувач дворівневий, то код має довжину $n = 1$, і $R = f_d$;
- оскільки $n = 1$, то відпадає необхідність синхронізації декодера.

Тема 5. Інформаційні характеристики каналів зв'язку

Швидкість передавання інформації каналом зв'язку це кількість переданої інформації за одиницю часу R_k дв.од./с. Швидкість передавання інформації каналом зв'язку визначається як різниця кількості інформації на вході каналу і кількість втраченої (зруйнованої) інформації в каналі зв'язку.

Основною інформаційною характеристикою заданого каналу зв'язку є його *пропускна здатність*, яка визначає максимальну швидкість передавання інформації цим каналом: $C = \max R_k$.

Пропускна здатність дискретного двійкового симетричного каналу (ймовірність помилок 0 і 1 однакова) $C_{д.к}$, дв.од./с, яким можна передавати 2 символи однакової тривалості, обчислюється за формулою

$$C_{д.к} = B[1 + (1 - p) \log_2(1 - p) + p \log_2 p], \quad (1)$$

де B – символна швидкість, Бод; p – імовірність помилки символу.

Приклад 1.

Обчислити пропускну здатність двійкового каналу при символній швидкості в ньому 150 Бод і ймовірності помилки символу $p = 10^{-4}$.

$$\begin{aligned} C_{д.к} &= B[1 + (1 - p) \log_2(1 - p) + p \log_2 p] = 150[1 + (1 - 10^{-4}) \log_2(1 - 10^{-4}) \\ &+ 10^{-4} \log_2 10^{-4}] = 150[1 + 0,0009 \log_2 0,0009 + 10^{-4} \log_2 10^{-4}] = 150[1 - 0,0091 - \\ &0,0013] = \\ &= 148,4 \text{ дв.од./с.} \end{aligned}$$

Пропускна здатність каналу без помилок $C_{д.к} = B \log_2 M$, якщо канал двійковий, то

$$C_{д.к} = B. \quad (2)$$

Якщо на вхід каналу подаються рівноймовірні незалежні символи.

Помилки, які виникають в каналі, призводять до того, що на виході каналу залишається деяка невизначеність щодо передаваного символу. Чим більша невизначеність тим менша пропускна здатність, тобто виникає явище втрати інформації через помилки, що виникають в каналі зв'язку. Як впливає з графіка, розрахованого за формулою (1), при $p = 0,5$ пропускна здатність двійкового

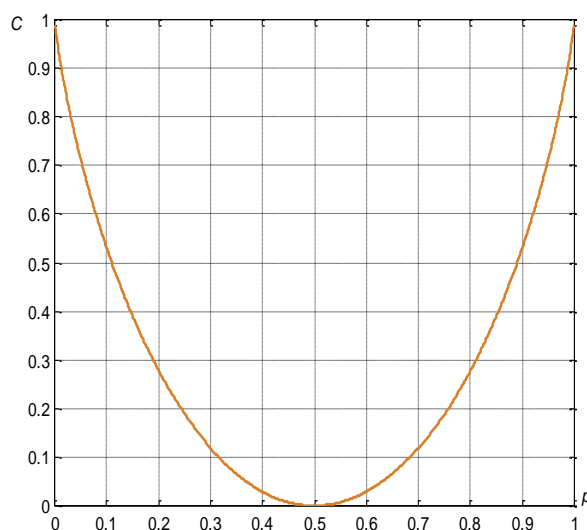


Рисунок 1 – Пропускна здатність ДСК $C_{д.к}$ в залежності від ймовірності помилки на символ p

каналу дорівнює нулю. Дійсно, якщо ймовірність помилки символу дорівнює 0,5, то по символу на виході каналу не можна нічого сказати про те, який символ було подано на вхід каналу – не отримуємо ніякої інформації. А при $p = 1$ пропускна здатність така ж, як і при $p = 0$ (канал без завад). Це пояснюється тим, що при $p = 1$ досить замінити нулі на одиниці і одиниці на нулі, щоб абсолютно правильно відновити переданий сигнал.

Під гауссовим каналом розуміємо неспотворюючий форму сигналу канал, в якому до сигналу додається АБГШ (аддитивний білий гауссів шум). Сигнал на виході каналу $z(t) = s(t) + n(t)$, де $s(t)$ – сигнал на вході каналу, а $n(t)$ – шум.

Пропускна здатність гауссового каналу $C_{г.к.}$, дв.од./с обчислюється за формулою

$$C_{г.к.} = F_{к} \log_2 (1 + P_s / (N_0 \cdot F_{к})) = F_{к} \log_2 (1 + P_s / P_n), \quad (3)$$

де $F_{к}$ – смуга пропускання каналу; $P_s / (N_0 \cdot F_{к})$ – відношення сигнал/шум; N_0 – потужність шуму у смузі 1 Гц.

Наявність шуму, діючого в каналі, призводять до того, що на виході каналу залишається деяка невизначеність щодо значення сигналу в певний момент часу. Чим більша невизначеність, тим менша пропускна здатність, тобто виникає явище втрати інформації через шум, що діє в каналі зв'язку.

Приклад 2.

Обчислити пропускну здатність гауссового каналу при його смузі пропускання $F_{к} = 2$ кГц, потужності сигналу $P_s = 10^{-3}$ В² і спектральній густині потужності шуму в ньому $N_0 = 10^{-6}$ В²/Гц.

$$P_s / P_n = P_s / (N_0 \cdot F_{к}) = 0,001 / (10^{-6} \cdot 2000) = 0,54;$$

$$C_{г.к.} = F_{к} \log_2 (1 + P_s / P_n) = 2000 \log_2 (1 + 0,54) = 1170 \text{ дв.од./с.}$$

З формули (3) видно, що при потужності сигналу $P_s \rightarrow \infty$ пропускна здатність каналу $C_{г.к.} \rightarrow \infty$, і пропускна здатність дорівнює нулю, якщо потужність дорівнює нулю. Формула Шеннона вказує на можливість обміну смугою пропускання каналу на потужність сигналу, і навпаки, якщо пропускна здатність фіксована.

Якщо $F_{к} \rightarrow 0$, то отримаємо невизначеність $C_{г.к.} \rightarrow 0 \cdot \infty$, але перший множник прямує до нуля швидше ніж другий до нескінченності, через наявність логарифму, тому $C_{г.к.} = 0$. Коли $F_{к} \rightarrow \infty$, пропускна здатність прямує

до сталої:

$$C_{\text{г.к.}} = 1,443 \frac{P_s}{N_0} \quad (4)$$

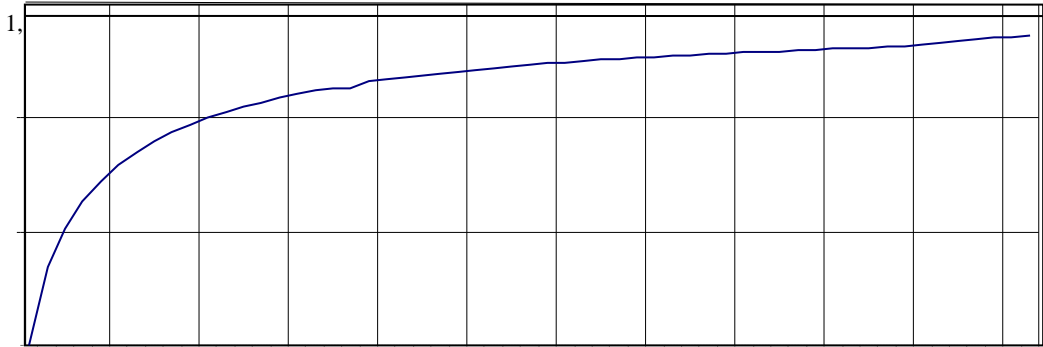


Рисунок 2 – Залежність пропускної здатності гауссового каналу $C_{\text{г.к.}}$ від смуги пропускання каналу F_k

При малих значеннях смуги пропускання пропускна здатність суттєво збільшується з ростом F_k . При подальшому збільшенні F_k зростання сповільнюється.

Приклад 3.

Якою має бути потужність сигналу для забезпечення пропускної здатності $C_{\text{г.к.}} = 12000$ дв.од./с за умови необмеженої смуги пропускання каналу і спектральної густини потужності шуму в ньому $N_0 = 10^{-9}$ В²/Гц.

Коли F_k прямує до нескінченності, пропускна здатність каналу прямує до сталої $C_{\text{г.к.}} = 1,443 \frac{P_s}{N_0}$. З останнього співвідношення випливає, що для передачі зі швидкістю $C_{\text{г.к.}} = 12000$ дв.од./с необхідно забезпечити потужність сигналу $P_s \geq \frac{C_{\text{г.к.}} N_0}{1,443} = 0,69 C_{\text{г.к.}} N_0 = 0,69 \cdot 12000 \cdot 10^{-9} = 8,28 \cdot 10^{-6}$ В².

Якщо сигнал на вході каналу має гауссів (нормальний) розподіл ймовірностей миттєвих значень, ширина спектра сигналу дорівнює F_k , спектр сигналу рівномірний і попадає в смугу пропускання каналу зв'язку.

Теорема Шеннона для каналу із завадами характеризує потенційні можливості передавання повідомлень. Формулювання теореми таке: якщо продуктивність джерела $R_{\text{дж}}$ менша за пропускну здатність каналу C_k , тобто $R_{\text{дж}} < C_k$, то існує спосіб кодування (перетворення повідомлення в сигнал на вході каналу) та декодування (перетворення сигналу в повідомлення на виході каналу), за яких точність відновлення (декодування) повідомлення може бути як завгодно високою.

Тема 6. Передавання інформації каналами зв'язку з завадами

Передавання інформації каналами зв'язку з завадами є одним із центральних питань теорії електричного зв'язку та теорії інформації. Реальний канал зв'язку завжди містить завади – випадкові процеси, що спотворюють корисний сигнал під час передачі. Це можуть бути тепловий шум, промислові перешкоди, атмосферні завади, імпульсні сплески, міжсимвольна інтерференція чи інші фактори, через які прийнятий сигнал відрізняється від переданого.

У класичній моделі Клода Шеннона (1948) канал з завадами розглядається як середовище, що додає шум до сигналу. Повідомлення від джерела кодується (стискається та захищається), модулюється в сигнал, який проходить канал і на виході отримується зашумлена версія. Завдання приймача – відновити оригінальне повідомлення з максимальною вірогідністю, попри спотворення.

Ключова ідея Шеннона полягає в тому, що навіть у каналі з завадами можливе надійне (майже безпомилкове) передавання інформації зі швидкістю, меншою за певну межу – пропускну здатність каналу (канальну ємність, channel capacity). Ця межа залежить від потужності сигналу, рівня завад і характеристик самого каналу, але не залежить від конкретних методів кодування — якщо швидкість передачі нижча за цю межу, то існують коди, які роблять ймовірність помилки скільки завгодно малою при достатній довжині блоку.

Для дискретних каналів, наприклад бінарного симетричного каналу (BSC), де кожен біт перекидається з імовірністю p , пропускну здатність дорівнює $1 - H(p)$, де $H(p)$ – ентропія Бернуллівського розподілу з параметром p . Для каналу з адитивним гауссовим білим шумом (AWGN), найпоширенішого в радіозв'язку, формула Шеннона має вигляд $C = B \log_2(1 + S/N)$, де B – ширина смуги частот, S/N – відношення сигнал/шум. Це означає, що збільшення потужності сигналу або зменшення шуму дозволяє передавати більше інформації, але є фундаментальна межа.

Щоб досягти цієї межі на практиці, застосовують завадостійке кодування – введення контрольної надлишковості, яка дозволяє виявляти та виправляти помилки. До таких кодів належать блокові (Хеммінга, Ріда-Соломона, LDPC, Turbo, Polar-коди в 5G) та згорткові коди. Кодер каналу додає біти, а декодер на приймальному боці використовує оптимальні алгоритми (наприклад, Вітербі для згорткових кодів або ітеративне декодування для Turbo/LDPC), щоб відновити дані.

Без кодування прості методи (повторення, примітивні перевірки) дають низьку ефективність: для зменшення помилок доводиться сильно знижувати швидкість. Шеннон довів, що це не обов'язково – розумне кодування дозволяє передавати з високою швидкістю і малою ймовірністю помилки одночасно.

У сучасних системах (Wi-Fi, 5G, супутниковий зв'язок, оптичні лінії) поєднують ефективно кодування джерела (стиснення), завадостійке кодування, модуляцію (наприклад, QAM з кодом Грея) та адаптивні методи (вибір модуляції залежно від поточного рівня шуму). Це дозволяє наблизитися до теоретичної межі Шеннона, роблячи передавання інформації через зашумлені канали надійним і ефективним навіть у складних умовах.

Розглянемо в цій темі більш детально цифрову модуляцію, а в наступних перейдемо до завадостійкого кодування.

Первинний цифровий сигнал $b(t)$ – це послідовність двійкових символів (бітів) 1 і 0, що слідує через інтервал T_6 . У цифрових пристроях прямокутний імпульс високого рівня відповідає 1, а імпульс низького рівня – 0. Основний параметр первинного цифрового сигналу – його швидкість R (біт/с).

Сигнал цифрової модуляції $s(t)$ – це послідовність радіоімпульсів, які відображають первинний сигнал і слідує через тактовий інтервал T :

$$s(t) = \sum_{k=-\infty}^{\infty} s_i^{(k)}(t - kT), \quad (1)$$

де $s_i(t)$, $i = 0, \dots, M - 1$ – каналні символи (радіоімпульси);

M – число каналних символів (рівнів модульованого сигналу);

$s_i^{(k)}(t - kT)$ – i -й каналний символ, що передається на k -му тактовому інтервалі.

Радіоімпульси можуть відрізнятися амплітудами, фазами чи частотами. Існують різні види цифрової модуляції: АМ- M , ФМ- M , АФМ- M , КАМ- M , ЧМ- M .

Якщо $M = 2$, то має місце двійковий сигнал $s(t)$, коли символ $s_0(t)$ використовується для передавання 0, а символ $s_1(t)$ – для передавання 1. Якщо $M > 2$, то має місце багатопозиційний (багаторівневий) сигнал $s(t)$. Як правило, $M = 4, 8, \dots, 2^n$, n – ціле число. Тут кожний символ $s_i(t)$ використовується для передавання $n = \log_2 M$ біт первинного сигналу $b(t)$. Яку саме послідовність біт переносить кожний символ установлює модуляційний код. Якщо в разі двійкових сигналів $T = T_6$, то в разі багатопозиційних сигналів тривалість тактового інтервалу збільшується: $T = T_6 \log_2 M$.

Для сигналів АМ- M і ФМ-2 каналні символи записуються:

$$s_i(t) = a_i A(t) \sqrt{2} \cos(2\pi f_0 t), \quad i = 0, 1, \dots, M - 1, \quad (2)$$

де a_i – число, що відображає n біт, які передаються символом $s_i(t)$;

$A(t)$ – функція, що визначає форму радіоімпульсів;

f_0 – частота носійного коливання.

Із виразу (2) випливає, що каналні символи є сигналами аналогової БМ і, тому спектр радіоімпульсу $s_i(t)$ складається з двох бокових смуг, зосереджених біля частоти носійної f_0 . Спектральні властивості радіоімпульсу $s_i(t)$ цілком визначаються функцією $A(t)$.

Якщо функція $A(t)$ – прямокутний імпульс тривалості T , то спектр радіоімпульсу буде широким, а для передавання цифрових сигналів важливо сформувати компактний спектр. Для того, щоб спектр радіоімпульсу $s_i(t)$ був компактним, і була відсутня міжсимвольна інтерференція, функція $A(t)$ повинна бути імпульсом Найквіста. Тоді бокові смуги частот будуть копіями спектра Найквіста (рис. 1), а ширина спектра сигналів АМ-М і ФМ-2:

$$\Delta F = 2f_n(1 + \alpha) = \frac{1 + \alpha}{T} = \frac{1 + \alpha}{T_0 \log_2 M} = \frac{R(1 + \alpha)}{\log_2 M}, \quad (3)$$

де $f_n = 0,5/T$ – частота Найквіста;

α – коефіцієнт скату спектра ($0 \leq \alpha \leq 1$).

Із виразу (3) випливає важливий висновок – збільшення числа позицій сигналу дозволяє зменшити ширину спектра каналних символів (2).

Канальні символи прийнято умовно зображати у вигляді сигнальних точок у деякому просторі. Сигнальні точки сигналів АМ-М і ФМ-2 розташовуються на числовій осі і тому ці сигнали називають одновимірними (рис. 2). Діаграми, на яких канальні символи зображені у вигляді сигнальних точок, називаються сигнальними сузір'ями.

Модуляційний код сигналу АМ-2: передачі 0 відповідає $a_0 = 0$, а передачі 1 відповідає $a_1 = a$.

Модуляційний код сигналу ФМ-2: $0 \rightarrow a_0 = -a$; $1 \rightarrow a_1 = a$.

Модуляційний код сигналу АМ-4: $00 \rightarrow a_0 = -a$; $01 \rightarrow a_1 = -3a$; $10 \rightarrow a_2 = a$; $11 \rightarrow a_3 = 3a$. Число a визначає енергії каналних символів.

Канальні символи $s_i(t)$ у разі ФМ-М ($M \geq 4$) і АФМ-М у загальному вигляді описуються за допомогою синфазної і квадратурної складових:

$$s_i(t) = a_{ci}A(t)\sqrt{2} \cos 2\pi f_0 t + a_{si}A(t)\sqrt{2} \sin 2\pi f_0 t, \quad i = 0, 1, \dots, M - 1, \quad (4)$$

де a_{ci} , a_{si} – коефіцієнти, що відображають послідовність із n біт, яка передається каналним символом $s_i(t)$.

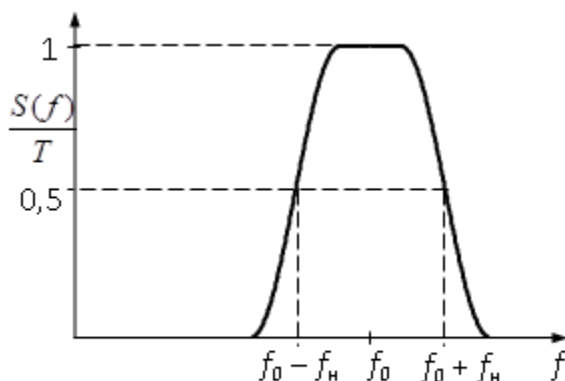


Рисунок 1 – Спектр елементарних сигналів у випадку АМ-М та ФМ-М ($\alpha = 0,6$)

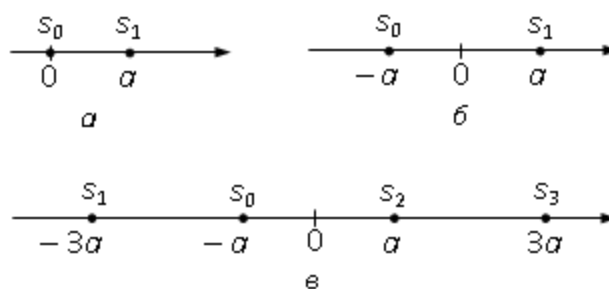


Рисунок 2 – Сигнальні сузір'я сигналів: а – АМ-2; б – ФМ-2; в – АМ-4

Тема 7. Блокові коректувальні коди

Одним із способів підвищення якості передавання цифрових сигналів каналами зв'язку з помилками є застосування коректувальних кодів, що дозволяють виявляти чи виправляти помилки, які виникають у каналах зв'язку.

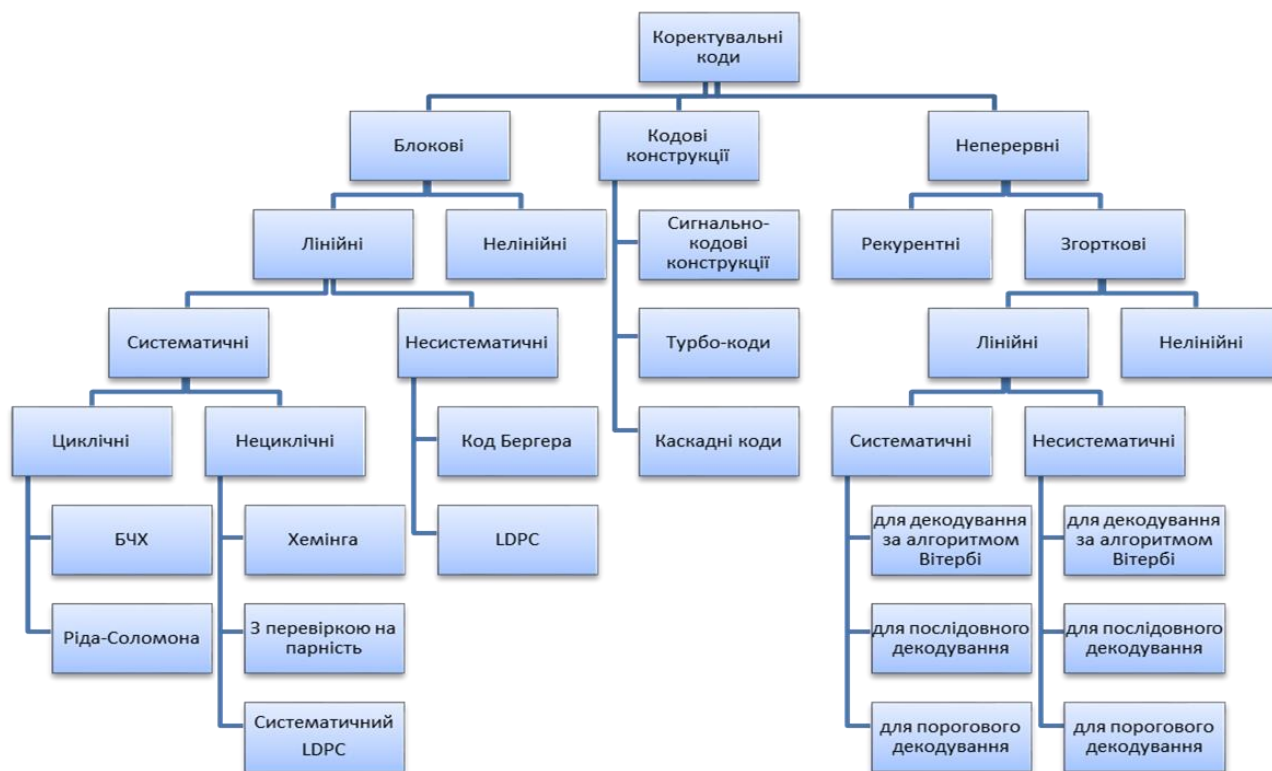


Рисунок 1 – Класифікація коректувальних кодів

Для опису коректувальних кодів вводяться наступні параметри.

Загальний принцип побудови коректувальних кодів досить простий. Із загального числа $M = 2^n$ можливих кодових комбінацій довжини n використовуються для передавання не всі, а тільки $M_0 = 2^k$ ($M_0 < M$). Використовувані кодові комбінації називаються *дозволеними*. Інші $M - M_0$ комбінацій вважаються *забороненими*, тобто вони не можуть подаватися в канал зв'язку, їхня поява на виході каналу свідчить про наявність помилок. Таким чином, завдяки наявності заборонених кодових комбінацій виникає можливість виявлення помилок, що виникають під час передавання. Отже, будь-який коректувальний код є кодом з надмірністю (каналом зв'язку передається $r = n - k$ надлишкових символів у кожній кодовій комбінації).

Віддаль Хеммінга d_{ij} показує степінь відмінності i -ї й j -ї кодових комбінацій. Для будь-яких двох двійкових кодових комбінацій віддаль дорівнює числу незбіжних у них символів.

Кодова віддаль d_{\min} – це мінімальна віддаль Хеммінга для заданого коду. Перебравши всі можливі пари дозволених кодових комбінацій і обчисливши для них віддалі d_{ij} , необхідно знайти серед них мінімальну, тобто $d_{\min} = \min d_{ij}$.

Швидкість коду $R_{\text{код}}$ показує відносне число інформаційних символів k у кодових комбінаціях довжини n і обчислюється $R_{\text{код}} = k/n$.

Здатність коду коректувати визначається кратністю помилок, що виявляються $q_{\text{вияв}}$, і кратністю помилок, що виправляються $q_{\text{вип}}$.

Кратність помилок, що гарантовано виявляються, $q_{\text{вияв}}$ – число помилок у кодовій комбінації, що гарантовано виявляються під час декодування, визначається: $q_{\text{вияв}} < d_{\text{мін}}$.

Кратність помилок, що виправляються, $q_{\text{вип}}$ – число помилок у кодовій комбінації, що виправляються під час декодування, визначається: $q_{\text{вип}} < d_{\text{мін}}/2$.

Призначення кодера і декодера полягає в наступному. На вхід кодера надходить комбінація простого коду \mathbf{A}_i довжини k , кодер перетворює її в комбінацію коректувального коду \mathbf{B}_i довжини n відповідно до правил кодування, причому, $n > k$. На вхід декодера надходить комбінація довжини n з каналу зв'язку:

$$\hat{\mathbf{B}}_j = \mathbf{B}_i \oplus \mathbf{E}, \quad (1)$$

де \mathbf{E} – комбінація помилок. Наприклад, $\mathbf{B}_i = 101000$; нехай помилка відбулася в другому і третьому символах, тоді $\mathbf{E} = 011000$, а $\hat{\mathbf{B}}_j = 110000$.

Залежно від здатності коду коректувати і мети його застосування декодер коректувального коду може працювати в режимі виявлення або в режимі виправлення помилок. У режимі виявлення помилок декодер аналізує: комбінація $\hat{\mathbf{B}}_j$ дозволена чи заборонена? Якщо комбінація $\hat{\mathbf{B}}_j$ дозволена, то декодер відповідно до правила декодування формує на своєму виході комбінацію \mathbf{A}_j довжини k . Якщо комбінація $\hat{\mathbf{B}}_j$ заборонена, то вона бракується декодером, на виході декодера комбінація відсутня, а на виході сигналу помилки (рис. 1) з'являється певний сигнал (наприклад, "1"). У режимі виправлення помилок декодер замість забороненої комбінації $\hat{\mathbf{B}}_j$ декодує найближчу до неї дозволена кодову комбінацію відповідно до правила декодування і видає комбінацію довжини k .

Найбільше поширення в системах передавання дістали систематичні коректувальні коди, кодові комбінації яких містять k інформаційних символів (символи комбінації простого коду, що надійшла на вхід кодера) і $r = n - k$ додаткових символів, сформованих кодером з інформаційних символів. У випадку лінійних кодів додаткові символи є лінійною комбінацією інформаційних символів.

Коректувальні коди з кодовою віддаллю $d_{\text{мін}} = 3$, що дозволяють під час декодування виправляти однократні помилки, називають кодами Хеммінга. Найбільш зручним для опису процесів кодування й декодування лінійних блокових кодів є матричний спосіб. Так, кодування систематичним кодом (n, k) , що полягає у введенні в кодові комбінації додаткових символів, описується матричним співвідношенням

$$\mathbf{A} \cdot \mathbf{G} = \mathbf{B}, \quad (2)$$

де $\mathbf{A} = (a_1 \ a_2 \ \dots \ a_k)$ – матриця-рядок розміром k , що відповідає комбінації простого коду на вході кодера;

$\mathbf{B} = (b_1 \ b_2 \ \dots \ b_k \ b_{k+1} \ \dots \ b_n)$ – матриця-рядок розміром n , що відповідає комбінації коректувального коду на виході кодера;

$$\mathbf{G} = \begin{matrix} \underbrace{\hspace{1.5cm}}_{\text{Інформ. матриця}} & \underbrace{\hspace{1.5cm}}_{\text{Матриця-доповнення}} \\ \left(\begin{array}{cccc} 1 & 0 \dots 0 & g_{1,k+1} \dots g_{1,n} \\ 0 & 1 \dots 0 & g_{2,k+1} \dots g_{2,n} \\ \dots & \dots & \dots \\ 0 & 0 \dots 1 & g_{k,k+1} \dots g_{k,n} \end{array} \right) \end{matrix} \quad (3)$$

матриця, що породжує, розміром $k \times n$, елементи якої g_{ij} приймають значення 1 або 0. Матрицю, що породжує, можна розглядати як об'єднання одиничної матриці порядку k (інформаційної матриці) і матриці-доповнення розміром $k \times r$.

З рівняння (2) випливає, що додаткові символи пов'язані з інформаційними символами співвідношенням

$$b_i = a_i, \quad i = 1, \dots, k; \quad b_i = \sum_{j=1}^k a_j g_{ji}, \quad i = k + 1, k + 2, \dots, n. \quad (4)$$

Тут додавання виконується за модулем 2. Таким чином, елементи i -го стовпця матриці, що породжує, визначають i -ий додатковий символ.

Рядки матриці \mathbf{G} повинні задовольняти наступним умовам:

1. Віддаль між будь-якими двома рядками не повинна бути меншою d_{\min} .
2. Кожний рядок повинен містити не менш d_{\min} одиниць.
3. Усі рядки повинні бути лінійно незалежні, тобто жоден з рядків не можна отримати шляхом підсумовування (за модулем 2) будь-яких інших.

Остання умова для матриці (3) виконується завдяки тому, що перші k стовпців утворюють одиничну матрицю. Отже, при побудові матриці-доповнення необхідно задовольнити умовам 1 і 2.

Якщо кодова віддаль $d_{\min} = 3$, то умови 1 і 2 для рядків матриці-доповнення можна сформулювати так.

1. Віддаль між будь-якою парою рядків повинна бути не меншою 1 (тобто досить, щоб рядки були незбіжними).
2. Кожний рядок повинен містити не менше двох одиниць.

Оскільки ці умови легко здійсненні, то й матриця, що породжує, може бути легко побудована. Наприклад, для коду (7, 4) матриця, що породжує, має вигляд

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (5)$$

Процес декодування полягає в обчисленні синдрому і в матричній формі записується так:

$$\mathbf{H} \cdot \hat{\mathbf{V}} = \mathbf{S}, \quad (6)$$

де \mathbf{H} – перевірна матриця розміром $r \times n$;

$\hat{\mathbf{V}}$ – матриця-стовпець розміру n , що відповідає кодовій комбінації на вході декодера;

\mathbf{S} – матриця-стовпець розміру r .

Із співвідношення (6) випливає, що обчислення синдрому зводиться до r перевірок

$$s_i = \sum_{j=1}^n h_{ij} \cdot \hat{b}_j, \quad i = 1, 2, \dots, r, \quad (7)$$

тобто, i -а перевірка визначається елементами i -го рядка перевірної матриці.

Перевірна матриця будується в такий спосіб. Спочатку будується матриця – транспонована матриця-доповнення, а потім до неї праворуч приписується одинична матриця:

$$\mathbf{H} = (h_{ij}) = \begin{pmatrix} g_{1,k+1} & g_{2,k+1} & 1 & 0 \dots 0 \\ g_{1,k+2} & g_{2,k+2} & 0 & 1 \dots 0 \\ \dots & \dots & \dots & \dots \\ g_{1,n} & g_{2,n} & 0 & 0 \dots 0 \end{pmatrix}. \quad (8)$$

Наприклад, для коду (7,4) з матрицею, що породжує, (4) перевірна матриця має вигляд:

$$\mathbf{H} = \left(\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right), \quad (9)$$

і перевірки декодера зводяться до наступних виразів:

$$\begin{aligned} s_1 &= \hat{b}_1 \oplus \hat{b}_3 \oplus \hat{b}_4 \oplus \hat{b}_5, \\ s_2 &= \hat{b}_1 \oplus \hat{b}_2 \oplus \hat{b}_3 \oplus \hat{b}_6, \\ s_3 &= \hat{b}_2 \oplus \hat{b}_3 \oplus \hat{b}_4 \oplus \hat{b}_7. \end{aligned} \quad (10)$$

На основі (10) легко скласти таблицю ненульових синдромів (табл. 1).

Таблиця 1 – Таблиця синдромів коду (7,4)

Синдром $s_1s_2s_3$	100	010	001	110	101	011	111
Помилковий символ	\hat{b}_5	\hat{b}_6	\hat{b}_7	\hat{b}_1	\hat{b}_4	\hat{b}_2	\hat{b}_3

Останні чотири колонки табл. 1 указують, як по виду синдрому визначається номер помилкового символу (додаткові символи не підлягають виправленню, тому що після виконання перевірок (10) вони не використовуються).

Правило формування додаткових символів (4) є основою алгоритму роботи кодера:

1. На вхід кодера надходять k інформаційних символів у паралельному коді або в послідовному коді (в останньому випадку потрібен регістр зсуву довжиною k).

2. Обчислюються $r = n - k$ додаткових символів за допомогою суматорів за модулем 2.

3. k інформаційних і r додаткових символів надходять на вихід кодера в паралельному або послідовному коді (в останньому випадку потрібен перетворювач паралельного коду в послідовний).

Алгоритм роботи декодера наступний:

1. На вхід декодера надходять n символів з каналу зв'язку.

2. На основі співвідношення (10) обчислюється синдром.

3. Аналізатор синдрому, побудований на основі таблиці синдромів, формує сигнали для виправлення помилок інформаційних символів.

4. Виконується виправлення помилкових символів, що полягає в їхній інверсії й виконується шляхом додавання за модулем 2 до символів, що виправляються, одиниць (нехай \hat{b}_i – деякий символ, тоді $\hat{b}_i \oplus 1 = \overline{\hat{b}_i}$).

5. На вихід декодера після виправлення помилки надходить комбінація простого коду з k символів.

Серед систематичних блокових кодів найбільш широке поширення одержали циклічні коди, завдяки простоті побудови кодера і декодера. Для опису циклічних кодів виявилось зручним представляти кодові комбінації поліномами – наприклад, комбінації $A_i = 10111$ відповідає поліном $a_i(x) = x^4 + x^2 + x + 1$ (символи кодової комбінації є коефіцієнтами при відповідних степенях фіктивної змінної x , причому символу, що записується першим, відповідає найбільш висока степінь x).

Будь-який циклічний код задається не тільки числами n і k , але і *твірним поліномом* $g(x)$ степеня r . Циклічним (n, k) кодом називається код, усі комбінації якого представляються поліномами степеня $n - 1$ і менше, що діляться без залишку на твірний поліном. У табл. 2 наведені твірні поліноми для $r = 3, 4$ і 5 .

Таблиця 2 – Поліноми, що породжують

r	$g(x)$
3	$x^3 + x^2 + 1$ $x^3 + x + 1$
4	$x^4 + x^3 + 1$ $x^4 + x + 1$
5	$x^5 + x^4 + x^2 + 1$

Робота кодера циклічного коду (n, k) зводиться до наступного. Нехай $a(x)$ – поліном, що відповідає комбінації простого коду, яка надійшла на вхід кодера. Поліном $a(x) \cdot x^r$ відповідає додаванню до вхідної комбінації r нулів праворуч. Виконується ділення полінома $a(x) \cdot x^r$ на твірний поліном $g(x)$ з метою визначення залишку від ділення $r(x)$. Залишок від ділення $r(x)$ і є додатковими символами. Поліном, що відповідає вихідній комбінації кодера, визначається як:

$$b(x) = a(x) \cdot x^r + r(x), \quad (11)$$

тобто r нулів, введених у комбінацію, заміщаються комбінацією, що відповідає залишку від ділення.

Легко показати, що поліном $b(x)$ ділиться без залишку на поліном $g(x)$:

$$\frac{b(x)}{g(x)} = \frac{a(x)x^r}{g(x)} + \frac{r(x)}{g(x)} = p(x) + \frac{r(x) \oplus r(x)}{g(x)} = p(x),$$

де $p(x)$ – ціла частина від ділення $a(x) \cdot x^r / g(x)$.

Слід пам'ятати, що додавання поліномів виконується за правилом додавання за модулем два (mod 2) коефіцієнтів при однакових степенях x .

Розглянемо приклад формування кодової комбінації коду (10, 5) з твірним поліномом $g(x) = x^5 + x^4 + x^2 + 1$. Нехай $A_i = 10110$, тоді $a_i(x) = x^4 + x^2 + x$, і $a_i(x) \cdot x^5 = x^9 + x^7 + x^6$. Виконаємо ділення з метою визначення залишку.

$$\begin{array}{r} x^9 + x^8 + x^7 + x^6 \\ \underline{x^9 + x^8 + x^7 + x^4} \\ x^8 + x^7 + x^5 + x^3 \\ \underline{x^8 + x^7 + x^5 + x^3} \\ x^5 + x^4 + x^3 \\ \underline{x^5 + x^4 + x^2 + 1} \\ x^3 + x^2 + 1 = r(x) \end{array}$$

Відповідно до (11)
 $b_i(x) = x^9 + x^7 + x^6 + x^3 + x^2 + 1$
 чи $B_i = 1011001101$.

У декодері циклічного коду виконується ділення отриманої комбінації $\hat{b}(x)$ на твірний поліном. Поліноми переданої комбінації $b(x)$, комбінації на вході декодера

$\hat{b}(x)$ і помилки $e(x)$ зв'язані співвідношенням, аналогічним (1): $\hat{b}(x) = b(x) \oplus e(x)$. Результат ділення $\hat{b}(x)$ на твірний поліном можна представити

$$\frac{\hat{b}(x)}{g(x)} = \frac{b(x)}{g(x)} \oplus \frac{e(x)}{g(x)} = p(x) \oplus \frac{e(x)}{g(x)} = p(x) \oplus v(x) \oplus \frac{s(x)}{g(x)},$$

звідки випливає, що залишок від ділення $s(x)$ залежить тільки від полінома помилки і не залежить від переданої комбінації ($v(x)$ – ціла частина від ділення $e(x)$ на $g(x)$). Залишок від ділення $s(x)$ є синдромом. Ненульовий залишок свідчить про те, що комбінація на вході декодера є забороненою (помилковою). Якщо декодер працює в режимі виправлення помилок, то номер помилкового символу (чи номери помилкових символів) визначається на основі аналізу

Таблиця 3 – Синдроми однократних помилок

Поліном помилки $e(x)$	Синдром $s(x)$
x^9	$x^4 + x^3 + 1$
x^8	$x^4 + x^2 + x$
x^7	$x^3 + x + 1$
x^6	$x^4 + x^3 + x^2 + x + 1$
x^5	$x^4 + x^2 + 1$
x^4	x^4
x^3	x^3
x^2	x^2
x	x
1	1

синдрому. Для коду (10, 5) з твірним поліномом $g(x) = x^5 + x^4 + x^2 + 1$ складемо таблицю синдромів для всіх однократних помилок, виконуючи ділення $e(x)$ на $g(x)$ і фіксуєючи в табл. 3 тільки залишки від ділення.

З табл. 3 випливає, що у випадку однократних помилок усі синдроми різні, тому кожен синдром однозначно вказує номер помилкового символу. Виправлення помилки декодером здійснюється за допомогою дешифратора,

побудованого відповідно до табл. 3, й інвертора, що виконує інверсію помилкового символу.

Розглянутий код (10, 5) має кодову віддаль $d_{\min} = 4$ і дозволяє виправляти тільки однократні помилки.

Тема 8. Згорткові коректувальні коди

Визначення та опис згорткових кодів. Як відомо, у разі блокових коректувальних кодів вхідна послідовність інформаційних символів (надалі бітів) розбивається на окремі блоки, що надалі кодуються незалежно один від одного. Таким чином, закодована послідовність є послідовністю незалежних кодових слів (комбінацій) однакової довжини.

Для неперервних кодів ситуація інша. Процес кодування провадиться неперервно і символи на виході кодера (так звані кодові символи) створюють одне напівнескінченне кодове слово (кодову комбінацію).

Згорткові коди є окремим випадком неперервних кодів. Назву вони отримали через таку властивість – послідовність кодових символів на виході кодера обчислюється як математична операція цифрової згортки вхідних інформаційних бітів з імпульсною характеристикою кодера.

Структура кодера згорткового коду (ЗК) і процес кодування (декодування) задаються поліномами, що породжують, $g^{(i)}$, де $i = 1, 2, \dots, n$. Як правило, поліноми записуються скорочено, позначаючи три двійкових коефіцієнти поліному як одну вісімкову цифру. Наприклад:

$$g^{(1)} = 7 \text{ означає } g^{(1)} = 111, \text{ тобто } g^{(1)} = D^2 + D + 1 \text{ чи } 1 + D + D^2;$$
$$g^{(2)} = 21 \text{ означає } g^{(2)} = 010101, \text{ тобто } g^{(2)} = D^4 + D^2 + 1 \text{ чи } 1 + D^2 + D^4,$$

де D – затримка на один такт.

Основні параметри згорткових кодів

Швидкість коду визначається як $R_{\text{код}} = k/n$, де k – кількість входів кодера, n – кількість виходів кодера. Швидкість коду показує, що на k вхідних інформаційних біт кодер видає n кодових символів.

Довжина кодового обмеження (ДКО) ν характеризує пам'ять кодера і дорівнює сумарній кількості комірок пам'яті, що містить кодер.

Імпульсна характеристика кодера – відгук кодера ЗК на інформаційну послідовність "1000000000000000...".

Вільна віддаль коду d_f – мінімальна віддаль Хеммінга між послідовністю нульових кодових символів й усіма іншими послідовностями кодових символів. Вільна віддаль d_f характеризує здатність ЗК виправляти помилки, а саме, кількість помилок $q_{\text{вип}}$, що виправляються декодером на довжині $N = (5 \dots 6)\nu$ послідовності кодових символів, що декодується. Співвідношення між d_f і $q_{\text{вип}}$ таке ж саме, як і для кодової віддалі блокового коректувального коду:

$$d_f \geq 2 q_{\text{вип}} + 1. \quad (1)$$

Кодер згорткового коду. Кодер згорткового коду містить тактований регістр пам'яті для збереження інформаційних символів, число яких дорівнює степені поліному, і перетворювач вхідної інформаційної послідовності у вихідну кодову послідовність (суматори). Структурна схема кодера ЗК (7, 5) показана на рис. 1. Швидкість цього коду $R_k = 1/2$. Кодер містить регістр зсуву з трьома комірками пам'яті D та два суматора за модулем два \oplus . Входи

суматорів за модулем два з'єднані з тими розрядами регістра, в яких

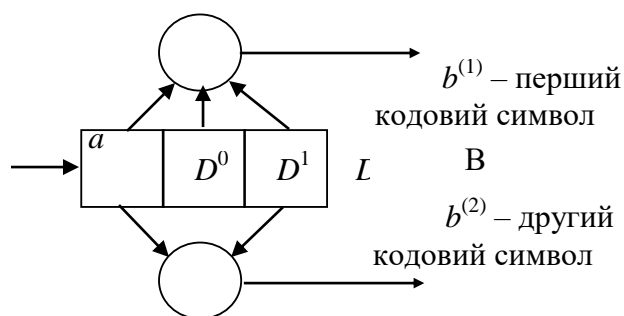


Рисунок 1 – Структурна схема кодера згорткового коду (7,5)

коефіцієнти поліномів, що породжують, дорівнюють одиниці.

Інформаційні біти a надходять до входу регістра. На кожний такт на виходах суматорів за модулем два утворюються кодові символи $b^{(1)}$ і $b^{(2)}$, тобто на один інформаційний біт на виході буде два кодових символи.

Для математичного опису згорткового кодування, тобто обчислення цифрової згортки, використовується декілька методів: графічний зв'язок, векторне подання, поліноми зв'язку, діаграма станів, деревовидна та решітчаста діаграми. Найбільш наочною є решітчаста діаграма, яка розглядається нижче.

Решітчаста діаграма ЗК

Решітчастою діаграмою ЗК (коротко – решіткою) називається орієнтований граф зі структурою "комірок", що періодично повторюються. Кожна комірка складається з колонок з однакового числа **вузлів** (вершин), з'єднаних **вітками** (ребрами) (рис. 2, а). Між процедурою кодування ЗК і решіткою є однозначна відповідність, яка задається такими правилами:

- кожний вузол відповідає внутрішньому стану кодера, як правило, це вміст v останніх комірок пам'яті в регістрі зсуву;
- кожна вітка відображає перехід кодера з одного стану в інший після надходження в кодер нового інформаційного символу: верхня вітка відповідає 0, а нижня – 1;
- над кожною віткою позначаються вихідні кодові символи, що з'являються на виході кодера, коли він переходить з одного стану в інший, які зв'язує дана вітка;
- послідовність віток називається **шляхом** на решітці, який визначається послідовністю інформаційних бітів і однозначно надає відповідну їй послідовність кодових символів.

Так, для кодера ЗК (7, 5), зображеного на рис. 1, решітка буде мати чотири стани (00, 10, 01 та 11) і її вид показано на рис. 2. Наочне правило обчислення вихідних кодових символів віток ілюструє рис. 3 для початкового стану кодера 00 і надходження до входу інформаційних біт 0 та 1. Обчислення вихідних кодових символів інших віток провадиться аналогічно для інших станів кодера.

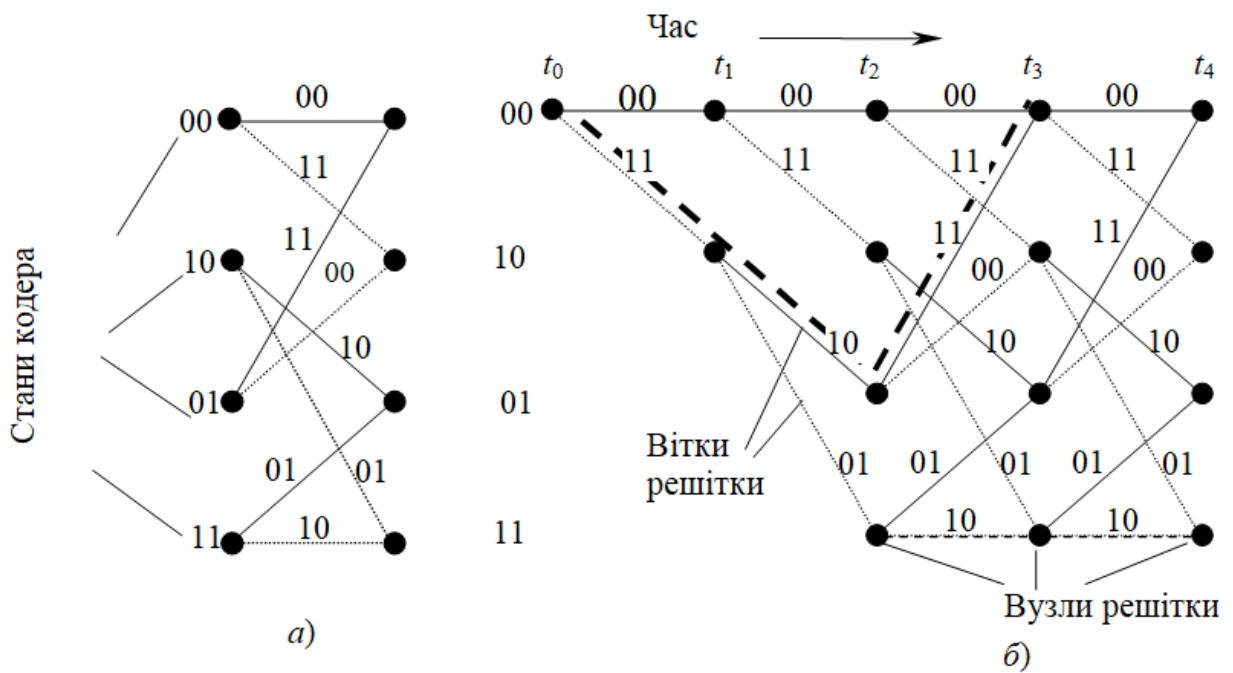


Рисунок 2 – Решітчаста діаграма кодера згорткового коду (7,5):
 а) – комірка решітки; б) – розгортка решітки за часом

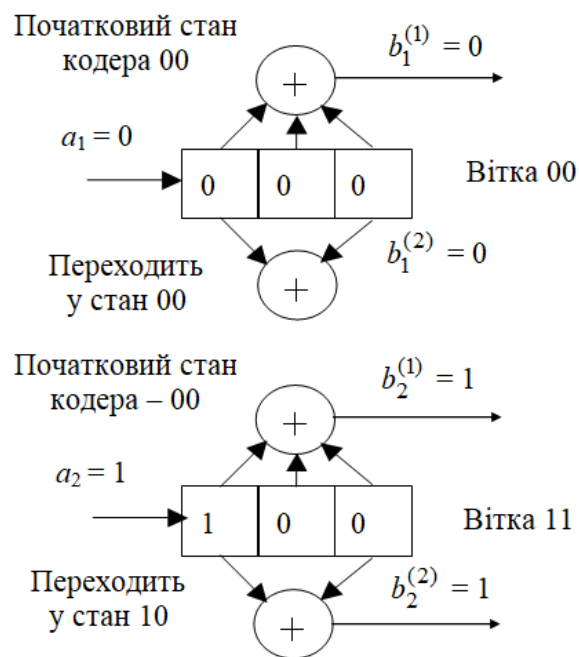


Рисунок 3 – Згорткове кодування для коду (7,5)

За решітчастою діаграмою *вільна віддаль* коду d_f обчислюється як вага (кількість одиниць) найкоротшого ненульового шляху, що починається і закінчується в нульовому стані (на рис. 2 для ЗК (7, 5) – це напівжирний пунктирний шлях).

Приклад 1. Для кодера ЗК (7, 5), зображеного на рис. 1, знайти послідовність кодових символів, якщо послідовність інформаційних біт $\mathbf{a} = 01101000$. Прийняти, що в початковому стані регістр містить нулі.

Рішення. Послідовність кодування детально описана в табл. 1, за якою на рис. 4 зображена решітчаста діаграма зі шляхом кодування, під яким розуміють послідовність проходження віток під час кодування.

Таблиця 1 – Процес кодування послідовності інформаційних бітів 01101000 ЗК (7, 5)

t_k	Інформаційні біти	Вміст регістра кодера	Стан кодера в момент t_k	Стан кодера в момент t_{k+1}	Вихідні кодові символи в момент t_k	
					$b^{(1)}$	$b^{(2)}$
0	0	000	00	00	0	0
1	1	100	00	10	1	1
2	1	110	10	11	0	1
3	0	011	11	01	0	1
4	1	101	01	10	0	0
5	0	010	10	01	1	0
6	0	001	01	00	1	1
7	0	000	00	00	0	0

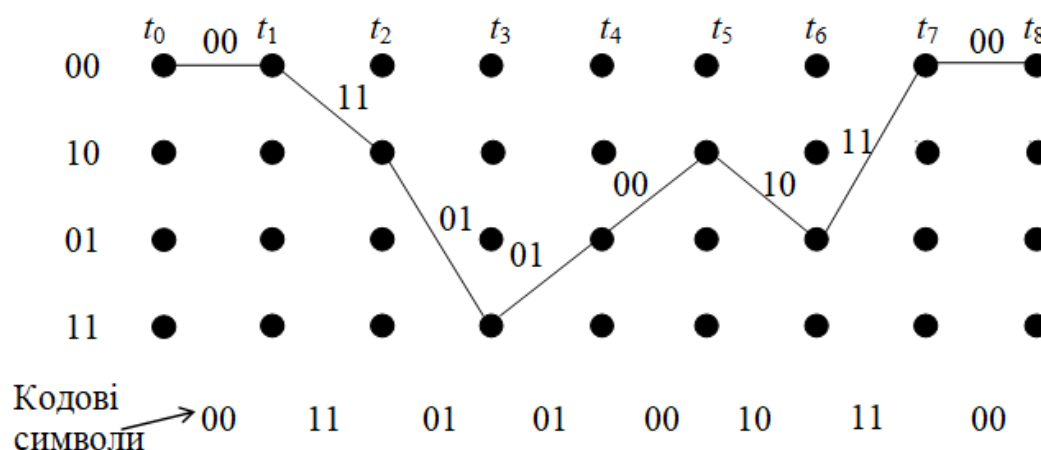


Рисунок 4 – Решітчаста діаграма кодера ЗК (7, 5) та шлях кодування на ній

Декодування згорткових кодів за алгоритмом Вітербі.

Цей алгоритм є найбільш поширеним. Алгоритм Вітербі передбачає на кожному k -му тактовому інтервалі перелічені нижче етапи декодування.

1. **Обчислення** віддалі між послідовністю символів на вході декодера і можливими послідовностями символами, що відповідають усім віткам решітки, які входять в кожний стан у момент t_k . Ця віддаль називається **метрикою вітки**.

2. **Побудову** решітчастої діаграми декодера, що аналогічна решітчастій діаграмі кодера, на якій зображують усі можливі вітки з їх метриками. Число віток і відповідних шляхів на решітці зростає при збільшенні кількості комірок решітки, що розглядаються (так звана **глибина простежування**, яка залежить від об'єму пам'яті декодера і має значення до 10 ДКО).

3. **Проріджування** решітки на кожному кроці її побудови. Проріджування решітки – це процедура вилучення одного із двох шляхів, що входять у кожний стан декодера, за правилом: вилучається шлях з більшою метрикою,

залишається – з меншою метрикою (якщо метрики однакові, то вилучається будь-який). Під *метрикою шляху* (чи *метрикою стану* решітки M_{ij} , де ij – номер стану декодера) розуміють сумарну метрику віток, якими проходить конкретний шлях на момент t_k до конкретного стану. Проріджування необхідне для зменшення числа шляхів декодера і відповідно ємності пам'яті.

4. *Знаходження оптимального* шляху по решітці після закінчення простежування і прийняття рішення про передані інформаційні біти, тобто виконання декодування. Оптимальним є шлях з найменшою метрикою і його називають *шляхом, що вижив*. Декодування проводиться за шляхом, що вижив: якщо він проходить по верхній вітці решітки, то інформаційний біт – "0", нижній – "1".

Процедуру декодування за алгоритмом Вітербі розглянемо на конкретному прикладі для двійкового симетричного каналу, тобто демодулятор видає "жорсткі" рішення у вигляді послідовності кодових символів \hat{b} (можливо з помилками).

Приклад 2. Провести декодування за алгоритмом Вітербі послідовності кодових символів $\hat{b} = 00\ 11\ 00\ 01\ 00\ 10$. Згортковий код (7, 5). Прийняти, що на початку декодування реєстр декодера знаходиться в нульовому стані.

Примітка. Послідовність на вході декодера \hat{b} – це фрагмент послідовності кодових символів кодера, розглянутого у прикладі 1.

Рішення. Перші два етапи декодування за алгоритмом Вітербі об'єднані під час побудови решітчастої діаграми декодера, наведеної на рис. 5 для моментів часу $t_0 - t_4$. Метрика віток на кожному k -му тактовому інтервалі обчислюється як віддаль Хеммінга між парою кодових символів на вході декодера $\hat{b}^{(1)}\hat{b}^{(2)}$ і кодовими символами віток решітки. Обчислена віддаль (0, 1 чи 2) показана біля кожної вітки на рис. 5. Із рис. 5 випливає, що з моменту t_2 число віток дорівнює восьми в кожній комірці решітки, а число можливих шляхів зростає експоненційно зі збільшенням глибини простежування. Процедура 3) проріджування решітки наведена на рис. 6 для моментів часу t_3 та t_4 .

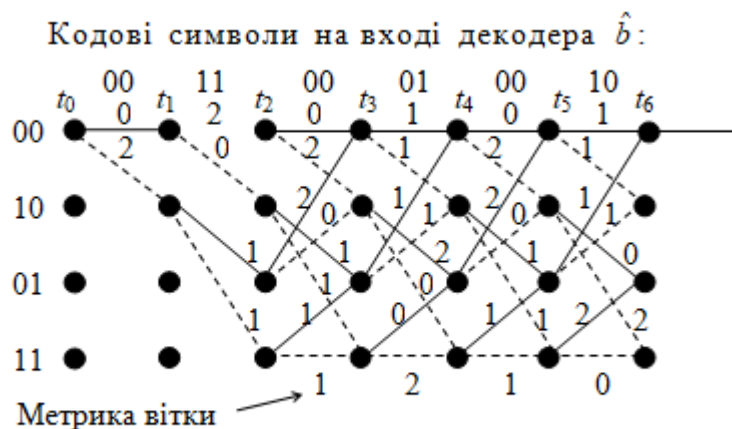


Рисунок 5 – Фрагмент решітчастої діаграми декодера ЗК (7, 5)

Тема 9. Ефективність систем передавання

Блокові коди. Найбільш поширеними серед блокових кодів є коди Боуза-Чоудхурі-Хоквінгема (БЧХ) – клас блокових лінійних систематичних циклічних кодів, що виправляють кратні помилки, тобто дві і більше ($d_{\min} \geq 5$). Теоретично коди БЧХ можуть виправляти довільну кількість помилок, але при цьому істотно збільшується довжина кодових комбінацій, що призводить до збільшення затримки передавання даних і ускладнення алгоритму декодування.

Якщо коректувальний код повинен забезпечувати виправлення помилок кратності $q_{\text{вип}}$, то необхідна кодова віддаль d_{\min} блокового коду (n, k) визначається за формулою

$$d_{\min} = 2q_{\text{вип}} + 1. \quad (1)$$

Основні параметри коду (n, k) зв'язані співвідношенням

$$k = n - m q_{\text{вип}}, \quad (2)$$

де m – найменше ціле, що задовольняє умові $2^m \geq n$.

Синдромне декодування. Сьогодні для блокових коректувальних кодів (n, k) синдромне декодування є основним для виявлення та виправлення помилок. Синдромний метод виправлення помилок базується на простому правилі: за синдромом кодової комбінації визначається місцезнаходження помилкових символів. Тому під синдромом кодової комбінації розуміють результат обчислення декодером за певними правилами числа $s(s_1, s_2, \dots, s_r)$, $r = n - k$, яке свідчить про наявність помилкових символів і визначає їх розміщення (конфігурацію) у кодовій комбінації. У разі двійкових кодів синдром записується у двійковій системі числення, тобто його розряди s_1, s_2, \dots, s_r приймають значення 0 і 1.

Нульовий синдром вказує на те, що кодова комбінація на вході декодера є дозволеною – помилок немає або вони не виявлені. Ненульовий синдром свідчить про наявність помилок – так використовується синдром при декодуванні з виявленням помилок. При декодуванні з виправленням помилок у декодері за синдромом декодованої кодової комбінації визначаються місця розташування помилок.

Якщо кратність помилок q у кодовій комбінації, що декодується, не перевищує кратність помилок, що виправляються $q_{\text{вип}}$, $q \leq q_{\text{вип}}$, то синдром однозначно визначає місце розташування помилкових символів у кодовій комбінації, і помилки виправляються декодером шляхом інверсії помилкових символів.

Якщо $q > q_{\text{вип}}$, то можливі дві ситуації:

– отриманий обчисленням синдром відповідає певній конфігурації помилок, яка зустрічається при $q \leq q_{\text{вип}}$, в цій ситуації декодер “виправляє” помилки, але виправлення не вірне, і, швидше за все, кількість помилкових символів у кодовій комбінації зростає;

– отриманий обчисленням синдром не відповідає жодній з конфігурацій помилок, які можуть бути при $q \leq q_{\text{вип}}$, в цій ситуації

декодер "відмовляється" від декодування і пропускає таку кодову комбінацію на вихід з помилками.

Таким чином, синдромне декодування двійкових кодів зводиться до обчислення синдрому та пошуку тим чи іншим способом конфігурації помилок, за якою і виправляються помилки. Виправлення помилки в деякому двійковому символі зводиться до його інверсії та виконується шляхом додавання до помилкового символу 1 за модулем 2.

Коректувальна здатність завадостійких кодів. Коректувальна здатність коду характеризується величиною кратності помилок, що виправляються

$$q_{\text{вип}} \leq (d_{\text{min}} - 1)/2 \quad (3)$$

та величиною кратності помилок, що гарантовано виявляються

$$q_{\text{вияв}} \leq d_{\text{min}} - 1, \quad (4)$$

де d_{min} – кодова віддаль коду.

Завадостійкість системи передавання з коректувальним кодом (так же, як і системи передавання без коректувального коду) описується залежністю ймовірності помилки біта від відношення сигнал/шум на вході демодулятора. Під відношенням сигнал/шум розуміють $h_6^2 = E_6/N_0$ – відношення енергії сигналу, що витрачається на передавання одного біта, до питомої потужності шуму. Використаємо позначення: p – ймовірність помилки біта на виході демодулятора; p_d – ймовірність помилки біта на виході декодера.

Приклад залежностей для системи передавання без завадостійкого кодування $p = f(h_6^2)$ та для системи передавання з кодом БЧХ (1023, 733) $p_d = f(h_6^2)$ наведено на рис. 2.

Завадостійкість системи передавання без коректувального коду детально обговорювалась в лабораторній роботі 3.4. В системі передавання з коректувальним кодом ймовірність помилки біта p на виході демодулятора для модуляції ФМ-2 визначається

$$p = Q(\sqrt{2R_{\text{код}} h_6}), \quad (3)$$

де

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} \exp(-\frac{t^2}{2}) dt - \quad (4)$$

гауссова Q -функція;

$R_{\text{код}} = k/n$ – швидкість коду.

При використанні коректувального коду залежність ймовірності помилки символів на виході декодера від ймовірності помилки на його вході наступна:

$$p_d = \frac{d_{\text{min}}}{n} \sum_{q=q_{\text{вип}}+1}^n C_n^q p^q (1-p)^{n-q}. \quad (5)$$

У сумі (5) для розрахунків при $p \ll 1$ достатньо використати тільки перший доданок та виключити множник $(1-p)^{n-q}$, який близький до одиниці:

$$P_d = \frac{d_{\min}}{n} C_n^{q_{\text{вип}}+1} p^{q_{\text{вип}}+1}. \quad (6)$$

Енергетичний виграш кодування. Узагальненою характеристикою завадостійкості коректувальних кодів у конкретній системі передавання є енергетичний виграш кодування (ЕВК), що показує різницю відношень сигнал/шум (дБ) на вході демодулятора без коректувального коду h_{61}^2 і з коректувальним кодом h_{62}^2 за однакової ймовірності помилки символів (рис. 2).

$$\text{ЕВК} = h_{61}^2 - h_{62}^2. \quad (7)$$

Нагадаємо, що відношення h_6^2 визначається як

$$h_6^2 = \frac{E_6}{N_0} = \frac{P_s T_6}{N_0} = \frac{P_s}{N_0 R}, \quad (9)$$

де P_s/N_0 – відношення середньої потужності сигналу до питомої потужності шуму на вході демодулятора;

R – швидкість цифрового сигналу на вході системи передавання.

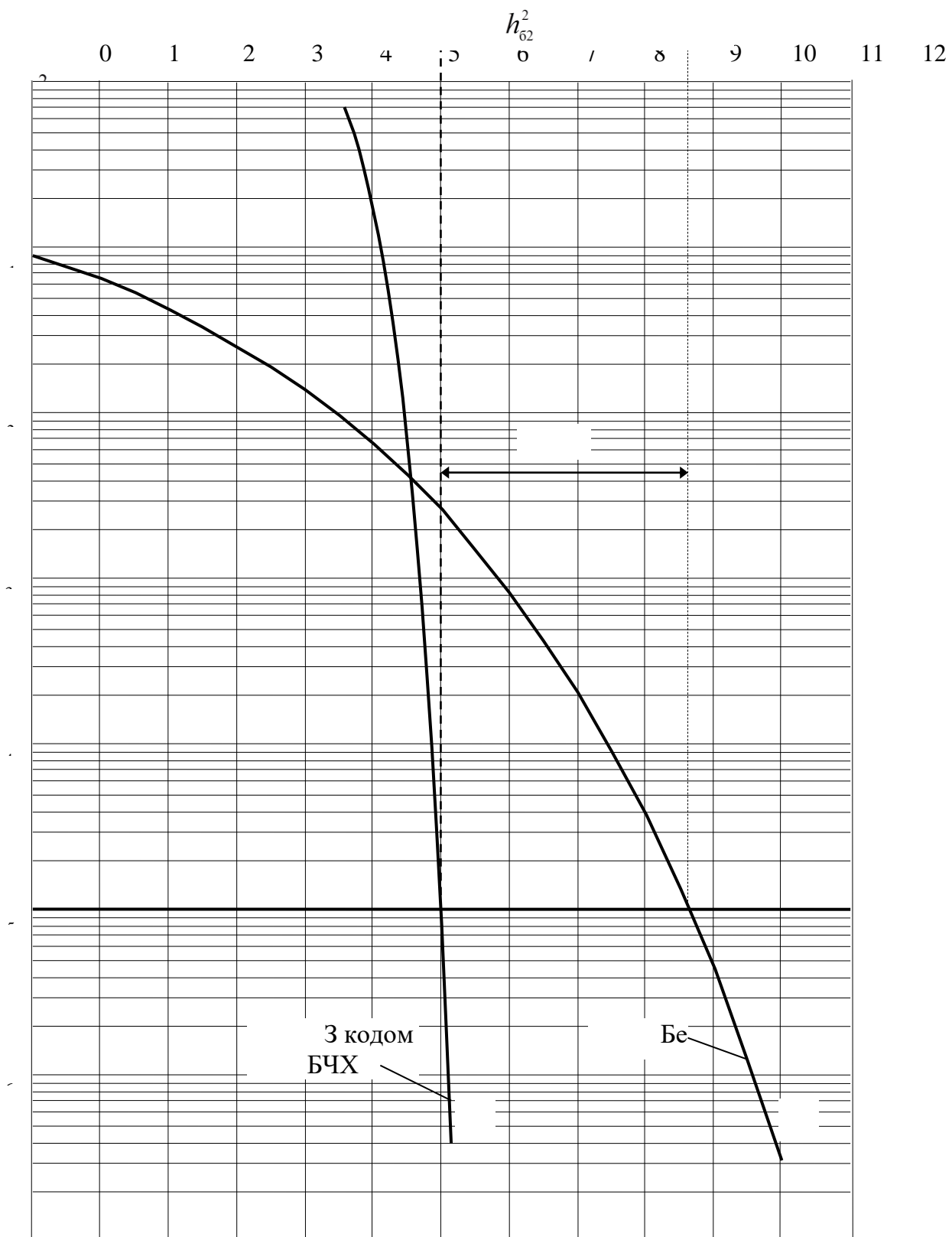


Рисунок 2 – Графіки завадостійкості системи передавання без коректувального коду та з коректувальним кодом

ЕВК можна розрахувати теоретично чи виміряти експериментально. Наприклад, у системі передавання потрібно забезпечити ймовірність помилки символу $p_{\text{доп}} = 10^{-5}$. За відсутності завадостійкого кодування згідно рис. 2

задана ймовірність помилки буде забезпечуватись при відношенні сигнал/шум $h_{61}^2 = 9,6$ дБ, у той час як використання коду БЧХ (1023, 733) дозволяє знизити необхідне відношення сигнал/шум до $h_{62}^2 = 6$ дБ, то відповідно, $ЕВК = 9,6 - 6 = 3,6$ дБ.

Якщо код забезпечує $ЕВК > 0$ (на рис. 2 при $p_{доп} < 4 \cdot 10^{-3}$), то такий коректувальний код дозволяє зменшити необхідне відношення сигнал/шум на вході демодулятора. Якщо ж $ЕВК \leq 0$, то такий коректувальний код не дозволяє зменшити необхідне відношення сигнал/шум на вході демодулятора, його використання тільки погіршує завадостійкість системи передавання.

Залежності $p = f(h_6^2)$ та $p_d = f(h_6^2)$ можуть бути отримані експериментально за методикою, що була описана в лабораторній роботі 3.4.

Методи боротьби з пакетами помилок. Необхідно зауважити, що коди БЧХ здатні ефективно боротися зі статистично незалежними помилками, тобто такими, які взаємно не залежні і, в кінцевому рахунку, загальне число помилкових символів виявляється рівномірно розподіленим по всій послідовності переданих даних (рис. 3, а).

Якщо помилки пакетуються (рис. 3, б) тобто спостерігається поява відразу декількох помилок поспіль, для ефективної боротьби з ними можуть застосовуватися коди БЧХ з перемеженням або блокові коди Ріда-Соломона.

Перемеження являє собою розкид біт в послідовності за певним законом. Найбільш простим методом перемеження є матричне перемеження (рис. 4), при якому після кодера в буфер (матрицю розміром $(X \times Y)$) ведеться запис символів по рядках і зчитування по стовпцях. У декодері навпаки: запис по стовпцях, зчитування по рядках (деперемеження). У результаті перемеження і деперемеження дві помилки, що стоять поруч в послідовності на виході демодулятора виявляються рознесеними на величину розміру буфера X , тобто помилки не створюють пакет. На рис. 3, в показано приклад деперемеження послідовності, заданої на рис. 4, б, деперемечувачем, зображеним на рис. 4.

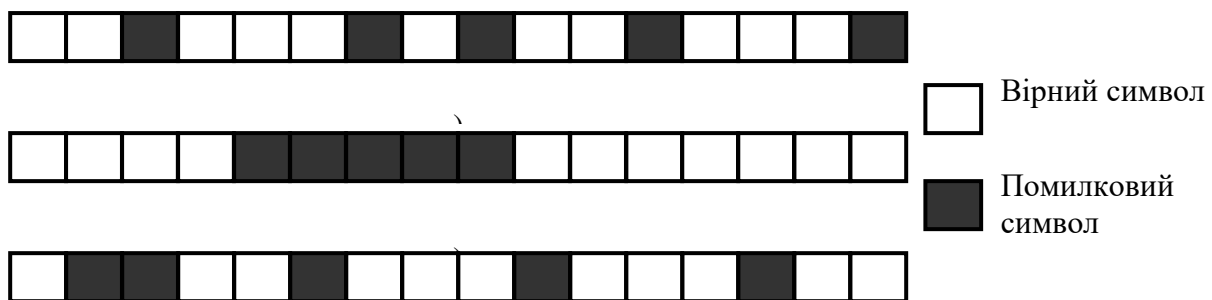


Рисунок 3 – Помилки: а) рівномірно розподілені, б) пакетовані, в) після перемеження послідовності б)

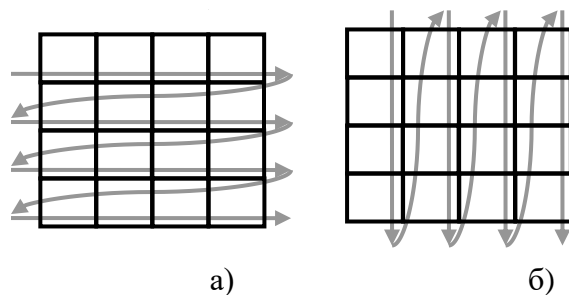


Рисунок 4 – Матричне перемеження

Робота з буфером при перемеженні
 а) порядок запису в буфер,
 б) порядок зчитування з буфера.
 Робота з буфером при деперемеженні
 б) порядок запису в буфер,
 а) порядок зчитування з буфера

Коди Ріда-Соломона – недвійкові циклічні коди, що дозволяють виправляти помилки в блоках даних. Елементами кодових комбінацій є не біти, а групи біт (блоки). Дуже поширені коди Ріда-Соломона, що працюють з байтами. Код Ріда-Соломона є окремим випадком коду БЧХ.

В даний час коди Ріда-Соломона широко використовуються в системах передавання з завадостійким кодуванням, в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень.

Довідкові дані. У табл. 1 наведено параметри коректувальних кодів: n – довжина кодового слова, k – кількість інформаційних біт, $q_{\text{вип}}$ – кількість помилок, що виправляються, C_n^q – число сполучень з n по q , $R_{\text{код}}$ – швидкість коду.

На рис. 6 наведено залежності енергетичного виграшу при використанні циклічних кодів в каналах з ФМ-2 в залежності від довжини блоку n і швидкості коду $R_{\text{код}}$.

Таблиця 1 – Характеристики циклічних кодів

n	k	$q_{\text{вип}}$	$C_n^{q_{\text{вип}}+1}$	$R_{\text{код}}$	n	k	$q_{\text{вип}}$	$C_n^{q_{\text{вип}}+1}$	$R_{\text{код}}$
7	4	1	21	0,57	127	120	1	8001	0,95
15	11	1	105	0,73		113	2	333375	0,89
	7	2	455	0,47		106	3	10334625	0,84
31	26	1	465	0,84		99	4	$2,54 \cdot 10^8$	0,78
	21	2	4495	0,68		92	5	$5,17 \cdot 10^9$	0,72
	16	3	31465	0,52	255	239	2	2731135	0,94
11	4	169911	0,35	231		3	$1,72 \cdot 10^8$	0,91	
63	57	1	1953	0,90		223	4	$8,64 \cdot 10^9$	0,87
	51	2	39711	0,81		215	5	$3,6 \cdot 10^{11}$	0,84
	45	3	595665	0,71	207	6	$1,28 \cdot 10^{13}$	0,81	
	39	4	7028847	0,62	511	493	2	$2,21 \cdot 10^7$	0,96
33	5	67945521	0,57	484		3	$2,81 \cdot 10^9$	0,95	

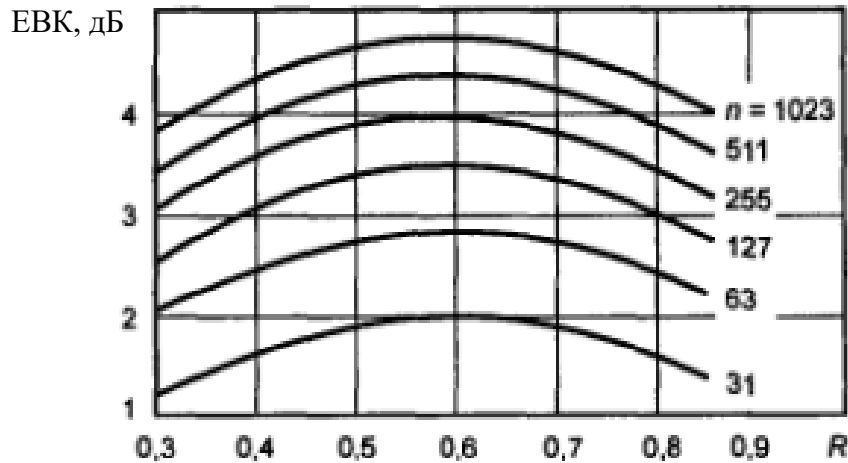


Рисунок 6 – Залежність ЕВК від швидкості коду $R_{\text{код}}$ для різних n

З поміж усіх алгоритмів декодування ЗК найчастіше у сучасних кодеках використовується алгоритм Вітербі. Визначимо завадостійкість, енергетичний вигравш та складність декодування згорткових кодів.

Завадостійкість згорткових кодів. Обчислення ймовірності помилки проводиться в припущенні, що помилкові події відбуваються рідко – це дозволяє скористатися верхньою межею. Верхня межа ймовірності помилки біта на виході декодера визначається виразом:

$$p \leq \sum_{k=df}^{\infty} C_k P_k, \quad (1)$$

де P_k – ймовірності помилкового вибору шляху, що має вагу k ;

C_k – спектр ваг згорткового коду, що показує кількість шляхів з вагою k (табл. 1).

Таким чином, розрахунок ймовірності помилкового відтворення двійкового символу на виході декодера зводиться до визначення спектру ваг і розрахунку ймовірностей помилкового вибору шляху.

Якщо двійкові кодові послідовності відрізняються в k символах, а в каналі зв'язку використовується ФМ-2, то віддаль між шляхами у метриці Евкліда між цими послідовностями $d_k = 2k\sqrt{E_s}$. Враховуючи, що енергія кожного кодового символу $E_s = E_b R_{\text{код}}$, віддаль можна визначити як $d_k = 2k\sqrt{E_b R_{\text{код}}}$.

Тоді ймовірності помилкового вибору шляху можна розрахувати за наступним виразом:

$$P_k = Q\left(\sqrt{2kR_{\text{код}} E_b / N_0}\right) \quad (2)$$

де $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp(-\frac{t^2}{2}) dt$ – гауссова Q -функція;

$R_{\text{код}}$ – швидкість коду;

E_b – енергія, що витрачається на передавання одного інформаційного символу;

N_0 – спектральна густина потужності АБГШ.

Така методика дає достатньо точні результати при $p < 10^{-2}$. Для розрахунків можна скористатися апроксимацією Q -функції.

Таблиця 1 – Параметри оптимальних згорткових кодів зі швидкістю 1/2

№ коду	v	поліноми, що породжують	d_f	Спектр ваг				
				C_d	C_{d+1}	C_{d+2}	C_{d+3}	C_{d+4}
1	2	5, 7	5	1	4	12	32	80
2	3	13, 7 (7, 13)	6	1	4	14	38	102
3	3	15, 17 (17, 15)	6	2	7	18	49	130
4	3	13, 15 (15, 13)	6	4	0	38	0	277
5	3	17, 13 (13, 17)	6	2	8	16	42	180
6	4	23, 35	8	4	12	20	72	225
7	4	31, 33	8	4	12	26	74	205
8	4	25, 37	8	2	0	18	0	148
9	5	53, 75	8	2	36	32	62	332
10	5	61, 73	8	6	0	60	0	469
11	5	43, 75	8	6	0	93	0	534
12	5	45, 73	8	5	0	98	0	446
13	5	71, 73	8	10	23	30	117	356
14	6	133, 171	10	36	0	211	0	1404
15	6	135, 163	10	46	0	332	0	1911
16	7	247, 371	11	2	20	60	148	340

Енергетичний вигравш від застосування кодування. Використання завадостійкого кодування у системах передавання дозволяє знизити вимоги до відношення сигнал/шум E_b/N_0 у каналі, тим самим одержати енергетичний вигравш від застосування кодування (ЕВК). Зменшення необхідного відношення сигнал/шум обчислюють відносно залежності для некодованої ФМ-2. Відношення сигнал/шум, що необхідне для забезпечення заданого значення ймовірності помилки після декодування, зменшується на величину ЕВК, який у залежності від швидкості згорткового коду складає 4...6 дБ для каналу з АБГШ.

Використовуючи значення вільної віддалі згорткового коду можна визначити асимптотичний енергетичний вигравш від застосування кодування (АЕВК) при ймовірності помилки на виході декодера p прямує до 0, що є верхньою межею реального ЕВК при $p \neq 0$ та розраховується за формулою:

$$\text{АЕВК} = 10 \lg(R_{\text{код}} \cdot d_f) \quad (3)$$

де d_f – вільна віддаль в метриці Хеммінга.

Згорткові коди при використанні алгоритму декодування Вітербі забезпечують значно більший енергетичний вигравш в порівнянні з блоковими кодами. Це пояснює широке використання згорткових кодів у телекомунікаційних системах для підвищення завадостійкості. Типовим тут є застосування коду (133, 171), що забезпечує АЕВК = 6,99 дБ при швидкості $R_{\text{код}}$

= 1/2, тобто при двократному розширенні смуги частот кодованого сигналу. Для кодеків такого коду розроблені і серійно випускаються декодери Вітербі у вигляді великих інтегральних схем. АЕВК найбільш розповсюджених ЗК наведені в табл. 2.

Таблиця 2 – Характеристики згорткових кодів

Швидкість $R_{\text{код}}$	Довжина кодового обмеження $v=4$		Довжина кодового обмеження $v=6$	
	Код	АЕВК, дБ	Код	АЕВК, дБ
1/3	25, 33, 37	6,02	133, 145, 175	6,99
1/2	31, 33	5,44	133, 171	6,99
2/3	31, 33, 31	5,23	133, 171, 133	6,02
3/4	25, 37, 37, 37	4,78	135, 163, 163, 163	6,73

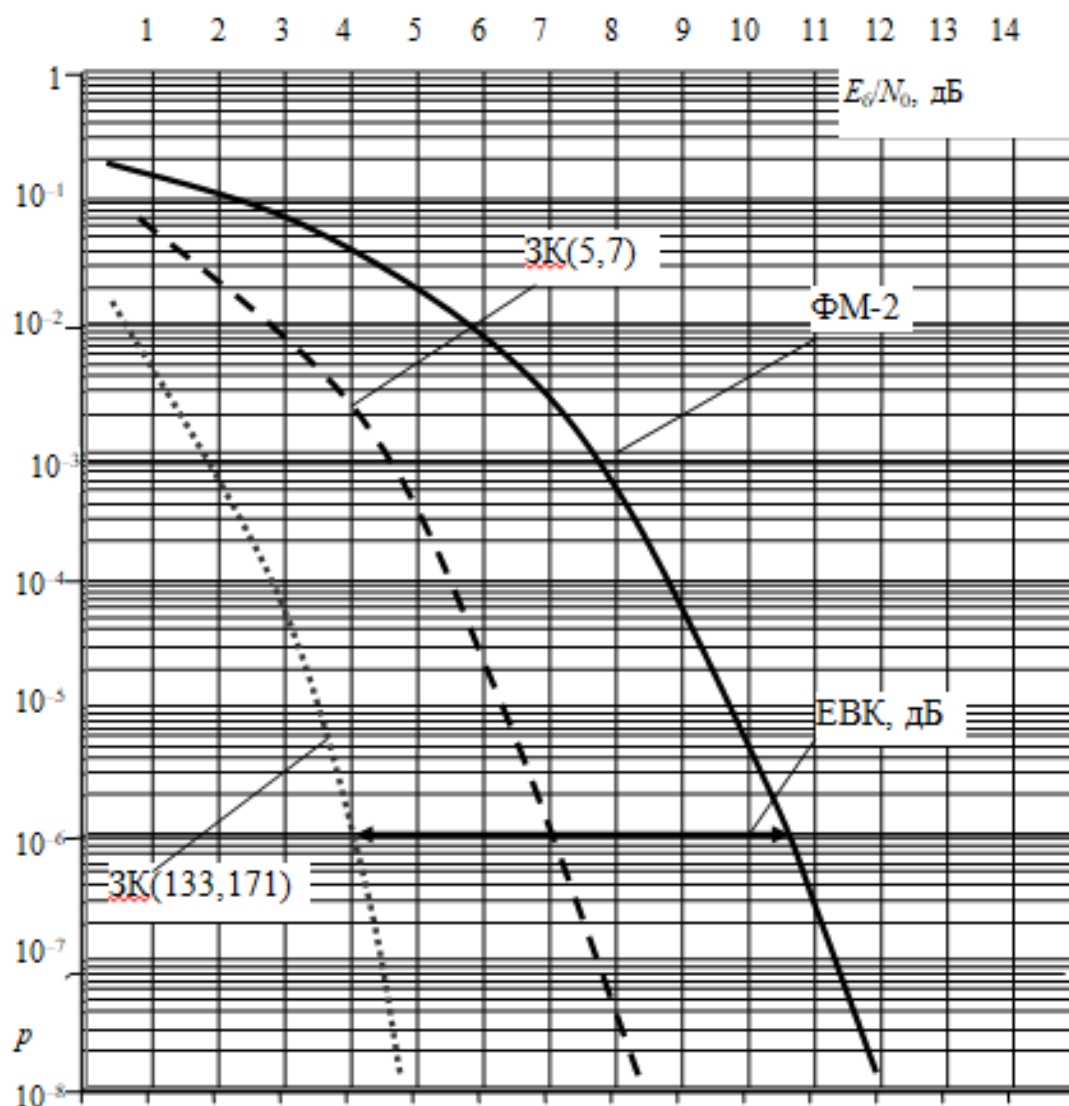


Рисунок 2 – Завадостійкість декодування ЗК

З табл. 2 видно, що АЕВК збільшується зі зменшенням швидкості коду та зі збільшенням довжини кодового обмеження. Тобто, для збільшення АЕВК, а отже і ЕВК, необхідно використовувати більш складні кодеки. На рис. 2

наведені залежності ймовірності помилки на виході декодера від відношення сигнал/шум для систем передавання з ФМ-2 без ЗК, із згортковими кодами (5, 7) та (133, 171), а також вираш від використання коду (133, 171). При ймовірності помилки 10^{-6} ЕВК складає 6,5 дБ.

Складність декодування. Оцінка складності декодування згорткових кодів необхідна для порівняння кодів з наступним відбором з метою їх реалізації. Складність декодування за алгоритмом Вітербі, як правило, оцінюють кількістю віток кодових решіток W , оброблювальних декодером перед прийняттям рішення. Величина W залежить від кількості розгалужених на кожному кроці віток декодування W_0 і від глибини перегляду шляхів L . Для швидкості коду $R_{\text{код}} = 1/2$ глибина декодування визначається як

$$L = (6 - 7)v, \quad (4)$$

де v – довжина кодового обмеження.

Кількість віток, оброблюваних на одному кроці декодування, оцінюється як $W_0 = S \cdot n$, де S – число станів кодової решітки ($S = m^v$, де m – основа коду), а n – число віток, що виходить з одного стану. Ґрунтуючись на викладеному вище, складність декодування визначимо як:

$$W = L \cdot W_0. \quad (5)$$

Навчальне видання

Розенвассер Денис Михайлович

ТЕОРІЯ ІНФОРМАЦІЇ ТА КОДУВАННЯ

Конспект лекцій

Українською мовою