

VIVERE!
VINCERE!
CREARE!

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний авіаційний університет

СИСТЕМИ МОБІЛЬНОГО РАДІОЗВ'ЯЗКУ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Київ 2023

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Національний авіаційний університет

СИСТЕМИ МОБІЛЬНОГО РАДІОЗВ'ЯЗКУ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»



Київ 2023

Укладачі: *О. Ю. Лавриненко* — канд. техн. наук, доцент кафедри ТКРС;
О. В. Жарова — канд. техн. наук, доцент кафедри ТКРС;
Д. І. Бахтіяров — канд. техн. наук, доцент кафедри ТКРС;
О. Г. Голубничий — д-р техн. наук, професор кафедри ТКРС
А. Г. Тараненко — канд. техн. наук, доцент кафедри ТКРС

Рецензент *В. М. Шутко* — д-р техн. наук, проф. завідувач кафедри електроніки, робототехніки і технологій моніторингу та інтернету речей (Національний авіаційний університет)

Затверджено Науково-методично-редакційною радою Національного авіаційного університету (протокол № 2/23 від 10.02.2023 р.).

Системи мобільного радіозв'язку: лабораторний практикум / уклад. : *О. Ю. Лавриненко, О. В. Жарова, Д. І. Бахтіяров та ін.* — К. : НАУ, 2023. — 96 с.

Наведено практичні вказівки до виконання лабораторних робіт у програмному комплексі MathWorks MATLAB з основними теоретичними відомостями, запитаннями та завданнями для самоконтролю. Лабораторні роботи спрямовано на здобування знань та набуття навичок, необхідних для комп'ютерного моделювання процесів, які є невід'ємною частиною у побудові систем мобільного радіозв'язку.

Практикум розроблений відповідно до програми дисципліни «Системи мобільного радіозв'язку», яка викладається для здобувачів вищої освіти бакалаврату на третьому та четвертому курсах спеціальності 172 «Телекомунікації та радіотехніка» під час навчального процесу, а також може бути корисним для викладачів та фахівців з радіотехніки й телекомунікацій.

ВСТУП

Мета лабораторних робіт – закріплення здобувачами вищої освіти теоретичних знань та набуття ними практичних навичок у комп'ютерному моделюванні процесів, які є невід'ємною частиною у побудові систем мобільного радіозв'язку, що являє собою базу в підготовці фахівців спеціальності 172 «Телекомунікації та радіотехніка».

У процесі виконання робіт здобувачі вищої освіти повинні дослідити особливості практичного застосування:

- дискретизації аналогових сигналів;
- квантування дискретних сигналів;
- кодування сигналів методом імпульсно-кодової модуляції;
- дискретної фільтрації сигналів;
- кодування мови методом довготривалого передбачення з регулярним імпульсним збудженням;
- циклічного кодування;
- згорткового кодування;
- блоково-діагонального перемежовування;
- модуляції на основі частотної маніпуляції з мінімальним зсувом;
- каналу зв'язку з адитивним білим шумом;
- низькочастотної, високочастотної, смугової і режекторної фільтрації фільтрами Баттерворта;

за допомогою комп'ютерного моделювання, використовуючи допоміжні функції з бібліотек Signal Processing Toolbox та Communications Toolbox, мови програмування MATLAB.

Опис кожної лабораторної роботи супроводжується викладенням основних теоретичних відомостей досліджуваних процесів та послідовності виконання практичної частини робіт у програмному середовищі MATLAB, після виконання яких потрібно оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу. Для більш поглибленого розуміння теоретичних та практичних аспектів досліджуваних процесів можна використовувати джерела, які вказані у списку літератури.

СПИСОК ЛІТЕРАТУРИ

1. Кравчук С.О., Голубничий О.Г., Тараненко А.Г., Потапов В.Г., Ткаліч О.П. Системи зв'язку з рухомими об'єктами: підручник. — К.: «Спринт-Сервіс», 2012. — 452 с.
2. Конахович Г.Ф. Системи радіозв'язку: навч. посіб. — К.: НАУ, 2004. — 311 с.
3. Конахович Г.Ф., Мачалін І.О., Пузиренко О.Ю. Теорія електричного зв'язку: навч. посіб. — К.: «Інтерсервіс», 2013. — 368 с.
4. Гусев О.Ю., Конахович Г.Ф., Корнієнко В.І., Кузнецов Г.В., Пузиренко О.Ю. Теорія електричного зв'язку: навч. посіб. — Львів: «Магнолія 2006», 2011. — 364 с.
5. Бортник Г.Г., Кичак В.М. Цифрова обробка сигналів в телекомунікаційних системах: підручник. — Вінниця: ВНТУ, 2014. — 232 с.
6. MathWorks MATLAB R2022b Documentation. Режим доступу: <https://www.mathworks.com/help/matlab/> — 02.12.2022.

ЗМІСТ

| | |
|-------------------------------------------------------------------------------------------------------------------|----|
| ВСТУП..... | 3 |
| Лабораторна робота 1. ДИСКРЕТИЗАЦІЯ АНАЛОГОВИХ СИГНАЛІВ..... | 4 |
| Лабораторна робота 2. КВАНТУВАННЯ ДИСКРЕТНИХ СИГНАЛІВ ... | 12 |
| Лабораторна робота 3. КОДУВАННЯ ТА ІМПУЛЬСНО-КОДОВА МОДУЛЯЦІЯ | 19 |
| Лабораторна робота 4. ДИСКРЕТНА ФІЛЬТРАЦІЯ | 26 |
| Лабораторна робота 5. КОДУВАННЯ МОВИ МЕТОДОМ ДОВГОТРИВАЛОГО ПЕРЕДБАЧЕННЯ З РЕГУЛЯРНИМ ІМПУЛЬСНИМ ЗБУДЖЕННЯМ | 33 |
| Лабораторна робота 6. ЦИКЛІЧНЕ КОДУВАННЯ..... | 41 |
| Лабораторна робота 7. ЗГОРТКОВЕ КОДУВАННЯ..... | 49 |
| Лабораторна робота 8. ЗАВАДОСТІЙКЕ КОДУВАННЯ В СИСТЕМІ GSM | 56 |
| Лабораторна робота 9. ЧАСТОТНА МАНІПУЛЯЦІЯ З МІНІМАЛЬНИМ ЗСУВОМ..... | 63 |
| Лабораторна робота 10. КАНАЛ ЗВ'ЯЗКУ З АДИТИВНИМ БЛИМ ШУМОМ | 71 |
| Лабораторна робота 11. НИЗЬКОЧАСТОТНИЙ, ВИСОКОЧАСТОТНИЙ, СМУГОВИЙ І РЕЖЕКТОРНИЙ ФІЛЬТРИ БАТТЕРВОРТА | 79 |
| ПІСЛЯМОВА | 93 |
| СПИСОК ЛІТЕРАТУРИ | 94 |

Навчальне видання

СИСТЕМИ МОБІЛЬНОГО РАДІОЗВ'ЯЗКУ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Укладачі:

ЛАВРИНЕНКО Олександр Юрійович
ЖАРОВА Олена Валеріївна
БАХТІЯРОВ Денис Ілшатович
ГОЛУБНИЧИЙ Олексій Георгійович
ТАРАНЕНКО Анатолій Григорович

В авторській редакції

Технічний редактор *А. І. Лавринович*

Надруковано з оригінал-макета автора

Підп. до друку 02.03.2023. Формат 60x84/16. Папір офс.
Офс. друк. Ум. друк. арк. 5,58. Обл.-вид. арк. 6,0.
Тираж 25 прим. Замовлення № 19-1.

Видавець і виготівник

Національний авіаційний університет
03058, Київ-58, проспект Любомира Гузара, 1.

Свідоцтво про внесення до Державного реєстру ДК № 7604 від 15.02.2022.

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Національний авіаційний університет

СИСТЕМИ МОБІЛЬНОГО РАДІОЗВ'ЯЗКУ

Лабораторний практикум
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Київ 2023

УДК 621.396.2 (076.5)

С 408

Укладачі:

О. Ю. Лавриненко — канд. техн. наук, доцент кафедри ТКРС

А. Г. Тараненко — канд. техн. наук, доцент кафедри ТКРС

Д. І. Бахтіяров — канд. техн. наук, доцент кафедри ТКРС

О. Г. Голубничий — д-р техн. наук, професор кафедри ТКРС

О. В. Жарова — канд. техн. наук, доцент кафедри ТКРС

Рецензент *В. М. Шутко* — д-р техн. наук, проф.
завідувач кафедри електроніки, робототехніки і
технологій моніторингу та інтернету речей
(Національний авіаційний університет)

Затверджено Науково-методично-редакційною радою Національного авіаційного університету (протокол № 2/23 від 10.02.2023 р.).

Системи мобільного радіозв'язку: лабораторний практикум / уклад. : О. Ю. Лавриненко, А. Г. Тараненко, Д. І. Бахтіяров, О. Г. Голубничий, О. В. Жарова. — К. : НАУ, 2023. — 96 с.

Наведено практичні вказівки до виконання лабораторних робіт у програмному комплексі MathWorks MATLAB з основними теоретичними відомостями, запитаннями та завданнями для самоконтролю. Лабораторні роботи спрямовано на здобування знань та набуття навичок, необхідних для комп'ютерного моделювання процесів, які є невід'ємною частиною у побудові систем мобільного радіозв'язку.

Лабораторний практикум розроблений відповідно до програми дисципліни «Системи мобільного радіозв'язку», яка викладається для здобувачів вищої освіти бакалаврату на третьому та четвертому курсах спеціальності 172 «Телекомунікації та радіотехніка» під час навчального процесу, а також даний практикум може бути корисним для викладачів та фахівців з радіотехніки й телекомунікацій.

ВСТУП

Мета лабораторних робіт – закріплення здобувачами вищої освіти теоретичних знань та набуття ними практичних навичок у комп'ютерному моделюванні процесів, які є невід'ємною частиною у побудові систем мобільного радіозв'язку, що являє собою базу в підготовці фахівців спеціальності 172 «Телекомунікації та радіотехніка».

У процесі виконання робіт здобувачі вищої освіти повинні дослідити особливості практичного застосування:

- дискретизації аналогових сигналів;
- квантування дискретних сигналів;
- кодування сигналів методом імпульсно-кодової модуляції;
- дискретної фільтрації сигналів;
- кодування мови методом довготривалого передбачення з регулярним імпульсним збудженням;
- циклічного кодування;
- згорткового кодування;
- блоково-діагонального перемережування;
- модуляції на основі частотної маніпуляції з мінімальним зсувом;
- каналу зв'язку з адитивним білим шумом;
- низькочастотної, високочастотної, смугової і режекторної фільтрації фільтрами Баттерворта;

за допомогою комп'ютерного моделювання, використовуючи допоміжні функції з бібліотек Signal Processing Toolbox та Communications Toolbox, мови програмування MATLAB.

Опис кожної лабораторної роботи супроводжується викладенням основних теоретичних відомостей досліджуваних процесів та послідовності виконання практичної частини робіт у програмному середовищі MATLAB, після виконання яких потрібно оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу. Для більш поглибленого розуміння теоретичних та практичних аспектів досліджуваних процесів можна використовувати джерела, які вказані у списку літератури.

Лабораторна робота 1

ДИСКРЕТИЗАЦІЯ АНАЛОГОВИХ СИГНАЛІВ

Мета роботи: дослідити процес дискретизації сигналів за допомогою комп'ютерного моделювання, використовуючи допоміжні функції **decimate** та **interp** з бібліотеки Signal Processing Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Представлення сигналу функцією часу $x(t)$ є надлишково докладним описом, якщо той має обмежену ширину спектра. Для передавання неперервної функції часу $x(t)$ з обмеженою шириною спектра F_{\max} зовсім не потрібно передавати усі миттєві значення функції. Достатньо передавати лише окремі її значення (рис. 1.1), що беруться не рідше певного інтервалу часу Δt . Миттєве значення сигналу у фіксований момент часу $k \cdot \Delta t$ називається *відліком* і позначається $x(k \cdot \Delta t)$, а крок часу Δt – *інтервалом дискретизації*.

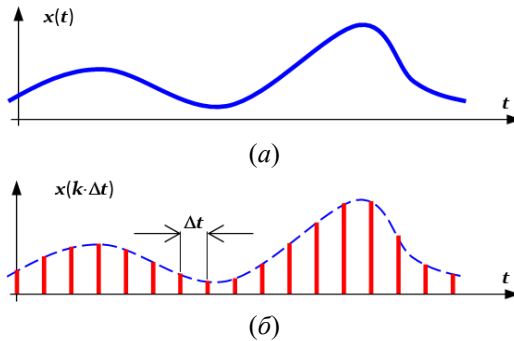


Рис. 1.1. Неперервна функція часу (а) і її відліки (б)

Якщо такі відліки напруги сигналу передати лінійно зв'язку, наприклад, у вигляді коротких імпульсів, амплітуда (або тривалість) яких пропорційна відлічуваному миттєвому значенню, то на приймальному боці первинна функція $x(t)$ з обмеженим спектром може бути відновлена *єдиним шляхом*. Для цього достатньо пропустити імпульси через ідеальний фільтр низьких частот (ФНЧ) з частотою зрізу $f_{зр.} = F_{\max}$, на виході якого буде отримана первинна функція $x(t)$ з обмеженим спектром. Пояснюється це тим, що під час передавання відліків і наступної фільтрації в прийнятому сигналі зберігаються складові первинного сигналу $x(t)$.

Теоретичним обґрунтуванням вищевикладеного є *теорема Котельникова* (відома також як *теорема відліків*): будь-яка неперервна функція $x(t)$, що не містить частот вище за F_{\max} , повністю визначається послідовністю своїх значень у моменти часу, що відстоять один від одного на інтервал $\Delta t = 1/(2 \cdot F_{\max})$.

Ця теорема показує, що між передаваннями дискретного і неперервного сигналів принципової різниці не існує. Обидва ці випадки зводяться до передавання окремих значень деякої фізичної величини, які можуть бути пронумеровані і закодовані. Як наслідок, можуть бути використані методи модуляції, єдині для передавання дискретної і неперервної інформації. Після демодуляції і декодування на приймальній стороні будуть одержані відліки, за якими передавана функція $x(t)$ може бути однозначно відновлена, якщо лише частота слідування цих відліків перевищує або, як мінімум, дорівнює частоті дискретизації $f_{\text{д}} = 2 \cdot F_{\max}$.

Функція $x(t)$ сигналу з обмеженою на частоті F_{\max} шириною спектру виражається через свої відліки $x(k \cdot \Delta t)$ наступним чином:

$$\begin{aligned} x(t) &= \sum_{k=0}^{\infty} x(k \cdot \Delta t) \cdot \frac{\sin[2 \cdot \pi \cdot F_{\max} \cdot (t - k \cdot \Delta t)]}{2 \cdot \pi \cdot F_{\max} \cdot (t - k \cdot \Delta t)} = \\ &= \sum_{k=0}^{\infty} x(k \cdot \Delta t) \cdot \frac{\sin[\Omega_{\max} \cdot (t - k \cdot \Delta t)]}{\Omega_{\max} \cdot (t - k \cdot \Delta t)}. \end{aligned} \quad (1.1)$$

Вираз (1.1) являє собою розклад сигналу $x(t)$ у нескінченний ряд ($k = 0, \pm 1, \pm 2, \dots$):

$$x(t) = \sum_{k=0}^{\infty} x(k \cdot \Delta t) \cdot \psi_k(t), \quad (1.2)$$

складений з функцій (рис. 1.2)

$$\psi_k(t) = \frac{\sin[2 \cdot \pi \cdot F_{\max} \cdot (t - k \cdot \Delta t)]}{2 \cdot \pi \cdot F_{\max} \cdot (t - k \cdot \Delta t)}.$$

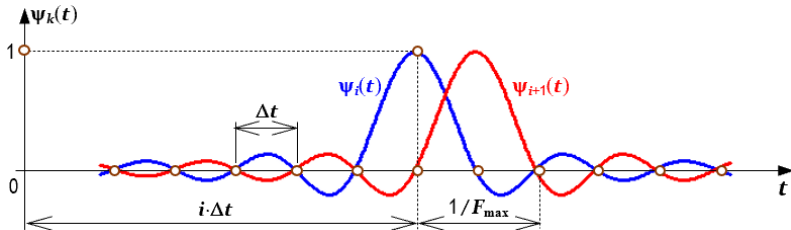


Рис. 1.2. Графік функцій відліків

Коефіцієнтами при функціях $\psi_k(t)$ у виразі (1.2) є значення (відліки) сигналу $x(t)$, що відстоять на $\Delta t = 1/(2 \cdot F_{\max})$ один від одного. Функція відліків $\psi_k(t)$ є імпульсним відгуком ідеального ФНЧ з частотою зрізу F_{\max} , якщо до його входу подати δ -функцію в момент часу $k \cdot \Delta t$.

Розклад функції сигналу $x(t)$ у ряд може бути виконаний також і на скінченному інтервалі часу T :

$$x(t) = \sum_{k=0}^{n-1} x(k \cdot \Delta t) \cdot \psi_k(t),$$

де n – кількість відліків функції $x(t)$, взятих за час T через інтервали $\Delta t = 1/(2 \cdot F_{\max})$, $n = T/\Delta t = 2 \cdot F_{\max} \cdot T$.

Виконаний розклад функції сигналу на скінченному інтервалі часу T принципово не може бути точним, оскільки з перетворень Фур'є слідує, що у природі не існує сигналів, які одночасно мають обмежену тривалість і обмежений спектр; реальні сигнали завжди обмежені у часі, а їх спектр є необмеженим (як відомо, нескінченно вузький, а отже й обмежений спектр має лише сигнал, постійний на проміжку $[-\infty, \infty]$). У результаті зневажання високочастотними складовими спектру, що перевищують деяку частоту F_{\max} , розклад $x(t)$ у ряд стає *наближеним*.

Також, що стосується теореми Котельникова, – вона є точною в тій мірі, в якій $x(t)$ може бути прийнято за функцію з обмеженим спектром. Проте обмеженість спектру функції $x(t)$ вимагає нескінченного продовження цієї функції у часі. Тому, фактично, теорема Котельникова є справедливою лише для сигналів, що нескінченно тривають у часі. Для точного відновлення функції за дискретними відліками необхідна нескінченна кількість таких відліків ($T \rightarrow \infty$, $n \rightarrow \infty$).

Але на практиці сигнал $x(t)$ повинен бути відновлений за дискретними відліками, кількість яких n обмежена через обмеженість часу спостереження T . У цьому випадку теорема Котельникова не може гарантувати точного відновлення первинної функції $x(t)$. У точках відліку функція $x(t)$ відновлюється точно, оскільки в цих точках усі функції $\psi_k(t)$, крім однієї, дорівнюють 0. Але у проміжках між відліками функція $x(t)$ не може бути відновлена точно, оскільки, обмежуючись інтервалом $[0, T]$, ми тим самим зневажаємо внесками, що вносяться відліками, які знаходяться за його ме-

жами. Оскільки великі значення функції $\psi_k(t)$ мають місце лише в околі точки відліку, відкидання відліків за межами $[0, T]$ внесе найбільшу похибку поблизу від країв інтервалу $[0, T]$ (рис. 1.2).

Отже, сигнал з обмеженим спектром $[0, F_{\max}]$, що спостерігається на обмеженому інтервалі часу $[0, T]$, може бути відновлений за $n = 2 \cdot F_{\max} \cdot T$ відліками лише приблизно. Похибка такого наближення дорівнює нулю у точках відліків, відмінна від нуля між точками відліків і зростає по мірі наближення до країв інтервалу $[0, T]$.

Можна, звичайно, спробувати ліквідувати внесок, що вноситься відліками, які знаходяться за межами інтервалу $[0, T]$, вважаючи, що за цими межами $x(t) = 0$. Але слід пам'ятати, що обмежений у часі сигнал має нескінченно широкий спектр і, таким чином, дискретність відліків при такому припущенні зникає.

У межах первинної смуги $[-F_{\max}, F_{\max}]$ спектр дискретного сигналу $x(k \cdot \Delta t)$ дорівнює, з точністю до постійного множника $(1/\Delta t)$, спектру первинного сигналу $x(t)$ (рис. 1.3, а). Крім того, спектр періодично повторюється по частоті з інтервалом f_d (рис. 1.3, б). Після обрання частоти дискретизації (наприклад, $f_d = 2 \cdot F_{\max}$, як на рис. 1.3, б) кожна спектральна копія відокремлюється від сусідніх копій смугою частот, яка дорівнює f_d і, теоретично, аналоговий сигнал можна повністю відновити за вибірками шляхом фільтрації. Проте, для виконання цієї задачі був би необхідний ідеальний фільтр з чітко обмеженою смугою прозорості (абсолютно крутими фронтами), який, нажаль, неможливо реалізувати фізично.

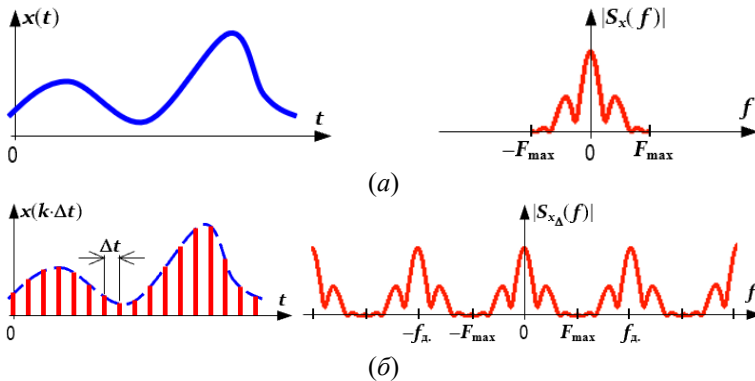


Рис. 1.3. Представлення у часовій і частотній областях: (а) – первинного сигналу $x(t)$ і його спектру; (б) – дискретного сигналу $x(k \cdot \Delta t)$ і його спектру

Очевидно, що при збільшенні частоти дискретизації $f_d > 2 \cdot F_{\max}$, спектральні копії первинного сигналу $x(t)$ віддалятимуться (у частотній області) одна від одної, як показано на рис. 1.4, а, і це полегшить операцію фільтрації. На рисунку також зображено типову характеристику ФНЧ, який використовується для виокремлення спектру сигналу. При зменшенні частоти дискретизації до $f_d < 2 \cdot F_{\max}$, спектральні копії починають перекриватися (рис. 1.4, б), у смугу частот $[-F_{\max}, F_{\max}]$ дискретизованого сигналу потрапляють спектральні компоненти суміжних областей і при відновленні функції за відліками інформація частково буде втрачена.

Явище, яке є результатом недостатньої дискретизації (вибірки здійснюються недостатньо часто), називається *накладанням*. Частота $f_d = 2 \cdot F_{\max}$ – це границя, нижче за яку відбувається накладання. Для зменшення накладання спектрів і, відповідно, похибки відновлення сигналу з відліків реальним ФНЧ, частоту дискретизації обирають на 15...20% вищою за граничну. Розрахувавши частоту дискретизації для 20%-го захисту від накладання спектрів, одержимо *інженерну формулу* для частоти дискретизації:

$$f_d \geq 2,2 \cdot F_{\max}.$$

Наприклад, у системах з імпульсним передаванням в якості робочої частоти дискретизації розмовного сигналу з $F_{\max} = 3400$ Гц рекомендується не $2 \cdot F_{\max} = 6800$ Гц, а $f_d = 8000$ Гц. При високоякісному оцифруванні музичного джерела з шириною спектра 20 кГц частота дискретизації становить 44,1 кГц.

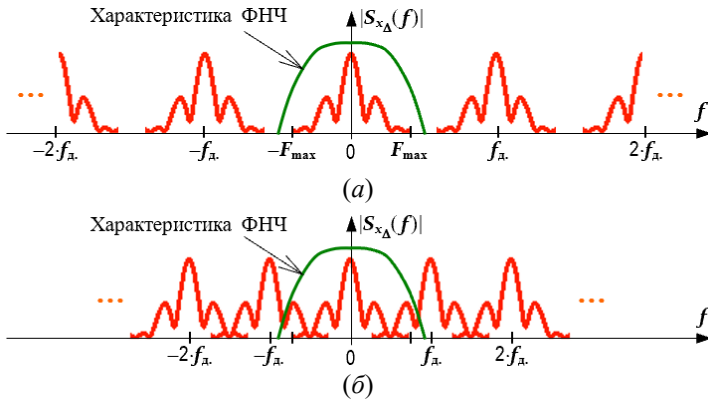


Рис. 1.4. Спектри для різних частот дискретизації:
(а) $f_d > 2 \cdot F_{\max}$; (б) $f_d < 2 \cdot F_{\max}$

Послідовність виконання роботи

1.1. Вивчити синтаксичні конструкції функцій **decimate** та **interp** мови програмування MATLAB і дати визначення їх параметрам **x**, **n**, **x_decim**, **x_interp**, **b**.

Синтаксис функцій **decimate** та **interp** для пониження і підвищення частоти дискретизації сигналів:

```
x_decim = decimate(x, n)
[x_interp, b] = interp(x_decim, n)
```

Опис параметрів функцій:

x – вихідний сигнал;

n – у скільки разів понижується або підвищується частота дискретизації сигналів **x** та **x_decim**, відповідно;

x_decim – сигнал з пониженою частотою дискретизації в **n** разів, що відповідно обумовлює зниження кількості відліків в **n** разів;

x_interp – сигнал з підвищеною частотою дискретизації в **n** разів, що відповідно обумовлює збільшення кількості відліків в **n** разів;

b – коефіцієнти нерекурсивної частини фільтра нижніх частот (чисельника функції передачі), використаного в процесі інтерполяції.

Процес пониження частоти дискретизації зводиться до пропускання вихідного сигналу **x** через фільтр нижніх частот Чебишева 1-го роду 8-го порядку з нормованою частотою зрізу $0,8/n$ та подальшого відкидання зайвих відліків, шляхом вибору кожного **n**-го відліку із відфільтрованого сигналу. Процес підвищення частоти дискретизації зводиться до вставки **n**-1 нулів між відліками вихідного сигналу і пропускання результату через нерекурсивний фільтр нижніх частот $8*n+1$ порядку, який забезпечує точне збереження значень вихідних відліків і мінімізацію середньоквадратичної помилки відхилення значень інтерпольованих відліків від ідеальних, при цьому інтерпольовані значення розраховуються по 8-ми вихідних відліках.

1.2. Змодельовати процес пониження і підвищення частоти дискретизації сигналів в програмному середовищі MATLAB за допомогою функцій **decimate** та **interp**.

Задайте частоту **Fs** (Гц) і крок **ts** (с) дискретизації сигналу **x** тривалістю **T** (с).

```
Fs = 4000;
ts = 1/Fs;
T = 0:ts:1;
```

Здайте амплітуду **A** (В), частоти **f1**, **f2** (Гц) і початкову фазу **phi** (рад) гармонічного коливання.

```
A = 1;  
f1 = 30;  
f2 = 60;  
phi = 0;
```

Згенеруйте дискретний сигнал **x** тривалістю **T** (с), який змінюється за гармонійним законом згідно заданої амплітуди **A** (В), частот **f1**, **f2** (Гц) і початкової фази **phi** (рад).

```
x = A*sin(2*pi*f1*T + phi) + A*sin(2*pi*f2*T + phi);
```

Виконайте процедуру пониження частоти дискретизації вихідного сигналу **x** в **n** разів, що обумовлює зниження кількості відліків у вихідному сигналі **x**, також в **n** разів за допомогою функції **decimate**.

```
n = 4;  
x_decim = decimate(x, n);
```

Виконайте процедуру підвищення частоти дискретизації сигналу **x_decim** в **n** разів, що обумовлює збільшення кількості відліків у сигналі **x_decim**, також в **n** разів за допомогою функції **interp**.

```
[x_interp, b] = interp(x_decim, n);
```

1.3. Побудувати графіки вихідного сигналу **x**, сигналу з пониженою частотою дискретизації **x_decim** та сигналу з підвищеною частотою дискретизації **x_interp**.

Побудуйте графік вихідного сигналу **x**.

```
figure(1)  
subplot(2,2,1)  
stem(0:120, x(1:121), 'filled', 'markersize', 3)  
title('Вихідний сигнал')  
ylabel('Амплітуда')  
xlabel('Відліки')  
grid on
```

Побудуйте графік сигналу з пониженою частотою дискретизації **x_decim**.

```
subplot(2,2,2)  
stem(0:30, x_decim(1:31), 'filled', 'markersize', 3)  
title('Сигнал з пониженою частотою дискретизації')  
ylabel('Амплітуда')  
xlabel('Відліки')  
grid on
```

Побудуйте графік сигналу з підвищеною частотою дискретизації x_{interp} .

```
subplot(2,2,3)
stem(0:120, x_interp(1:121), 'filled', 'markersize', 3)
title('Сигнал з підвищеною частотою дискретизації')
ylabel('Амплітуда')
xlabel('Відліки')
grid on
```

1.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Навіщо потрібно дискретизувати аналогові сигнали?
2. Розкрийте сутність розкладу сигналів у ряд Котельникова.
3. Сформулюйте теорему відліків.
4. Поясніть процес пониження і підвищення частоти дискретизації.
5. Нижче якої частоти дискретизації аналоговий сигнал без спотворень не може бути відновлено?
6. Поясніть причину використання на практиці інженерної формули визначення частоти дискретизації.
7. Відомо, що максимальна частота дискретизації цифрової системи – 0,75 МГц. Якою є максимальна частота звукового сигналу, який буде можливо точно відтворити за допомогою такої системи?
8. Відомо, що неперервна функція $z(t)$ не містить складових з частотою вище за 0,0001 ГГц. Визначте, з яким інтервалом мають братися відліки, щоб за ними можна було відтворити задану функцію згідно інженерної форми запису теореми відліків.

9. Як називається явище, яке є результатом недостатньої дискретизації (вибірки здійснюються недостатньо часто)?

10. Записати та пояснити синтаксичні конструкції функцій **decimate** та **interp** мови програмування MATLAB і дати визначення їх параметрам **x**, **n**, **x_decim**, **x_interp**, **b**.

Лабораторна робота 2

КВАНТУВАННЯ ДИСКРЕТНИХ СИГНАЛІВ

Мета роботи: дослідити процес квантування дискретних сигналів за допомогою комп'ютерного моделювання використовуючи допоміжну функцію **quantiz** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Первинний неперервний сигнал $A(t)$ з обмеженням на інтервалі $[0, F_{\max}]$ спектром *дискретизується* з інтервалом $\Delta t = (2 \cdot F_{\max})^{-1}$, де $2 \cdot F_{\max} = f_d$ – частота дискретизації. Одержується дискретна послідовність відліків сигналу у моменти часу $t_k = (k-1) \cdot \Delta t$. Інтервал дискретизації, як правило, обирають у відповідності до теореми Котельникова (див. лаб. роб. 1). Одержані відліки $A_{d,k} = A[(k-1) \cdot \Delta t]$ дискретизуються за рівнем – операція *квантування*.

Сутність квантування як нелінійного перетворення полягає у тому, що всі відліки, які потрапили в інтервал квантування Δq_i (крок між рівнями квантування q_{i-1} та q_i , рис. 2.1), представляються одним значенням d_{q_i} , яке називають квантованим.

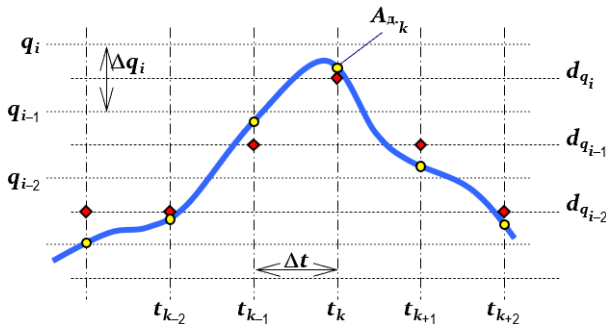


Рис. 2.1. Приклад квантування рівня відліків
(● – значення дискретних відліків; ◆ – квантовані значення)

Характерною особливістю квантування є те, що через похибки квантування копія сигналу відрізняється від оригіналу навіть за повної відсутності завад у каналі зв'язку. Тому вводять таку характеристику, як *шуми квантування* $\varepsilon(t)$:

$$A^*(t) = A(t) + \varepsilon(t).$$

Процес квантування є визначеним, якщо є заданою *характеристика квантування*, що пов'язує інтервали квантування і квантовані значення: кожному інтервалу квантування Δq_i поставлено у відповідність квантоване значення d_{q_i} . Якщо інтервал квантування постійний, то саме квантування є *рівномірним* (з постійним кроком). Подібна система буде неекономічною: багато значень квантування використовуватимуться досить рідко, а шум квантування $\varepsilon(t)$ буде однаковим для всіх амплітуд сигналу (у межах від $-\Delta q/2$ до $\Delta q/2$).

Таким чином, відношення «сигнал/шум» буде гіршим для сигналів низьких рівнів, порівняно із сигналами високих рівнів. Для того, щоб шум квантування був пропорційним рівню сигналу, використовують *нерівномірне квантування*, яке забезпечує краще (більш детальне) квантування слабких сигналів і грубе квантування сильних. Результатом є підвищення загального відношення «сигнал/шум» – зменшення шуму для домінуючих слабких сигналів за рахунок підвищення шуму для значно менш імовірних сильних сигналів. Одним із способів одержання нерівномірного квантування є використання пристрою з характеристикою, зображеною на рис. 2.2,а.

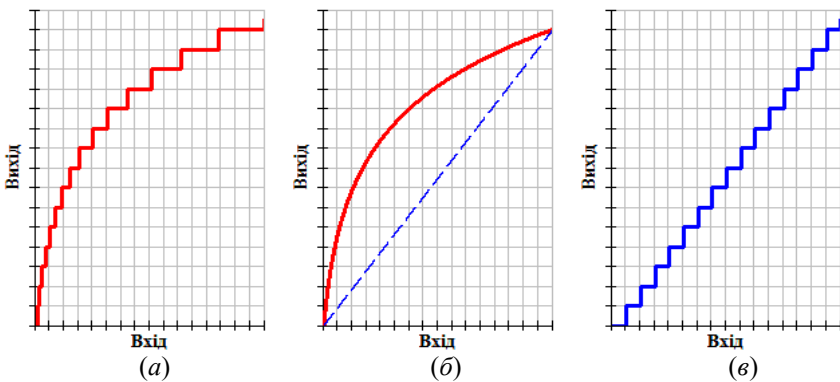


Рис. 2.2. Характеристики пристроїв компресії

Зазвичай нерівномірне квантування реалізується наступним чином: спочатку первинний сигнал деформується за допомогою пристрою, який має логарифмічну характеристику компресії, зображену на рис. 2.2,б (суцільна лінія), а потім деформований сигнал пропускається через пристрій, що має рівномірний крок квантування (характеристика такого лінійного пристрою квантування зображена на рис. 2.2, в).

Для сигналів малої амплітуди характеристика компресії має більш крутий фронт, ніж для сигналів великої амплітуди. Отже, зміна даного сигналу при малих амплітудах задіє більшу кількість рівномірно розміщених рівнів квантування, порівняно з аналогічною зміною при великих амплітудах. Таким чином, нелінійна характеристика компресії ефективно змінює розподіл амплітуд вхідного сигналу, і на виході системи вже не існуватиме домінування сигналів малих амплітуд.

Після прийому сигнал пропускається через пристрій з характеристикою, оберненою до зображеної на рис. 2.2, б. Дана операція називається *розширенням* або *експандуванням*, а описана пара етапів обробки сигналів (компресія/розширення) у сукупності називається *командуванням*.

Більшість сучасних систем використовують кусочно-лінійну апроксимацію функції логарифмічної компресії. Північною Америкою запропонована до використання так звана *μ -рівнева характеристика компресії*, яка описується наступним законом:

$$B = B_{\max} \cdot \frac{\ln \left[1 + \mu \cdot \left(\frac{|A|}{A_{\max}} \right) \right]}{\ln(1 + \mu)} \cdot \operatorname{sgn}(A),$$

де $\operatorname{sgn}(A) = \begin{cases} +1 & \text{при } A \geq 0; \\ -1 & \text{при } A < 0 \end{cases}$ – знакова функція; μ – додатна константа;

A, B – значення рівнів на вході і виході пристрою компресії відповідно; A_{\max}, B_{\max} – відповідно, максимальні додатні амплітуди напруг. Характеристика пристрою компресії для декількох значень параметру μ зображена на рис. 2.3, а. Стандартним значенням для μ є 255. Зазначимо, що $\mu = 0$ відповідає лінійному перетворенню (рівномірному квантуванню).

У Європі для опису характеристики пристрою компресії використовується *A -рівнева характеристика компресії*:

$$B = \begin{cases} B_{\max} \cdot \frac{A \cdot \frac{|A|}{A_{\max}}}{1 + \ln(A)} \cdot \text{sgn}(A), & \text{при } 0 < \frac{|A|}{A_{\max}} \leq \frac{1}{A}; \\ B_{\max} \cdot \frac{1 + \ln\left[A \cdot \frac{|A|}{A_{\max}}\right]}{1 + \ln(A)} \cdot \text{sgn}(A), & \text{при } \frac{1}{A} < \frac{|A|}{A_{\max}} \leq 1, \end{cases}$$

де A – додатна константа (стандартним є значення $A = 87,6$, а значення $A = 1$ відповідає рівномірному квантуванню). На рис. 2.3, б зображені характеристики пристроїв компресії для декількох значень параметру A . Отримані після операцій дискретизації і квантування відліки розглядаються в обраній системі числення (переважно двійковій) як числа, які кодуються для перетворення їх на кодові комбінації електричних сигналів. Іншими словами, кожен аналоговий відлік перетворюється на послідовність, яка складається з групи двійкових розрядів (бітів). Розмір слова можна виразити через кількість рівнів квантування L , дозволених для кожного відліку; він же дорівнює кількості значень, яку може набувати послідовність бітів.

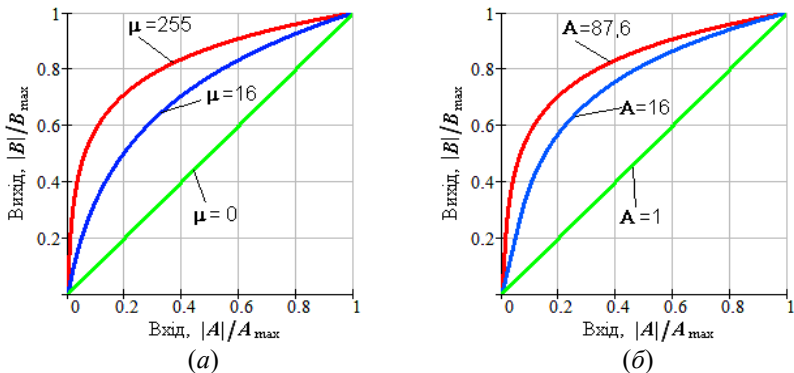


Рис. 2.3. Характеристики пристроїв компресії:
(а) – для μ -закону; (б) – для A -закону

Зв'язок між кількістю рівнів квантування на відлік (L) і кількістю біт, необхідних для представлення всього набору відліків (b), у випадку двійкової системи числення наступний:

$$L = 2^b.$$

Кількість біт, достатня для кодування кожного рівня, залежить від того, яке спотворення викликане квантуванням є припустимим.

Нехай величина помилки внаслідок квантування, $|\varepsilon|$, визначається як частина p розмаху амплітуди напруги первинного аналогового сигналу $A(t)$:

$$|\varepsilon| \leq p \cdot U_A. \quad (2.1)$$

Оскільки абсолютне значення помилки квантування не може бути більшим за $\Delta q/2$ (див. рис. 2.1), можемо записати:

$$|\varepsilon_{\max}| = \frac{\Delta q}{2} = \frac{U_A}{2 \cdot (L-1)}. \quad (2.2)$$

Отже, з формул (2.1) і (2.2) маємо:

$$\frac{U_A}{2 \cdot (L-1)} \leq p \cdot U_A, \quad \frac{1}{2 \cdot (L-1)} \leq p, \quad (L-1) \geq \frac{1}{2 \cdot p},$$

$$L = 2^b \geq \left(\frac{1}{2 \cdot p} + 1 \right) \text{ рівнів}$$

або

$$b \geq \log_2 \left(\frac{1}{2 \cdot p} + 1 \right) \text{ біт.}$$

Послідовність виконання роботи

2.1. Вивчити синтаксичну конструкцію функції **quantiz** мови програмування MATLAB і дати визначення її параметрам **x**, **partition**, **codebook**, **index**, **quants**, **distor**.

Синтаксис функції **quantiz** для квантування дискретних сигналів:

[index, quants, distor] = quantiz(x, partition, codebook)

Опис параметрів функції:

x – дискретний сигнал;

partition – значення рівнів квантування, які задаються згідно запису **min(x)+q:q:max(x)**, де крок квантування **q** для заданої кількості рівнів квантування **L** знаходиться наступним виразом **q = (max(x)-min(x))/L**;

codebook – квантовані значення рівнів квантування, які задаються згідно запису **min(x)+q:q:max(x)+q**;

index – квантований дискретний сигнал (номера рівнів квантування квантованих значень відліків дискретного сигналу **quants**);

quants – квантовані значення відліків дискретного сигналу;

distor – середньоквадратична помилка квантування, яка оцінює ступінь спотворення дискретного сигналу **x** після процесу квантування.

2.2. Змоделювати процес квантування дискретного сигналу x в програмному середовищі MATLAB за допомогою функції **quantiz**.

Задайте частоту F_d (Гц) і крок td (с) дискретизації сигналу тривалістю T (с).

```
Fd = 40;  
td = 1/Fd;  
T = 0:td:1;
```

Задайте амплітуду A (В), частоту f (Гц) і початкову фазу ϕ (рад) гармонічного коливання.

```
A = 2;  
f = 4;  
phi = pi/4;
```

Згенеруйте дискретний сигнал x тривалістю T (с), який змінюється за гармонійним законом згідно заданої амплітуди A (В), частоти f (Гц) і початкової фази ϕ (рад).

```
x = A*cos(2*pi*f*T + phi);
```

Задайте кількість рівнів квантування L .

```
L = 4;
```

Задайте крок квантування q .

```
q = (max(x)-min(x))/L;
```

Задайте значення рівнів квантування **partition**.

```
partition = min(x)+q:q:max(x);
```

Задайте квантовані значення рівнів квантування **codebook**.

```
codebook = min(x)+q:q:max(x)+q;
```

Виконайте квантування дискретного сигналу x по заданому набору рівнів **partition** за допомогою функції **quantiz**.

```
[index, quants, distor] = quantiz(x, partition, codebook);
```

2.3. Побудувати графіки дискретного сигналу x , квантованих значень відліків дискретного сигналу **quants** та квантованого дискретного сигналу **index**.

Побудуйте графік дискретного сигналу x .

```
figure(1)  
subplot(2,2,1)  
stem(T, x, 'linewidth', 2)  
title('Дискретний сигнал')  
ylabel('Амплітуда')  
xlabel('Час (с)')
```

```
grid on
```

Побудуйте графік квантованих значень відліків дискретного сигналу **quants**.

```
subplot(2,2,2)
```

```
stem(T, quants, 'linewidth', 2)
```

```
title('Квантовані значення відліків дискретного сигналу')
```

```
ylabel('Амплітуда')
```

```
xlabel('Час (с)')
```

```
grid on
```

Побудуйте графік квантованого дискретного сигналу **index**.

```
subplot(2,2,3)
```

```
stem(T, index, 'linewidth', 2)
```

```
title('Квантований дискретний сигнал')
```

```
ylabel('Амплітуда')
```

```
xlabel('Час (с)')
```

```
grid on
```

2.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Поясніть призначення операції квантування.
2. Як визначити крок квантування?
3. У чому кардинальна відмінність між операціями дискретизації та квантування сигналів?
4. Розкрийте сутність і відмінність між А- та μ -рівневою характеристиками компресії.
5. Що таке рівні квантування, та як їх знайти?

6. Наведіть характеристики рівномірного і нерівномірного квантування.

7. Чи буде сигнал спотворений після операції квантування, якщо при передачі його, у каналі зв'язку завади відсутні?

8. Який зв'язок між кількістю рівнів квантування та розміром кодового слова на один дискретизований відлік?

9. Визначте мінімально достатню кількість біт для двійкового кодування окремого рівня квантування, щоб мати можливість представити весь набір відліків, якщо відомо, що загальна кількість рівнів квантування відліків сигналу $s(t)$ дорівнює 220.

10. Записати та пояснити синтаксичну конструкцію функції **quantiz** мови програмування MATLAB і дати визначення її параметрам **x**, **partition**, **codebook**, **index**, **quants**, **distor**.

Лабораторна робота 3

КОДУВАННЯ ТА ІМПУЛЬСНО-КODOVA МОДУЛЯЦІЯ

Мета роботи: дослідити процес кодування сигналів методом імпульсно-кової модуляції за допомогою комп'ютерного моделювання використовуючи допоміжні функції **de2bi** і **bi2de** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Для передавання неперервних повідомлень дискретними методами використовують *цифрові методи модуляції* (ЦММ). Сутність цифрової модуляції полягає у тому, що неперервний сигнал $x(t)$, який необхідно передати, попередньо дискретизується у часі, квантується за рівнем, а отримані в результаті відліки розглядаються в обраній системі числення як числа, які кодуються для перетворення їх на кодові комбінації електричних сигналів.

Отже, ЦММ оснований на трьох необхідних перетвореннях інформаційних корисних сигналів: дискретизації, квантуванні і кодуванні. Четверте перетворення – модуляцію – використовують при передаванні сигналів і у даному випадку нами розглядатися не буде.

Найпоширенішими ЦММ є *імпульсно-кова модуляція* (ІКМ).

Узагальнена структурна схема системи ІКМ наведена на рис. 3.1.

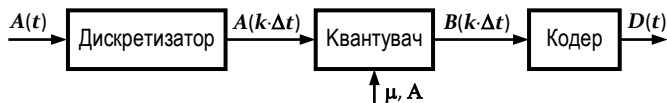


Рис. 3.1. Структурна схема системи ІКМ

Розглянемо сутність операцій дискретизації, квантування і кодування у відповідності до імпульсно-кодової модуляції неперервних сигналів (рис. 3.2).

Первинний неперервний сигнал $A(t)$ з обмеженням на інтервалі $[0, F_{\max}]$ спектром дискретизується з інтервалом $\Delta t = (2 \cdot F_{\max})^{-1}$, де $2 \cdot F_{\max} = f_d$ – частота дискретизації. Одержується дискретна послідовність відліків сигналу у моменти часу $t_k = (k-1) \cdot \Delta t$. Інтервал дискретизації, як правило, обирають у відповідності до теореми Котельникова (див. лаб. роб. 1).

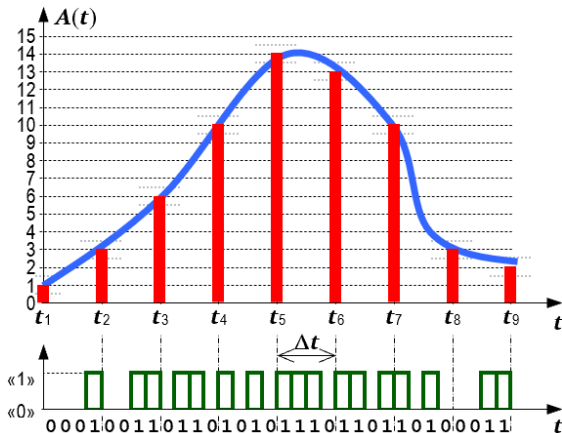


Рис. 3.2. Часові діаграми, що пояснюють сутність операцій дискретизації, квантування і кодування

Одержані відліки $A_{д,k} = A[(k-1) \cdot \Delta t]$ дискретизуються за рівнем – операція квантування. Сутність квантування як нелінійного перетворення полягає у тому, що всі відліки, які потрапили в інтервал квантування Δq_i (крок між рівнями квантування q_{i-1} та q_i), представляються одним значенням d_{q_i} , яке називають квантованим (див. лаб. роб. 2). Під кодуванням у розширеному значенні розуміють процес відображення повідомлення у сигнал для передавання його каналом зв'язку (КЗ). У вузькому значенні, кодування – це процес перетворення елементів дискретного повідомлення у послідовність

кодових символів (первинний електричний сигнал) для передавання їх дискретним каналом зв'язку.

У випадку найпростішого кодування кожен символ дискретного повідомлення пронумеровується і передавання повідомлень зводиться до передавання послідовності чисел, записаних в певній системі числення (СЧ). Наприклад, для передавання літер українського алфавіту достатньо пересилати числа від 1 до 32. Позначимо через s основу СЧ (наприклад, $s = 2$ для двійкової СЧ). Тоді будь-яке n -розрядне число χ можна записати у вигляді поліному (багаточлена):

$$\chi = \sum_{i=0}^{n-1} a_i \cdot s^i,$$

де a_i – цілі числа, що задовольняють умові $0 \leq a_i \leq s - 1$. Наприклад, десяткове число 158 у двійковій СЧ записується наступним чином:

$$\begin{aligned} (158)_{s=2} &= \\ &= 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = \\ &= 1001\ 1110. \end{aligned}$$

Переваги двійкового кодування: 1) для передавання каналом зв'язку послідовності елементів дискретного повідомлення, перетвореної у послідовність двійкових чисел, є необхідними лише два елементарних кодових сигнали: «0» і «1»; 2) при передаванні двійкових сигналів необхідна система лише з двома стійкими станами; 3) арифметичні операції в двійковій СЧ легко реалізуються за допомогою комбінацій найпростіших логічних схем обчислювальної техніки: *додавання* (схема Або) і *добуток* (схема І). Повну сукупність умовних символів, яку застосовують для кодування повідомлень, і алгоритм, що їх пов'язує, називають *кодом*. Кількість різних символів у коді називають *основою коду* t_k . Послідовність кодових символів, що відповідають окремому елементу дискретного повідомлення (число, записане у СЧ з основою $s = t_k$), називають *ковою комбінацією* (КК). *Значністю* (*розрядністю* або, рідше, *довжиною*) *коду* n називають кількість символів у КК. На кодову комбінацію, що присвоюється кожному окремому елементу повідомлення, вказує *оператор кодування*. Якщо $t_k = 2$, код називають *двійковим* (*бінарним*); при $t_k > 2$ – *багатопозиційним*. Якщо всі КК містять однакову кількість символів, яка дорівнює значності коду, то код називають *рівномірним*; в іншому випадку – *нерівномірним*.

Для рівномірного коду загальна кількість різних КК $N_{\Sigma} = m_k^n$. Наприклад, для рівномірного коду Бодо $m_k = 2$, $n = 5$, а $N_{\Sigma} = 2^5 = 32$. Цієї кількості комбінацій достатньо, наприклад, для кодування всіх літер українського алфавіту. Прикладом нерівномірного коду можна назвати код Морзе, в якому літерам, що зустрічаються найчастіше, присвоюються більш короткі КК (як буде показано у подальшому, це зроблено для збільшення середньої швидкості передавання інформації). Розглянемо процес аналого-цифрового (АЦП) та цифро-аналогового (ЦАП) перетворень мовних сигналів. Перетворення аналогового мовного сигналу в цифровий сигнал складається з трьох операцій: дискретизації, квантування та кодування. Дискретизація необхідна для отримання миттєвих амплітуд або відліків. Інтервал дискретизації дорівнює:

$$\Delta t_d = \frac{1}{f_d} \leq \frac{1}{2F_B},$$

де f_d – частота дискретизації; F_B – верхня частота спектру аналогового сигналу.

У телефонії мовний кодер обмежується смугою (300...3400) Гц, оскільки передача цих частот забезпечує достатню розбірливість мови. При такій смузі зазвичай приймають $f_d = 8$ кГц, тоді $\Delta t_d = 125$ мкс. Квантування полягає в тому, що кожний амплітудний відлік замінюється номером найближчого фіксованого рівня. Кількість цих рівнів $N_{\text{кв}}$ має забезпечувати задану похибку наступного цифро-аналогового перетворення. Кодування полягає у відображенні номерів двійковим n -розрядним кодом. Кількість комбінацій двійкового коду дорівнює $N_{\text{код}} = 2^n$. Використання всіх кодівих комбінацій досягається при $N_{\text{кв}} = N_{\text{код}}$. При $f_d = 8$ кГц на виході АЦП формується послідовність n -бітових двійкових слів, що надходять з такою ж частотою. Швидкість передачі такого цифрового сигналу визначається значенням n . Наприклад, при $n = 13$ швидкість дорівнює:

$$\frac{n}{\Delta t} = \frac{n}{\Delta t_d} = \frac{13 \text{ біт}}{125 \cdot 10^{-6} \text{ с}} = 104 \text{ кбіт/с}. \quad (3.1)$$

У приймачі ЦАП формує квантовані рівні з двійкових слів мовного декодера. При цьому виконуються операції, відповідні перетворенню двійкових чисел в десяткові, а саме: 1) попереднє форму-

вання аналогових рівнів, пропорційних вагам двійкових розрядів: $2^0 = 1$, $2^1 = 2$, $2^2 = 4$ тощо; 2) множення цих рівнів на одиниці чи нулі у відповідних розрядах двійкового числа; 3) додавання отриманих множників. Отримана сума являє собою вихідний рівень ЦАП, постійний на інтервалі дискретизації. Стрибки рівнів згладжуються за допомогою фільтру нижніх частот, на виході якого отримують аналоговий мовний сигнал.

Послідовність виконання роботи

3.1. Вивчити синтаксичну конструкцію функцій **de2bi** та **bi2de** мови програмування MATLAB і дати визначення їх параметрам **S**, **n**, **p**, **'left-msb'**, **Mp**, **Sdec**.

Синтаксис функцій **de2bi** і **bi2de** для перетворення чисел із десяткової в задану систему числення та оберненого перетворення:

```
Mp = de2bi(S, n, p, 'left-msb')
```

```
Sdec = bi2de(Mp, p, 'left-msb')
```

Опис параметрів функцій:

S – числа в десятковій системі числення (значення сигналу);

n – кількість значень **p**-ї системи числення, що припадають на одне число десяткової системи числення, де максимально можливе число **Lmax**, яке може бути записане в **n** значеннях відповідно до заданої системи числення **p**, розраховується наступним виразом **Lmax = p^n - 1**;

p – основа заданої системи числення, згідно якої буде відбуватися перетворення чисел десяткової системи числення (значень вихідного сигналу) **S**;

'left-msb' – запускає формування матриці **Mp** таким чином, що перший її стовпець відповідає старшим розрядам чисел **S** в заданій системі числення **p**;

Mp – матриця, кожен рядок якої відповідає числу **S** в заданій системі числення **p**;

Sdec – обернено-перетворені числа в десятковій системі числення (значення обернено-перетвореного сигналу).

Для формування перетвореного сигналу **Sp** довжиною **k = m*n** із значень матриці **Mp** розмірністю **m*n** **p**-ї системи числення, було потрібно задіяти функцію **reshape**, яка дозволяє перетворити матрицю в вектор, а також здійснити обернене перетворення згідно з відповідними записами **Sp = reshape(Mp', k, 1)**, **Mp = reshape(Sp, n, m)'**, де символ **'** – транспонування матриці.

3.2. Змоделювати процес перетворення чисел (значень сигналу) S із десяткової в задану систему числення p і їх подальшого оберненого перетворення в програмному середовищі MATLAB за допомогою функцій **de2bi** та **bi2de** відповідно.

Задайте основу системи числення p , згідно якої буде відбуватися перетворення значень вихідного сигналу S .

$$p = 2;$$

Задайте кількість значень p -ї системи числення, що припадають на одне значення вихідного сигналу S (для двійкової системи числення, n – кількість бітів на один відлік вихідного сигналу S).

$$n = 8;$$

Розрахуйте максимально можливі значення L_{max} вихідного сигналу S , яке може бути записане в n значеннях відповідно до заданої системи числення p .

$$L_{max} = p^n - 1;$$

Установіть довжину m вихідного сигналу S .

$$m = 5;$$

Згенеруйте вихідний сигнал S довжиною m , який складається з рівномірно розподілених випадкових цілих чисел десяткової системи числення в діапазоні від 0 до L_{max} .

$$S = \text{randi}([0 L_{max}], m, 1);$$

Виконайте перетворення значень вихідного сигналу S із десяткової системи числення в матрицю значень M_p розмірністю $m \times n$ згідно заданої системи числення p .

$$M_p = \text{de2bi}(S, n, p, \text{'left-msb'});$$

Розрахуйте довжину k перетвореного сигналу S_p .

$$k = m * n;$$

Сформууйте перетворений сигнал S_p в p -ї системі числення з допомогою перетворення матриці M_p розмірністю $m \times n$ у вектор-стовпець довжиною k за рядками.

$$S_p = \text{reshape}(M_p', k, 1);$$

Виконайте обернене перетворення сформованого сигналу S_p довжиною k в матрицю значень сигналу M_{Sp} розмірністю $m \times n$ за стовпцями.

$$M_{Sp} = \text{reshape}(S_p, n, m);$$

Виконайте обернене перетворення значень матриці M_{Sp} розмірністю $m \times n$ із p -ї системи числення в значення обернено-перетвореного сигналу S_{dec} довжиною m десяткової системи числення.

```
Sdec = bi2de(MSp, p, 'left-msb');
```

Обчисліть кількість помилкових значень, порівнюючи вихідний **S** та обернено-перетворений **Sdec** сигнали.

```
numerr = biterr(S, Sdec);
```

3.3. Побудувати графіки вихідного **S**, перетвореного **Sp** та обернено-перетвореного **Sdec** сигналів.

Побудуйте графік вихідного сигналу **S**.

```
figure(1)
subplot(2,2,1)
stairs([0:length(S)], [S; S(length(S))], 'linewidth', 2)
title('Вихідний сигнал')
ylabel('Сигнал в десятковій системі числення')
xlabel('Послідовність відліків сигналу')
grid on
```

Побудуйте графік перетвореного сигналу **Sp**.

```
subplot(2,2,2)
stairs([0:length(Sp)], [Sp; Sp(length(Sp))], 'linewidth', 2)
title('Перетворений сигнал')
ylabel('Сигнал в p-й системі числення')
xlabel('Послідовність відліків сигналу')
grid on
```

Побудуйте графік обернено-перетвореного сигналу **Sdec**.

```
subplot(2,2,3)
stairs([0:length(Sdec)], [Sdec; Sdec(length(Sdec))], 'linewidth', 2)
title('Обернено-перетворений сигнал')
ylabel('Сигнал в десятковій системі числення')
xlabel('Послідовність відліків сигналу')
grid on
```

3.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.

4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Розкрийте сутність цифрових методів модуляції.
2. Яку роль відіграє кодування в ІКМ?
3. У чому різниця між цифровими та аналоговими сигналами?
4. Наведіть структурну схему і опишіть принцип роботи системи з імпульсно-ковою модуляцією.
5. Опишіть процес АЦП и ЦАП мовних сигналів.
6. Які види кодування ви знаєте, та чим вони принципово відрізняються один від одного?
7. Дайте визначення коду, кодування, кодової комбінації, значності коду.
8. Наведіть приклади кодів, виходячи з основи коду і довжини кодових комбінацій.
9. Визначте, чому дорівнюватиме максимальна швидкість передачі інформації при ІКМ сигналу $s(t)$ з обмеженим на частоті 36 кГц спектром при кодуванні кожного відліку одним байтом.
10. Записати та пояснити синтаксичні конструкції функцій **de2bi** та **bi2de** мови програмування MATLAB і дати визначення їх параметрам **S**, **n**, **p**, **'left-msb'**, **Mp**, **Sdec**.

Лабораторна робота 4

ДИСКРЕТНА ФІЛЬТРАЦІЯ

Мета роботи: дослідити процес дискретної фільтрації сигналів за допомогою комп'ютерного моделювання використовуючи допоміжну функцію **filter** з бібліотеки Signal Processing Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Фільтрацією називається процес зміни частотного спектра сигналу в певному бажаному напрямку. Цифровим фільтром називається цифрова система, яку можна використовувати для фільтрації дискретних сигналів. Він може бути реалізований програмним методом або за допомогою спеціальної апаратури, і в кожному з цих випадків цифровий фільтр можна застосувати для фільтрації сигнала.

лів як реального часу, так і попередньо записаних. Цифровий фільтр можна подати структурною схемою, зображеною на рис. 4.1. На цій схемі $x(n)$ і $y(n)$ – відповідно, вхідний вплив і реакція фільтра на цей вплив. Функціонально вони пов’язані співвідношенням

$$y(n) = \Phi[x(n)],$$

де вид оператора $\Phi[\cdot]$ залежить від властивостей конкретної системи.

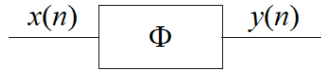


Рис. 4.1. Структурна схема цифрового фільтра

Реакцію цифрового фільтра на довільну дію можна подати за допомогою імпульсної характеристики фільтра. Припустимо, що $x(n)$ – вхідна, а $y(n)$ – вихідна послідовності фільтра і нехай $h(n)$ – відгук на одиничний імпульс, званий імпульсною характеристикою. Тоді

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m).$$

Таким чином, $x(n)$ і $y(n)$ пов’язані співвідношенням типу згортки. Частотна характеристика фільтра визначається наступним виразом:

$$H(e^{j\omega}) = \sum_{m=-\infty}^{\infty} h(m)e^{-j\omega m}. \quad (4.1)$$

Оскільки частотна характеристика є періодичною функцією частоти ω , рівність (4.1) можна розглядати як розкладання $H(e^{j\omega})$ до ряду Фур’є, причому коефіцієнти є одночасно відліками імпульсної характеристики. Відповідно до теорії рядів Фур’є, коефіцієнти $h(n)$ можуть бути виражені через $H(e^{j\omega})$:

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega.$$

З цього співвідношення видно, що $h(n)$ по суті є суперпозицією синусоїд $e^{j\omega n}$ з амплітудами $H(e^{j\omega})$, які можна представити так:

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j \arg H(e^{j\omega})}.$$

Вираз $|H(e^{j\omega})|$ називають амплітудною характеристикою фільтра, а $e^{j \arg H(e^{j\omega})}$ фазовою характеристикою фільтра.

Дамо кілька визначень, присвячених цифровим фільтрам: 1) цифровий фільтр називається стаціонарним, якщо його параметри не змінюються в часі, тобто попередньо збуджений фільтр, в якому $x(n) = y(n) = 0$ при всіх $n < 0$ називають стаціонарним тоді і тільки тоді, коли $\Phi[x(n - n_0)] = y(n - n_0)$ для всіх можливих впливів; 2) цифровий фільтр називають лінійним тоді і тільки тоді, коли $\Phi[\alpha x_1(n) + \beta x_2(n)] = \alpha \Phi[x_1(n)] + \beta \Phi[x_2(n)]$ для всіх α і β довільних постійних і для всіх допустимих впливів $x_1(n)$ і $x_2(n)$; 3) цифровий фільтр називають фізично реалізованим, якщо величина відгуку при $n = n_0$ залежить від значень вхідної послідовності з номерами $n \leq n_0$. Це означає, що імпульсна характеристика $h(n)$ дорівнює нулю при $n < 0$; 4) цифровий фільтр називається стійким тоді і тільки тоді, коли реакція на обмежену дію обмежена, тобто якщо $|x(n)| \leq M < \infty$ при всіх n слід $|y(n)| \leq M_1 < \infty$ при всіх n . Необхідною та достатньою умовою стійкості фільтра є наступна вимога до його імпульсної характеристики:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty.$$

Цифровий фільтр у загальному вигляді представляється так як різницеве рівняння:

$$y(n) = - \sum_{j=1}^{M-1} a_j y(n-j) + \sum_{i=1}^N b_i x(n-i), \quad (4.2)$$

де a_j і b_i – дійсні або комплексні коефіцієнти. Цифрові фільтри прийнято ділити на два класи: нерекурсивні (НФ) та рекурсивні (РФ). Якщо в (4.2) усі коефіцієнти $a_j = 0$, що відповідає відсутності зворотного зв'язку, то фільтр є нерекурсивним та описується рівнянням

$$y(n) = \sum_{i=0}^N b_i x(n-i). \quad (4.3)$$

Якщо в (4.3) хоча б один із коефіцієнтів $a_j \neq 0$, то цифровий фільтр є рекурсивним і являє собою пристрій із зворотним зв'язком. Таким чином, для рекурсивних фільтрів співвідношення

між вхідною послідовністю $\{x(n)\}$ і відгуком фільтра $\{y(n)\}$ може бути записано наступним чином:

$$y(n) = F[y(n-1), y(n-2), \dots, x(n), x(n-1), \dots], \quad (4.4)$$

тобто поточний відлік відгуку $y(n)$ визначається не тільки поточним та попереднім значеннями вхідної послідовності, а й попередніми відліками відгуку. У нерекурсивних фільтрах зв'язок між вхідною послідовністю та відгуком має вигляд

$$y(n) = F[x(n), x(n-1), \dots],$$

тобто поточний відлік відгуку залежить від поточного та попередніх значень вхідної послідовності. Для аналізу систем, що описуються різницевиими рівняннями, широко застосовується z -перетворення. Пряме z -перетворення $X(z)$ послідовності $x(n)$ визначається формулою

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}. \quad (4.5)$$

У різницевих рівняннях істотною операцією є одинична затримка, що описується оператором $1/z$, або z^{-1} (тобто для послідовності $x(n-1)$ z -перетворення матиме вигляд $z^{-1}X(z)$). Передавальною (системною) функцією $H(z)$ цифрового фільтра називається відношення z -перетворень вихідного $Y(z)$ та вхідного $X(z)$ сигнали фільтра. Для рекурсивного та нерекурсивного фільтрів з (4.3) та (4.4), використовуючи (4.5), отримуємо:

$$H_p(z) = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^M a_j z^{-j}};$$

$$H_n(z) = \sum_{i=0}^N b_i z^{-i}.$$

Комплексна частотна характеристика цифрового фільтра, представленого у вигляді різницевого рівняння (4.2), може бути отримана підстановкою для передавальної функції значення $z = e^{j\omega T}$. Для рекурсивного фільтра загального виду частотна характеристика матиме вигляд

$$H_p(\omega) = \frac{\sum_{i=0}^N b_i e^{-j\omega T}}{1 + \sum_{j=1}^M a_j z^{-j\omega T}}.$$

Аналогічно, для нерекурсивного фільтра маємо:

$$H_n(\omega) = \sum_{i=0}^N b_i e^{-i\omega T}.$$

Послідовність виконання роботи

4.1. Вивчити синтаксичну конструкцію функції **filter** мови програмування MATLAB і дати визначення її параметрам **b**, **a**, **x**, **x_filter**.

Синтаксис функції **filter** для дискретної фільтрації сигналів:

```
x_filter = filter(b, a, x)
```

Опис параметрів функції:

b – коефіцієнти поліномів чисельника передавальної функції необхідного дискретного фільтра (для фільтра ковзного середнього з вікном довжиною **windowSize**, коефіцієнти задаються згідно запису **b = (1/windowSize)*ones(1, windowSize)**);

a – коефіцієнти поліномів знаменника передавальної функції необхідного дискретного фільтра (для фільтра ковзного середнього, коефіцієнт приймає значення **a = 1**);

x – вихідний сигнал (в якості вихідного сигналу **x** в даній роботі приймався зашумлений сигнал **x_noise**, який є сумою вихідного сигналу **x** з сигналом білого шуму **noise**, що задається функцією **wgn** відповідно до запису **noise = wgn(size(x, 1), size(x, 2), Pnoise, 'linear')**, де **Pnoise** – потужність шуму (дисперсія));

x_filter – фільтрований сигнал (в якості фільтрованого сигналу **x_filter** в даній роботі розуміється сигнал очищений від впливу білого шуму **x_noise_filter**).

4.2. Змоделювати процес дискретної фільтрації зашумленого сигналу **x_noise** в програмному середовищі MATLAB за допомогою функції **filter**.

Задайте частоту **Fs** (Гц) і крок **ts** (с) дискретизації сигналу **x** тривалістю **T** (с).

```
Fs = 200;
```

```
ts = 1/Fs;
```

```
T = 0:ts:1;
```

Задайте амплітуду **A** (В), частоту **f** (Гц) і початкову фазу **phi** (рад) гармонічного коливання.

```
A = 2;
```

```
f = 6;
```

```
phi = 0;
```

Згенеруйте дискретний сигнал x тривалістю T (с), який змінюється за гармонійним законом згідно заданої амплітуди A (В), частоти f (Гц) і початкової фази ϕ (рад).

```
x = A*sin(2*pi*f*T + phi);
```

Згенеруйте дискретний білий гаусівський (нормальний) шум з потужністю (дисперсія), яка дорівнює **Pnoise**.

```
Pnoise = 0.25;
```

```
noise = wgn(size(x, 1), size(x, 2), Pnoise, 'linear');
```

Згенеруйте дискретний сигнал під впливом білого шуму.

```
x_noise = x + noise;
```

Синтезуйте коефіцієнти поліномів чисельника b і знаменника a передавальної функції дискретного фільтра ковзного середнього з вікном довжиною **windowSize**, який використовується для згладжування (фільтрації) сигналів під впливом шумів.

```
windowSize = 5;
```

```
b = (1/windowSize)*ones(1, windowSize);
```

```
a = 1;
```

Виконайте дискретну фільтрацію зашумленого сигналу x_noise методом ковзного середнього з вікном довжиною **windowSize**, згідно розрахованих коефіцієнтів b і a за допомогою функції **filter**.

```
x_noise_filter = filter(b, a, x_noise);
```

4.3. Побудувати графіки вихідного сигналу x , білого шуму **noise**, зашумленого сигналу x_noise та фільтрованого сигналу x_noise_filter .

Побудуйте графік вихідного сигналу x .

```
figure(1)
```

```
subplot(2,2,1)
```

```
plot(T, x, 'linewidth', 2)
```

```
title('Вихідний сигнал')
```

```
ylabel('Амплітуда')
```

```
xlabel('Час (с)')
```

```
grid on
```

Побудуйте графік білого шуму **noise**.

```
subplot(2,2,2)
```

```
plot(T, noise, 'linewidth', 2)
```

```
title('Білий шум')
```

```
ylabel('Амплітуда')
```

```
xlabel('Час (с)')
```

```
grid on
```

Побудуйте графік зашумленого сигналу **x_noise**.

```
subplot(2,2,4)
plot(T, x_noise, 'linewidth', 2)
title('Зашумлений сигнал')
ylabel('Амплітуда')
xlabel('Час (с)')
grid on
```

Побудуйте графік фільтрованого сигналу **x_noise_filter**.

```
subplot(2,2,3)
plot(T, x_noise_filter, 'linewidth', 2)
title('Фільтрований сигнал')
ylabel('Амплітуда')
xlabel('Час (с)')
grid on
```

4.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Опишіть процес фільтрації дискретних сигналів.
2. Що відбувається з частотною складовою дискретного сигналу у процесі фільтрації?
3. Що таке амплітудно-частотна та фазово-частотна характеристики?
4. Який цифровий фільтр називається стаціонарним?
5. Дайте визначення імпульсній характеристиці фільтра.
6. Якими основними якостями володіє цифровий фільтр?
7. Дайте визначення рекурсивним та нерекурсивним фільтрам. У чому їхня відмінність?

8. Що таке передавальна характеристика фільтру?
9. Яким чином пов'язані між собою цифрова фільтрація та математична згортка?
10. Записати та пояснити синтаксичну конструкцію функції `filter` мови програмування MATLAB і дати визначення її параметрам `b`, `a`, `x`, `x_filter`.

Лабораторна робота 5

КОДУВАННЯ МОВИ МЕТОДОМ ДОВГОТРИВАЛОГО ПЕРЕДБАЧЕННЯ З РЕГУЛЯРНИМ ІМПУЛЬСНИМ ЗБУДЖЕННЯМ

Мета роботи: дослідити процес кодування мови методом довготривалого передбачення з регулярним імпульсним збудженням за допомогою комп'ютерного моделювання використовуючи допоміжну функцію `lpc` з бібліотеки Signal Processing Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Кодування мови виконується згідно методу *RPE-LTP* (*Regular Pulse Excitation Long Term Prediction* – довготривале передбачення з регулярним імпульсним збудженням). Структурна схема кодера показана на рис. 5.1, а параметри кодування представленні в табл. 5.1.

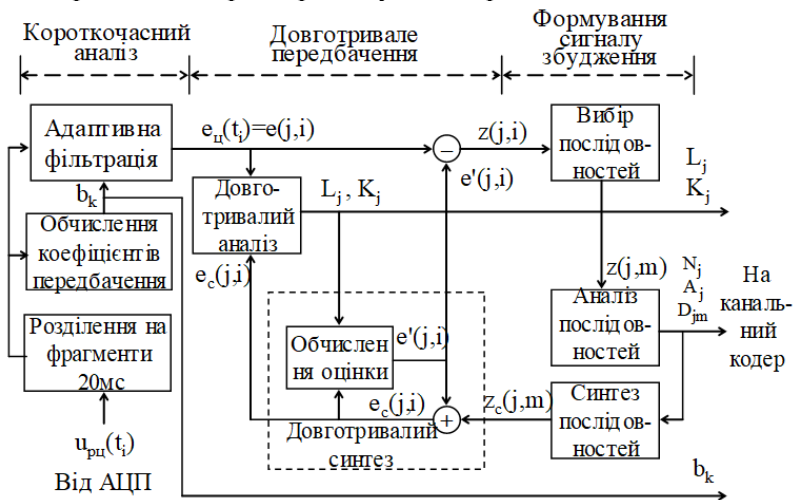


Рис. 5.1. Мовний кодер

Таблиця 5.1

Склад вихідного слова кодера

| Вихідні параметри | Розрядність параметрів | Число біт для фрагменту 20мс |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------|
| Коефіцієнти передбачення b_k | b_1, b_2 - 6 біт, b_3, b_4 - 5 біт, b_5, b_6 - 4 біта, b_7, b_8 - 3 біта | 36 |
| Параметри довготривалого аналізу L_j, K_j ($j = 0, 1, 2, 3$) | L_j - 7 біт, K_j - 2 біта, | 36 |
| Параметри сигналу збудження N_j, A_j, D_{jm} ($m = 0, \dots, 12$) | N_j - 2 біта, A_j - 6 біт, D_{jm} - 3 біта. | 188 |
| Довжина вихідного слова | | 260 |

На вхід кодера від АЦП надходить сигнал у вигляді двійкових 13-розрядних відліків зі швидкістю передачі 104 кбіт/с, визначеною за формулою (3.1).

Кодування включає 3 операції.

1. Короткочасний аналіз. Вхідний сигнал розділяється на фрагменти, тривалість яких дорівнює інтервалу квазістаціонарності мови 20 мс. Кількість відліків в одному інтервалі дорівнює $20 \text{ мс} / 125 \text{ мкс} = 160$ відліків. Короткочасний аналіз полягає в лінійному передбаченні сигналу з використанням математичної моделі 8-го порядку. В кожному інтервалі 20 мс обчислюються коефіцієнти передбачення $b_k, k = \overline{1, 8}$, по яких налаштовується адаптивний фільтр-аналізатор.

На виході цього фільтру формується 160 цифрових відліки залишку передбачення $e_u(t_i)$. Вихідними параметрами короткочасного аналізу є коефіцієнти b_k . Для наступних операцій поточний інтервал 20 мс розділяється на 4 частини по 5 мс, тобто 160 відліків залишку розділяються на 4 групи по 40 відліків. Сигнал залишку в поточному інтервалі позначимо як $e(i, j), j = \overline{0, 3}, i = \overline{0, 39}$, де j – номер групи, i – номер відліку в групі.

2. Довготривале передбачення. Дана операція включає аналіз та синтез. У результаті довготривалого синтезу формується приблизний залишок лінійного передбачення $e_c(j, i) \approx e(j, i)$. Такий самий сигнал згодом синтезується в мовному декодері приймача. Розглянемо виконання довготривалого аналізу, враховуючи, що сигнал

$e_c(j,i)$ вже сформований. У даному сигналі виділяються фрагменти, зміщені на $(-1\dots-120)$ відліків відносно кожної групи сигналу $e(j,i)$. У цих фрагментах проводиться пошук 40 відсотків, найбільш схожих на відповідну групу. Під час пошуку використовується вікно шириною 40 відліків, які зміщуються з кроком один відлік. При кожному зміщенні обчислюється коефіцієнт кореляції відліків i у j -ій групі та у вікні.

$$r_j = \sum_{i=0}^{39} e(j,i) \cdot e_c(j,i)_{\text{у вікні}}, j = \overline{0,3}.$$

При максимальному коефіцієнті $r_{j\max}$ вікно фіксується і визначається зміщення між j -ю групою і вікном, рівним L_j відліків. Позначимо відліки у такому вікні через $e_L(j,i)$. Коефіцієнт згладжування розраховується за формулою:

$$R_j = \frac{r_{j\max}}{\sum_{i=0}^{39} e_L^2(j,i)} = \frac{\sum_{i=0}^{39} e(j,i) \cdot e_L(j,i)}{\sum_{i=0}^{39} e_L^2(j,i)}, j = \overline{0,3}.$$

Значення R_j квантується чотирма рівнями з номерами K_j (табл. 5.2).

Таблиця 5.2

Рівні квантування

| | | | | |
|------------------|---------|-----------|-----------|------|
| R_j | 0...0,2 | 0,2...0,5 | 0,5...0,8 | >0,8 |
| $R_{j\text{кв}}$ | 0,1 | 0,35 | 0,65 | 1 |
| K_j | 0 | 1 | 2 | 3 |

Величини L_j та K_j є вихідними параметрами довготривалого передбачення. Зміщення L_j змінюється в діапазоні $(40\dots120)$ відліків і кодується 7 бітами: $L_j = 40\dots120 = (0101000\dots1111000)_2$. Номер K_j кодується 2 бітами: $K_j = 0\dots3 = (00\dots11)_2$.

Розглянемо роботу блока обчислення оцінки. З сигналу $e_c(j,i)$ з урахуваннями зміщення L_j обираються ті самі 40 відліків $e_L(j,i)$, що потрапили у вікно при виконанні довготривалого аналізу. Ці відліки помножуються на коефіцієнт $R_{j\text{кв}}$, що обирається за номером K_j . У результаті формується сигнал $e'(j,i) = R_{j\text{кв}} \cdot e_L(j,i)$, який є приблизною копією або оцінкою сигналу $e(j,i)$. Похибка синтезу або залишок довготривалого передбачення розраховується за формулою:

$$Z(j,i) = e(j,i) - e'(j,i) = e(j,i) - R_{j\text{кв}} \cdot e_L(j,i).$$

3. Формування сигналу збудження. Кожна група залишку $Z(j,i)$, $j = \overline{0,3}$, $i = \overline{0,39}$ розділяється на 4 підгрупи по 13 відліків, що повторюються через однакові проміжки часу. Послідовність з номером N_j позначимо через $Z(j,m)$, де $m = N_j + 3l$, $l = \overline{0,12}$. Для подальшої обробки обирається одна з чотирьох послідовностей з максимальною потужністю $P_j = \max_m \sum Z^2(j,m)$. Серед тринадцяти відліків визначається максимальний відлік $|Z(j,m)_{\max}|$, який використовується для нормування:

$$B(j,m) = \frac{Z(j,m)}{|Z(j,m)_{\max}|}.$$

Квантовані відліки $|Z(j,m)_{\max}|_{\text{кв}}$ мають 64 значення з номерами $A_j = 0 \dots 63$, квантовані відліки $B(j,m)_{\text{кв}}$ мають 8 значень з номерами $D_j = 0 \dots 7$.

Вихідними параметрами є номери N_j , A_j і D_{jm} , які кодуються 2, 6 і 3 бітами:

$$\begin{aligned} N_j &= (0 \dots 3) = (00 \dots 11)_2; \\ A_j &= (0 \dots 63) = (000000 \dots 111111)_2; \\ D_{jm} &= (0 \dots 7) = (000 \dots 111)_2. \end{aligned}$$

Ці величини також використовуються при формуванні сигналу $e_c(j,i)$. Для цього спочатку виконується синтез цих послідовностей, кожна з яких являє собою 13 відліків $Z_c(j,m)$. Часове місце знаходження цих відліків задається відповідно до номерів N_j . За номерами A_j і D_{jm} обираються квантовані рівні та обчислюється значення $Z_c(j,m) = |Z(j,m)_{\max}|_{\text{кв}} \cdot B(j,m)_{\text{кв}}$. Далі виконується довготривалий синтез, за якого відліки $Z_c(j,m)$ сумуються з оцінкою $e'(j,i)$: $e_c(j,i) = Z_c(j,m) + e'(j,i)$.

В результаті формується приблизний залишок лінійного передбачення $e_c(j,i) \approx e(j,i)$. Цей сигнал в наступному інтервалі 20 мс буде використаний при довготривалому аналізі і при формуванні оцінки $e'(j,i)$. Кодування в першому інтервалі 20 мс потребує нульового сигналу $e_c(j,i)$ до початку кодування. Для цього одразу після увімкнення передавача на вхід кодера подається визначена цифрова послідовність, що встановлює кодер в потрібне початкове положення.

Тоді згідно табл. 5.1, де приведений склад вихідного слова кодера для кожного мовного фрагменту в часовому діапазоні 20 мс з розрядністю кодового слово 260 біт – розрахуємо характеристики кодера:

– частота формування слів: 1 слово/20 мс=50 слів/с;

– швидкість передавання вихідного сигналу:

$$n/\Delta t = 260 \text{ біт}/20 \text{ мс} = 13 \text{ кбіт/с};$$

– коефіцієнт стиснення сигналу АЦП: $\frac{104 \text{ кбіт/с}}{13 \text{ кбіт/с}} = 8$, тобто ко-

дер формує за одиницю часу у 8 разів менше біт, ніж надходить на його вхід.

Структурна схема декодера показана на рис. 5.2.

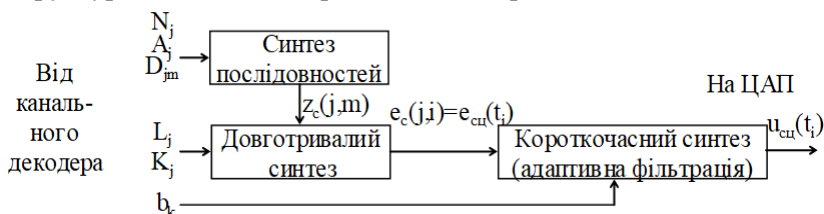


Рис. 5.2. Мовний декодер

Синтез послідовностей та довготривалий синтез виконуються аналогічно операціям кодера. Сигнал $e_c(j,i)$ в декодері співпадає з таким самим сигналом у кодері, є приблизним залишком лінійного передбачення і служить сигналом збудження для синтезуючого фільтра. Фільтр налаштовується по коефіцієнтам b_k . На виході декодера отримують синтезований цифровий мовний сигнал $U_{\text{cut}}(t_i)$, що подається на ЦАП. Цей сигнал відрізняється від сигналу $U_{\text{rc}}(t_i)$, що передається, внаслідок нерівності $e_c(j,i) \neq e(j,i)$, яка обумовлена: 1) вибором 13 відліків $Z(j,m)$ з 40 відліків $Z(j,i)$; 2) квантуванням величин $|Z(j,m)_{\text{max}}|$ і $B(j,m)$.

Похибка мовного синтезу зменшується через операції довготривалого передбачення, яка забезпечує гарну якість мови на виході приймача GSM. Вид кодованого звукового сигналу (мова або шум) не впливає на роботу розглянутого кодера.

Послідовність виконання роботи

5.1. Вивчити синтаксичну конструкцію функції **lpc** мови програмування MATLAB і дати визначення її параметрам **x**, **p**, **b**, **g**.

Синтаксис функції **lpc** для розрахунку коефіцієнтів фільтра лінійного передбачення автокореляційним методом:

$$[\mathbf{b}, \mathbf{g}] = \text{lpc}(\mathbf{x}, \mathbf{p})$$

Опис параметрів функції:

x – вихідний сигнал;

p – порядок фільтра лінійного передбачення;

b – коефіцієнти нерекурсивної частини фільтра лінійного передбачення **p**-го порядку (чисельник передавальної функції), які знаходяться шляхом мінімізації середньоквадратичної помилки передбачення **g** (метод найменших квадратів), що найкращим чином передбачає поточне значення сигналу **x**, по його попередніх відліках;

g – середньоквадратична помилка передбачення.

5.2. Змодельувати процес дискретної фільтрації вихідного **x**, шумового **noise** та зашумленого **x_noise** сигналів на основі лінійного передбачення в програмному середовищі MATLAB за допомогою функції **lpc**.

Задайте частоту **F_s** (Гц) і крок **ts** (с) дискретизації сигналу **x** тривалістю **T** (с).

```
Fs = 200;
```

```
ts = 1/Fs;
```

```
T = 0:ts:1;
```

Задайте амплітуду **A** (В), частоту **f** (Гц) і початкову фазу **phi** (рад) гармонічного коливання.

```
A = 2;
```

```
f = 5;
```

```
phi = pi/3;
```

Згенеруйте дискретний сигнал **x** тривалістю **T** (с), який змінюється за гармонійним законом згідно заданої амплітуди **A** (В), частоти **f** (Гц) і початкової фази **phi** (рад).

```
x = A*sin(2*pi*f*T + phi);
```

Згенеруйте дискретний білий гаусівський (нормальний) шум з потужністю (дисперсія), яка дорівнює **Pnoise**.

```
Pnoise = 0.5;
```

```
noise = wgn(size(x, 1), size(x, 2), Pnoise, 'linear');
```

Згенеруйте дискретний сигнал під впливом білого шуму.

```
x_noise = x + noise;
```

Розрахуйте коефіцієнти нерекурсивної частини фільтрів лінійного передбачення **p**-го порядку **b_x**, **b_noise** та **b_x_noise** (чисельники передавальних функцій) шляхом мінімізації середньоквадратичних помилок передбачення **g_x**, **g_noise** та **g_x_noise** (метод найменших квадратів), що найкращим чином передбачає поточне значення сигналів **x**, **noise** та **x_noise** по їх попередніх відліках.

```
p = 8;
```

```
[b_x, g_x] = lpc(x, p);
[b_noise, g_noise] = lpc(noise, p);
[b_x_noise, g_x_noise] = lpc(x_noise, p);
```

Виконайте обробку вихідного **x**, шумового **noise** та зашумленого **x_noise** сигналів фільтрами лінійного передбачення **p**-го порядку, згідно розрахованих коефіцієнтів нерекурсивної частини фільтрів **b_x**, **b_noise**, **b_x_noise** (чисельники передавальних функцій) та рекурсивної частини фільтрів **a_x**, **a_noise**, **a_x_noise** (знаменники передавальних функцій) за допомогою функції **filter**.

```
b_x = [0 -b_x(2:end)];
a_x = 1;
x_filter = filter(b_x, a_x, x);
b_noise = [0 -b_noise(2:end)];
a_noise = 1;
noise_filter = filter(b_noise, a_noise, noise);
b_x_noise = [0 -b_x_noise(2:end)];
a_x_noise = 1;
x_noise_filter = filter(b_x_noise, a_x_noise, x_noise);
```

Розрахуйте помилки передбачення вихідного **e_x**, шумового **e_noise** та зашумленого **e_x_noise** сигналів.

```
e_x = x - x_filter;
e_noise = noise - noise_filter;
e_x_noise = x_noise - x_noise_filter;
```

5.3. Побудувати графіки вихідного **x_filter**, шумового **noise_filter** та зашумленого **x_noise_filter** сигналів передбачення та їх помилок передбачення **e_x**, **e_noise**, **e_x_noise**.

Побудуйте графіки вихідного **x** та передбаченого **x_filter** сигналів.

```
figure(1)
subplot(2,2,1)
plot(T, x, T, x_filter, '--', 'linewidth', 2)
title('Вихідний і передбачений сигнали')
ylabel('Амплітуда')
xlabel('Час (с)')
legend('Вихідний сигнал', 'Передбачений сигнал')
grid on
```

Побудуйте графіки шумового **noise** та передбаченого **noise_filter** сигналів.

```
figure(1)
```

```

subplot(2,2,2)
plot(T, noise, T, noise_filter, '--', 'linewidth', 2)
title('Шумовий і передбачений сигнали')
ylabel('Амплітуда')
xlabel('Час (с)')
legend('Шумовий сигнал', 'Передбачений сигнал')
grid on

```

Побудуйте графіки зашумленого **x_noise** та передбаченого **x_noise_filter** сигналів.

```

figure(1)
subplot(2,2,3)
plot(T, x_noise, T, x_noise_filter, '--', 'linewidth', 2)
title('Зашумлений і передбачений сигнали')
ylabel('Амплітуда')
xlabel('Час (с)')
legend('Зашумлений сигнал', 'Передбачений сигнал')
grid on

```

Побудуйте графіки помилок передбачення вихідного **e_x**, шумового **e_noise** та зашумленого **e_x_noise** сигналів.

```

figure(1)
subplot(2,2,4)
plot(T, e_x, T, e_noise, 'r--', T, e_x_noise, 'g:', 'linewidth', 2)
title('Помилки передбачення вихідного, шумового і зашумленого сигналів')
ylabel('Амплітуда')
xlabel('Час (с)')
legend('Помилка передбачення вихідного сигналу', 'Помилка передбачення шумового сигналу', 'Помилка передбачення зашумленого сигналу')
grid on

```

5.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.

3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.

4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. З яких етапів складається процес кодування мови, яке виконується згідно методу RPE-LTP?

2. Як ви розумієте поняття довготривалий синтез, та для яких цілей він використовується в методі кодування мови RPE-LTP?

3. Дайте визначення терміну короткочасний аналіз сигналів.

4. Поясніть необхідність зменшувати швидкість передачі мови.

5. У чому суть методу лінійного передбачення?

6. Яку роль виконує короткочасний аналіз, довготривале передбачення та формування сигналу збудження у процесі кодування мови?

7. Яка швидкість передавання та коефіцієнт стиснення вихідного мовного потоку даних в кодері RPE-LTP.

8. Яким чином формується сигнал збудження в RPE-LTP?

9. Поясніть процес квантування згідно методу RPE-LTP.

10. Записати та пояснити синтаксичну конструкцію функції `lpc` мови програмування MATLAB і дати визначення її параметрам **x**, **p**, **b**, **g**.

Лабораторна робота 6

ЦИКЛІЧНЕ КОДУВАННЯ

Мета роботи: дослідити процес циклічного кодування та декодування за допомогою комп'ютерного моделювання використовуючи допоміжні функції **encode** і **decode** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

У *циклічних кодах* основною операцією кодування і декодування є *циклічний зсув (перестановка)* – наприклад, останній символ кодової комбінації займає місце першого, перший зсувається на місце другого, другий – на місце третього і т.д. до тих пір, поки передостанній символ не займе місце останнього.

На рис. 6.1 представлено приклади операції циклічного зсуву на один розряд.

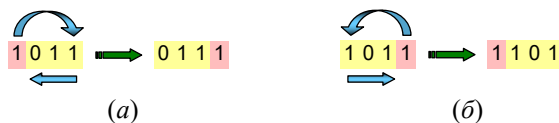


Рис. 6.1. Приклади циклічного зсуву кодової комбінації 1011 на один розряд ліворуч (а) і праворуч (б)

Лінійний $[n, k]$ -код є *циклічним*, якщо в результаті будь-якого циклічного зсуву вектора дозволеної кодової комбінації отримують вектор іншої дозволеної комбінації, тобто якщо $S = s_0, s_1, \dots, s_{n-2}, s_{n-1}$ – це вектор дозволеної кодової комбінації, то $S(1) = s_{n-1}, s_0, s_1, \dots, s_{n-2}$ – також є вектором дозволеної кодової комбінації. У загальному випадку для циклічних кодів є справедливим запис дозволеної кодової комбінації, одержаної шляхом i зсувів:

$$S(i) = s_{n-i}, s_{n-i+1}, \dots, s_{n-1}, s_0, s_1, s_2, \dots, s_{n-i-1}.$$

У теорії циклічних кодів використовується представлення векторів кодових комбінацій у вигляді степеневих поліномів. Арифметичні операції додавання і віднімання під час множення або ділення поліномів виконуються за модулем 2.

Оскільки

$$x^v \oplus x^v = x^v \cdot (1 \oplus 1) = 0, \text{ то } x^v = -x^v,$$

тобто символи двійкової комбінації (мономи) знаку не мають.

Усі дозволені кодові комбінації циклічного коду переходять одна в одну шляхом циклічних зсувів. При цьому усі вони діляться без залишку на свій породжуючий поліном. Для побудови дозволених КК використовують *неприводимі поліноми по модулю 2*. Останні не розкладаються на добуток поліномів меншого степеня і, таким чином, діляться націло лише на себе і, звичайно, на 1. У контексті розглянутих вище кодів такі поліноми називають також *утворюючими (генеруючими або породжуючими)* поліномами циклічного коду. Загальний запис генеруючого поліному $G(x)$ степеня r :

$$G(x) = g_0 \cdot x^0 + g_1 \cdot x^1 + g_2 \cdot x^2 + \dots + g_{r-1} \cdot x^{r-1} + g_r \cdot x^r,$$

де g – коефіцієнти з поля $GF(m_k)$. Якщо код побудований над полем $GF(2)$, то коефіцієнти приймають значення 0 або 1 і код є двійковим. При цьому є обов'язковою виконання умов: $g_0 = 1$ і $g_r = 1$. В іншому випадку, якщо, наприклад, $g_0 = 0$, поліном $G(x)$ крім себе і 1 буде ділитися націло ще й на x і, таким чином, не буде неприводимим. Якщо ж $g_r = 0$, навіть якщо поліном і примітивний, його степінь є меншою за r . Крім того, поліном $G(x)$ степеня $r = n - k$

буде генеруючим поліномом циклічного коду лише у випадку, коли на нього ділитиметься без залишку біном x^n+1 .

Розглянемо алгоритм формування дозовлених КК циклічного коду. Нехай заздалегідь обраний примітивний, неприводимий поліном $G(x)$ має найвищу степінь r . Тоді правило отримання дозовлених комбінацій може бути представлено тотожністю

$$\frac{A(x) \cdot x^r}{G(x)} = T(x) + \frac{M(x)}{G(x)}, \quad (6.1)$$

де $A(x)$ – інформаційний поліном; $T(x)$ – ціла частина від ділення; $M(x)$ – залишок від ділення.

Помножимо обидві частини (6.1) на $G(x)$:

$$A(x) \cdot x^r = T(x) \cdot G(x) + M(x), \quad (6.2)$$

звідки дозовлена кодова комбінація визначається виразом:

$$S(x) = T(x) \cdot G(x) = A(x) \cdot x^r + M(x). \quad (6.3)$$

Розглянемо приклад. Нехай розрядність інформаційного повідомлення $k = 4$, а саме воно виражається комбінацією $A = \| 1011 \|$ або поліномом $A(x) = 1 + x^2 + x^3$. Генеруючий поліном, наприклад, такий ($r = 3$):

$$G(x) = 1 + x + x^3.$$

Необхідно знайти дозовану кодову комбінацію $S(x)$ циклічного коду для повідомлення $A(x)$.

Спочатку визначимо ліву частину співвідношення (6.2), помноживши інформаційний поліном $A(x)$ на $x^{r=3}$, що відповідає зсуву вектора A у бік старших розрядів на r позицій

$$A(x) \cdot x^r = (1 + x^2 + x^3) \cdot x^3 = x^3 + x^5 + x^6. \quad (6.4)$$

Тоді ліва частина тотожності (6.1):

$$\begin{aligned} \frac{A(x) \cdot x^r}{G(x)} = & \begin{array}{l} \oplus \frac{x^6 + x^5 + x^3}{x^6 + x^4 + x^3} \left| \begin{array}{l} x^3 + x + 1 \\ x^3 + x^2 + x + 1 \end{array} \right. \leftarrow G(x) \\ \oplus \frac{x^5 + x^4}{x^5 + x^3 + x^2} \\ \oplus \frac{x^4 + x^3 + x^2}{x^4 + x^2 + x} \\ \oplus \frac{x^3 + x}{x^3 + x + 1} \\ \oplus 1 \leftarrow M(x) \end{array} \end{aligned}$$

Враховуючи отриманий після ділення залишок $M(x) = 1$, а також результат (6.4), за формулою (6.3) одержуємо поліном дозволеної кодової комбінації – шляхом *конкатенації* (об'єднання) отриманого вектора залишку $M(x)$ та інформаційного поліному $A(x)$:

$$S(x) = M(x) + A(x) \cdot x^r = 1 + x^3 + x^5 + x^6,$$

або, у векторному записі, $S = \left\| \underbrace{1 \ 0 \ 0}_{M(x)} \underbrace{1 \ 0 \ 1 \ 1}_{A(x)} \right\|$.

Виконаємо перевірку (при цьому ділення $S(x)$ на $G(x)$ повинне відбутися без залишку):

$$\begin{aligned} \frac{S(x)}{G(x)} = T(x) = & \oplus \frac{x^6 + x^5 + x^3 + 1}{x^6 + x^4 + x^3} \Big| \frac{x^3 + x + 1}{x^3 + x^2 + x + 1} \leftarrow G(x) \\ & \oplus \frac{x^5 + x^4 + 1}{x^5 + x^3 + x^2} \\ & \oplus \frac{x^4 + x^3 + x^2 + 1}{x^4 + x^2 + x} \\ & \oplus \frac{x^3 + x + 1}{x^3 + x + 1} \\ & 0 \end{aligned}$$

Отже, кодова комбінація $S(x)$ належить циклічному коду [7, 4] на основі поліному $G(x) = 1 + x + x^3$ і є дозволеною.

Розглянемо *алгоритм декодування циклічного коду*. Виконавши ділення прийнятої кодової комбінації на генеруючий примітивний поліном, можна *виявити* наявність помилок – помилки є, якщо присутній залишок від ділення.

Для *виправлення* помилок використовують наступний алгоритм:

1) Прийнятну кодову комбінацію $Z(x) = S(x) + E(x)$ ділять на генеруючий поліном $G(x)$.

2) Визначають вагу залишку від ділення $M(x) = w$.

3) Якщо $w = 0$, то приймається рішення про відсутність помилок у прийнятій КК.

4) Якщо $0 < w \leq q_c$, то прийнятну кодову комбінацію $Z(x)$ додають по модулю 2 до отриманого у п.1 залишку $M(x)$. Результат додавання буде виправленою комбінацією, тобто $S(x) = Z(x) + M(x)$.

5) Якщо $w > q_c$, виконують циклічний зсув кодової комбінації $Z(x)$ *ліворуч* (у бік молодших розрядів) на один розряд. Одержану

при цьому комбінацію ділять на генеруючий поліном $G(x)$ і визначають вагу нового залишку від ділення $M_1(x) - w_1$.

6) Якщо $w_1 \leq q_c$, то отриману циклічним зсувом кодову комбінацію $Z_1(x)$ додають по модулю 2 до залишку $M_1(x)$ і виконують циклічний зсув *праворуч* (у бік старших розрядів) на один розряд, одержуючи в результаті виправлену КК.

7) Якщо в результаті першого циклічного зсуву (п.5) $w_1 > q_c$, то операцію циклічного зсуву ліворуч і ділення на $G(x)$ одержаних в результаті комбінацій повторюють доти, доки вага залишку w_v не зрівняється з q_c .

8) Коли вага залишку стала рівною q_c , останню отриману циклічними зсувами кодову комбінацію $Z_v(x)$ додають по модулю 2 до останнього залишку $M_v(x)$ і отриманий результат циклічно зсувають праворуч на стільки розрядів v , на скільки раніше зсували ліворуч прийняту комбінацію $Z(x)$. У результаті отримують виправлену кодову комбінацію.

Розглянемо приклад циклічного декодування. Припустимо, що в результаті дії вектору завад $E = \|0\ 0\ 0\ 1\ 0\ 0\ 0\|$ на передану кодову комбінацію $S = \|1\ 0\ 0\ 1\ 0\ 1\ 1\|$ було прийнято спотворену кодову комбінацію $Z = \|1\ 0\ 0\ 0\ 0\ 1\ 1\|$, що відповідає поліному $Z(x) = 1 + x^5 + x^6$. Код системи (код Хемінга, типу [7, 4]) гарантовано виявляє дві ($q_d = 2$) і виправляє одну ($q_c = 1$) помилку. Генеруючий поліном системи – $G(x) = 1 + x + x^3$.

Розділимо прийняту кодову комбінацію на генеруючий поліном:

$$\begin{array}{l} \frac{Z(x)}{G(x)} = \\ \oplus \frac{x^6 + x^5 + 1}{x^6 + x^4 + x^3} \quad \left| \frac{x^3 + x + 1}{x^3 + x^2 + x} \leftarrow G(x) \right. \\ \oplus \frac{x^5 + x^4 + x^3 + 1}{x^5 + x^3 + x^2} \\ \oplus \frac{x^4 + x^2 + 1}{x^4 + x^2 + x} \\ x + 1 \quad \leftarrow M(x) \end{array}$$

Оскільки вага залишку $M(x)$ $w = 2 > q_c$, то виконуємо циклічний зсув комбінації $Z(x)$ на один розряд ліворуч: $Z_1 = \|0\ 0\ 0\ 1\ 1\ 1\|$ або $Z_1(x) = x^4 + x^5 + x^6$, і ділимо отриману комбінацію $Z_1(x)$ на поліном $G(x)$:

$$\frac{Z_1(x)}{G(x)} = \begin{array}{l} \oplus \frac{x^6 + x^5 + x^4}{x^6 + x^4 + x^3} \left| \frac{x^3 + x + 1}{x^3 + x^2 + x} \leftarrow G(x) \right. \\ \oplus \frac{x^5 + x^3}{x^5 + x^3 + x^2} \\ \left. x^2 \leftarrow M_1(x) \right. \end{array}$$

Вага отриманого залишку $M_1(x)$ $w_1 = 1 = q_c$.

Тому кодову комбінацію $Z_1(x)$ додаємо по модулю 2 до залишку $M_1(x)$:

$$Z_1(x) + M_1(x) = x^2 + x^4 + x^5 + x^6,$$

або

$$Z_1 + M_1 = \|0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1\|.$$

Одержану кодову комбінацію $Z_1 + M_1$ циклічно зсуваємо праворуч на один розряд, одержуючи в результаті виправлену кодову комбінацію:

$$\|1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1\| = S.$$

Послідовність виконання роботи

6.1. Вивчити синтаксичну конструкцію функцій **encode** та **decode** мови програмування MATLAB і дати визначення їх параметрам **data**, **n**, **k**, **'cyclic/binary'**, **gpol**, **encData**, **trt**, **decData**.

Синтаксис функцій **encode** і **decode** для циклічного кодування та декодування:

encData = **encode(data, n, k, 'cyclic/binary', gpol)**

decData = **decode(encData, n, k, 'cyclic/binary', gpol, trt)**

Опис параметрів функцій:

data – вихідне повідомлення;

n – довжина кодового слова (кодової комбінації);

k – довжина блоку повідомлення (інформаційної комбінації);

'cyclic/binary' – запускає циклічне кодування, де вихідне, кодоване та декодоване повідомлення знаходяться в двійковій системі числення (бітовий потік);

gpol – породжувальний поліном для циклічного коду, який генерується за допомогою функції **cyclpoly(n, k)**;

encData – кодоване повідомлення;

trt – таблиця декодування, яка генерується за допомогою комбінації функції **syndtable(parmat)**, де функція **parmat** = **cyclgen(n, gpol)** формує перевірочну матрицю для циклічного коду;

decData – декодоване повідомлення.

6.2. Змоделювати процес циклічного кодування цифрового сигналу (бітового потоку) **data** і його подальшого декодування в програмному середовищі MATLAB за допомогою функцій **encode** та **decode** відповідно.

Установіть довжину кодової комбінації **n** циклічного коду та довжину блоку повідомлення **k**.

```
n = 15;
```

```
k = 5;
```

Згенеруйте вихідне повідомлення **data** (потік бітів) довжиною **k**.

```
data = randi([0 1], k, 1);
```

Згенеруйте породжувальний поліном **gpoly** для циклічного коду.

```
gpoly = cyclpoly(n, k);
```

Сформуйте перевірочну матрицю **parmat** для циклічного коду.

```
parmat = cyclgen(n, gpoly);
```

Створіть таблицю декодування **trt**, використовуючи перевірочну матрицю **parmat**.

```
trt = syndtable(parmat);
```

Виконайте циклічне кодування вихідного повідомлення **data** за допомогою функції **encode**.

```
encData = encode(data, n, k, 'cyclic/binary', gpoly);
```

Введіть помилки в 4-й і 7-й біти кодованого повідомлення **encData**.

```
encDataError = encData;
```

```
encDataError(4) = ~encDataError(4);
```

```
encDataError(7) = ~encDataError(7);
```

Виконайте циклічне декодування кодованого повідомлення з бітовими помилками **encDataError** за допомогою функції **decode**.

```
decData = decode(encDataError, n, k, 'cyclic/binary', gpoly, trt);
```

Обчисліть кількість помилкових біт, порівнюючи вихідне **data** та декодоване **decData** повідомлення.

```
numerr = biterr(data, decData);
```

6.3. Побудувати графіки вихідного **data**, кодованого **encData**, кодованого з бітовими помилками **encDataError** та декодованого **decData** повідомлень.

Побудуйте графік вихідного повідомлення **data**.

```
figure(1)
```

```
subplot(2,2,1)
```

```
stairs([0:length(data)], [data; data(length(data))], 'linewidth', 2)
title('Вихідне повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(data) -1 2])
grid on
```

Побудуйте графік кодованого повідомлення **encData**.

```
subplot(2,2,2)
stairs([0:length(encData)], [encData;
encData(length(encData))], 'linewidth', 2)
title('Кодоване повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(encData) -1 2])
grid on
```

Побудуйте графік кодованого повідомлення з бітовими помилками **encDataError**.

```
subplot(2,2,4)
stairs([0:length(encDataError)], [encDataError;
encDataError(length(encDataError))], 'linewidth', 2)
title('Кодоване повідомлення з бітовими помилками')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(encDataError) -1 2])
grid on
```

Побудуйте графік декодованого повідомлення **decData**.

```
subplot(2,2,3)
stairs([0:length(decData)], [decData;
decData(length(decData))], 'linewidth', 2)
title('Декодоване повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(decData) -1 2])
grid on
```

6.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Наведіть алгоритм формування дозволених кодових комбінацій циклічного коду.
2. Що таке породжуючий поліном в циклічному кодуванні?
3. Як формуються дозволени кодові комбінації циклічного коду?
4. Сформулюйте правило отримання дозволених кодових комбінацій.
5. Проаналізуйте вплив структури циклічного $[n, k]$ коду на його корегувальні властивості.
6. Наведіть опис алгоритму виправлення помилки у прийнятій кодовій комбінації циклічного коду.
7. Розкрийте сутність циклічного кодування і вимоги до кодуєчого поліному циклічного коду.
8. Знайдіть дозволена кодова комбінація циклічного коду для інформаційного вектора $A = 10010000001$, якщо кодуєчий поліном $G(x) = 1 + x + x^4$. Виконайте перевірку одержаної КК на дозволенисть.
9. Виправте помилку, якщо прийнята кодова комбінація циклічного коду $Z = 1000101$, а породжуючий поліном – $G(x) = 1 + x + x^3$.
10. Записати та пояснити синтаксичні конструкції функцій **encode** та **decode** мови програмування MATLAB і дати визначення їх параметрам **data**, **n**, **k**, **'cyclic/binary'**, **gpol**, **encData**, **trt**, **decData**.

Лабораторна робота 7

ЗГОРТКОВЕ КОДУВАННЯ

Мета роботи: дослідити процес згорткового кодування та декодування за допомогою комп'ютерного моделювання використовуючи допоміжні функції **convenc** і **vitdec** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

У неперервних згорткових кодах (інші назви – *рекурентні* або *ланцюгові* коди) інформаційна послідовність на окремі блоки (кодові комбінації) не розбивається, а перевірочні розряди розміщуються у певному (рекурентному) порядку між інформаційними. Процеси кодування і декодування також здійснюються у неперервному режимі й підпорядковуються одному й тому самому рекурентному співвідношенню.

Суттєвою особливістю неперервних кодів є більша порівняно з блоковими кодами можливість обрання інформаційних символів, які беруть участь у формуванні перевірочних символів. Якщо у блокових кодах перевірочні символи й контрольовані ними інформаційні символи мають бути зосереджені у межах кодової комбінації, то у неперервних кодах є можливість їх «розосередити» по всій довжині інформаційної послідовності (за такою ознакою блокові коди належать до *зосереджених*, а неперервні – до *розосереджених* кодів).

У випадку впливу *некорельованих* завад, коли помилки в окремих символах інформаційної послідовності можна розглядати як випадкові незалежні події, відмінності між блоковими і неперервними кодами за ознакою рівня зосередженості перевірочних символів є несуттєвими. Але якщо на канал зв'язку діє *корельована* завада, що призводить до появи згрупованих помилок, які зачіпають цілий ланцюг символів, розосереджені коди мають суттєві переваги: якщо забезпечити достатнє рознесення між інформаційними й відповідними їм перевірочними символами, то стає можливим виправити всі помилки у межах цієї групи помилок незалежно від їх кратності.

Ідея побудови рекурентного коду Фінка полягає в наступному: до потоку інформаційних символів $A = a_1 a_2 a_3 \dots a_k \dots$ включаються корегувальні символи ρ_i таким чином, що між кожними двома інформаційними символами розміщується один корегувальний:

$$S = a_1 \rho_1 a_2 \rho_2 a_3 \rho_3 a_4 \dots a_{k-1} \rho_{k-1} a_k \rho_k a_{k+1} \rho_{k+1} a_{k+2} \dots \quad (7.1)$$

Є очевидним, що при цьому *надмірність* коду r/n дорівнює $1/2$.

Інформаційні символи коду визначаються повідомленням, яке передається, а корегувальні формуються за наступним правилом:

$$\rho_k = a_{k-\Delta} \oplus a_{k+\Delta+1}, \quad (7.2)$$

де Δ – довільне, визначене алгоритмом кодування ціле число, яке зветься *кроком коду*, $\Delta = 0, 1, 2, \dots$. Права частина рекурентного спів-

відношення (7.2) являє собою «згортку» відповідної ділянки інформаційної послідовності, тому неперервні коди також називають *згортковими*. Очевидно, що у випадку помилкового прийому деякого корегувального символу ρ_i у прийнятій послідовності формула (7.2) не виконуватиметься лише для одного значення $k = i$. У випадку ж прийому з помилкою інформаційного символу a_i формула (7.2) не виконуватиметься при двох значеннях k :

$$k' = i - \Delta - 1; \quad k'' = i + \Delta,$$

оскільки від значення одного символу a_i залежать два корегувальні символи. Звідси легко вивести *правило виправлення помилок* при декодуванні рекурентного коду. У прийнятій кодовій послідовності для кожного ρ_k перевіряється співвідношення (7.2). Якщо воно не виконується при двох значеннях k ($k = k'$ і $k = k''$) і при цьому $k'' - k' = 2 \cdot \Delta + 1$, то інформаційний символ $a_{i=k'+\Delta+1=k''-\Delta}$ повинен бути змінений на протилежний (інверсний). За способом декодування згорткові коди можна поділити на два класи: коди із *синдромним корегуванням* і коди *послідовного декодування*. Синдромне корегування використовується тільки систематичними неперервними кодами, до яких, зокрема, належить і код Фінка. Спосіб корегування при цьому ґрунтується на формуванні з прийнятої кодової послідовності виду (7.1) *контрольних сум*, утворюваних за наступним правилом:

$$c_k = \rho'_k \oplus a'_{k-\Delta} \oplus a'_{k+\Delta+1}. \quad (7.3)$$

З виразів (7.2) і (7.3) видно, що перевірочні символи ρ_k одержуються підсумовуванням симетрично розміщених відносно відповідних їм інформаційних символів $a'_{k-\Delta}$ і $a'_{k+\Delta+1}$ кодової послідовності, віддалених один від одного на $k + \Delta + 1 - (k - \Delta) = 2 \cdot \Delta + 1$:

$$\dots a'_{k-2\Delta-1} \dots \rho'_{k-\Delta-1} \dots a'_k \dots \rho'_{k+\Delta} \dots a'_{k+2\Delta+1} \dots$$

Оберемо крок Δ таким чином, щоб виконувалася наступна умова: якщо інформаційний символ a_k зазнає впливу групи помилок, тоді ні ця, ні сусідні групи помилок не повинні захоплювати ані *перевірочні* символи $\rho_{k+\Delta}$ і $\rho_{k-\Delta-1}$ (у формуванні яких бере участь символ a_k), ані *інформаційні* символи $a_{i+2\Delta+1}$ і $a_{i-2\Delta-1}$ (які беруть участь у формуванні тих самих перевірочних символів). Як видно з рис. 7.1, для того, щоб одна й та сама група помилок одночасно з

символом a_k не захопила ще й символ $\rho_{k+\Delta}$ або $\rho_{k-\Delta-1}$, її *максимальна довжина* l_{\max} повинна задовольняти умові

$$l_{\max} \leq 2 \cdot \Delta. \quad (7.4)$$

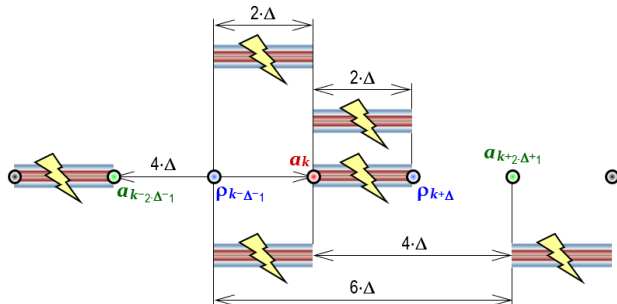


Рис. 7.1. Діаграма побудови коду Фінка

Для того щоб кінець попередньої і початок наступної групи помилок не змогли захопити інформаційні символи $a_{k-2\Delta-1}$ і a_k або a_k і $a_{k+2\Delta+1}$, *мінімальний інтервал* між початками сусідніх груп має задовольняти умові

$$L_{\min} \geq 6 \cdot \Delta + 1. \quad (7.5)$$

У разі виконання умов (7.4) і (7.5), спотворення інформаційного символу a_k буде водночас супроводжуватися одночасним оберненням на одиницю пари складових синдрому помилок:

$$c_{k-\Delta-1} = \rho'_{k-\Delta-1} \oplus a'_{k-2\Delta-1} \oplus a'_k; \quad (7.6)$$

$$c_{k+\Delta} = \rho'_{k+\Delta} \oplus a'_k \oplus a'_{k+2\Delta+1}, \quad (7.7)$$

індекси яких відрізняються на $2 \cdot \Delta + 1$ (решта символів у правих частинах рівнянь (7.6) і (7.7) не будуть підлягати дії груп помилок). Неважко переконатися, що у випадку дотримання умов (7.4) і (7.5) спотворення перевірного символу зумовлює обернення на одиницю лише складової синдрому з тим самим індексом, а складові синдрому з індексом, який відрізняється у той чи інший бік на $2 \cdot \Delta + 1$, зберігають нульове значення.

Отже, *правила корегування коду Фінка* можна звести до наступного: якщо складова синдрому $c_j \neq 0$, а складова $c_{j-2\Delta-1} = 0$, то виправлення раніше прийнятих інформаційних символів не проводиться; якщо ж $c_j \neq 0$ і $c_{j-2\Delta-1} \neq 0$, то необхідно виправити інформаційний символ $a'_{j-\Delta}$ на інверсний.

Код Фінка дозволяє виправляти групи помилок, за умови, якщо мінімальний інтервал L_{\min} між ними більш ніж втричі перевищує їх максимальну довжину l_{\max} , в іншому випадку стає неможливим обрати достатній крок Δ коду, який би одночасно задовольняв умовам (7.4) і (7.5).

Послідовність виконання роботи

7.1. Вивчити синтаксичну конструкцію функцій **convenc** та **vitdec** мови програмування MATLAB і дати визначення їх параметрам **data**, **trellis**, **encData**, **tb**, **'trunc'**, **'hard'**, **decData**.

Синтаксис функцій **convenc** і **vitdec** для згортального кодування та декодування:

```
encData = convenc(data, trellis)
```

```
decData = vitdec(encData, trellis, tb, 'trunc', 'hard')
```

Опис параметрів функцій:

data – вихідне повідомлення;

trellis – таблиця переходів для згортального коду зі швидкістю k/n без зворотних зв'язків, яка знаходиться за допомогою функції **poly2trellis(ConstraintLength, CodeGenerator)**, де параметр **ConstraintLength** вказує число елементів пам'яті, що використовується для кожного з k вхідних бітів, а параметр **CodeGenerator** описує поліноміальних зв'язок між кожним з n бітів вихідного символу і кожним з k бітів вхідного символу;

encData – кодоване повідомлення;

tb – глибина перегляду (трасування) таблиці переходів **trellis** при декодуванні згортального коду за допомогою алгоритму Вітербі;

'trunc' – задає початковий внутрішній стан згортального кодеру нульовим, де декодер здійснює зворотне трасування таблиці переходів **trellis** від стану з найкращою метрикою;

'hard' – запускає згортальне декодування кодованого повідомлення **encData**, яке знаходиться в двійковій системі числення (бітовий потік);

decData – декодоване повідомлення.

7.2. Змоделювати процес згортального кодування цифрового сигналу (бітового потоку) **data** і його подальшого декодування в програмному середовищі MATLAB за допомогою функцій **convenc** та **vitdec** відповідно.

Установить довжину блоку **m** вихідного повідомлення **data**.

```
m = 6;
```

Згенеруйте вихідне повідомлення **data** (потік бітів) довжиною **m**.

```
data = randi([0 1], m, 1);
```

Установіть кількість елементів пам'яті **ConstraintLength**, яку використовують для кожного з 2-х вхідних бітів.

```
ConstraintLength = [5 4];
```

Задайте матрицю **CodeGenerator**, яка описує поліноміальних зв'язок між кожним з 3-х бітів вихідного символу і кожним з 2-х бітів вхідного символу.

```
CodeGenerator = [23 35 0; 0 5 13];
```

Сформууйте таблицю переходів **trellis** для згортального коду зі швидкістю 2/3 без зворотних зв'язків, згідно параметрів **ConstraintLength** та **CodeGenerator**.

```
trellis = poly2trellis(ConstraintLength, CodeGenerator);
```

Виконайте згортальне кодування вихідного повідомлення **data** за допомогою функції **convenc**.

```
encData = convenc(data, trellis);
```

Введіть помилку в 4-й біт кодованого повідомлення **encData**.

```
encDataError = encData;
```

```
encDataError(4) = ~encDataError(4);
```

Установіть глибину перегляду (трасування) **tb** таблиці переходів **trellis** під час декодування за допомогою алгоритму Вітербі.

```
tb = 3;
```

Виконайте згортальне декодування кодованого повідомлення з бітовими помилками **encDataError** за допомогою функції **vitdec**.

```
decData = vitdec(encDataError, trellis, tb, 'trunc', 'hard');
```

Обчисліть кількість помилкових біт, порівнюючи вихідне **data** та декодоване **decData** повідомлення.

```
numerr = biterr(data, decData);
```

7.3. Побудувати графіки вихідного **data**, кодованого **encData**, кодованого з бітовими помилками **encDataError** та декодованого **decData** повідомлень.

Побудуйте графік вихідного повідомлення **data**.

```
figure(1)
```

```
subplot(2,2,1)
```

```
stairs([0:length(data)], [data; data(length(data))], 'linewidth', 2)
```

```
title('Вихідне повідомлення')
```

```
ylabel('Двійковий цифровий сигнал')
```

```
xlabel('Бітовий потік повідомлення')
```

```
axis([0 length(data) -1 2])  
grid on
```

Побудуйте графік кодованого повідомлення **encData**.

```
subplot(2,2,2)  
stairs([0:length(encData)], [encData;  
encData(length(encData))], 'linewidth', 2)  
title('Кодоване повідомлення')  
ylabel('Двійковий цифровий сигнал')  
xlabel('Бітовий потік повідомлення')  
axis([0 length(encData) -1 2])  
grid on
```

Побудуйте графік кодованого повідомлення з бітовими помилками **encDataError**.

```
subplot(2,2,4)  
stairs([0:length(encDataError)], [encDataError;  
encDataError(length(encDataError))], 'linewidth', 2)  
title('Кодоване повідомлення з бітовими помилками')  
ylabel('Двійковий цифровий сигнал')  
xlabel('Бітовий потік повідомлення')  
axis([0 length(encDataError) -1 2])  
grid on
```

Побудуйте графік декодованого повідомлення **decData**.

```
subplot(2,2,3)  
stairs([0:length(decData)], [decData;  
decData(length(decData))], 'linewidth', 2)  
title('Декодоване повідомлення')  
ylabel('Двійковий цифровий сигнал')  
xlabel('Бітовий потік повідомлення')  
axis([0 length(decData) -1 2])  
grid on
```

7.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.

3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.

4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Розкрийте зміст основних етапів завадостійкого кодування.
2. Розкрийте сутність синдромного методу декодування корегувальних кодів.
3. Проаналізуйте відмінність блочних кодів від неперервних та яка суттєва особливість неперервних кодів.
4. Сформулюйте принципи корегування коду Фінка.
5. Які є класи згорткових кодів за способом декодування та в чому їх принципіальна різниця?
6. Розкрийте сутність неперервного кодування і правило виправлення помилок при декодуванні на прикладі коду Фінка.
7. Поясніть діаграму побудови коду Фінка.
8. Яка ідея побудови рекурентного коду?
9. Опишіть принципи формування перевірочних розрядів, а також виявлення і виправлення помилок лінійними кодами.
10. Записати та пояснити синтаксичні конструкції функцій **convenc** та **vitdec** мови програмування MATLAB і дати визначення їх параметрам **data**, **trellis**, **encData**, **tb**, **'trunc'**, **'hard'**, **decData**.

Лабораторна робота 8

ЗАВАДОСТІЙКЕ КОДУВАННЯ В СИСТЕМІ GSM

Мета роботи: дослідити процес завадостійкого кодування в системі GSM та блоково-діагонального перемешування за допомогою комп'ютерного моделювання використовуючи допоміжні функції **helscanintrlv** і **helscandintrlv** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Канальний кодер забезпечує підвищення завадостійкості передачі повідомлень за рахунок циклічного та згортального кодування, а також перемешування. Згідно рис. 8.1 виконання цих перетворень залежить від виду логічного каналу.

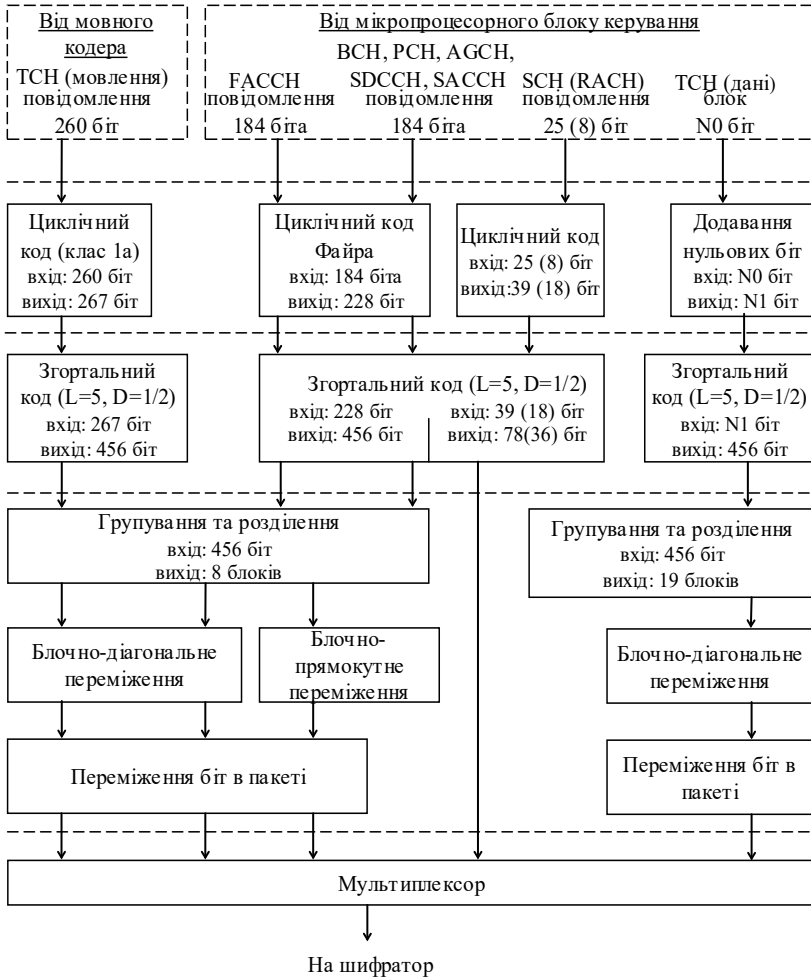
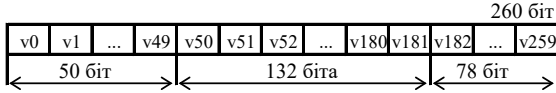


Рис. 8.1. Завадостійке кодування сигналів трафіка та управління

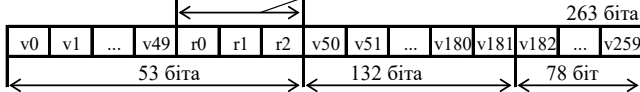
Для каналу *TCH* кодування мовного сигналу показано на рис. 8.2. Процеси, показані на рис. 8.2, а, полягають у наступному. З виходу мовного кодера кожні 20 мс надходить слово довжиною 260 біт. Це слово поділяється на три класи символів:

– клас 1а – це 50 біт, помилки в яких сильно погіршують якість синтезованої мови; в цей клас входять параметри короткочасного аналізу $b_5...b_8$ та параметри довготривалого аналізу L_j та K_j ;

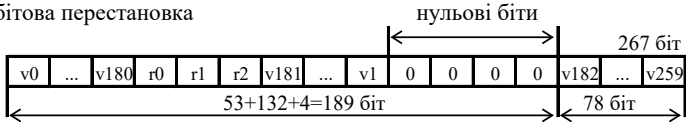
а) слово мовного кодера



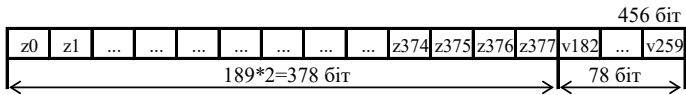
б) циклічний код (клас 1а)



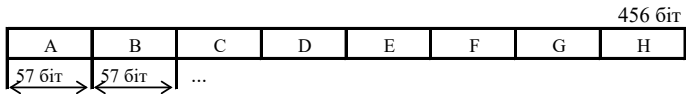
в) бітова перестановка



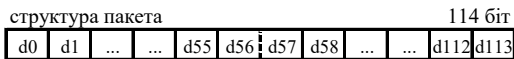
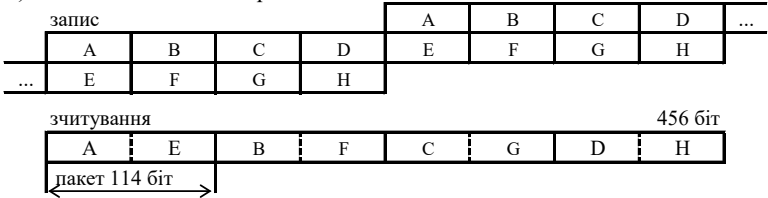
г) згортальний код (клас 1)



д) розділення на блоки



е) блочно-діагональне перемеження



ж) перемеження в пакеті

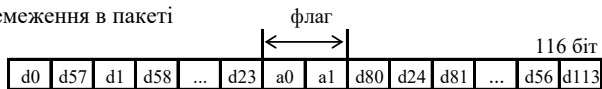


Рис. 8.2. Завадостійке кодування в мовному каналі TCH

– клас 1б – це 132 біта з помірним впливом помилок на якість мови; в цей клас входять параметри короткочасного аналізу $b_1 \dots b_4$, параметри збудження N_j , A_j та половина параметрів D_{jm} ;

– клас 2 – це 78 біт, помилки в яких слабо впливають на якість мови; в цей клас входить друга половина параметрів D_{jm} .

Процеси, показані на рис. 8.2, б, полягають у наступному. Біти класу 1а кодується циклічним кодом з породжуючим поліномом 3-го ступеня $p(x) = x^3 + x + 1$, тому кодова комбінація містить 53 біта. Даний код є слабким та застосовується лише для визначення помилок циклічним декодером. Біти інших класів не кодується, тому довжина слова складає 263 біта.

Процеси, показані на рис. 8.2, в, полягають у наступному. У слові виконується перестановлення перших 85 біт в наступному порядку: парні біти, перевірочні біти, непарні біти в зворотньому запису. Таке перестановлення аналогічне перемеженню. Після додавання чотирьох нульових біт довжина слова складає 267 біт. Далі слово надходить до згортального кодера, за виключенням біт класу 2.

Процеси, показані на рис. 8.2, г, полягають у наступному. Згортальний кодер має чотирьохрозрядний регістр та швидкість 1/2. Вхідні 189 біт перетворюються у вихідні 378 біт. Довжина слова складає 456 біт. На рис. 8.2, д показано, що слово розділяється на 8 блоків по 57 біт. На рис. 8.2, е отримані блоки піддаються блочно-діагональному перемеженню. З виходу перемежувача зчитуються 4 види подвійних блоків або пакетів довжиною по 114 біт.

Заключною операцією кодера (рис. 8.2, ж) є перемеження біт в кожному пакеті з додаванням флага. Число біт в пакеті дорівнює довжині двох частин повідомлення в слоті NB , тобто 114 біт. Тому слово довжиною 456 біт у вигляді чотирьох пакетів передається у виділеному слоті в чотирьох послідовних кадрах зі швидкістю

$$\frac{n}{\Delta t} = \frac{456 \text{ біт}}{4T_{\phi}} = \frac{456}{4 \cdot 4,61535 \cdot 10^{-3}} = 24,7 \text{ кбіт/с},$$

де T_{ϕ} – тривалість фрейму.

Канальний декодер мовного каналу виконує зворотні перетворення. Після деперемеження згортальний декодер виправляє помилки в бітах класів 1а та 1б. Циклічний кодер перевіряє наявність помилок, що залишилися. Якщо такі помилки знаходяться, то 260 біт поточного слова не надходять в мовний декодер, а замінюються аналогічним словом з попереднього інтервалу 20 мс.

Розглянемо канал даних *TCH*, показаний на рис. 8.1. Передача даних допускає затримку повідомлення в порівнянні з передачею мови. Тому перемерження виконується після розділення вихідних біт згортального кодера не на 8, а на 19 блоків.

Заключне бітове перемерження також виконується згідно більш складного алгоритму. У результаті розміщення біт стає псевдовипадковим, але відомим деперемержувачу для виконання зворотнього перетворення.

Розглянемо канали управління, також показані на рис. 8.1. Повідомлення каналів *RACH* та *SCH* кодується слабким циклічним кодом та згортальним кодом. Перемерження не застосовується, оскільки повідомлення містить однотипову інформацію, що повторюється досить рідко. Тому впливом пакетів помилок можна знехтувати. Повідомлення інших каналів довжиною 184 біта кодується за загальною схемою. Висока завадостійкість передачі цих повідомлень забезпечується циклічним кодом з породжуючим поліномом 40-го ступеня: $p(x) = (x^{23}+1)(x^{17}+x^3+1) = x^{40} + \dots + 1$. Вихідна комбінація циклічного кодера містить 224 біта. Після додавання чотирьох нульових біт отримують слово довжиною 228 біт, що надходить до згортального кодера. Вихідне слово згортального кодера довжиною 456 біт розділяється на 8 блоків з наступними блочним і внутрішньопакетним перемерженням.

Циклічний код з породжуючим поліномом виду $p(x) = (x^c+1)(x^m+\dots+1)$ називається кодом Файра. Такий код знаходить в кодовій комбінації пакет помилок довжиною $t_{\text{знах}}$ та одночасно виправляє інший пакет помилок довжиною $t_{\text{вип}}$ при умові $t_{\text{знах}} \leq m$, $t_{\text{вип}} \leq c - t_{\text{знах}} + 1$.

Послідовність виконання роботи

8.1. Вивчити синтаксичну конструкцію функцій **helscanintrlv** та **helscandintrlv** мови програмування MATLAB і дати визначення їх параметрам **data**, **Nrows**, **Ncols**, **hstep**, **intrlvData**, **deintrlvData**.

Синтаксис функцій **helscanintrlv** і **helscandintrlv** для блоково-діагонального перемержовування та деперемержовування:

```
intrlvData = helscanintrlv(data, Nrows, Ncols, hstep)
```

```
deintrlvData = helscandintrlv(intrlvData, Nrows, Ncols, hstep)
```

Опис параметрів функцій:

data – вихідне повідомлення;

Nrows – кількість рядків в проміжній матриці;

Ncols – кількість стовпців в проміжній матриці;

hstep – нахил діагоналі проміжної матриці, який задається невід'ємним цілим числом, меншим ніж **Nrows**;

intrlvData – перемержовуване повідомлення;

deintrlvData – деперемержовуване повідомлення.

Параметри **data**, **Nrows** та **Ncols** потрібно задавати згідно міркування, де добуток кількості рядків **Nrows** на кількість стовпців **Ncols** проміжної матриці має дорівнювати довжині вихідного повідомлення **data**, тобто кількість елементів у вихідному повідомленні **data** повинно дорівнювати кількості елементів в проміжній матриці розмірністю **Nrows** × **Ncols**.

8.2. Змоделювати процес блоково-діагонального перемержовування цифрового сигналу (бітового потоку) **data** і його подальшого деперемержовування в програмному середовищі MATLAB за допомогою функцій **helscanintrlv** та **helscandeintrlv** відповідно.

Установіть довжину блоку **m** вихідного повідомлення **data**.

```
m = 9;
```

Згенеруйте вихідне повідомлення **data** (потік бітів) довжиною **m**.

```
data = randi([0 1], m, 1);
```

Задайте кількість рядків **Nrows** і стовпців **Ncols** проміжної матриці.

```
Nrows = 3;
```

```
Ncols = 3;
```

Задайте нахил діагоналі **hstep** проміжної матриці.

```
hstep = 1;
```

Виконайте блоково-діагональне перемержовування вихідного повідомлення **data** за допомогою функції **helscanintrlv**.

```
intrlvData = helscanintrlv(data, Nrows, Ncols, hstep);
```

Виконайте блоково-діагональне деперемержовування перемержовуваного повідомлення **intrlvData** за допомогою функції **helscandeintrlv**.

```
deintrlvData = helscandeintrlv(intrlvData, Nrows, Ncols, hstep);
```

Обчисліть кількість помилкових біт, порівнюючи вихідне **data** та деперемержовуване **deintrlvData** повідомлення.

```
numerr = biterr(data, deintrlvData);
```

8.3. Побудувати графіки вихідного **data**, перемержовуваного **intrlvData** та деперемержовуваного **deintrlvData** повідомлень.

Побудуйте графік вихідного повідомлення **data**.

```
figure(1)
```

```
subplot(2,2,1)
```

```
stairs([0:length(data)], [data; data(length(data))], 'linewidth', 2)
title('Вихідне повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(data) -1 2])
grid on
```

Побудуйте графік перемешованого повідомлення **intrlvData**.

```
subplot(2,2,2)
stairs([0:length(intrlvData)], [intrlvData;
intrlvData(length(intrlvData))], 'linewidth', 2)
title('Перемешоване повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(intrlvData) -1 2])
grid on
```

Побудуйте графік деперемешованого повідомлення

deintrlvData.

```
subplot(2,2,3)
stairs([0:length(deintrlvData)], [deintrlvData;
deintrlvData(length(deintrlvData))], 'linewidth', 2)
title('Деперемешоване повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(deintrlvData) -1 2])
grid on
```

8.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Вкажіть мету і сфери використання завадостійкого кодування та чим воно відрізняється від ентропійного кодування.
2. Сформулюйте теоретичну базу побудови завадостійких кодів.
3. Дайте визначення лінійним завадостійким кодам та наведіть основні принципи їх побудови.
4. Опишіть процес завадостійкого кодування в мовному каналі *TCH* та кодування сигналів трафіка та управління.
5. Який породжуючий поліном використовується в каналі *RACH* та *SCH* системи GSM.
6. Опишіть принцип формування дозволених кодових комбінацій лінійних кодів і наведіть відомі вам приклади лінійних кодів.
7. Проаналізуйте вплив завадостійкого кодування з аспектів збільшення вірності передачі і ускладнення апаратури зв'язку.
8. Наведіть класифікацію корегувальних кодів і розкрийте особливості кожного з класів.
9. Розкрийте умови, які повинні виконуватися при формуванні базисних дозволених кодових комбінацій лінійних кодів.
10. Записати та пояснити синтаксичні конструкції функцій **helscanintrlv** та **helscandintrlv** мови програмування MATLAB і дати визначення їх параметрам **data**, **Nrows**, **Ncols**, **hstep**, **intrlvData**, **deintrlvData**.

Лабораторна робота 9

ЧАСТОТНА МАНПУЛЯЦІЯ З МІНІМАЛЬНИМ ЗСУВОМ

Мета роботи: дослідити процес модуляції на основі частотної маніпуляції з мінімальним зсувом та демодуляції за допомогою комп'ютерного моделювання використовуючи допоміжні функції **mskmod** і **mskdemod** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Біти, що передаються, модулюють радіосигнал за методом *MSK* (*Minimum Shift Keying*). У результаті такої модуляції частота радіосигналу дорівнює верхній або нижній частоті

$$f_B = f_0 + \frac{F_0}{4}, \quad f_H = f_0 - \frac{F_0}{4}, \quad (9.1)$$

де f_0 – частота немодульованого сигналу; F_6 – частота повторення біт.

Порядок чергування частот f_B і f_H залежить від значення біт.

Назва методу обумовлена тим, що $f_B f_H = F_6/2$ – мінімальне частотне зміщення, за якого гармонічні сигнали з частотами f_B і f_H ортогональні на інтервалі $T_6 = 1/F_6$.

Розглянемо принцип модуляції. Гармонічний сигнал визначається наступним чином:

$$\begin{aligned} U(t) &= u_m \sin \varphi(t) = u_m \sin(\omega_0 t + \varphi_0) = \\ &= u_m \cos \varphi_0 \sin \omega_0 t + u_m \sin \varphi_0 \cos \omega_0 t = \\ &= I(t) + Q(t), \end{aligned} \quad (9.2)$$

де u_m – амплітуда; $\varphi(t)$ – миттєва фаза; $\omega_0 = 2\pi f_0$ – частота; φ_0 – початкова фаза.

Синфазний сигнал $I(t)$ та квадратурний сигнал $Q(t)$ є ортогональними складовими сигналу $U(t)$. Прийнемо частоту цього сигналу

$f_0 = \frac{\omega_0}{2\pi}$ в якості частоти f_0 в формулах (9.1). Розділимо бітову послідовність на непарні біти і парні біти $d_{2i-1}, d_{2i}, i = 1, 2, 3 \dots$

Два сусідніх біта як один символ називаються дібітом. Дібіт має 4 значення: 11, 01, 10, 00. Відобразимо ці значення чотирма варіантами сигналу, що визначається за формулою (9.2) у табл. 9.1.

Таблиця 9.1

Відображення дібітів гармонічного сигналу

| d_{2i-1} | d_{2i} | $U(t)$ |
|------------|----------|----------------------------------------------------|
| 1 | 1 | $u_m \sin(\omega_0 t + \varphi_0) = I(t) + Q(t)$ |
| 0 | 1 | $u_m \sin(\omega_0 t - \varphi_0) = I(t) - Q(t)$ |
| 1 | 0 | $-u_m \sin(\omega_0 t - \varphi_0) = -I(t) + Q(t)$ |
| 0 | 0 | $-u_m \sin(\omega_0 t + \varphi_0) = -I(t) - Q(t)$ |

У табл. 9.2 показане аналогічне відображення для сигналу *MSK*, але сигнал $U(t)$ відрізняється тим, що його початкова фаза залежить від часу $\varphi_0(t) = \frac{\pi t}{2T_6}$, де $T_6 = \frac{1}{F_6}$ – тривалість біта.

При цьому:

1) коефіцієнти $\cos \varphi_0$ і $\sin \varphi_0$ у формулі (9.2) замінюються на гармонічні функції $\cos \varphi_0(t)$ і $\sin \varphi_0(t)$;

Таблиця 9.2

Модуляція MSK

| d_{2i-1} | d_{2i} | Модульований сигнал $U(t)$ |
|------------|----------|---------------------------------------------------------------|
| 1 | 1 | $U_m \sin(\omega_0 t + \pi t / 2T_0) = U_m \sin 2\pi f_B t$ |
| 0 | 1 | $U_m \sin(\omega_0 t - \pi t / 2T_0) = U_m \sin 2\pi f_H t$ |
| 1 | 0 | $-U_m \sin(\omega_0 t - \pi t / 2T_0) = -U_m \sin 2\pi f_H t$ |
| 0 | 0 | $-U_m \sin(\omega_0 t + \pi t / 2T_0) = -U_m \sin 2\pi f_B t$ |

2) миттєва фаза в формулі (9.2) дорівнює

$$\varphi(t) = \omega_0 t \pm \varphi_0(t) = \omega_0 t \pm \frac{\pi t}{2T}; \quad (9.3)$$

3) частота сигналу $U(t)$ задовільняє формулі (9.1):

$$\begin{aligned} u(t) &= \pm u_m \sin(\omega_0 t \pm \varphi_0(t)) = \pm u_m \sin(\omega_0 t \pm \frac{\pi t}{2T_0}) = \\ &= \pm u_m \sin 2\pi t (f_0 \pm \frac{1}{4T_0}) = \pm u_m \sin 2\pi (f_0 \pm \frac{F_0}{4}) t = \begin{cases} \pm u_m 2\pi f_B t \\ \pm u_m 2\pi f_H t \end{cases} \end{aligned}$$

Табл. 9.2 є розгорнутим представленням сигналу MSK. Цей сигнал також визначається виразом:

$$u(t) = \pm I(t) \pm Q(t). \quad (9.4)$$

У даній формулі знак складової $I(t)$ згідно табл. 9.1 та 9.2 залежить від значення парного біта, а знак $Q(t)$ від значення непарного біта. Тому для реалізації формули (9.4) необхідно:

- 1) сформулювати складові $I(t)$ і $Q(t)$;
- 2) розділити бітову послідовність на непарні та парні біти;
- 3) представити біти додатними та від'ємними рівнями;
- 4) помножити ці рівні на поточні значення складових, при цьому складова змінює знак, якщо біт є логічним нулем;
- 5) просумувати результати множення.

На рис. 9.1 показана схема модулятора, який виконує дані перетворення.

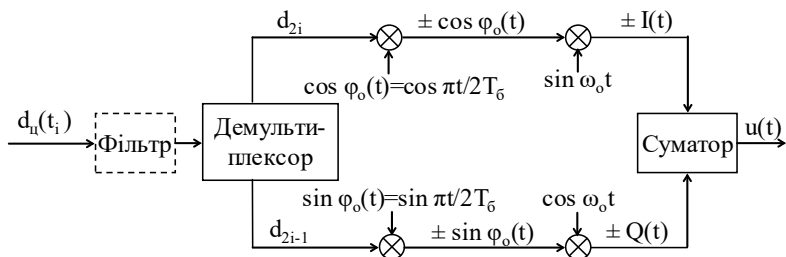


Рис. 9.1. Модулятор

Демультимплексор розділяє послідовність, що передається, на парні та непарні біти. На перші перемножувачі подаються отримані біти та низькочастотні сигнали $\sin \varphi_0(t)$ та $\cos \varphi_0(t)$ з періодом $4T_b$.

При нульових бітах ці сигнали на виході перемножувачів змінюють знак. На другі перемножувачі подаються сигнали $\pm \sin \varphi_0(t)$ та $\pm \cos \varphi_0(t)$, а також високочастотні $\sin \omega_0(t)$ та $\cos \omega_0(t)$. На виході других перемножувачів отримують сигнали $\pm I(t)$ і $\pm Q(t)$, що входять у формулу (9.4). Суматор формує радіосигнал $U(t)$, модульований згідно методу *MSK*. Дібітами є рівні d_{2i-1} , d_{2i} в кожному інтервалі T_b . Накопичування фазових зміщень у відповідності до значень біт проявляється в тому, що сигнал *MSK* «стискується» або «розтягується» по вісі часу t відносно гармонічного сигналу з частотою f_0 . При цьому частота сигналу стрибком або збільшується до f_B , або зменшується до f_H .

Визначимо, які параметри сигналу *MSK* залежать від значення дібіту. Табл. 9.2 показує, що сигнал з частотою f_B відображає одну пару дібітів 11 та 00, а з частотою f_H – іншу пару 10 та 01. Згідно цієї таблиці, дібіти в кожній двійці розрізняються за загальними знаками сигналу $U(t)$. Отже, 4 дібіта розрізняють за рахунок зміни миттєвої фази (знака) та початкової фази (частоти) модульованого сигналу.

Розрахуємо індекс частотної модуляції.

Частота слідування непарних і парних біт, які модулюють ортогональні складові, дорівнює $F_{mod} = \frac{1}{2T_b} = \frac{F_b}{2}$.

Девіація частоти радіосигналу згідно формули (9.1) дорівнює

$$\Delta f = \frac{F_{\sigma}}{4}.$$

Індекс частотної модуляції дорівнює $m = \frac{\Delta f}{F_{\text{мод}}} = 0,5$.

У системі *GSM* застосовується модуляція *GMSK*. За такої модуляції цифровий сигнал пропускається крізь вхідний фільтр, який має амплітудно-частотну характеристику (АЧХ) виду

$$K(f) = e^{-\left(\frac{f}{f_{0,5}}\right)^{\ln 2}},$$

де $f_{0,5}$ – частота зрізу на рівні 0,5 амплітуди сигналу.

Цей фільтр згладжує перепади рівнів цифрового сигналу, тому:

1) згладжуються гострі вершини сигналів $\pm \cos \varphi_0(t)$ та $\pm \sin \varphi_0(t)$;

2) згладжуються гострі вершини складових $\pm I(t)$, $\pm Q(t)$;

3) згладжується радіосигнал в моменти зміни його частоти.

У результаті ширина спектру радіосигналу зменшується.

Фільтр має смугу пропускання $(0 \dots f_{0,5}) = 83,1 \text{ кГц}$. Нормована смуга дорівнює

$$B_{0,5} = \frac{f_{0,5}}{F_{\sigma}} = \frac{81,3}{270,833} = 0,3.$$

Переваги модуляції *GMSK*:

1) радіосигнал на виході передавача має відносно вузький спектр, що знижує рівень завад в сусідніх частотних каналах;

2) радіосигнал має постійний рівень огинаючої, тому в передавачах можна використовувати економічні підсилювачі потужності класу С, що є гарною практикою.

Розглянемо принцип демодуляції. Схема демодулятора показана на рис. 9.2.

Опорні сигнали формуються згідно формули (9.2) з урахуванням формули (9.3):

$$S_1(t) = S_{1m} \cos \varphi_0(t) \sin \omega_0 t,$$

$$S_2(t) = S_{2m} \sin \varphi_0(t) \cos \omega_0 t.$$

Сигнал $S_1(t)$ є копією синфазної складової $I(t)$, сигнал $S_2(t)$ є копією квадратурної складової $Q(t)$.

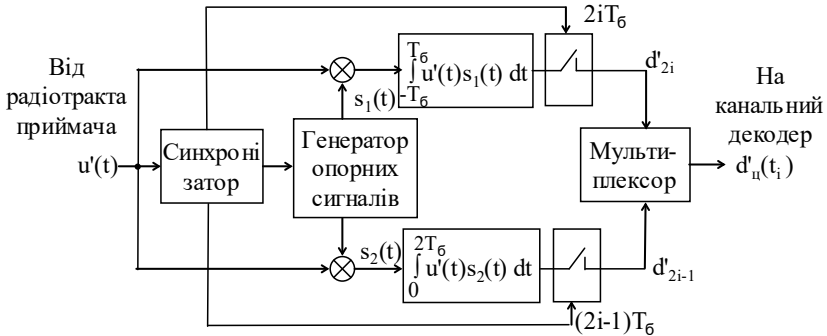


Рис. 9.2. Демодулятор

Інтегрування відбувається зі зміщенням на T_6 , оскільки в модуляторі непарні та парні біти також зміщені на T_6 . Ключі, якими управляють синхронізатори, служать для відліку рівнів в кінці інтервалів інтегрування. Відліки двох ключів є копіями парних та непарних біт, що передаються.

Мультиплексор формує демодульований цифровий сигнал. В каналі зв'язку з завадами прийнятий сигнал не співпадає з переданим, тому демодуляція може виконуватись з помилками. Ці помилки виправляє каналний декодер.

Послідовність виконання роботи

9.1. Вивчити синтаксичну конструкцію функцій **mskmod** та **mskdemod** мови програмування MATLAB і дати визначення їх параметрам **data**, **nsamp**, **modData**, **demodData**.

Синтаксис функцій **mskmod** і **mskdemod** для модуляції на основі частотної маніпуляції з мінімальним зсувом та демодуляції:

modData = **mskmod**(**data**, **nsamp**)

demodData = **mskdemod**(**modData**, **nsamp**)

Опис параметрів функцій:

data – вихідне повідомлення;

nsamp – кількість відліків, що припадають на один інформаційний символ вихідного повідомлення **data**;

modData – модульоване повідомлення;

demodData – демодульоване повідомлення.

Початкова фаза модульованого повідомлення **modData** дорівнює нулю.

9.2. Змоделювати процес модуляції на основі частотної маніпуляції з мінімальним зсувом цифрового сигналу (бітового потоку) **data** і його подальшої демодуляції в програмному середовищі MATLAB за допомогою функцій **mskmod** та **mskdemod** відповідно.

Установіть довжину блоку **m** вихідного повідомлення **data**.

```
m = 10;
```

Згенеруйте вихідне повідомлення **data** (потік бітів) довжиною **m**.

```
data = randi([0 1], m, 1);
```

Установіть кількість відліків **nsamp**, що припадають на один інформаційний символ вихідного повідомлення **data**.

```
nsamp = 5;
```

Виконайте модуляцію на основі частотної маніпуляції з мінімальним зсувом вихідного повідомлення **data** за допомогою функції **mskmod**.

```
modData = mskmod(data, nsamp);
```

Виконайте демодуляцію частотно-маніпульованого з мінімальним зсувом повідомлення **modData** за допомогою функції **mskdemod**.

```
demodData = mskdemod(modData, nsamp);
```

Обчисліть кількість помилкових біт, порівнюючи вихідне **data** та демодульоване **demodData** повідомлення.

```
numerr = biterr(data, demodData);
```

9.3. Побудувати графіки вихідного **data**, модульованого **modData** та демодульованого **demodData** повідомлень.

Побудуйте графік вихідного повідомлення **data**.

```
figure(1)
```

```
subplot(2,2,1)
```

```
stairs([0:length(data)], [data; data(length(data))], 'linewidth', 2)
```

```
title('Вихідне повідомлення')
```

```
ylabel('Двійковий цифровий сигнал')
```

```
xlabel('Бітовий потік повідомлення')
```

```
axis([0 length(data) -1 2])
```

```
grid on
```

Побудуйте графік модульованого повідомлення **modData**.

```
subplot(2,2,2)
```

```
plot(0:length(modData)-1, real(modData), 'linewidth', 2)
```

```
title('Модульоване повідомлення')
```

```
ylabel('Амплітуда')
```

```
xlabel('Час')
axis([0 length(modData)-1 -2 2])
grid on
```

Побудуйте графік демодульованого повідомлення **demodData**.

```
subplot(2,2,3)
stairs([0:length(demodData)], [demodData;
demodData(length(demodData))], 'linewidth', 2)
title('Демодульоване повідомлення')
ylabel('Двійковий цифровий сигнал')
xlabel('Бітовий потік повідомлення')
axis([0 length(demodData) -1 2])
grid on
```

9.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Опишіть основні етапи модуляції та демодуляції згідно методу частотної маніпуляції з мінімальним зсувом.
2. Які переваги має модуляція *GMSK* в порівнянні з іншими видами модуляцій?
3. Чим відрізняється модуляція *GMSK* в порівнянні з *MSK* модуляцією? Назвати концептуальні відмінності.
4. Яку головну роль відіграє процес модуляції у загальній системі передачі сигналів радіоканалом зв'язку?
5. Який вид маніпуляції використовується у системі GSM та в чому її переваги?
6. Опишіть розрахунок індексу частотної модуляції.

7. Яку роль відіграє мультиплексор в демодуляторі згідно методу частотної маніпуляції з мінімальним зсувом?

8. Яку роль відіграє суматор в модуляторі згідно методу частотної маніпуляції з мінімальним зсувом?

9. Як ви розумієте поняття – девіація частоти радіосигналу?

10. Записати та пояснити синтаксичні конструкції функцій **mskmod** та **mskdemod** мови програмування MATLAB і дати визначення їх параметрам **data**, **nsamp**, **modData**, **demodData**.

Лабораторна робота 10

КАНАЛ ЗВ'ЯЗКУ З АДИТИВНИМ БІЛИМ ШУМОМ

Мета роботи: дослідити процес формування каналу зв'язку з адитивним білим шумом за допомогою комп'ютерного моделювання використовуючи допоміжну функцію **awgn** з бібліотеки Communications Toolbox мови програмування MATLAB.

Основні теоретичні відомості

Якщо сигнали $s(t)$ і $z(t)$ розглядати як переданий і прийнятий сигнали, а $\xi(t)$ – як адитивну заваду, що діє в неперервному КЗ, то для *гаусівського* КЗ з *неперервним часом* $W_{\text{дж}} = 2 \cdot \Delta F_c$ з флукутаційною (адитивною) завадою, пропускну здатність можна знайти наступним виразом:

$$C = \Delta F_c \cdot \log_* \left(1 + \frac{\sigma_S^2}{\sigma_\xi^2} \right) \text{ [біт/сек.]} \quad (10.1)$$

Формула (10.1) пов'язує між собою смугу частот сигналу ΔF_c , відношення сигнал/шум $\sigma_S^2 / \sigma_\xi^2$ у каналі зв'язку та пропускну здатність останнього – C . Сутність формули (10.1) – одну й ту саму пропускну здатність C можна одержати за різних співвідношень між ΔF_c і $\sigma_S^2 / \sigma_\xi^2$. Оскільки пропускну здатність лінійно залежить від смуги частот сигналу, то потужність сигналу σ_S^2 доцільно обмінювати на цю смугу і задану пропускну здатність забезпечувати за рахунок використання широкосмугових (шумоподібних) сигналів.

Проведемо аналіз формули Шеннона (10.1). Теоретично (при використанні достатньо складної схеми кодування) інформацію по каналу зв'язку можна передавати з *будь-якою* швидкістю $R \leq C$ з як *завгодно малою* імовірністю виникнення помилки. Якщо ж $R > C$,

то коду, на основі якого можна одержати як завгодно малу ймовірність виникнення помилки, *не існує*.

У роботах Шеннона показано, що величини σ_S^2 , σ_ξ^2 та ΔF_c встановлюють не ймовірності появи помилки, а *граничні значення швидкості передавання*. Шеннон використовував рівняння (10.1) для графічного представлення доступних границь продуктивності прикладних систем. Якщо $\Delta F_c = const$, а відношення σ_S^2/σ_ξ^2 зростає, то пропускна здатність також зростатиме, але це зростання буде досить повільним, оскільки воно підпорядковується логарифмічному закону.

Тому, наприклад (рис. 10.1), якщо при смузі частот 1000 Гц і відношенні $\sigma_S^2/\sigma_\xi^2 = 10^3 = 30$ дБ пропускна здатність $C \approx 10\,000$ біт/сек., а її необхідно збільшити вдвічі при збереженні попередньої смуги частот 1000 Гц, то цього можна досягти лише збільшивши відношення сигнал/шум до $10^6 = 60$ дБ (тобто у 1000 разів).

Розглянемо тепер залежність пропускної здатності каналу C від смуги частот ΔF_c при $\sigma_S^2/\sigma_\xi^2 = const$. З курсу радіотехніки відомо, що середня потужність шуму є пропорційною смузі частот сигналу:

$$\sigma_\xi^2 = \Omega_0 \cdot \Delta F_c, \quad (10.2)$$

де Ω_0 – спектральна щільність потужності білого шуму, Вт/Гц.

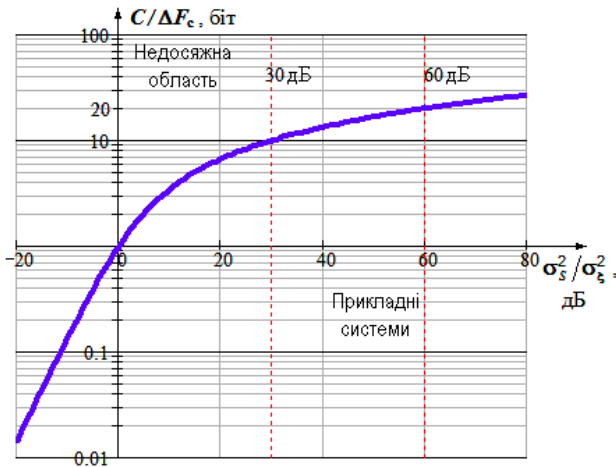


Рис. 10.1. Залежність нормованої по смузі частот сигналу пропускної здатності каналу від відношення сигнал/шум у каналі

Тому зручно ввести *еквівалентну смугу шуму*

$$\Delta F_0 = \sigma_S^2 / \Omega_0, \quad (10.3)$$

де ΔF_0 – смуга спектру такого шуму, який при спектральній щільності Ω_0 має потужність, яка дорівнює потужності сигналу $s(t)$.

Підставивши (10.2) і (10.3) до (10.1), одержимо

$$C = \Delta F_c \cdot \log_* \left(1 + \frac{\Delta F_0}{\Delta F_c} \right).$$

Після нормування по еквівалентній смузі шуму маємо:

$$\frac{C}{\Delta F_0} = \frac{\Delta F_c}{\Delta F_0} \cdot \log_* \left(1 + \frac{\Delta F_0}{\Delta F_c} \right).$$

На рис. 10.2 зображено графік залежності нормованої по еквівалентній смузі шуму пропускної здатності від відношення $\Delta F_c / \Delta F_0$.

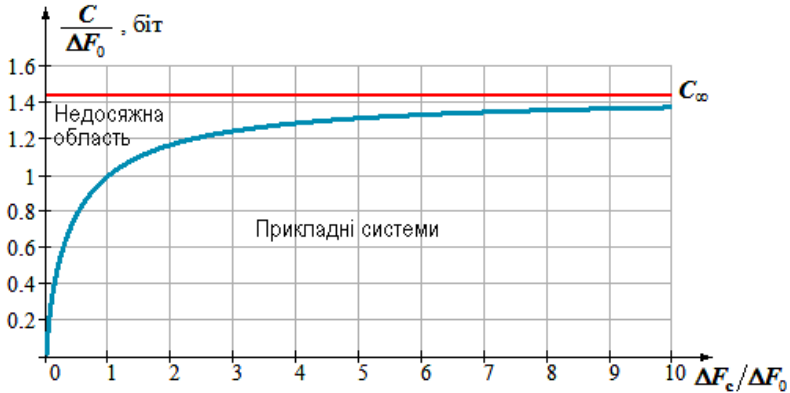


Рис. 10.2. Залежність нормованої по еквівалентній смузі шуму пропускної здатності від відношення $\Delta F_c / \Delta F_0$

Як видно, нормована пропускна здатність при цьому є монотонно зростаючою функцією, яка при $\Delta F_c \rightarrow \infty$, враховуючи, що за другою члудовою границею

$$\lim_{x \rightarrow \infty} \left[x \cdot \log_* \left(1 + \frac{1}{x} \right) \right] = \log_*(e),$$

асимптотично наближується до величини

$$\begin{aligned} \lim_{\Delta F_c \rightarrow \infty} \frac{C}{\Delta F_0} &= \lim_{\Delta F_c \rightarrow \infty} \frac{\Delta F_c}{\Delta F_0} \cdot \log_* \left(1 + \frac{\Delta F_0}{\Delta F_c} \right) = \\ &= \log_*(e) = C_\infty \approx 1,4427 \text{ біт.} \end{aligned}$$

Наслідком аналізу графіка рис. 10.2 є висновок, що чим ширшим є спектр сигналу ΔF_c (тобто чим більше сигнал наближається до білого шуму), тим сильніше пропускна здатність каналу C наближається до гранично можливої (C_∞). У цьому полягає одна з переваг використання шумоподібних (широкосмугових) сигналів при організації зв'язку. Проте, хоча із зростанням смуги частот ΔF_c можливості НКЗ стосовно передавання інформації збільшуються, такий крок не дає необмежених можливостей по збільшенню пропускної здатності C . Наприклад, при відношенні $\Delta F_c/\Delta F_0 = 2$ вже реалізується приблизно 81% від C_∞ ; при $\Delta F_c/\Delta F_0 = 3$ – 86,3%; при $\Delta F_c/\Delta F_0 = 5$ – 91,5%. Подібні результати можна отримати і з наступного аналізу. В аналоговому зв'язку, як відомо, використовується критерій якості, що зветься *відношенням середньої потужності сигналу до середньої потужності шуму*: σ_S^2/σ_ξ^2 . У цифровому зв'язку як критерій якості частіше використовується *нормована версія сигнал/шуму*, яка позначається як E_b/Ω_0 , де E_b – це енергія біта, яку можна описати як потужність сигналу σ_S^2 , помножену на час передавання біту τ_b ; Ω_0 – спектральна щільність потужності білого шуму, яку можна виразити як потужність шуму σ_ξ^2 , поділену на ширину смуги частот сигналу ΔF_c . Оскільки час передавання біту τ_b і швидкість передавання біту R є взаємозворотними, то τ_b можна замінити на $1/R$:

$$\frac{E_b}{\Omega_0} = \frac{\sigma_S^2 \cdot \tau_b}{\sigma_\xi^2 / \Delta F_c} = \frac{\sigma_S^2 / R}{\sigma_\xi^2 / \Delta F_c} = \frac{\sigma_S^2}{\sigma_\xi^2} \cdot \frac{\Delta F_c}{R}. \quad (10.4)$$

Якщо бітова швидкість передавання дорівнює пропускній здатності каналу зв'язку ($R = C$), то за допомогою виразу (10.4) можна записати наступне:

$$\frac{\sigma_S^2}{\sigma_\xi^2} = \frac{E_b}{\Omega_0} \cdot \frac{C}{\Delta F_c}.$$

Отже, рівняння (10.1) можна модифікувати наступним чином:

$$\frac{C}{\Delta F_c} = \log_* \left(1 + \frac{E_b}{\Omega_0} \cdot \frac{C}{\Delta F_c} \right), \quad (10.5)$$

якщо основа логарифму дорівнює 2, то

$$2^{C/\Delta F_c} = 1 + \frac{E_b}{\Omega_0} \cdot \frac{C}{\Delta F_c};$$

$$\frac{E_b}{\Omega_0} = \left(2^{C/\Delta F_c} - 1\right) \cdot \frac{\Delta F_c}{C}. \quad (10.6)$$

На рис. 10.3 представлено графік залежності $\Delta F_c/C$ від E_b/Ω_0 , яка описується формулою (10.6).



Рис. 10.3. Залежність нормованої по пропускній здатності смуги пропускання каналу від відношення E_b/Ω_0

Асимптотична поведінка даної кривої при $\Delta F_c/C \rightarrow \infty$ (або $C/\Delta F_c \rightarrow 0$) являє собою так звану *границю Шеннона* – нижнє граничне значення E_b/Ω_0 , при якому за будь-якої швидкості передавання R безпомилкове передавання інформації здійснити неможливо.

За допомогою співвідношення $\lim_{x \rightarrow 0} (1+x)^{1/x} = e$ можна обчислити це граничне значення E_b/Ω_0 .

Нехай $x = \frac{E_b}{\Omega_0} \cdot \frac{C}{\Delta F_c}$. Тоді, використавши рівняння (10.5) при основі логарифму 2, маємо:

$$\frac{C}{\Delta F_c} = x \cdot \log_2 \left[(1+x)^{1/x} \right]$$

або, розкривши перший x ,

$$1 = \frac{E_b}{\Omega_0} \cdot \log_2 \left[(1+x)^{1/x} \right].$$

У граничному випадку, при $C/\Delta F_c \rightarrow 0$ (або $x \rightarrow 0$), маємо

$$\frac{E_b}{\Omega_0} = \frac{1}{\log_2(e)} \approx 0,69315$$

або

$$\frac{E_b}{\Omega_0} = -1,59 \text{ дБ.}$$

Диференціальна ентропія гаусівського білого шуму визначається як ентропія n -мірного гаусівського розподілу з незалежними відліками:

$$H_{\delta_n} = \log_* \left[\left(\sqrt{2 \cdot \pi \cdot e \cdot \sigma^2} \right)^n \right] = n \cdot H_{\delta_{\max}}.$$

Отже, диференціальна ентропія гаусівського білого шуму у n разів більше за ентропію окремого незалежного відліку і є максимальною для неперервних сигналів обмеженої потужності.

Гаусівський білий шум, маючи найбільшу ентропію, є найбільш шкідливою завадою, оскільки переносить максимально можливу кількість шкідливої інформації; якщо в ролі несних використовувати гаусівські сигнали, які за властивостями близькі до білого шуму, можна за однакової середньої потужності передати максимальну кількість інформації. З огляду на це, білий шум широко використовують як модель несних і завад.

Послідовність виконання роботи

10.1. Вивчити синтаксичну конструкцію функції **awgn** мови програмування MATLAB і дати визначення її параметрам **x**, **snr_linear**, **snr_dB**, **'measured'**, **'linear'**, **'dB'**, **x_noise_linear**, **x_noise_dB**.

Синтаксис функції **awgn** для додавання до сигналу адитивного білого гаусівського шуму з необхідним відношенням сигнал/шум:

```
x_noise_linear = awgn(x, snr_linear, 'measured', 'linear')
x_noise_dB = awgn(x, snr_dB, 'measured', 'dB')
```

Опис параметрів функції:

x – вихідний сигнал;

snr_linear – відношення сигнал/шум в разях;

snr_dB – відношення сигнал/шум в децибелах;

'measured' – запускає автоматичне вимірювання потужності вихідного сигналу **x**;

'linear' – задає одиницю виміру відношення сигнал/шум **snr_linear** в разях;

'dB' – задає одиницю виміру відношення сигнал/шум **snr_dB** в децибелах;

x_noise_linear – результуючий зашумлений сигнал, в якому дотримано задане співвідношення потужності сигналу до потужності шуму **snr_linear** в одиниці виміру рази;

x_noise_dB – результуючий зашумлений сигнал, в якому дотримано задане співвідношення потужності сигналу до потужності шуму **snr_dB** в одиниці виміру децибелі.

10.2. Змоделювати процес додавання до вихідного сигналу **x** адитивного білого гаусівського нормального шуму з необхідним відношенням сигнал/шум в програмному середовищі MATLAB за допомогою функції **awgn**.

Задайте частоту **F_s** (Гц) і крок **ts** (с) дискретизації сигналу **x** тривалістю **T** (с).

```
Fs = 200;  
ts = 1/Fs;  
T = 0:ts:1;
```

Задайте амплітуду **A** (В), частоту **f** (Гц) і початкову фазу **phi** (рад) гармонічного коливання.

```
A = 2;  
f = 6;  
phi = pi/2;
```

Згенеруйте дискретний сигнал **x** тривалістю **T** (с), який змінюється за гармонійним законом згідно заданої амплітуди **A** (В), частоти **f** (Гц) і початкової фази **phi** (рад).

```
x = A*sin(2*pi*f*T + phi);
```

Виконайте додавання до вихідного сигналу **x** адитивного білого гаусівського шуму за допомогою функції **awgn**, таким чином, що результуючі зашумлені сигнали **x_noise_linear** та **x_noise_dB** матимуть відношення сигнал/шум **snr_linear** в разях або **snr_dB** в децибелах, відповідно.

```
snr_linear = 20;  
snr_dB = 20;  
x_noise_linear = awgn(x, snr_linear, 'measured', 'linear');  
x_noise_dB = awgn(x, snr_dB, 'measured', 'dB');
```

10.3. Побудувати графіки вихідного сигналу **x** та зашумлених сигналів **x_noise_linear** та **x_noise_dB** із заданими відношеннями сигнал/шум в разях **snr_linear** та децибелах **snr_dB**, відповідно.

Побудуйте графіки вихідного сигналу **x** та зашумленого сигналу **x_noise_linear** із заданим відношенням сигнал/шум **snr_linear** в разях.

```

figure(1)
subplot(2,2,1)
plot(T, x, T, x_noise_linear, 'linewidth', 2)
title('Вихідний та зашумлений із заданим відношенням сиг-
нал/шум в разях сигнали')
ylabel('Амплітуда')
xlabel('Час (с)')
legend('Вихідний сигнал', 'Зашумлений сигнал')
grid on

```

Побудуйте графіки вихідного сигналу x та зашумленого сигналу x_noise_dB із заданим відношенням сигнал/шум snr_dB в децибелах.

```

figure(1)
subplot(2,2,3)
plot(T, x, T, x_noise_dB, 'linewidth', 2)
title('Вихідний та зашумлений із заданим відношенням сиг-
нал/шум в децибелах сигнали')
ylabel('Амплітуда')
xlabel('Час (с)')
legend('Вихідний сигнал', 'Зашумлений сигнал')
grid on

```

10.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.
4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Наведіть і розкрийте сутність основних інформаційних характеристик джерел неперервних повідомлень.

2. Виконайте оцінку використання пропускної здатності неперервного каналу зв'язку при передаванні дискретних і неперервних повідомлень.

3. Поясніть залежність нормованої по смузі частот сигналу пропускної здатності каналу від відношення сигнал/шум у каналі.

4. Поясніть залежність нормованої по пропускній здатності смуги пропускання каналу від відношення E_b/Ω_0 .

5. Поясніть залежність нормованої по еквівалентній смузі шуму пропускної здатності від відношення $\Delta F_c/\Delta F_0$.

6. Як визначається нормована версія сигнал/шуму в каналі зв'язку?

7. Дайте визначення поняттю – диференціальна ентропія гаусівського білого шуму.

8. Порівняйте нормовану по смузі частот каналу пропуску здатність гаусівського неперервного каналу зв'язку з пропусковими здатностями побудованих на його основі двійкового симетричного і багатопозиційного дискретних каналів зв'язку без пам'яті. Наведіть пояснюючі графіки.

9. Вважаючи, що відношення сигнал/шум у каналі зв'язку становить 48 дБ, а смуга частот передаваного сигналу дорівнює 0,032 ГГц, обчисліть еквівалентну смугу шуму.

10. Записати та пояснити синтаксичну конструкцію функції **awgn** мови програмування MATLAB і дати визначення її параметрам **x**, **snr_linear**, **snr_dB**, **'measured'**, **'linear'**, **'dB'**, **x_noise_linear**, **x_noise_dB**.

Лабораторна робота 11

НИЗЬКОЧАСТОТНИЙ, ВИСОКОЧАСТОТНИЙ, СМУГОВИЙ І РЕЖЕКТОРНИЙ ФІЛЬТРИ БАТТЕРВОРТА

Мета роботи: дослідити процес низькочастотної, високочастотної, смугової і режекторної фільтрації фільтрами Баттерворта за допомогою комп'ютерного моделювання використовуючи допоміжну функцію **butter** з бібліотеки Signal Processing Toolbox мови програмування MATLAB.

Основні теоретичні відомості

За формою АЧХ розрізняють фільтри нижніх частот (ФНЧ), фільтри верхніх частот (ФВЧ), фільтри смугові (СФ) і режекторні

фільтри (РФ). Приклади АЧХ для наведених типів фільтрів показано на рис. 11.1.

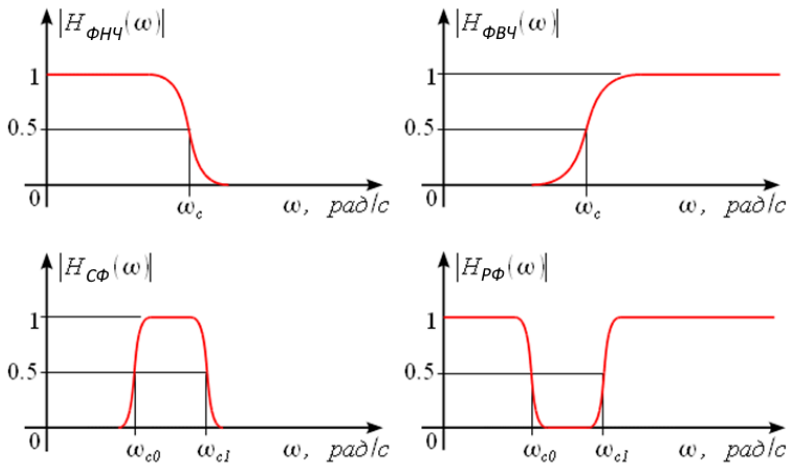


Рис. 11.1. Приклади АЧХ різних фільтрів

Розглянемо постановку завдання розрахунку фільтра з прикладу ФНЧ. В ідеалі ми хотіли б отримати фільтр, який пропускає без спотворень всі частоти нижче ω_0 і повністю придушує всі частоти вище ω_0 . Такий ФНЧ називають ідеальним, і він не може бути реалізований на практиці. Реалізовані ж ФНЧ завжди вносять якісь спотворення в смугі пропускання і не до кінця пригнічують смугу загородження.

На рис. 11.2 показані ідеальна та реальна АЧХ ФНЧ. Прямокутник – АЧХ ідеального фільтра, згладжений прямокутник – реального.

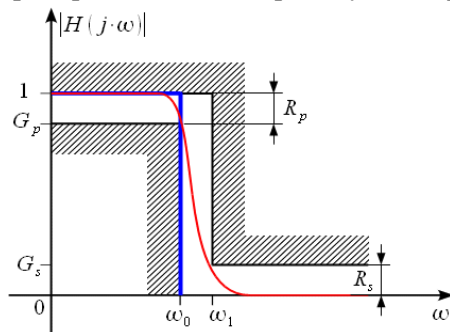


Рис. 11.2. Ідеальна та реальна АЧХ ФНЧ

Смуга частот від 0 до ω_0 називається смугою пропускання ФНЧ, смуга частот від ω_1 і вище називається смугою придушення або смугою загородження. Смуга між ω_0 і ω_1 називається перехідною смугою фільтра. Параметр

$$G_p = 1 / \sqrt{1 + \varepsilon_p^2}$$

визначає максимальне спотворення сигналу в смузі пропускання, а параметр

$$G_s = 1 / \sqrt{1 + \varepsilon_s^2}$$

задає необхідне придушення у смузі загородження. Таким чином, отримали такий «вигнутий коридор», в який має поміститися АЧХ нашого фільтра. При цьому, чим «коридор ужчий», тим ε_p менший параметр, а параметр ε_s більший. Прийнято спотворення у смузі пропускання та необхідне придушення виражати в децибелах, тоді:

$$R_p = 10 \cdot \log(1 + \varepsilon_p^2) = -20 \cdot \log(G_p) \quad \text{дБ};$$

$$R_s = 10 \cdot \log(1 + \varepsilon_s^2) = -20 \cdot \log(G_s) \quad \text{дБ}.$$

Звідки можна сказати:

$$\varepsilon_p = \sqrt{10^{R_p/10} - 1};$$

$$\varepsilon_s = \sqrt{10^{R_s/10} - 1}.$$

Таким чином, для розрахунку фільтра достатньо задати коридор АЧХ шляхом завдання вищенаведених параметрів.

Часто при розрахунку фільтра використовують ще два параметри, які ми також будемо надалі використовувати:

$$k = \frac{\omega_0}{\omega_1}; \quad k_1 = \frac{\varepsilon_p}{\varepsilon_s}.$$

Параметр k визначає селективні властивості фільтра. Якщо звужувати перехідну смугу, то k наблизатиметься до одиниці. З іншого боку параметр k_1 визначає ступінь придушення фільтра з урахуванням спотворень, що вносяться. Так, якщо коефіцієнт придушення у смузі загородження зростає, то k_1 наблизатиметься до нуля. Аналогічно k_1 наблизатиметься до нуля, якщо коефіцієнт нерівномірності у смузі пропускання наблизатиметься до нуля.

Фільтри Баттерворта нижніх частот характеризуються тим, що мають максимально гладку амплітудну характеристику на початку координат s -площини. Квадрат амплітудної характеристики нормованого (тобто має частоту зрізу 1 рад/с) фільтра Баттерворта дорівнює

$$|H(\omega)|^2 = 1 / (1 + (\omega^2)^P),$$

де P – порядок фільтра. Аналітично продовжуючи функцію на всю s -площину, отримуємо:

$$H(s)H(-s) = 1 / (1 - (-s^2)^P).$$

Усі полюси цієї функції знаходяться на одиничному колі на однаковій відстані один від одного в s -площині. Виразимо передатну функцію $H(s)$ через полюси, що розташовуються в лівій напівплощині s :

$$H(s) = k_0 / \prod_{k=1}^P (s - s_k),$$

де $s_k = e^{j\pi[1/2+(2k-1)/2P]}$, $k=1,2,\dots,P$; k_0 – константа нормування.

Можна сформулювати кілька властивостей фільтрів Баттерворта нижніх частот:

– фільтри Баттерворта мають лише полюси (всі нулі передавальних функцій цих фільтрів розташовані на нескінченності).

– на частоті $\omega = 1$ рад/с коефіцієнт передачі фільтра дорівнює $1/\sqrt{2}$ (тобто на частоті зрізу їх амплітудна характеристика знижується на 3 дБ).

– порядок фільтра *повністю* визначає весь фільтр.

Цифрові фільтри із заданою функцією передачі можна побудувати різними способами. У будь-якому реальному фільтрі шуми та похибки, що з'являються при квантуванні, суттєво залежать від структури фільтра. Насамперед усі фільтри можна розділити на два великі класи – рекурсивні, нерекурсивні.

Для рекурсивних фільтрів співвідношення між вхідною послідовністю $\{x(n)\}$ та відгуком фільтра $\{y(n)\}$ може бути записано наступним чином: $y(n) = F[y(n-1), y(n-2), \dots, x(n), x(n-1), \dots]$, тобто поточний відлік відгуку $y(n)$ визначається не тільки поточним та попереднім значеннями вхідної послідовності, а й попередніми відліками відгуку.

У нерекурсивних фільтрах зв'язок між вхідною послідовністю та відгуком має вигляд $y(n) = F[x(n), x(n-1), \dots]$, тобто поточний відлік відгуку залежить від поточного та попередніх значень вхідної послідовності.

Розглянемо передатну функцію N -го порядку виду

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^N a_i z^{-i} \Big/ \sum_{i=0}^N b_i z^{-i},$$

причому $b_0 = 1$.

Привівши цю рівність до спільного знаменника, отримаємо:

$$Y(z) \sum_{i=0}^N b_i z^{-i} = X(z) \sum_{i=0}^N a_i z^{-i},$$

або

$$\sum_{i=0}^N b_i z^{-i} Y(z) = \sum_{i=0}^N a_i z^{-i} X(z).$$

Якщо розглядати члени виду $z^{-k} Y(z)$ як зворотні z перетворення послідовностей $y(n-k)$, то, взявши зворотні z перетворення обох частин останньої рівності, можна отримати шукане різницеве рівняння

$$\sum_{i=0}^N b_i y(n-i) = \sum_{i=0}^N a_i x(n-i).$$

Оскільки $b_0 = 1$ різницеве рівняння можна вирішити щодо $y(n)$:

$$y(n) = \sum_{i=0}^N a_i x(n-i) - \sum_{i=1}^N b_i x(n-i).$$

Проста структура реалізації даного різничевого рівняння, звана прямою формою, показана на рис. 11.3.

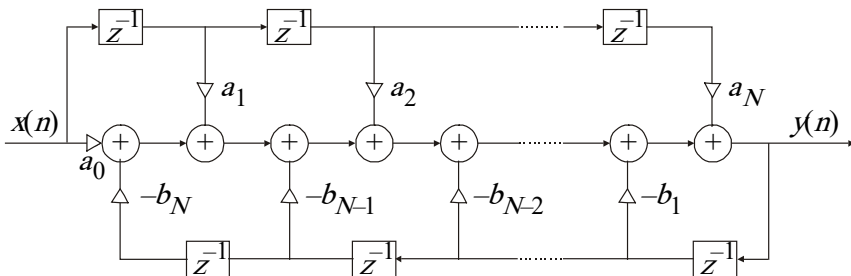


Рис. 11.3. Структура реалізації різничевого рівняння

У ньому для утворення ланцюгів використовуються окремі елементи затримки. Характерними рисами цієї структури є її простота і безпосередній зв'язок з z -перетворенням. Однак ця структура дуже чутлива до квантування коефіцієнтів.

За аналогією з фільтрами безперервного часу існує кілька простих перетворень цифрового фільтра нижніх частот (з частотою зрізу ω_c) в інший фільтр нижніх частот (з іншою частотою зрізу ω_u), а також цифровий фільтр верхніх частот, смуговий або режекторний. Далі наведено формули цих перетворень.

1. Фільтр нижніх частот \rightarrow фільтр нижніх частот:

$$z^{-1} \rightarrow \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}},$$

де $\alpha = \frac{\sin \{[(\omega_c - \omega_u) / 2]T\}}{\sin \{[(\omega_c + \omega_u) / 2]T\}}$; ω_u – задана частота зрізу фільтра

нижніх частот.

2. Фільтр нижніх частот \rightarrow фільтр верхніх частот:

$$z^{-1} \rightarrow -\left(\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}\right),$$

де $\alpha = -\frac{\cos \{[(\omega_c - \omega_u) / 2]T\}}{\cos \{[(\omega_c + \omega_u) / 2]T\}}$; ω_u – задана частота зрізу фільт-

ра верхніх частот.

3. Фільтр нижніх частот \rightarrow смуговий фільтр:

$$z^{-1} \rightarrow -\frac{\{z^{-2} - [2\alpha k / (k+1)]z^{-1} + (k-1) / (k+1)\}}{\{[(k-1) / (k+1)]z^{-2} - [2\alpha k / (k+1)]z^{-1} + 1\}}.$$

Тут

$$\alpha = \cos(\omega_0 T) = \frac{\cos \{[(\omega_u + \omega_l) / 2]T\}}{\cos \{[(\omega_u - \omega_l) / 2]T\}}; \quad k = ctg \left[\left(\frac{\omega_u - \omega_l}{2} \right) T \right] tg \left(\frac{\omega_c T}{2} \right);$$

ω_0 – центральна частота смугового фільтра.

4. Фільтр нижніх частот \rightarrow режекторний фільтр:

$$z^{-1} \rightarrow \frac{\{z^{-2} [2\alpha / (1+k)]z^{-1} + (1-k) / (1+k)\}}{\{[(1-k) / (1+k)]z^{-2} - [2\alpha / (1+k)]z^{-1} + 1\}}.$$

Тут

$$\alpha = \cos(\omega_0 T) = \frac{\cos\left\{\left[\left(\omega_u + \omega_l\right) / 2\right] T\right\}}{\cos\left\{\left[\left(\omega_u - \omega_l\right) / 2\right] T\right\}}; \quad k = \operatorname{tg}\left[\left(\frac{\omega_u - \omega_l}{2}\right) T\right] \operatorname{tg}\left(\frac{\omega_c T}{2}\right);$$

ω_0 – центральна частота режекторного фільтра.

Послідовність виконання роботи

11.1. Вивчити синтаксичну конструкцію функції **butter** мови програмування MATLAB і дати визначення її параметрам **n**, **Wn_lowpass**, **b_lowpass**, **a_lowpass**, **Wn_highpass**, **'high'**, **b_highpass**, **a_highpass**, **Wn_bandpass**, **b_bandpass**, **a_bandpass**, **Wn_bandstop**, **'stop'**, **b_bandstop**, **a_bandstop**.

Синтаксис функції **butter** для синтезу низькочастотного, високочастотного, смугового і режекторного фільтрів Баттерворта:

```
[b_lowpass, a_lowpass] = butter(n, Wn_lowpass)  
[b_highpass, a_highpass] = butter(n, Wn_highpass, 'high')  
[b_bandpass, a_bandpass] = butter(n, Wn_bandpass)  
[b_bandstop, a_bandstop] = butter(n, Wn_bandstop, 'stop')
```

Опис параметрів функції:

n – порядок фільтра Баттерворта;

Wn_lowpass – нормована частота зрізу для фільтра нижніх частот Баттерворта, яка знаходиться відповідно до запису **Wn_lowpass = fc_lowpass/(Fs/2)**, де **fc_lowpass** (Гц) – частота зрізу в герцах, **Fs** (Гц) – частота дискретизації;

b_lowpass, **a_lowpass** – коефіцієнти поліномів чисельника **b_lowpass** і знаменника **a_lowpass** функції передачі дискретного фільтра нижніх частот Баттерворта **n**-го порядку з частотою зрізу, рівною **fc_lowpass** (Гц), що відповідає нормованій частоті зрізу **Wn_lowpass**;

Wn_highpass – нормована частота зрізу для фільтра верхніх частот Баттерворта, яка знаходиться відповідно до запису **Wn_highpass = fc_highpass/(Fs/2)**, де **fc_highpass** (Гц) – частота зрізу в герцах, **Fs** (Гц) – частота дискретизації;

'high' – запускає фільтр верхніх частот Баттерворта;

b_highpass, **a_highpass** – коефіцієнти поліномів чисельника **b_highpass** і знаменника **a_highpass** функції передачі дискретного фільтра верхніх частот Баттерворта **n**-го порядку з частотою зрізу, рівною **fc_highpass** (Гц), що відповідає нормованій частоті зрізу **Wn_highpass**;

Wn_bandpass – нормована частотна смуга пропускання для смугового фільтра Баттерворта, яка знаходиться відповідно до запису $\mathbf{Wn_bandpass} = \mathbf{fc_bandpass}/(\mathbf{Fs}/2)$, де $\mathbf{fc_bandpass} = [\mathbf{fc_bandpass_lower} \ \mathbf{fc_bandpass_upper}]$ (Гц) – частотна смуга пропускання в герцах з нижньою $\mathbf{fc_bandpass_lower}$ (Гц) та верхньою $\mathbf{fc_bandpass_upper}$ (Гц) частотами зрізу, \mathbf{Fs} (Гц) – частота дискретизації;

b_bandpass, **a_bandpass** – коефіцієнти поліномів чисельника **b_bandpass** і знаменника **a_bandpass** функції передачі смугового дискретного фільтра Баттерворта n -го порядку з нижньою $\mathbf{fc_bandpass_lower}$ (Гц) та верхньою $\mathbf{fc_bandpass_upper}$ (Гц) частотами зрізу смуги пропускання $\mathbf{fc_bandpass}$ (Гц), що відповідає нормованій смузі пропускання **Wn_bandpass**;

Wn_bandstop – нормована частотна смуга затримування для режекторного фільтра Баттерворта, яка знаходиться відповідно до запису $\mathbf{Wn_bandstop} = \mathbf{fc_bandstop}/(\mathbf{Fs}/2)$, де $\mathbf{fc_bandstop} = [\mathbf{fc_bandstop_lower} \ \mathbf{fc_bandstop_upper}]$ (Гц) – частотна смуга затримування в герцах з нижньою $\mathbf{fc_bandstop_lower}$ (Гц) та верхньою $\mathbf{fc_bandstop_upper}$ (Гц) частотами зрізу, \mathbf{Fs} (Гц) – частота дискретизації;

'stop' – запускає режекторний фільтр Баттерворта;

b_bandstop, **a_bandstop** – коефіцієнти поліномів чисельника **b_bandstop** і знаменника **a_bandstop** функції передачі режекторного дискретного фільтра Баттерворта n -го порядку з нижньою $\mathbf{fc_bandstop_lower}$ (Гц) та верхньою $\mathbf{fc_bandstop_upper}$ (Гц) частотами зрізу смуги затримування $\mathbf{fc_bandstop}$ (Гц), що відповідає нормованій смузі затримування **Wn_bandstop**.

11.2. Змоделювати процес низькочастотної, високочастотної, смугової і режекторної дискретної фільтрації Баттерворта зашумленого сигналу **x_noise** в програмному середовищі MATLAB за допомогою функції **butter**.

Задайте частоту \mathbf{Fs} (Гц) і крок \mathbf{ts} (с) дискретизації сигналу тривалістю \mathbf{T} (с).

```
Fs = 1000;
```

```
ts = 1/Fs;
```

```
T = 0:ts:1;
```

Задайте амплітуду \mathbf{A} (В), частоту \mathbf{f} (Гц) і початкову фазу $\mathbf{\phi}$ (рад) гармонічного коливання.

```
A = 2;  
f = 10;  
phi = 0;
```

Згенеруйте дискретний сигнал **x** тривалістю **T** (с), який змінюється за гармонійним законом згідно заданої амплітуди **A** (В), частоти **f** (Гц) і початкової фази **phi** (рад).

```
x = A*sin(2*pi*f*T + phi);
```

Згенеруйте дискретний білий гаусівський (нормальний) шум з потужністю (дисперсія), яка дорівнює 0.25.

```
noise = wgn(size(x, 1), size(x, 2), 0.25, 'linear');
```

Згенеруйте дискретний сигнал під впливом білого шуму.

```
x_noise = x + noise;
```

Синтезуйте коефіцієнти поліномів чисельника **b_lowpass** і знаменника **a_lowpass** функції передачі дискретного фільтра нижніх частот Баттерворта **n**-го порядку з частотою зрізу, рівною **fc_lowpass** (Гц), що відповідає нормованій частоті зрізу **Wn_lowpass**.

```
n = 10;
```

```
fc_lowpass = 30;
```

```
Wn_lowpass = fc_lowpass/(Fs/2);
```

```
[b_lowpass, a_lowpass] = butter(n, Wn_lowpass);
```

Виконайте низькочастотну дискретну фільтрацію сигналу під впливом білого шуму **x_noise** згідно заданої частоти зрізу **fc_lowpass** (Гц) за допомогою функції **filter**.

```
x_noise_lowpass = filter(b_lowpass, a_lowpass, x_noise);
```

Синтезуйте коефіцієнти поліномів чисельника **b_highpass** і знаменника **a_highpass** функції передачі дискретного фільтра верхніх частот Баттерворта **n**-го порядку з частотою зрізу, рівною **fc_highpass** (Гц), що відповідає нормованій частоті зрізу **Wn_highpass**.

```
fc_highpass = 300;
```

```
Wn_highpass = fc_highpass/(Fs/2);
```

```
[b_highpass, a_highpass] = butter(n, Wn_highpass, 'high');
```

Виконайте високочастотну дискретну фільтрацію сигналу під впливом білого шуму **x_noise** згідно заданої частоти зрізу **fc_highpass** (Гц) за допомогою функції **filter**.

```
x_noise_highpass = filter(b_highpass, a_highpass, x_noise);
```

Синтезуйте коефіцієнти поліномів чисельника **b_bandpass** і знаменника **a_bandpass** функції передачі смугового дискретного фільтра Баттерворта **n**-го порядку з нижньою **fc_bandpass_lower** (Гц) та верхньою **fc_bandpass_upper** (Гц) частотами зрізу смуги пропускання **fc_bandpass** (Гц), що відповідає нормованій смузі пропускання **Wn_bandpass**.

```
fc_bandpass_lower = 100;
fc_bandpass_upper = 200;
fc_bandpass = [fc_bandpass_lower fc_bandpass_upper];
Wn_bandpass = fc_bandpass/(Fs/2);
[b_bandpass, a_bandpass] = butter(n, Wn_bandpass);
```

Виконайте смугову дискретну фільтрацію сигналу під впливом білого шуму **x_noise** згідно заданої смуги пропускання **fc_bandpass** (Гц) за допомогою функції **filter**.

```
x_noise_bandpass = filter(b_bandpass, a_bandpass, x_noise);
```

Синтезуйте коефіцієнти поліномів чисельника **b_bandstop** і знаменника **a_bandstop** функції передачі режекторного дискретного фільтра Баттерворта **n**-го порядку з нижньою **fc_bandstop_lower** (Гц) та верхньою **fc_bandstop_upper** (Гц) частотами зрізу смуги затримування **fc_bandstop** (Гц), що відповідає нормованій смузі затримування **Wn_bandstop**.

```
fc_bandstop_lower = 100;
fc_bandstop_upper = 300;
fc_bandstop = [fc_bandstop_lower fc_bandstop_upper];
Wn_bandstop = fc_bandstop/(Fs/2);
[b_bandstop, a_bandstop] = butter(n, Wn_bandstop, 'stop');
```

Виконайте режекторну дискретну фільтрацію сигналу під впливом білого шуму **x_noise** згідно заданої смуги затримування **fc_bandstop** (Гц) за допомогою функції **filter**.

```
x_noise_bandstop = filter(b_bandstop, a_bandstop, x_noise);
```

11.3. Побудувати графіки зашумленого сигналу після обробки низькочастотним **x_noise_lowpass**, високочастотним **x_noise_highpass**, смуговим **x_noise_bandpass** і режекторним **x_noise_bandstop** дискретними фільтрами Баттерворта та їх спектрів.

Побудуйте графіки зашумленого **x_noise** і обробленого низькочастотним фільтром **x_noise_lowpass** сигналів.

```
figure(1)
subplot(2,2,1)
```

```

plot(T, x_noise)
hold on
plot(T, x_noise_lowpass)
legend('Зашумлений сигнал', 'Фільтрований сигнал')
xlabel('Час (с)')
ylabel('Амплітуда')
title('Зашумлений і оброблений низькочастотним фільтром сиг-
нали')
grid on

```

Побудуйте графіки спектрів зашумленого **x_noise** і обробленого низькочастотним фільтром **x_noise_lowpass** сигналів.

```

figure(1)
subplot(2,2,3)
pspectrum(x_noise, T)
hold on
pspectrum(x_noise_lowpass, T)
legend('Спектр зашумленого сигналу', 'Спектр фільтрованого си-
гналу')
xlabel('Частота (Гц)')
ylabel('Потужність спектра (дБ)')
title('Спектри зашумленого і обробленого НЧФ сигналів')
grid on

```

Побудуйте графіки зашумленого **x_noise** і обробленого високочастотним фільтром **x_noise_highpass** сигналів.

```

figure(1)
subplot(2,2,2)
plot(T, x_noise)
hold on
plot(T, x_noise_highpass)
legend('Зашумлений сигнал', 'Фільтрований сигнал')
xlabel('Час (с)')
ylabel('Амплітуда')
title('Зашумлений і оброблений високочастотним фільтром сиг-
нали')
grid on

```

Побудуйте графіки спектрів зашумленого **x_noise** і обробленого високочастотним фільтром **x_noise_highpass** сигналів.

```
figure(1)
```

```

subplot(2,2,4)
pspectrum(x_noise, T)
hold on
pspectrum(x_noise_highpass, T)
legend('Спектр зашумленого сигналу', 'Спектр фільтрованого си-
гналу')
xlabel('Частота (Гц)')
ylabel('Потужність спектра (дБ)')
title('Спектри зашумленого і обробленого височастотним фі-
льтром сигналів')
grid on

```

Побудуйте графіки зашумленого **x_noise** і обробленого смуго-
вим фільтром **x_noise_bandpass** сигналів.

```

figure(2)
subplot(2,2,1)
plot(T, x_noise)
hold on
plot(T, x_noise_bandpass)
legend('Зашумлений сигнал', 'Фільтрований сигнал')
xlabel('Час (с)')
ylabel('Амплітуда')
title('Зашумлений і оброблений смуговим фільтром сигнали')
grid on

```

Побудуйте графіки спектрів зашумленого **x_noise** і обробленого
смуговим фільтром **x_noise_bandpass** сигналів.

```

figure(2)
subplot(2,2,3)
pspectrum(x_noise, T)
hold on
pspectrum(x_noise_bandpass, T)
legend('Спектр зашумленого сигналу', 'Спектр фільтрованого си-
гналу')
xlabel('Частота (Гц)')
ylabel('Потужність спектра (дБ)')
title('Спектри зашумленого і обробленого смуговим фільтром
сигналів')
grid on

```

Побудуйте графіки зашумленого **x_noise** і обробленого режекторним фільтром **x_noise_bandstop** сигналів.

```
figure(2)
subplot(2,2,2)
plot(T, x_noise)
hold on
plot(T, x_noise_bandstop)
legend('Зашумлений сигнал', 'Фільтрований сигнал')
xlabel('Час (с)')
ylabel('Амплітуда')
title('Зашумлений і оброблений режекторним фільтром сигнали')
grid on
```

Побудуйте графіки спектрів зашумленого **x_noise** і обробленого режекторним фільтром **x_noise_bandstop** сигналів.

```
figure(2)
subplot(2,2,4)
pspectrum(x_noise, T)
hold on
pspectrum(x_noise_bandstop, T)
legend('Спектр зашумленого сигналу', 'Спектр фільтрованого сигналу')
xlabel('Частота (Гц)')
ylabel('Потужність спектра (дБ)')
title('Спектри зашумленого і обробленого режекторним фільтром сигналів')
grid on
```

11.4. Оформити звіт за результатами виконаної практичної роботи і вивченого теоретичного матеріалу.

Звіт з лабораторної роботи повинен містити інформацію згідно наступних пунктів:

1. Тема лабораторної роботи та її мета.
2. Знімки екрану робочих вікон Command Window і Workspace програмного середовища MATLAB з виконаною практичною частиною лабораторної роботи.
3. Побудовані графіки згідно завдання практичної частини лабораторної роботи.

4. Проаналізувати теоретичну а також практичну частину лабораторної роботи, і на підставі чого написати обґрунтовані висновки власного судження в рамках вивченого матеріалу.

Запитання та завдання для самоконтролю

1. Які типи фільтрів існують, та як вони відрізняються за формою АЧХ?
2. Поясніть структуру реалізації різницевого рівняння.
3. Як ви розумієте поняття – ідеальна та реальна АЧХ ФНЧ, та в чому їх відмінність?
4. Які параметри використовують при розрахунку фільтра.
5. Що називається перехідною смугою фільтра?
6. Яку роль відіграє АЧХ в розрахунку цифрових фільтрів?
7. Які властивості фільтрів Баттерворта нижніх частот ви знаєте?
8. Як ви розумієте поняття – порядок фільтра, та яка його роль в розрахунку фільтрів?
9. Чи можливо маючи розраховані коефіцієнти НЧФ, розрахувати банк фільтрів ФВЧ, СФ та РФ?
10. Записати та пояснити синтаксичну конструкцію функції **butter** мови програмування MATLAB і дати визначення її параметрам **n**, **Wn_lowpass**, **b_lowpass**, **a_lowpass**, **Wn_highpass**, **'high'**, **b_highpass**, **a_highpass**, **Wn_bandpass**, **b_bandpass**, **a_bandpass**, **Wn_bandstop**, **'stop'**, **b_bandstop**, **a_bandstop**.

ПІСЛЯМОВА

Виконання лабораторних робіт в системі вивчення дисципліни «Системи мобільного радіозв'язку» сприяє набуттю базових практичних навичок в побудові сучасних цифрових систем зв'язку з рухомими об'єктами за допомогою комп'ютерного моделювання використовуючи допоміжні функції з бібліотек Signal Processing Toolbox та Communications Toolbox мови програмування MATLAB, що дозволяє набагато ефективніше засвоїти теоретичні викладення лабораторного практикуму та лекційного матеріалу.

Слід зауважити, що успішне виконання даного курсу лабораторних робіт неможливе без знань та практичних умінь використовувати сучасний програмний комплекс MathWorks MATLAB, який є лідером у вирішенні технічних, інженерних та наукових завдань.

Навчання за допомогою даного лабораторного практикуму сприяє формуванню у здобувачів вищої освіти професійних знань та навичок, необхідних для комп'ютерного моделювання процесів, які є невід'ємною частиною у побудові систем мобільного радіозв'язку, а саме – дискретизація, квантування та кодування сигналів; дискретна фільтрація; кодування мови методом довготривалого передбачення з регулярним імпульсним збудженням; циклічне та згорткове кодування; блоково-діагональне перемешування; частотна маніпуляція з мінімальним зсувом; канал зв'язку з адитивним білим шумом; низькочастотні, високочастотні, смугові та режкторні фільтри Баттерворта.

Опанування теоретичними та практичними основами, які викладені у даному лабораторному практикумі дозволить здобувачу вищої освіти сформувати базове розуміння процесу побудови систем мобільного радіозв'язку, що є основоположним моментом в підготовці фахівців спеціальності 172 «Телекомунікації та радіотехніка».

СПИСОК ЛІТЕРАТУРИ

1. Кравчук С.О., Голубничий О.Г., Тараненко А.Г., Потапов В.Г., Ткаліч О.П. Системи зв'язку з рухомими об'єктами: підручник. — К.: «Спринт-Сервіс», 2012. — 452 с.
2. Конахович Г.Ф. Системи радіозв'язку: навч. посіб. — К.: НАУ, 2004. — 311 с.
3. Конахович Г.Ф., Мачалін І.О., Пузиренко О.Ю. Теорія електричного зв'язку: навч. посіб. — К.: «Інтерсервіс», 2013. — 368 с.
4. Гусев О.Ю., Конахович Г.Ф., Корнієнко В.І., Кузнецов Г.В., Пузиренко О.Ю. Теорія електричного зв'язку: навч. посіб. — Львів: «Магнолія 2006», 2011. — 364 с.
5. Бортник Г.Г., Кичак В.М. Цифрова обробка сигналів в телекомунікаційних системах: підручник. — Вінниця: ВНТУ, 2014. — 232 с.
6. MathWorks MATLAB R2022b Documentation. Режим доступу: <https://www.mathworks.com/help/matlab/> — 02.12.2022.

ЗМІСТ

| | |
|-------------------------------------------------------------------------------------------------------------------|----|
| Вступ | 3 |
| Лабораторна робота 1. Дискретизація аналогових сигналів | 4 |
| Лабораторна робота 2. Квантування дискретних сигналів | 12 |
| Лабораторна робота 3. Кодування та імпульсно-кодова модуляція | 19 |
| Лабораторна робота 4. Дискретна фільтрація | 26 |
| Лабораторна робота 5. Кодування мови методом довготривалого передбачення з регулярним імпульсним збудженням | 33 |
| Лабораторна робота 6. Циклічне кодування | 41 |
| Лабораторна робота 7. Згорткове кодування | 49 |
| Лабораторна робота 8. Завадостійке кодування в системі GSM..... | 56 |
| Лабораторна робота 9. Частотна маніпуляція з мінімальним зсувом | 63 |
| Лабораторна робота 10. Канал зв'язку з адитивним білим шумом | 71 |
| Лабораторна робота 11. Низькочастотний, високочастотний, смуговий і режекторний фільтри Баттерворта | 79 |
| Післямова | 93 |
| Список літератури | 94 |

Навчальне видання

СИСТЕМИ МОБІЛЬНОГО РАДІОЗВ'ЯЗКУ

ЛАБОРАТОРНИЙ ПРАКТИКУМ
для здобувачів вищої освіти
ОС «Бакалавр» спеціальності 172
«Телекомунікації та радіотехніка»

Укладачі:

ЛАВРИНЕНКО Олександр Юрійович
ТАРАНЕНКО Анатолій Григорович
БАХТІЯРОВ Денис Ілшатович
ГОЛУБНИЧИЙ Олексій Георгійович
ЖАРОВА Олена Валеріївна

В авторській редакції

Технічний редактор *А. І. Лавринович*
Комп'ютерна верстка *Н. В. Черної*