

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Г.Є. ФІЛАТОВА

**МАТЕМАТИЧНІ ОСНОВИ ОБРОБКИ
СИГНАЛІВ ТА ЗОБРАЖЕНЬ:
ТЕОРІЯ ТА ПРАКТИКА**

Підручник

Затверджено
вченою радою НТУ «ХП»,
протокол № 7 від 02.07.2021р.

Харків
НТУ «ХП»
2021

Рецензенти: А.А. Коваленко, д-р техн. наук, проф., завідувач каф. «Електронні обчислювальні машини» Харківського національного університету радіоелектроніки;
С.М. Порошин, д-р. техн. наук, проф., завідувач каф. «Мультимедійні інформаційні технології і системи» Національного технічного університету «Харківський політехнічний інститут»

Філатова Г.Є.

Ф 51 Математичні основи обробки сигналів та зображень: теорія та практика: підручник / Г.Є. Філатова. – Х. : НТУ «ХПІ», 2021. – 237 с.

У підручнику розглянуто математичні основи обробки сигналів (ряди та перетворення Фур'є, згортки, z -перетворення, проектування цифрових фільтрів) та зображень (двовірне перетворення Фур'є, поелементні перетворення, двовірне цифрова фільтрація, косинусне та вейвлет-перетворення). Теоретичні відомості та розрахунки супроводжуються прикладами програм в ППП MatLab.

Призначено для студентів спеціальності 123 «Комп'ютерна інженерія».

Іл. 67. Табл. 18. Бібліогр.: 10 назв.

УДК 004.032.2

© Г.Є. Філатова 2021
© НТУ «ХПІ», 2021

СОДЕРЖАНИЕ

1. РЯДИ ФУР'Є БЕЗПЕРЕРВНИХ СИГНАЛІВ	8
1.1. Класифікація безперервних сигналів	8
1.2. Ряди Фур'є безперервних сигналів	9
1.2.1. Розкладення в ряд Фур'є періодичних сигналів	9
1.2.2. Розкладення в ряд Фур'є неперіодичних сигналів	10
1.3. Приклади до розділу 1	12
1.4. Завдання до розділу 1 для самостійної роботи	16
2. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗКЛАДЕННЯ В РЯД ФУР'Є ДЕЯКИХ СИГНАЛІВ	18
2.1. Ряд Фур'є послідовності прямокутних імпульсів	18
2.2. Меандр	22
2.3. Пілкоподібний сигнал	24
2.4. Послідовність трикутних імпульсів	26
2.5. Завдання до розділу 2 для самостійної роботи	29
3. ПРЯМЕ ТА ЗВОРОТНЕ ДИСКРЕТНЕ ПЕРЕТВОРЕННЯ ФУР'Є	30
3.1. Дискретне перетворення Фур'є	30
3.2. Властивості дискретного перетворення Фур'є	31
3.3. Приклади до розділу 3	40
3.4. Завдання до розділу 3 для самостійної роботи	43
4. ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є	45
4.1. Алгоритми швидкого перетворення Фур'є	45
4.1.1. Швидке перетворення Фур'є з проріджуванням за часом	45
4.1.2. Швидке перетворення Фур'є з проріджуванням за частотою	46
4.1.3. Алгоритм зворотного швидкого перетворення Фур'є з проріджуванням за частотою	47
4.2. Приклади до розділу 4	48
4.3. Завдання до розділу 4 для самостійної роботи	53

5. КРУГОВА І ЛІНІЙНА ЗГОРТКИ	54
5.1. Визначення згортки	54
5.1.1. Кругова згортка	54
5.1.2. Лінійна згортка	55
5.2. Приклади до розділу 5	55
5.3. Завдання до розділу 5 для самостійної роботи	59
6. КОРЕЛЯЦІЙНА ФУНКЦІЯ.....	61
6.1. Автокореляційна функція та її властивості	61
6.2. Взаємнокореляційна функція та її властивості	63
6.3. Приклади до розділу 6	65
6.4. Завдання до розділу 6 для самостійної роботи	70
7. Z-ПЕРЕТВОРЕННЯ	71
7.1. Пряме z-перетворення	71
7.2. Зворотнє z-перетворення	71
7.3. Властивості z-перетворення	72
7.4. Приклади до розділу 7	73
7.5. Завдання до розділу 7 для самостійної роботи	77
8. МЕТОДИ ЗВОРОТНОГО Z-ПЕРЕТВОРЕННЯ.....	79
8.1. Метод розкладання на елементарні дроби	79
8.2. Метод лишків	81
8.3. Порівняння методів зворотного z-перетворення	82
8.4. Приклади до розділу 8	83
8.5. Завдання до розділу 8 для самостійної роботи	91
9. ПРОЕКТУВАННЯ ФІЛЬТРІВ ЗІ СКІНЧЕННОЮ ІМПУЛЬСНОЮ ХАРАКТЕРИСТИКОЮ МЕТОДОМ ЗВАЖУВАННЯ	92
9.1. Введення в цифрові фільтри	92
9.2. Обчислення коефіцієнтів методом зважування	95
9.3. Приклади до розділу 9	99
9.4. Завдання до розділу 9 для самостійної роботи	102

10. ПРОЕКТУВАННЯ ФІЛЬТРІВ ЗА ДОПОМОГОЮ ПАКЕТА	
ПРИКЛАДНИХ ПРОГРАМ MATLAB	104
10.1. Характеристика пакета програм FDATool	104
10.1.1. Задання специфікації і розрахунок фільтра	105
10.1.2. Перегляд характеристик фільтра	106
10.1.3. Експорт та імпорт опису фільтра	107
10.2. Характеристика пакета програм Simulink.....	110
10.2.1. Оглядач бібліотеки блоків.....	111
10.2.2. Створення моделі	112
10.2.3. Установка параметрів моделювання	114
10.2.4. Запуск процесу моделювання	116
10.3. Завдання до розділу 10 для самостійної роботи.....	117
11. ПЕРВИННА (ПОПЕРЕДНЯ) ОБРОБКА ЗОБРАЖЕНЬ.....	119
11.1. Дискретизація зображень	119
11.2. Робота із зображеннями в системі MATLAB	122
11.2.1. Характеристика пакета Image Processing Toolbox.....	122
11.2.2. Растрова графіка в системі MATLAB	122
11.2.3. Типи растрових зображень.....	123
11.2.4. Вивід на екран графічного об'єкту	124
11.2.5. Вивід зображення на екран	125
11.2.6. Вивід на екран декількох зображень в одному вікні	127
11.2.7. Читання зображення з файлу	129
11.2.8. Двомірне дискретне перетворення Фур'є	130
11.3. Приклади розрахунку двомірного ДПФ	133
11.4. Завдання до розділу 11 для самостійної роботи.....	142
12. ПОЕЛЕМЕНТНІ ПЕРЕТВОРЕННЯ ЗОБРАЖЕНЬ	144
12.1. Поліпшення візуальної якості зображень шляхом	
поелементного перетворення.....	144
12.2. Лінійне контрастування зображення.....	145
12.3. Соляризація зображення.....	146
12.4. Препарування зображення	147

12.5. Поелементна обробка зображень в MATLAB	150
12.5.1. Бінаризація відсіканням за порогом яскравості	150
12.5.2. Вирівнювання гістограми зображення.....	153
12.5.3. Зміна яскравості палітри	155
12.5.4. Соляризація.....	159
12.6. Приклади до розділу 12	160
12.7. Завдання до розділу 12 для самостійної роботи.....	165
 13. ЛІНІЙНА ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ	169
13.1. Основні поняття	169
13.2. Лінійні фільтри.....	170
13.2.1. Фільтри згладжування	171
13.2.2. Фільтри підвищення контрастності.....	172
13.2.3. Різницьові фільтри	173
13.3. Лінійна фільтрація в MATLAB.....	176
13.3.1. Усереднюючий фільтр.....	176
13.3.2. Фільтр Гауса	178
13.3.3. Фільтр підвищення різкості	180
13.3.4. Фільтр Собеля	182
13.3.5. Фільтр Прюїтта.....	183
13.3.6. Фільтр Лапласа.....	185
13.3.7. Фільтр Лапласа-Гауса	186
13.4. Приклади до розділу 13	188
13.5. Завдання до розділу 13 для самостійної роботи.....	198
 14. НЕЛІНІЙНА ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ	201
14.1. Основні поняття	201
14.2. Порогова фільтрація	202
14.3. Медіанна фільтрація	203
14.4. Фільтри екстремумів.....	203
14.5. Нелінійна фільтрація зображень в MATLAB	204
14.5.1. Медіанна фільтрація зображення	204
14.5.2. Рангова фільтрація	206

14.6. Приклади до розділу 14	208
14.7. Завдання до розділу 14 для самостійної роботи	211
15. СТИСНЕННЯ ЗОБРАЖЕНЬ.....	212
15.1. Дискретне косинусне перетворення	212
15.2. Застосування дискретного косинусного перетворення для стиснення зображень	216
15.2.1. Передумови появи стандарту стиснення зображень	216
15.2.2. Дискретне косинусне перетворення в стандарті JPEG	216
15.2.3. Кодування зображення	218
15.3. Основи вейвлет-перетворень	221
15.4. Двомірне вейвлет-перетворення	224
15.5. Приклади до розділу 15	226
15.6. Завдання до розділу 15 для самостійної роботи	234
СПИСОК ЛІТЕРАТУРИ	236

1. РЯДИ ФУР'Є БЕЗПЕРЕРВНИХ СИГНАЛІВ

1.1. Класифікація безперервних сигналів

Під сигналом зазвичай мають на увазі величину, що відображає стан фізичної системи. Саме тому природно розглядати сигнали як функції, що задані у фізичних координатах. Прикладом можуть служити одновимірні сигнали, що задаються функцією часу $x(t)$, двовірні сигнали $I(x, y)$, задані на площині тощо. Надалі ми розглядатимемо в основному сигнали як дійсні функції часу $x(t)$.

Аналогові сигнали описуються безперервними та кусково-безперервними функціями $x_a(t)$, причому як власне функція, так і її аргумент можуть приймати будь-які значення в межах деяких інтервалів:

$$t_{\min} \leq t \leq t_{\max}, \quad x_{a\min} \leq x_a \leq x_{a\max}.$$

Фінітний сигнал характеризується тим, що відрізняється від нуля лише на скінченному інтервалі T . Дуже важливим є клас сигналів з *фінітним спектром*. У таких сигналів спектральна функція (перетворення Фур'є) $X(f)$ перетворюється в нуль поза деяким скінченим інтервалом частот, наприклад, у діапазоні $[-f_d, f_d]$.

Задамо *випадковий сигнал* як вибірккову функцію деякого випадкового процесу, що визначається ансамблем реалізацій, тобто сукупністю реалізацій, що розглядаються разом з ймовірностями їх появи. Невипадкові сигнали називаються *детермінованими* та описуються відомими функціями, що задані на скінчених або нескінченних інтервалах.

Каузальний сигнал (від англ. causal – причинний) – це сигнал, який розглядається починаючи з деякого моменту та не існує до цього моменту, тобто $x(t) = 0$ при $t < 0$.

1.2. Ряди Фур'є безперервних сигналів

1.2.1. Розкладення в ряд Фур'є періодичних сигналів

Припустимо, що у нас є періодичний сигнал $x(t)$, що має період величиною $T = 2\pi/\omega_0$, таким чином частота цього сигналу буде $f = \omega_0/2\pi$, звідки отримуємо: $\omega_0 = 2\pi/T$.

Під розкладенням у ряд Фур'є мають на увазі суму гармонічних функцій або комплексних експонент з частотами, що створюють арифметичну прогресію.

Існування такого ряду можливе лише коли фрагмент сигналу відповідає наступним умовам Діріхле:

- ✓ не повинно бути розривів 2-го роду;
- ✓ кількість розривів 1-го роду повинно бути скінченним;
- ✓ число екстремумів повинно бути скінченним.

Існує декілька наступних форм запису ряду Фур'є:

- тригонометрична;
- дійсна;
- комплексна.

Розглянемо тригонометричну форму, яку можна записати наступним чином:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k \cos\left(k \frac{2\pi}{T} t\right) + \sum_{k=1}^{+\infty} b_k \sin\left(k \frac{2\pi}{T} t\right), \quad (1.1)$$

де a_0 , a_k , b_k – коефіцієнти тригонометричного ряду; $a_0/2$ – середнє значення сигналу на періоді.

Ці сталі визначаються за наступними формулами:

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} x(t) dt ;$$

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos\left(k \frac{2\pi}{T} t\right) dt;$$

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin\left(k \frac{2\pi}{T} t\right) dt.$$

Якщо функція $x(t)$ є парною, то коефіцієнти b_k вироджуються в 0, а якщо непарною – a_k стають нулями.

1.2.2. Розкладення в ряд Фур'є неперіодичних сигналів

Математичний апарат для розкладення періодичних функцій можна застосувати і до неперіодичних, побудувавши періодичне продовження цієї функції на всю числову вісь. При цьому необхідно, щоб отримана функція задовольняла вищенаведеним умовам теореми Діріхле. В такому разі в точках безперервності ця функція може бути подана рядом Фур'є, що збігатиметься з початковою майже всюди на скінченному проміжку.

Нехай функція $x(t)$ визначена на скінченному проміжку $[a; b]$ та $x(a) = x(b)$. Якщо $x(a) \neq x(b)$, то необхідно вибрати проміжок $[a; b)$, щоб продовження функції мало одне значення.

Розглянемо кусково-безперервний сигнал $f(x)$, який задано на інтервалі $[-L; L]$. Використаємо підстановку $x = \frac{Ly}{\pi}$, де $y \in [-\pi; \pi]$.

Перетворимо його таким чином:

$$F(y) = f\left(\frac{Ly}{\pi}\right).$$

Тоді сигнал $F(y)$ визначений та може бути проінтегрованим на

проміжку $[-\pi; \pi]$. Розкладення в ряд Фур'є для сигналу $F(y)$ має вигляд:

$$F(y) = f\left(\frac{Ly}{\pi}\right) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k \cos ky + \sum_{k=1}^{+\infty} b_k \sin ky.$$

Коефіцієнти Фур'є для даного сигналу визначаються формулами:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} F(y) dy; \quad a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} F(y) \cos ky dy; \quad b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} F(y) \sin ky dy.$$

Повертаючись до оригінальних змінних отримуємо наступні вирази для ряду Фур'є вихідного сигналу $f(x)$:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k \cos \frac{k\pi x}{L} + \sum_{k=1}^{+\infty} b_k \sin \frac{k\pi x}{L}, \quad (1.2)$$

$$\text{де } a_0 = \frac{1}{L} \int_{-L}^L f(x) dx; \quad a_k = \frac{1}{L} \int_{-L}^L f(x) \cos \frac{k\pi x}{L} dx; \quad b_k = \frac{1}{L} \int_{-L}^L f(x) \sin \frac{k\pi x}{L} dx.$$

Якщо сигнал $f(x)$ визначений на проміжку $[a; b]$, то його розкладення в ряд Фур'є визначене за виразом (1.2), де $L = \frac{b-a}{2}$, а коефіцієнти обчислюються наступним чином:

$$a_0 = \frac{1}{L} \int_a^b f(x) dx; \quad a_k = \frac{1}{L} \int_a^b f(x) \cos \frac{k\pi x}{L} dx; \quad b_k = \frac{1}{L} \int_a^b f(x) \sin \frac{k\pi x}{L} dx.$$

1.3. Приклади до розділу 1

Приклад №1.1. Знайти розкладення в ряд Фур'є функції:

$$f(x) = \begin{cases} A, & 0 \leq x \leq L; \\ 0, & L < x \leq 2L. \end{cases}$$

Розв'язок.

Визначимо коефіцієнти розкладення:

$$a_0 = \frac{1}{L} \int_a^b f(x) dx = \frac{1}{L} \int_0^L A dx = \frac{A}{L} x \Big|_0^L = A;$$

$$\begin{aligned} a_k &= \frac{1}{L} \int_a^b f(x) \cos \frac{k\pi x}{L} dx = \frac{1}{L} \int_0^L A \cos \frac{k\pi x}{L} dx = \frac{A}{L} \left[\frac{L}{k\pi} \sin \frac{k\pi x}{L} \right]_0^L = \\ &= \frac{A}{k\pi} (\sin k\pi - \sin 0) = 0; \end{aligned}$$

$$\begin{aligned} b_k &= \frac{1}{L} \int_a^b f(x) \sin \frac{k\pi x}{L} dx = \frac{1}{L} \int_0^L A \sin \frac{k\pi x}{L} dx = \frac{A}{L} \left[-\frac{L}{k\pi} \cos \frac{k\pi x}{L} \right]_0^L = \\ &= \frac{A}{k\pi} (-\cos k\pi + \cos 0) = \frac{A}{k\pi} [1 - (-1)^k] = \frac{A}{k\pi} [1 + (-1)^{k+1}]. \end{aligned}$$

Можна помітити, що для парних $k = 2n$, $n = 1, 2, 3, \dots$

$$b_{2n} = \frac{A}{2n\pi} [1 + (-1)^{2n+1}] = 0.$$

Для непарних $k = 2n - 1$, $n = 1, 2, 3, \dots$

$$b_{2n-1} = \frac{A}{(2n-1)\pi} \left[1 + (-1)^{2n} \right] = \frac{2A}{(2n-1)\pi}.$$

Отже, розкладення в ряд Фур'є отримувє вигляд:

$$f(x) = \frac{A}{2} + \frac{2A}{\pi} \sum_{n=1}^{+\infty} \frac{1}{2n-1} \sin\left(\frac{2n-1}{L} \pi x\right).$$

Приклад №1.2. Знайти розкладення в ряд Фур'є функції:

$$f(x) = \begin{cases} 0, & -1 \leq x \leq 0; \\ x, & 0 < x \leq 1. \end{cases}$$

Розв'язок.

Знайдемо півперіод $L = (1 - (-1)) / 2 = 1$. Визначимо коефіцієнти розкладення:

$$a_0 = \frac{1}{L} \int_a^b f(x) dx = \int_{-1}^1 f(x) dx = \int_0^1 x dx = \frac{x^2}{2} \Big|_0^1 = \frac{1}{2};$$

$$a_k = \frac{1}{L} \int_a^b f(x) \cos \frac{k\pi x}{L} dx = \int_0^1 x \cos k\pi x dx.$$

Для знаходження визначеного інтеграла від добутку функцій скористаємось методом інтегрування частинами:

$$\int_a^b U(x) dV(x) = U(x)V(x) \Big|_a^b - \int_a^b V(x) dU(x).$$

Тоді

$$\begin{aligned}
 a_k &= \int_0^1 x \cos k\pi x dx = \left| \begin{array}{l} U(x) = x; dU = dx; \\ dV(x) = \cos k\pi x dx; \\ V(x) = \frac{1}{k\pi} \sin k\pi x \end{array} \right| = \left[\frac{1}{k\pi} x \sin k\pi x \right]_0^1 - \\
 &- \frac{1}{k\pi} \int_0^1 \sin k\pi x dx = \frac{1}{\underbrace{k\pi}_{=0}} \sin k\pi + \frac{1}{k^2 \pi^2} \cos k\pi \Big|_0^1 = \frac{1}{k^2 \pi^2} \left[\underbrace{\cos k\pi}_{=(-1)^k} - \underbrace{\cos 0}_{=1} \right] = \\
 &= \frac{1}{k^2 \pi^2} [(-1)^k - 1] = \frac{(-1)^k - 1}{k^2 \pi^2};
 \end{aligned}$$

$$\begin{aligned}
 b_k &= \frac{1}{L} \int_a^b f(x) \sin \frac{k\pi x}{L} dx = \int_0^1 x \sin k\pi x dx = \left| \begin{array}{l} U(x) = x; dU = dx; \\ dV(x) = \sin k\pi x dx; \\ V(x) = -\frac{1}{k\pi} \cos k\pi x \end{array} \right| = \\
 &= \left[-\frac{1}{k\pi} x \cos k\pi x \right]_0^1 + \frac{1}{k\pi} \int_0^1 \cos k\pi x dx = -\frac{1}{k\pi} \left[\underbrace{\cos k\pi}_{=(-1)^k} - \underbrace{0 \cdot \cos 0}_{=0} \right] + \\
 &+ \frac{1}{k^2 \pi^2} \sin k\pi x \Big|_0^1 = \frac{-(-1)^k}{k\pi} + \frac{1}{k^2 \pi^2} \left[\underbrace{\sin k\pi}_{=0} - \underbrace{\sin 0}_{=0} \right] = \frac{(-1)^{k+1}}{k\pi}.
 \end{aligned}$$

Очевидно, розкладення в ряд Фур'є має наступний вигляд:

$$f(x) = \frac{1}{4} + \sum_{k=1}^{+\infty} \frac{(-1)^k - 1}{k^2 \pi^2} \cos k\pi x + \sum_{k=1}^{+\infty} \frac{(-1)^{k+1}}{k\pi} \sin k\pi x.$$

Приклад №1.3. Знайти розкладення в ряд Фур'є функції

$$f(x) = \begin{cases} x, & 0 \leq x \leq 1; \\ 1, & 1 < x \leq 2; \\ 3-x, & 2 < x \leq 3. \end{cases}$$

Розв'язок.

Знайдемо півперіод $L = (3-0)/2 = 3/2$. Обчислимо коефіцієнти розкладення:

$$\begin{aligned} a_0 &= \frac{1}{L} \int_a^b f(x) dx = \frac{2}{3} \int_0^3 f(x) dx = \frac{2}{3} \left(\int_0^1 x dx + \int_1^2 dx + \int_2^3 (3-x) dx \right) = \\ &= \frac{2}{3} \left[\frac{x^2}{2} \Big|_0^1 + x \Big|_1^2 + \left(3x - \frac{x^2}{2} \right) \Big|_2^3 \right] = \frac{2}{3} \left[\frac{1}{2} + 2 - 1 + 9 - \frac{9}{2} - 6 + \frac{4}{2} \right] = \frac{4}{3}; \\ a_k &= \frac{1}{L} \int_a^b f(x) \cos \frac{2k\pi x}{L} dx = \frac{2}{3} \int_0^3 f(x) \cos \frac{2k\pi x}{3} dx = \frac{2}{3} \underbrace{\left[\int_0^1 x \cos \frac{2k\pi x}{3} dx + \right.}_{= \frac{3}{2k\pi} x \sin \frac{2k\pi x}{3} \Big|_0^1 - \frac{3}{2k\pi} \int_0^1 \sin \frac{2k\pi x}{3} dx} \\ &\quad \left. + \int_1^2 \cos \frac{2k\pi x}{3} dx + \underbrace{\int_2^3 (3-x) \cos \frac{2k\pi x}{3} dx}_{= \frac{3}{2k\pi} (3-x) \sin \frac{2k\pi x}{3} \Big|_2^3 + \frac{3}{2k\pi} \int_2^3 \sin \frac{2k\pi x}{3} dx} \right] = \\ &= \frac{2}{3} \left\{ \left[\frac{3}{2k\pi} x \sin \frac{2k\pi x}{3} \Big|_0^1 - \frac{3}{2k\pi} \int_0^1 \sin \frac{2k\pi x}{3} dx \right] + \frac{3}{2k\pi} \sin \frac{2k\pi x}{3} \Big|_1^2 + \right. \\ &\quad \left. + \left[\frac{3}{2k\pi} (3-x) \sin \frac{2k\pi x}{3} \Big|_2^3 + \frac{3}{2k\pi} \int_2^3 \sin \frac{2k\pi x}{3} dx \right] \right\} = \end{aligned}$$

$$\begin{aligned}
&= \frac{2}{3} \left\{ \frac{3}{2k\pi} \left[\sin \frac{2k\pi}{3} - \underbrace{\sin 0}_{=0} \right] + \frac{9}{4k^2\pi^2} \cos \frac{2k\pi x}{3} \right|_0^1 + \frac{3}{2k\pi} \left[\sin \frac{4k\pi}{3} - \sin \frac{2k\pi}{3} \right] + \\
&+ \frac{3}{2k\pi} \left[\underbrace{(3-3) \sin \frac{6k\pi}{3}}_{=0} - (3-2) \sin \frac{4k\pi}{3} \right] - \frac{9}{4k^2\pi^2} \cos \frac{2k\pi x}{3} \Big|_2^3 \Big\} = \\
&= \frac{2}{3} \left\{ \frac{9}{4k^2\pi^2} \left(\cos \frac{2k\pi}{3} - \underbrace{\cos 0}_{=1} \right) - \frac{9}{4k^2\pi^2} \left(\underbrace{\cos \frac{6k\pi}{3}}_{=\cos 2k\pi=1} - \underbrace{\cos \frac{4k\pi}{3}}_{=\cos(2k\pi - \frac{2k\pi}{3}) = \cos \frac{2k\pi}{3}} \right) \right\} = \\
&= \frac{2}{3} \frac{9}{4k^2\pi^2} \left[\cos \frac{2k\pi}{3} - 1 + \cos \frac{2k\pi}{3} - 1 \right] = \frac{3}{k^2\pi^2} \left[\cos \frac{2k\pi}{3} - 1 \right], \quad k = 1, 2, 3, \dots
\end{aligned}$$

Оскільки функція парна на заданому інтервалі $[0,3]$, то $b_k = 0$.
Тоді розкладення в ряд Фур'є має вид:

$$f(x) = \frac{2}{3} + \frac{3}{\pi^2} \sum_{k=1}^{+\infty} \frac{1}{k^2} \left(\cos \frac{2k\pi}{3} - 1 \right) \cos k\pi x = \frac{2}{3} + \frac{3}{\pi^2} \sum_{k=1}^{+\infty} \frac{\cos \frac{2k\pi}{3} - 1}{k^2} \cos k\pi x.$$

1.4. Завдання до розділу 1 для самостійної роботи

Знайти ряд Фур'є сигналу $s(t) = \begin{cases} b, & a \leq t \leq b \\ t, & b < t \leq c \end{cases}$, що зображений на

рис. 1.1. Параметри сигналу необхідно вибрати згідно номера за журналом з табл. 1.1 (N – номер студента за журналом).

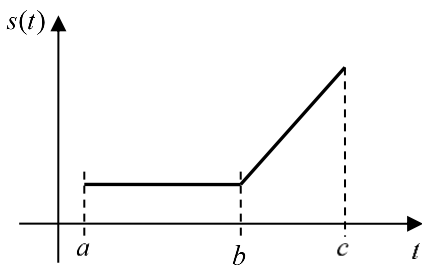


Рисунок 1.1 – Неперіодичний сигнал, заданий на інтервалі $[a; c]$

Таблиця 1.1 – Параметри сигналу

N	a	b	c	N	a	b	c	N	a	b	c
1	-9	-5	0	11	-7	-4	-1	21	-5	-3	0
2	-8	-4	1	12	-6	-3	0	22	-4	-2	1
3	-7	-3	2	13	-5	-2	1	23	-3	-1	2
4	-6	-2	3	14	-4	-1	2	24	-2	1	3
5	-5	-1	4	15	-3	1	3	25	-1	1	4
6	-4	1	5	16	-2	1	4	26	0	2	5
7	-3	1	6	17	-1	2	5	27	1	3	6
8	-2	2	7	18	0	3	6	28	2	4	7
9	-1	3	8	19	1	4	7	29	3	5	8
10	0	4	9	20	2	5	8	30	4	6	9

2. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗКЛАДЕННЯ В РЯД ФУР'Є ДЕЯКИХ СИГНАЛІВ

2.1. Ряд Фур'є послідовності прямокутних імпульсів

Розглянемо послідовність прямокутних імпульсів з амплітудою A , довжиною τ , періодом повтору T . Початком відріку часу задамо середину імпульсу (рис. 2.1)

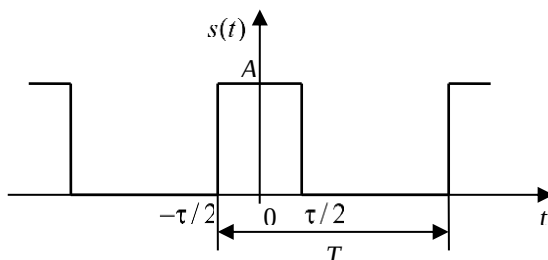


Рисунок 2.1 – Періодична послідовність прямокутних імпульсів

Даний сигнал є парною функцією, тому для його представлення зручніше використовувати тригонометричну форму ряду Фур'є – у ній будуть присутні тільки косинусні складові a_k :

$$a_k = \frac{2}{T} \int_{-\tau/2}^{\tau/2} A \cos\left(\frac{2\pi k}{T} t\right) dt = \frac{2A}{T} \sin \frac{\pi k \tau}{T}.$$

Із отриманого виразу видно, що довжина імпульсів та їх період використовуються у вигляді відношення. Відношення періоду до довжини імпульсів називають шпаруватістю послідовності імпульсів та позначають буквою q : $q = T/\tau$. Тоді зі врахуванням шпаруватості коефіцієнти a_k можна записати наступним чином:

$$a_k = \frac{2A}{\pi k} \sin \frac{\pi k}{q} = \frac{2A}{q} \frac{\sin \frac{\pi k}{q}}{\frac{\pi k}{q}}. \quad (2.1)$$

Оскільки при $x \rightarrow 0$ $\frac{\sin x}{x} \rightarrow 1$, то значення сталої ряду обчислюється за виразом:

$$\frac{a_0}{2} = \frac{A}{q} = \frac{A\tau}{T}.$$

Таким чином, ряд Фур'є послідовності прямокутних імпульсів має наступний вигляд:

$$s(t) = \frac{A}{q} + \sum_{k=1}^{+\infty} \frac{2A}{\pi k} \sin\left(\frac{\pi k}{q}\right) \cos\left(\frac{2\pi k}{T}t\right).$$

Амплітуди гармонічних додатків ряду залежать від номеру гармоніки за законом $\frac{\sin x}{x}$ (рис. 2.2). Графік функції $\frac{\sin x}{x}$ має пелюстковий характер. Необхідно підкреслити, що для графіків дискретних спектрів періодичних сигналів можливі два варіанти градування горизонтальної осі – в номерах гармонік та в частотах. На рис. 2.2 градування осі відповідає номерам гармонік, а частотні параметри спектру нанесені на графік за допомогою розмірних ліній.

Ширина пелюсток, виміряна в кількості гармонік, дорівнює шпаруватості послідовності (при $k = nq$ маємо $\sin(\pi k / q) = 0$, якщо $n \neq 0$). Звідси впливає важлива властивість спектру послідовності прямокутних імпульсів – у ньому відсутні (мають нульові амплітуди) гармоніки з номерами, що кратні шпаруватості.

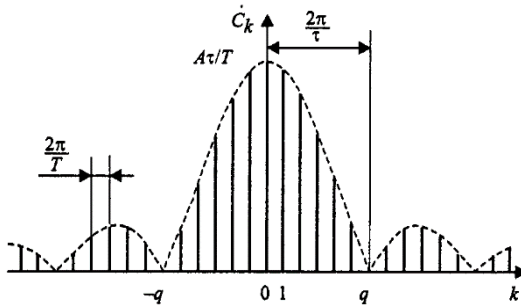


Рисунок 2.2 – Коефіцієнти ряду Фур'є для послідовності прямокутних імпульсів

Відстань за частотою між сусідніми гармоніками дорівнює частоті надходження імпульсів, тобто $2\pi/T$. Ширина пелюсток спектру, виміряна в одиницях частоти, дорівнює $2\pi/\tau$, тобто обернено пропорційна довжині імпульсів. Це результат загального закону: *чим коротший сигнал, тим ширший його спектр*.

Нижче наведена програма на MatLab, що показує, як заданий сигнал складається з окремих гармонік (результати роботи програми наведені на рис. 2.3):

```
%Програма №2.1
N=16; %число ненульових гармонік
t=-1:0.01:1; %вектор моментів часу
A=1; %амплітуда
T=1; %період
q=5; %шпаруватість
nh=1:N; %номера ненульових гармонік
harmonics=cos(2*pi*nh'*t/T); %гармоніки
Am=2*A*sin(pi*nh/q)/pi./nh; %амплітуди гармонік
%член ряду Фур'є
s1=harmonics.*repmat(Am',1,length(t));
s2=A/q+cumsum(s1); %часткові суми гармонік
%вивід графіків на екран
```

```

%червоний графік - поточна ненульова гармоніка
%синій графік - результат підсумовування гармонік
%(ряд Фур'є)
for k=1:N
    subplot(4,4,k),
    plot(t,s2(k,:),t,s1(k,:))
end

```

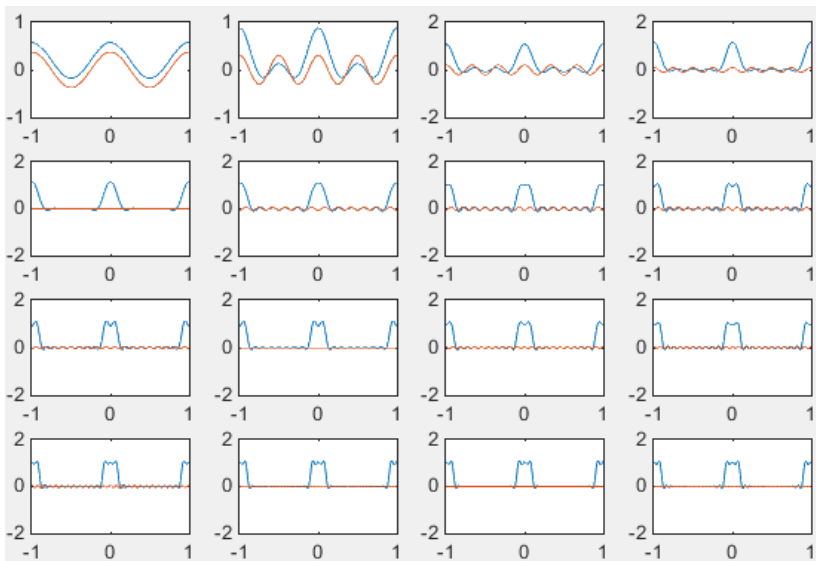


Рисунок 2.3 – Гармоніки для послідовності прямокутних імпульсів з $q = 5$

Взагалі послідовність прямокутних імпульсів погано підходить для зображення рядом Фур'є, оскільки містить стрибки, а сума будь-якого числа гармонічних складових з будь-якими амплітудами буде функцією безперервною. Саме тому поведінка ряду Фур'є в околі розривів несе особливий інтерес. На графіках добре видно, що в околі точки розриву підсумовування ряду Фур'є дає похилий проміжок, причому крутизна нахилу зростає з ростом числа підсумованих гармо-

нік.

На прилеглих до розриву проміжках сума ряду Фур'є дає помітні пульсації, причому на графіках помітно, що амплітуда цих пульсацій не зменшується з ростом числа підсумованих гармонік. Пульсації лише стискаються по горизонталі, наближаючись до точки розриву. Це явище з розривами першого роду (стрибками) для рядів Фур'є носить назву *ефекту Гібса*. Можна показати, що амплітуда першого найбільшого викиду складає 9 % від величини стрибка.

2.2. Меандр

Важливим окремим випадком попереднього сигналу є меандр – послідовність прямокутних імпульсів зі шпаруватістю, що дорівнює двом, при тому що довжина імпульсів та проміжки між ними стають рівними (рис. 2.4).

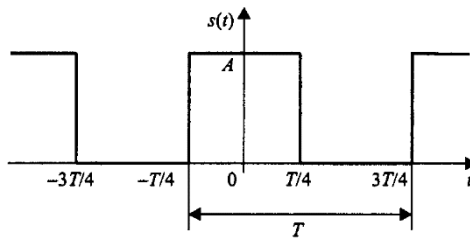


Рисунок 2.4 – Меандр

Підставивши $q = 2$ у вираз (2.1), отримаємо:

$$a_k = A \frac{\sin(\pi k / 2)}{\pi k / 2} = \begin{cases} A, & k = 0 \\ 0, & k = 2m, m \neq 0 \\ \frac{2A}{\pi k}, & k = 4m + 1 \\ -\frac{2A}{\pi k}, & k = 4m - 1 \end{cases}, \quad (2.2)$$

де t – будь-яке ціле.

Таким чином, у спектрі меандру присутні лише непарні гармоніки. Це узгоджується зі загальним правилом, що наведене вище. Представлення меандру в вигляді ряду Фур'є зі врахуванням виразу (2.2) може бути записане наступним чином:

$$s(t) = \frac{A}{2} + \frac{2A}{\pi} \left(\cos\left(\frac{2\pi}{T}t\right) - \frac{1}{3} \cos\left(3\frac{2\pi}{T}t\right) + \frac{1}{5} \cos\left(5\frac{2\pi}{T}t\right) - \dots \right).$$

Гармонічні складові, з яких складається меандр, мають амплітуди, що обернено пропорційні до номерів гармонік та знаків, що чергуються.

Нижче наводиться програма, що демонструє, як меандр складається з окремих гармонік (результати роботи програми наведені на рис. 2.5):

```
%Програма №2.2
N=16; %число ненульових гармонік
t=-1:0.01:1; %вектор моментів часу
A=1; %амплітуда
T=1; %період
nh=(1:N)*2-1; %номера ненульових гармонік
harmonics=cos(2*pi*nh*t/T); %гармоніки
Am=2*A/pi./nh; %амплітуди гармонік
Am(2:2:end)=-Am(2:2:end); %чергування знаків
%член ряду Фур'є
s1=harmonics.*repmat(Am',1,length(t));
s2=A/2+cumsum(s1); %часткові суми гармонік
%вивід графіків на екран
%червоний графік - поточна ненульова гармоніка
%синій графік - результат підсумовування гармонік
%(ряд Фур'є)
for k=1:N
    subplot(4,4,k),      plot(t,s2(k,:),t,s1(k,:))
end
```

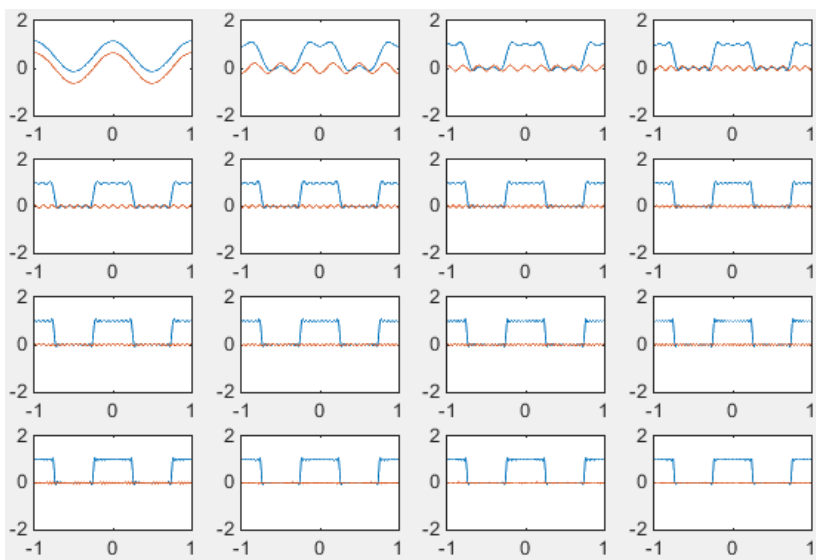


Рисунок 2.5 – Гармоніки для меандру

2.3. Пілкоподібний сигнал

Пілкоподібний сигнал (рис. 2.6) в рамках періоду описується лінійною функцією:

$$s(t) = \frac{2A}{T}(t - kT), \quad (k - 1/2)T < t \leq (k + 1/2)T.$$

Даний сигнал є непарною функцією, тому його ряд Фур'є в тригонометричній формі (1.1) містить лише тільки косинусні складові:

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} \frac{2A}{T} t \sin\left(\frac{2\pi k}{T} t\right) dt = -\frac{2A}{\pi k} (-1)^k.$$

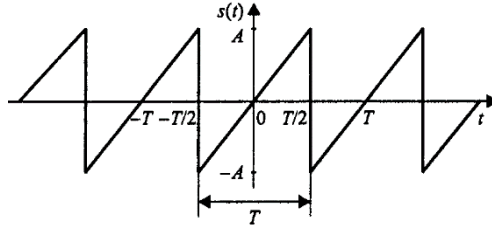


Рисунок 2.6 – Пилкоподібний сигнал

Ряд Фур'є для пилкоподібного сигналу виглядатиме наступним чином:

$$s(t) = \frac{2A}{\pi} \left(\sin\left(\frac{2\pi}{T}t\right) - \frac{1}{2} \sin\left(2 \frac{2\pi}{T}t\right) + \frac{1}{3} \sin\left(3 \frac{2\pi}{T}t\right) - \frac{1}{4} \sin\left(4 \frac{2\pi}{T}t\right) + \dots \right).$$

У розглянутих вище спектрів прямокутного та пилкоподібного періодичних сигналів є одна загальна риса – амплітуди гармонік з ростом їх номерів зменшуються пропорційно до k .

Нижче наводиться програма, що показує, як пилкоподібний сигнал складається з окремих гармонік (результати роботи програми наведені на рис. 2.7):

```
%Програма №2.3
N=16; %число ненульових гармонік
t=-1:0.01:1; %вектор моментів часу
A=1; %амплітуда
T=1; %період
nh=1:N; %номера ненульових гармонік
harmonics=sin(2*pi*nh'*t/T); %гармоніки
Am=2*A/pi./nh; %амплітуди гармонік
Am(2:2:end)=-Am(2:2:end); %чергування знаків
%член ряду Фур'є
s1=harmonics.*repmat(Am',1,length(t));
s2=cumsum(s1); %часткові суми гармонік
```

```

%вивід графіків на екран
%червоний графік - поточна ненульова гармоніка
%синій графік - результат підсумовування гармонік
%(ряд Фур'є)
for k=1:N
    subplot(4,4,k), plot(t,s2(k,:),t,s1(k,:))
end

```

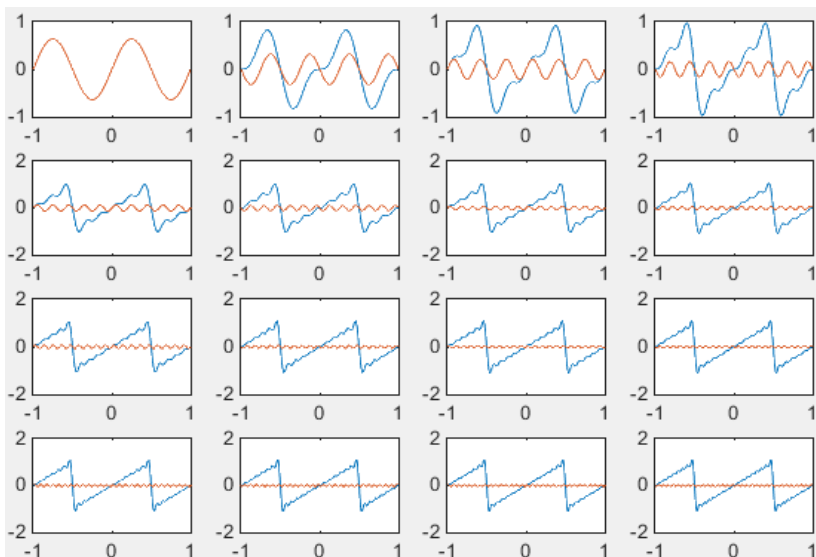


Рисунок 2.7 – Гармоніки для пілкоподібного періодичного сигналу

2.4. Послідовність трикутних імпульсів

Розглянемо періодичну послідовність трикутних імпульсів, що мають симетричну форму (рис. 2.8):

$$s(t) = a \left(1 - 4 \frac{|t - kT|}{T} \right), \quad (k - 1/2)T \leq t < (k + 1/2)T.$$

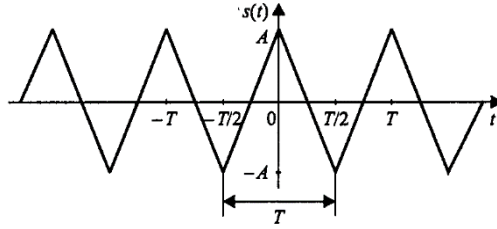


Рисунок 2.8 – Послідовність трикутних імпульсів

Обчислимо коефіцієнти ряду Фур'є. При цьому, сигнал є парною функцією, тому у тригонометричній формі ряду (1.1) будуть присутні лише косинусні складові:

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} A \left(1 - 4 \frac{|t|}{T} \right) \cos \left(\frac{2\pi k}{T} t \right) dt = \frac{4A}{(\pi k)^2} [1 - (-1)^k] =$$

$$= \begin{cases} 0, & k = 2m; \\ \frac{8A}{(\pi k)^2}, & k = 2m + 1. \end{cases}$$

Як і у випадку меандру, тут присутні лише непарні гармоніки. Власне ряд Фур'є має наступний вигляд:

$$s(t) = \frac{8A}{\pi^2} \left(\cos \left(\frac{2\pi}{T} t \right) + \frac{1}{3^2} \cos \left(3 \frac{2\pi}{T} t \right) + \frac{1}{5^2} \cos \left(5 \frac{2\pi}{T} t \right) + \dots \right).$$

Таким чином, на відміну від послідовностей прямокутних та пилкоподібних імпульсів, для трикутного періодичного сигналу амплітуди гармонік якого спадають пропорційно другій степені номерів гармонік k . Це прояв загального правила – *швидкість спадання спектру залежить від степені гладкості сигналу*.

Прямокутний та пилкоподібний сигнал мають розриви першого

роду (стрибки), и в їх спектрах присутній множник $1/k$. Трикутний сигнал є безперервною функцією, але її перша похідна містить розриви, та амплітуди гармонік його ряду Фур'є містять множник $1/k^2$.

Екстраполювавши цю залежність, отримаємо наступне правило: якщо N – номер останньої безперервної похідної, то спектр цього сигналу буде спадати зі швидкістю $1/k^{N-2}$.

Граничним випадком буде гармонічний сигнал, диференціювати який без втрати безперервності можна нескінченно. Згідно до загального правила, це дасть нескінченну швидкість спадання спектру, що повністю відповідає дійсності (ряд Фур'є для гармонічного сигналу містить лише одну гармоніку).

Нижче наводиться програма, що демонструє як послідовність трикутних імпульсів складається з окремих гармонік (результати роботи програми наведені на рис. 2.9):

```
%Програма №2.4
N=8; %число ненульових гармонік
t=-1:0.01:1; %вектор моментів часу
A=1; %амплітуда
T=1; %період
nh=(1:N)*2-1; %номера ненульових гармонік
harmonics=cos(2*pi*nh'*t/T); %гармоніки
Am=8*A/(pi^2)./nh./nh; %амплітуди гармонік
%член ряду Фур'є
s1=harmonics.*repmat(Am',1,length(t));
s2=cumsum(s1); %часткові суми гармонік
%вивід графіків на екран
%червоний графік - поточна ненульова гармоніка
%синій графік - результат підсумовування гармонік
%(ряд Фур'є)
for k=1:N
    subplot(4,2,k),plot(t,s2(k,:),t,s1(k,:))
end
```

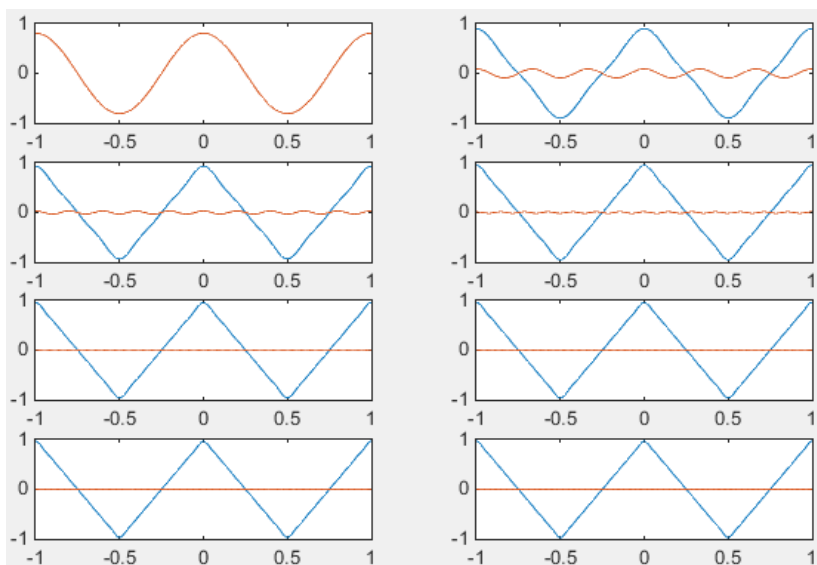


Рисунок 2.9 – Гармоніки для послідовності трикутних імпульсів

2.5. Завдання до розділу 2 для самостійної роботи

Використовуючи розрахунки індивідуального завдання підрозділу 1.4, розробити програму, що демонструє, як зображений на рис. 1.1 сигнал складається з окремих гармонік.

3. ПРЯМЕ ТА ЗВОРОТНЕ ДИСКРЕТНЕ ПЕРЕТВОРЕННЯ ФУР'Є

3.1. Дискретне перетворення Фур'є

При обробці реальних сигналів ми маємо справу з послідовністю скінченного числа відліків. Для зображення сигналу у вигляді суми синусоїд, він має бути періодичним, тобто нескінченно повторюваним в часі. Цю проблему можна розв'язати, якщо помістити нескінченне число однакових груп відліків до і після оброблюваної групи, утворюючи таким чином математичну, але не реальну періодичність.

Вираз для прямого дискретного перетворення Фур'є (ДПФ) має наступний вигляд:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}, \quad (3.1)$$

де множник $e^{-j \frac{2\pi}{N} kn}$ називається множителем повороту і позначається W_N . Величина $X[k]$ є частотним виходом ДПФ в k -й точці спектру, де k знаходиться в діапазоні від 0 до $N-1$. Значення $x[n]$ є n -м відліком в часовій області, де n також знаходиться в діапазоні від 0 до $N-1$.

Вираз для зворотного дискретного перетворення Фур'є (ЗДПФ) має наступну форму:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} kn}. \quad (3.2)$$

Підстановка множників повороту W_N в (3.1) і (3.2) призводить до визначення ДПФ та ЗДПФ відповідно у такій формі:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} ;$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} .$$

Отримані вирази дозволяють перетворювати часову область дискретного періодичного сигналу на частотну і назад.

Як видно, спектр періодичного аналогового сигналу має дискретний характер, а спектр дискретного сигналу, визначеного за допомогою послідовності відліків, є дискретним і періодичним. Іншими словами, коли дискретизації піддається спектр $X[k]$, це призводить до періодичного повторення сигналу $x[n]$.

3.2. Властивості дискретного перетворення Фур'є

1) *Лінійність.*

Якщо $X[k]$ і $Y[k]$ – це ДПФ послідовностей $x[n]$ і $y[n]$ відповідно, то ДПФ послідовності $z[n] = ax[n] + by[n]$, де a , b – довільні константи, дорівнює

$$Z[k] = aX[k] + bY[k].$$

Доведення:

$$z[n] = ax[n] + by[n] = a \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} + b \frac{1}{N} \sum_{k=0}^{N-1} Y[k] W_N^{-kn} =$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} (aX[k] + bY[k]) W_N^{-kn} = \frac{1}{N} \sum_{k=0}^{N-1} Z[k] W_N^{-kn} .$$

2) Зсув.

Нехай $X[k]$ – ДПФ послідовності $x[n]$, а послідовність $y[n]$ виходить шляхом зсуву (в разі скінченної послідовності – кругового зсуву) на n_0 відліків. Тоді ДПФ послідовності $y[n]$ дорівнює

$$Y[k] = X[k]W_N^{kn_0}. \quad (3.3)$$

Доведення:

$$\begin{aligned} Y[k] &= \sum_{n=0}^{N-1} y[n]W_N^{kn} = \sum_{n=0}^{N-1} x[n-n_0]W_N^{kn} = \sum_{n=0}^{N-1} x[n-n_0]W_N^{k(n-n_0+n_0)} = \\ &= \sum_{n=0}^{N-1} x[n-n_0]W_N^{k(n-n_0)}W_N^{kn_0} = X[k]W_N^{kn_0}. \end{aligned}$$

Аналогічний результат справедливий для зсуву коефіцієнтів ДПФ. Якщо $X[k]$ і $Y[k]$ – це ДПФ послідовностей $x[n]$ і $y[n]$ відповідно та $Y[k] = X[k-k_0]$, то

$$y[n] = x[n]W_N^{-n k_0}.$$

3) Властивість симетрії.

Якщо послідовність $x[n]$ є дійсною, то її ДПФ задовольняє таким умовам симетрії:

$$\operatorname{Re}(X[k]) = \operatorname{Re}(X[N-k]);$$

$$\operatorname{Im}(X[k]) = -\operatorname{Im}(X[N-k]);$$

$$|X[k]| = |X[N-k]|;$$

$$\arg X[k] = -\arg X[N-k].$$

ДПФ симетричної послідовності $x[n] = x[N - n]$ є дійсним.

4) Кругова згортка.

Нехай $x[n]$ і $y[n]$ мають ДПФ $X[k]$ і $Y[k]$ відповідно. За означенням кругова згортки двох послідовностей дорівнює

$$g[n] = \sum_{m=0}^{N-1} x[m]y[n-m], \quad n = \overline{0, N-1}.$$

Тоді ДПФ послідовності $g[n]$ дорівнює

$$G[k] = X[k]Y[k].$$

Доведення:

$$\begin{aligned} G[k] &= \sum_{n=0}^{N-1} g[n]W_N^{nk} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x[m]y[n-m]W_N^{nk} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x[m]y[n-m]W_N^{(n-m)k} = \\ &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x[m]W_N^{mk} y[n-m]W_N^{(n-m)k} = \sum_{m=0}^{N-1} x[m]W_N^{mk} \sum_{n=0}^{N-1} y[n-m]W_N^{(n-m)k} = X[k]Y[k]. \end{aligned}$$

Якщо $x[n]$ і $y[n]$ мають ДПФ $X[k]$ і $Y[k]$ відповідно, то ДПФ послідовності $g[n] = x[n]y[n]$ з точністю до постійного множника дорівнює круговій згортці $X[k]$ і $Y[k]$:

$$G[k] = \frac{1}{N} \sum_{m=0}^{N-1} X[m]Y[k-m], \quad k = \overline{0, N-1}.$$

Розглянемо декілька прикладів програм для ілюстрації властивостей ДПФ (кругова згортка більш детально розглянута у розділі 5).

Нижче наведений приклад використання функції **fft()** для розрахунку спектру гармонічного сигналу (результати роботи програми

зображені на рис. 3.1):

```
%Програма №3.1
%Спектр гармонічного сигналу
N=16; %кількість відліків дискретного сигналу
T=1; %період сигналу
f=2; %частота сигналу
A=1; %амплітуда сигналу
td=0:T/N:T-(T/N); %відліки дискретного сигналу
%відліки для емуляції аналогового сигналу
ta=0:0.01:T;
xd=A*cos(2*pi*f*td); %дискретний сигнал
xa=A*cos(2*pi*f*ta); %емуляція аналогового сигналу
y=fft(xd); %дискретне перетворення Фур'є
%графік дискретного сигналу
subplot(3,1,1);
stem(td,xd);
%графік аналогового сигналу
hold on;
plot(ta,xa,'--');
hold off;
%номера відліків спектру дискретного сигналу
k=0:length(td)-1;
%графік амплітудного спектру дискретного сигналу
subplot(3,1,2);
stem(k,abs(y));
%графік фазового спектру дискретного сигналу
subplot(3,1,3);
stem(k,angle(y));
```

На рис. 3.1 можна побачити графік дискретної функції $u[nT_c] = \cos(2\pi f n T_c)$, де $T_c = 1/16\text{с}$ – час дискретизації; $f = 2\text{Гц}$ – частота гармонічного коливання, а також графіки її амплітудного і фазового спектру. На графіку амплітудного спектру можна побачити тільки одну спектральну лінію, яка розташована на частоті 2Гц і відповідає початкової функції косинуса. При цьому у фазовому спектрі на частоті

2Гц спостерігається спектральна лінія нульової амплітуди. Якщо початкову функцію змінити на синус з такими ж значеннями амплітуди та частоти, то амплітудний спектр не зміниться, а зміниться лише фазовий спектр (на частоті 2Гц з'явиться спектральна лінія з ненульовою амплітудою).

З наведених на рис. 3.1. графіків легко побачити, що спектр дискретного сигналу, визначеного за допомогою послідовності відліків, є дискретним і періодичним.

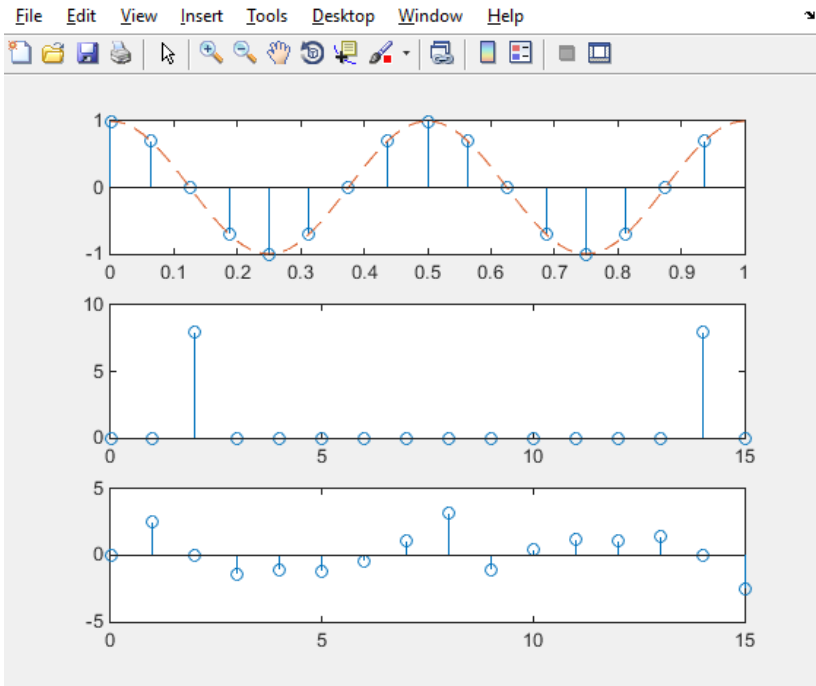


Рисунок 3.1 – Графіки гармонійного сигналу та його амплітудного і фазового спектрів

Тепер розглянемо приклад, який демонструє таку властивість ДПФ як лінійність (результати роботи програми зображені на рис. 3.2):

```

%Програма №3.2
N=16; %кількість відліків дискретного сигналу
T=1; % період сигналу
f1=2; %частота 1-го гармонічного коливання
f2=3; %частота 2-го гармонічного коливання
A1=1; %амплітуда 1-го гармонічного коливання
A2=0.5; %амплітуда 2-го гармонічного коливання
td=0:T/N:T-(T/N); %відліки дискретного сигналу
%відліки для емуляції аналогового сигналу
ta=0:0.01:T;
%дискретний сигнал
xd=A1*cos(2*pi*f1*td)+A2*sin(2*pi*f2*td);
%емуляція аналогового сигналу
xa=A1*cos(2*pi*f1*ta)+A2*sin(2*pi*f2*ta);
y=fft(xd); %ДПФ
%графік дискретного сигналу
subplot(3,1,1);
stem(td,xd);
%графік аналогового сигналу
hold on;
plot(ta,xa,'--');
hold off;
%номера відліків спектру дискретного сигналу
k=0:length(td)-1;
%графік амплітудного спектру дискретного сигналу
subplot(3,1,2);
stem(k,abs(y));
%графік фазового спектру дискретного сигналу
subplot(3,1,3);
stem(td,angle(y));

```

Верхній графік на рис. 3.2 відповідає дискретній функції, яка є лінійною комбінацією двох гармонічних коливань (косинусу з частотою $f_1 = 2\text{Гц}$ та амплітудою 1 та синусу з частотою $f_2 = 3\text{Гц}$ та ампліту-

дою 0.5): $y[nT_c] = \cos(2\pi f_1 nT_c) + 0.5\sin(2\pi f_2 nT_c)$, де $T_c = 1/16\text{с}$ – час дискретизації. Тобто дискретна функція $y[nT_c]$ є лінійною комбінацією двох функцій, одна з котрих розглянута у попередньому прикладі.

Два нижніх графіка на рис. 3.2 відповідають амплітудному і фазовому спектрам функції $y[nT_c]$.

Якщо порівняти спектральні характеристики сигналів на рис. 3.1 та 3.2, то легко помітити, що до спектральній лінії на частоті 2Гц, що відповідає косинусу, додана ще одна спектральна лінія на частоті 3Гц, яка відповідає синусу. Тобто в частотній області спектр суми гармонічних коливань відповідає сумі спектрів цих функцій, що демонструє лінійність ДПФ.

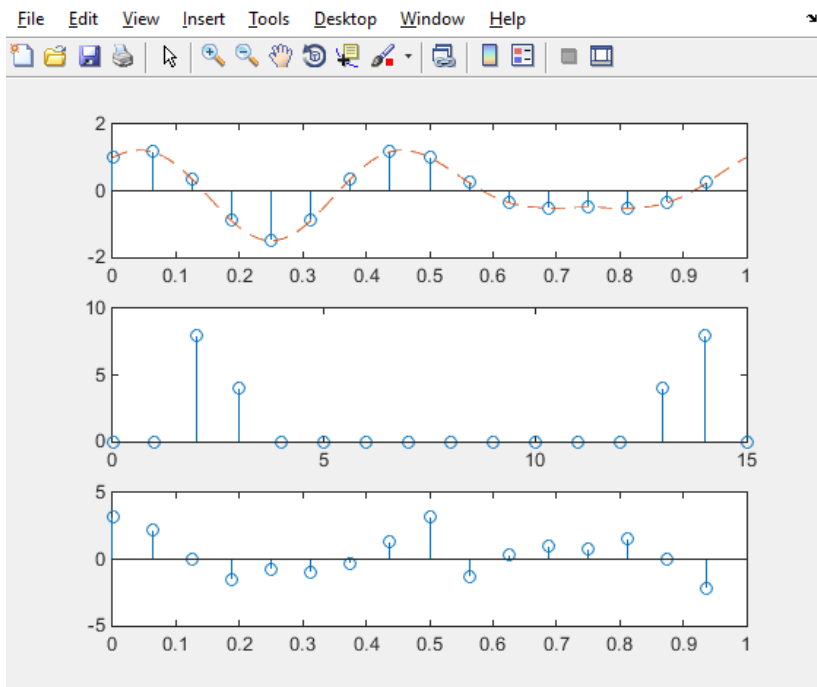


Рисунок 3.2 – Графіки суми гармонійних коливань та її

амплітудного і фазового спектрів

Нижче наведений приклад демонструє таку властивість ДПФ як зсув (результати роботи програми зображені на рис. 3.3):

```
%Програма №3.3
N=16; %кількість відліків дискретного сигналу
T=1; % період сигналу
f1=2; %частота 1-го гармонічного коливання
f2=3; %частота 2-го гармонічного коливання
A1=1; %амплітуда 1-го гармонічного коливання
A2=0.5; %амплітуда 2-го гармонічного коливання
t=0:T/N:T-(T/N); %відліки дискретного сигналу
%дискретний сигнал
x=A1*cos(2*pi*f1*t)+A2*sin(2*pi*f2*t);
y=fft(x); %ДПФ
%графік дискретного сигналу
subplot(3,2,1);
stem(t,x);
%номера відліків спектру дискретного сигналу
k=0:length(t)-1;
%графік амплітудного спектру початкового сигналу
subplot(3,2,3); s
tem(k,abs(y));
%графік фазового спектра початкового сигналу
subplot(3,2,5);
stem(t,angle(y));
n0=2; %кількість відліків зсуву
%зсув сигналу за виразом (3.3) на два відліки
y1=y.*exp(-2*pi*i/N.*k*n0);
x1=ifft(y1); %ЗДПФ
%графік дискретного сигналу після зсуву
subplot(3,2,2);
stem(t,x1);
%графік амплітудного спектру сигналу після зсуву
subplot(3,2,4);
stem(k,abs(y1));
```

```
%графік фазового спектру сигналу після зсуву
subplot(3,2,6);
stem(t,angle(y1));
```

У наведеній програмі як початковий сигнал використовується сума гармонічних коливань з попереднього прикладу. Потім цей сигнал зсувається на два відліки з використанням властивості ДПФ за виразом (3.3). На рис. 3.3 у лівій частині зображені графіки початкового сигналу, та його амплітудного та фазового спектрів, а в правій частині – графіки сигналу після зсуву, та його амплітудного та фазового спектрів.

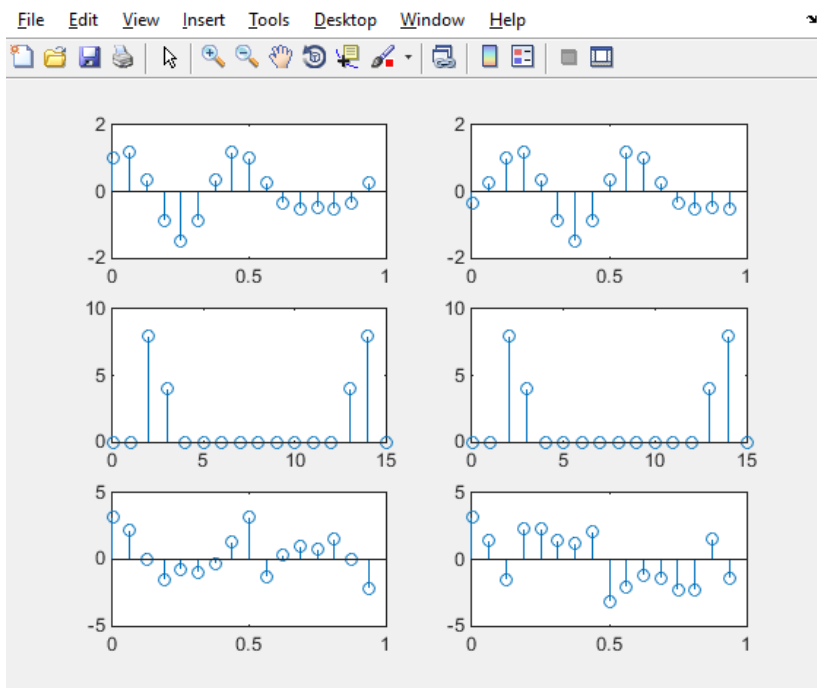


Рисунок 3.3 – Результати роботи програми №3.3

Порівняння спектрів початкового сигналу та сигналу після зсуву показують, що амплітудний спектр не змінився, а змінився лише фазовий спектр, тому що при зсуві змінюється лише фаза сигналу.

3.3. Приклади до розділу 3

Приклад №3.1. Розрахуємо спектр дискретного сигналу $x[n] = (a; 0; 0; 0; 0)$.

Розв'язок.

Скористаємося виразом (3.4) для обчислення спектру дискретного неперіодичного сигналу, в який підставимо значення $x[n]$ заданого сигналу

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn} = a e^{-j \frac{2\pi}{N} k \cdot 0} = a.$$

Приклад №3.2. Дійсний сигнал $x[n]$ має наступні значення відліків: $x[n] = (1; -1)$. Цей сигнал зображений на рис. 3.4. Необхідно розрахувати перші дві гармоніки спектру цього сигналу.

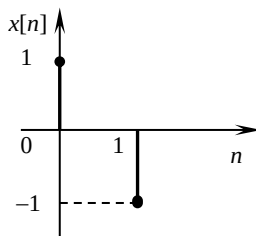


Рисунок 3.4 – Графік вихідного сигналу

Розв'язок.

Є два ($N = 2$) відліки. Щоб знайти комплексні величини, необхідно обчислити значення рівняння ДПФ при $k = 0, 1$.

При $k = 0$ отримаємо:

$$X[0] = \sum_{n=0}^1 x[n] e^{-j\frac{2\pi}{2}0n} = \sum_{n=0}^1 x[n] e^{-j0} = \sum_{n=0}^1 x[n];$$

$$X[0] = x[0] + x[1] = 1 - 1 = 0.$$

При $k = 1$ отримаємо:

$$X[1] = \sum_{n=0}^1 x[n] e^{-j\frac{2\pi}{2}1n} = \sum_{n=0}^1 x[n] e^{-j\pi n} = x[0]e^{-j\pi 0} + x[1]e^{-j\pi 1},$$

$$X[1] = 1 e^{-j0} + (-1) e^{-j\pi} = 1 + (-1)(\cos(\pi) - j \sin(\pi)) = 1 + 1 = 2.$$

Отже, перші дві гармоніки спектру дорівнюють $X[k] = (0; 2)$.

Приклад №3.3. Розрахуємо ДПФ дискретного періодичного сигналу, що заданий трьома відліками $x[n] = (0; 1; 2)$.

Розв'язок.

$$X[0] = x[0]e^{-j\frac{2\pi}{3}0 \cdot 0} + x[1]e^{-j\frac{2\pi}{3}1 \cdot 0} + x[2]e^{-j\frac{2\pi}{3}2 \cdot 0} = 0 + 1 + 2 = 3;$$

$$X[1] = x[0]e^{-j\frac{2\pi}{3}0 \cdot 1} + x[1]e^{-j\frac{2\pi}{3}1 \cdot 1} + x[2]e^{-j\frac{2\pi}{3}2 \cdot 1} = 0 + 1e^{-j\frac{2\pi}{3}} + 2e^{-j\frac{4\pi}{3}};$$

$$X[2] = x[0]e^{-j\frac{2\pi}{3}0 \cdot 2} + x[1]e^{-j\frac{2\pi}{3}1 \cdot 2} + x[2]e^{-j\frac{2\pi}{3}2 \cdot 2} = 0e^{-j0} + 1e^{-j\frac{4\pi}{3}} + 2e^{-j\frac{8\pi}{3}}.$$

$$\text{Оскільки } e^{-j\frac{2\pi}{3}} = e^{-j\frac{8\pi}{3}} = \frac{-1}{2} - j\frac{\sqrt{3}}{2} \text{ та } e^{-j\frac{4\pi}{3}} = \frac{-1}{2} + j\frac{\sqrt{3}}{2}, \text{ то}$$

$$X[1] = \frac{-3 + j\sqrt{3}}{2} \text{ та } X[2] = \frac{-3 - j\sqrt{3}}{2}.$$

Графіки заданого дискретного періодичного сигналу $x[n]$ і розрахованого дискретного періодичного спектру амплітуд $X[k]$ приведені на рис. 3.5 (амплітуда A комплексного числа $X = a + jb$ обчислюється за виразом $A = \sqrt{a^2 + b^2}$).

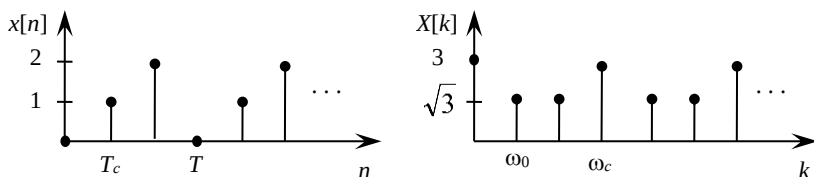


Рисунок 3.5 – Графік дискретного сигналу $x[n]$
та його спектр амплітуд $X[k]$

Приклад №3.4. Розрахуємо значення дискретного сигналу $x[n]$, при заданому ДПФ у наступній формі $X[k] = (0;1;0;1)$.

Розв'язок.

$$x[0] = \frac{1}{4}(X[0]e^{j0^\circ} + X[1]e^{j0^\circ} + X[2]e^{j0^\circ} + X[3]e^{j0^\circ}) = 0,5;$$

$$x[1] = \frac{1}{4}(X[0]e^{j0^\circ} + X[1]e^{j\frac{\pi}{2}} + X[2]e^{j\pi} + X[3]e^{j\frac{3\pi}{2}}) = \frac{1}{4}(0 + j1 + 0 - j1) = 0;$$

$$x[2] = \frac{1}{4}(X[0]e^{j0^\circ} + X[1]e^{j\pi} + X[2]e^{j2\pi} + X[3]e^{j3\pi}) = \frac{1}{4}(0 - 1 + 0 - 1) = -0,5;$$

$$x[3] = \frac{1}{4}(X[0]e^{j0^\circ} + X[1]e^{j\frac{3\pi}{2}} + X[2]e^{j3\pi} + X[3]e^{j\frac{9\pi}{2}}) = \frac{1}{4}(0 - j + 0 + j) = 0.$$

Графік послідовності $x[n] = (0,5; 0; -0,5; 0)$ наведено на рис. 3.6.

Графік показує, що сигнал $x[n]$ дискретний та періодичний.

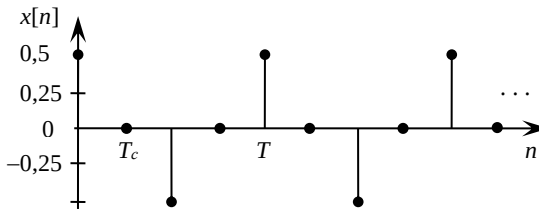


Рисунок 3.6 – Графік сигналу, відновленого за формулою ЗДПФ

3.4. Завдання до розділу 3 для самостійної роботи

1) Розрахувати ДПФ дискретного сигналу, заданого своїми відліками. Побудувати графіки сигналу і амплітудного спектру. Вхідні дані див. у табл. 3.1 (N – номер студента за журналом).

Таблиця 3.1 – Варіанти дискретних сигналів

N	$x[n]$	N	$x[n]$	N	$x[n]$
1	$(-2; -1; 1; 2)$	11	$(0; 2; 2; 0)$	21	$(1; -2; 3; -4)$
2	$(-2; 1; -1; 2)$	12	$(2; 0; 3; 0)$	22	$(1; -2; -3; 4)$
3	$(-2; 1; 2; -1)$	13	$(2; 0; 3; 0)$	23	$(-3; -1; 0; 2)$
4	$(-2; -1; 2; 1)$	14	$(-2; 0; 3; 0)$	24	$(-3; 0; 1; 3)$
5	$(1; 2; 3; 4)$	15	$(2; 0; -3; 0)$	25	$(-3; 0; 0; 3)$
6	$(1; -2; 3; -4)$	16	$(-2; -3; 1; 2)$	26	$(0; 2; 2; 0)$
7	$(1; -2; -3; 4)$	17	$(-2; 1; -1; 4)$	27	$(2; 0; 3; 0)$
8	$(-3; -1; 0; 2)$	18	$(-4; 1; 2; -1)$	28	$(2; 0; 3; 0)$
9	$(-3; 0; 1; 3)$	19	$(-2; -1; 5; 1)$	29	$(-2; 0; 3; 0)$
10	$(-3; 0; 0; 3)$	20	$(1; 2; 3; 4)$	30	$(2; 0; -3; 0)$

2) Розрахувати значення дискретного сигналу на базі наявного

ДПФ. Побудувати графіки амплітудного спектру і сигналу. Вхідні дані див. у табл. 3.2 (N – номер студента за журналом).

Таблиця 3.2 – Варіанти спектрів

N	$X[k]$	N	$X[k]$	N	$X[k]$
1	(-2;-1;1;2)	11	(0;2;2;0)	21	(1;-2;3;-4)
2	(-2;1;-1;2)	12	(2;0;3;0)	22	(1;-2;-3;4)
3	(-2;1;2;-1)	13	(2;0;3;0)	23	(-3;-1;0;2)
4	(-2;-1;2;1)	14	(-2;0;3;0)	24	(-3;0;1;3)
5	(1;2;3;4)	15	(2;0;-3;0)	25	(-3;0;0;3)
6	(1;-2;3;-4)	16	(-2;-3;1;2)	26	(0;2;2;0)
7	(1;-2;-3;4)	17	(-2;1;-1;4)	27	(2;0;3;0)
8	(-3;-1;0;2)	18	(-4;1;2;-1)	28	(2;0;3;0)
9	(-3;0;1;3)	19	(-2;-1;5;1)	29	(-2;0;3;0)
10	(-3;0;0;3)	20	(1;2;3;4)	30	(2;0;-3;0)

4. ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є

4.1. Алгоритми швидкого перетворення Фур'є

Швидке перетворення Фур'є (ШПФ) є загальною назвою будь-якого методу зменшення обчислювальної складності ДПФ. Існує безліч різноманітних методів зменшення складності. Найбільш поширеними алгоритмами є ШПФ по основі 2. Алгоритм ШПФ по основі 2 розділяє повне обчислення ДПФ на комбінацію 2-точкових ДПФ. Кожне 2-точкове ДПФ містить базову операцію множення з накопиченням, яка носить назву «метелик». Існують алгоритми ШПФ з проріджуванням за часом і з проріджуванням за частотою.

4.1.1. Швидке перетворення Фур'є з проріджуванням за часом

На рис. 4.1 зображена базова операція для ШПФ з проріджуванням за часом, а на рис. 4.2 – приклад для 4-точкового ШПФ.

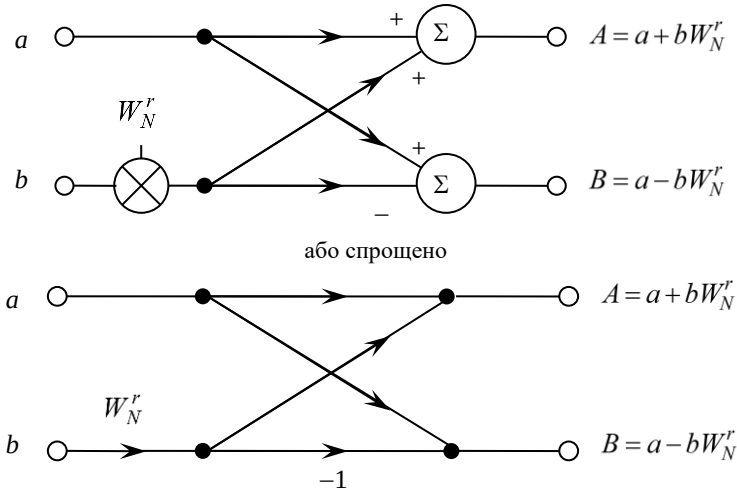


Рисунок 4.1 – Базова операція в алгоритмі ШПФ із проріджуванням за часом

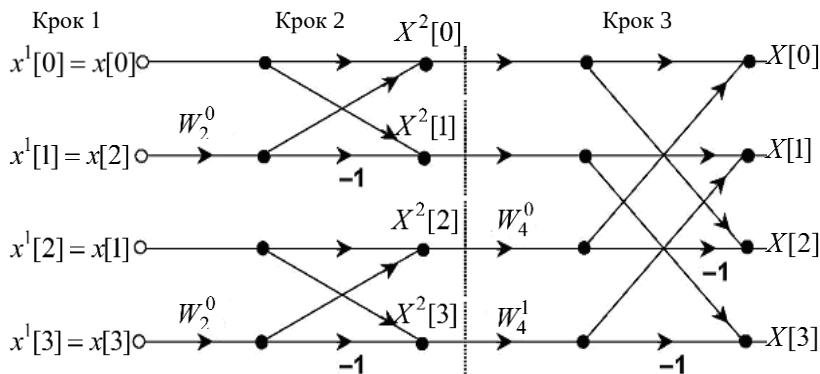


Рисунок 4.2 – Алгоритм 4-точкового ШПФ із прорідженням за часом

Приклад алгоритму реверсування біт, який застосовується для реалізації проріджування за часом, наведений в табл. 4.1. Десятковий індекс n представляється його двійковим еквівалентом, потім двійкові розряди розташовуються в зворотному порядку і перетворюються назад на десяткове число.

Таблиця 4.1 – Реверсування біт в алгоритмі проріджування за часом

Десяткове число	0	1	2	3	4	5	6	7
Двійковий еквівалент	000	001	010	011	100	101	110	111
Двійковий еквівалент з реверсуванням	000	100	010	110	001	101	011	111
Десятковий еквівалент	0	4	2	6	1	5	3	7

4.1.2. Швидке перетворення Фур'є з проріджуванням за частотою

На рис. 4.3 зображена базова операція для ШПФ з проріджуванням за частотою, а на рис. 4.4 приклад для 4-точкового ШПФ.

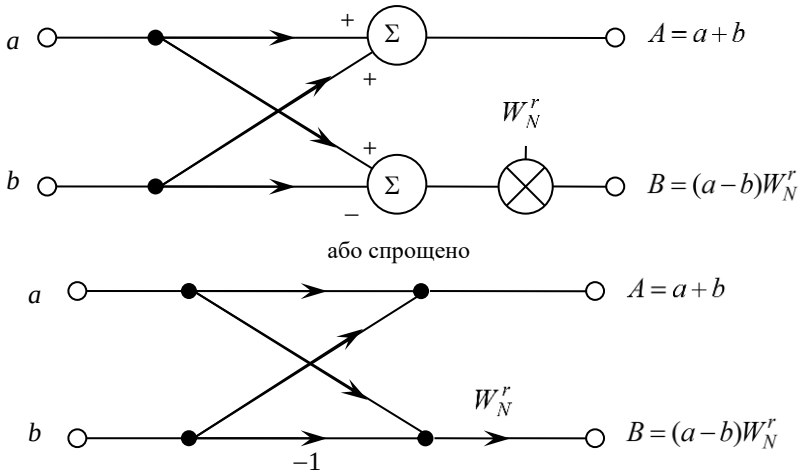


Рисунок 4.3 – Базова операція в алгоритмі ШПФ із проріджуванням за частотою

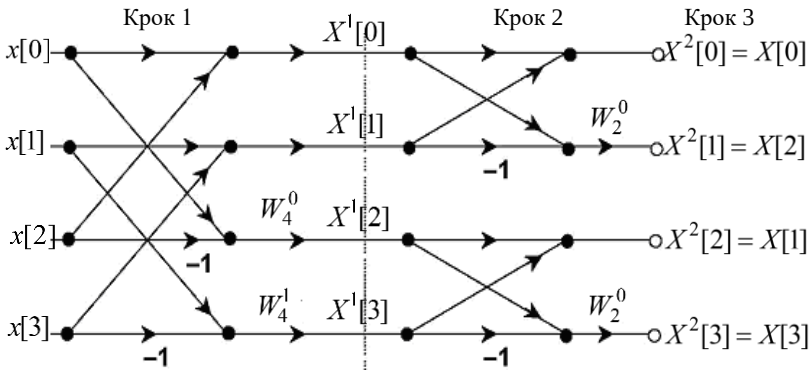


Рисунок 4.4 – Алгоритм 4-точкового ШПФ із прорідженням за частотою

4.1.3. Алгоритм зворотного швидкого перетворення Фур'є з проріджуванням за частотою

На рис. 4.5 зображений приклад для 4-точкового зворотного

ШПФ (ЗШПФ) з проріджуванням за частотою.

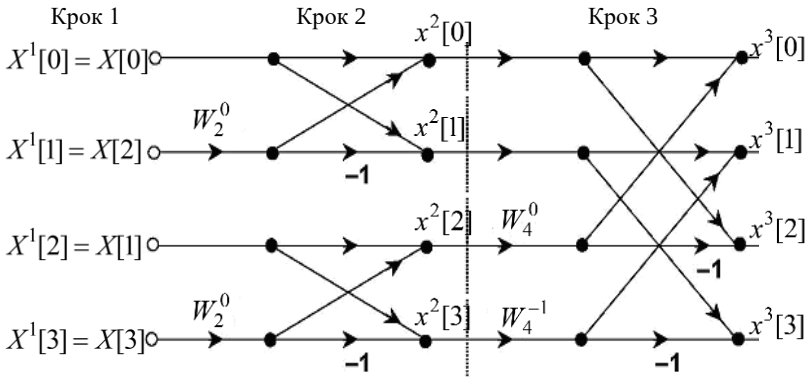


Рисунок 4.5 – Алгоритм 4-х точкового ЗШПФ з проріджуванням за частотою

4.2. Приклади до розділу 4

Приклад №4.1. Порівняємо спектри наступних сигналів $x[n] = (1; 0; -1)$ та $x[n] = (1; 0; -1; 0)$.

Розв'язок.

Розрахуємо спектр дискретного сигналу $x[n] = (1; 0; -1)$.

При $k = 0$:

$$X[0] = \sum_{n=0}^2 x[n] e^{-j \frac{2\pi}{3} 0n} = \sum_{n=0}^2 x[n], \text{ отже}$$

$$X[0] = x[0] + x[1] + x[2] = 1 + 0 - 1 = 0.$$

При $k = 1$:

$$X[1] = \sum_{n=0}^2 x[n] e^{-j \frac{2\pi}{3} 1n} = x[0] e^{-j \frac{2\pi}{3} 0} + x[1] e^{-j \frac{2\pi}{3} 1} + x[2] e^{-j \frac{2\pi}{3} 2},$$

$$X[1] = 1e^{-j0} + 0e^{-j\frac{2\pi}{3}} - 1e^{-j\frac{4\pi}{3}} = 1 - \frac{-1 + j\sqrt{3}}{2} = \frac{3 - j\sqrt{3}}{2}.$$

При $k = 2$:

$$X[2] = \sum_{n=0}^2 x[n] e^{-j\frac{2\pi}{3}2n} = x[0]e^{-j\frac{2\pi}{3}0} + x[1]e^{-j\frac{2\pi}{3}2} + x[2]e^{-j\frac{2\pi}{3}4},$$

$$X[2] = 1e^{-j0} + 0e^{-j\frac{4\pi}{3}} - 1e^{-j\frac{8\pi}{3}} = 1 - \frac{-1 - j\sqrt{3}}{2} = \frac{3 + j\sqrt{3}}{2}.$$

Отже, перші три гармоніки спектру наступні:

$$X[k] = \left(0; \frac{3 - j\sqrt{3}}{2}; \frac{3 + j\sqrt{3}}{2}\right).$$

Додамо у вихідний сигнал один відлік, що дорівнює 0 та розрахуємо спектр отриманого сигналу $x[n] = (1; 0; -1; 0)$.

При $k = 0$:

$$X[0] = \sum_{n=0}^3 x[n] e^{-j\frac{2\pi}{4}0n} = \sum_{n=0}^3 x[n], \quad X[0] = 1 + 0 - 1 + 0 = 0.$$

При $k = 1$:

$$X[1] = \sum_{n=0}^3 x[n] e^{-j\frac{2\pi}{4}1n} = x[0]e^{-j\frac{\pi}{2}0} + x[1]e^{-j\frac{\pi}{2}1} + x[2]e^{-j\frac{\pi}{2}2} + x[3]e^{-j\frac{\pi}{2}3},$$

$$X[1] = 1e^{-j0} + 0e^{-j\frac{\pi}{2}} - 1e^{-j\frac{\pi}{2}2} + 0e^{-j\frac{3\pi}{2}} = 1 + 1 = 2.$$

При $k = 2$:

$$X[2] = \sum_{n=0}^3 x[n] e^{-j\frac{2\pi}{4}2n} = x[0]e^{-j\frac{\pi}{2}0} + x[1]e^{-j\frac{\pi}{2}2} + x[2]e^{-j\frac{\pi}{2}4} + x[3]e^{-j\frac{\pi}{2}6}$$

$$X[2] = 1e^{-j0} + 0e^{-j\pi} - 1e^{-j2\pi} + 0e^{-j3\pi} = 1 - 1 = 0.$$

При $k=3$:

$$X[2] = \sum_{n=0}^3 x[n] e^{-j\frac{2\pi}{4}3n} = x[0]e^{-j\frac{\pi}{2}0} + x[1]e^{-j\frac{\pi}{2}3} + x[2]e^{-j\frac{\pi}{2}6} + x[3]e^{-j\frac{\pi}{2}9}$$

$$X[3] = 1e^{-j0} + 0e^{-j\frac{3\pi}{2}} - 1e^{-j3\pi} + 0e^{-j\frac{9\pi}{2}} = 1+1=2.$$

Отже, перші чотири гармоніки спектру дорівнюють $X[k] = (0; 2; 0; 2)$.

Таким чином, спектр містить більшу кількість гармонік, причому перші 3 гармоніки сигналів не співпадають.

Приклад №4.2. Розрахуємо ШПФ дискретного періодичного сигналу, заданого відліками $x[n] = (-2; -1; 0; 1)$. Порівняємо результати розрахунків двома алгоритмами.

Розв'язок.

Алгоритм проріджування за часом

Крок 1. Перестановка відліків

Вихідні номери	Двійковий вигляд вихідних номерів	Реверс бітів	Новий порядок відліків для реалізації алгоритму
0	00	00	0
1	01	10	2
2	10	01	1
3	11	11	3

Таким чином, отримуємо наступну послідовність:

$$x^1[0] = x[0] = -2, \quad x^1[1] = x[2] = 0, \quad x^1[2] = x[1] = -1, \quad x^1[3] = x[3] = 1.$$

Крок 2.

$$X^2[0] = x^1[0] + W_2^0 x^1[1] = -2 + 0 = -2;$$

$$X^2[1] = x^1[0] - W_2^0 x^1[1] = -2 - 0 = -2;$$

$$X^2[2] = x^1[2] + W_2^0 x^1[3] = -1 + 1 = 0;$$

$$X^2[3] = x^1[2] - W_2^0 x^1[3] = -1 - 1 = -2.$$

Крок 3.

$$X[0] = X^2[0] + W_4^0 X^2[2] = -2 + 0 = -2;$$

$$\begin{aligned} X[1] &= X^2[1] + W_4^1 X^2[3] = -2 - 2e^{-j\frac{2\pi}{4}} = -2 - 2(\cos\frac{\pi}{2} - j\sin\frac{\pi}{2}) = \\ &= -2 + 2j; \end{aligned}$$

$$X[2] = X^2[0] - W_4^0 X^2[2] = -2 + 0 = -2;$$

$$\begin{aligned} X[3] &= X^2[1] - W_4^1 X^2[3] = -2 + 2e^{-j\frac{2\pi}{4}} = -2 + 2(\cos\frac{\pi}{2} - j\sin\frac{\pi}{2}) = \\ &= -2 - 2j. \end{aligned}$$

Алгоритм прорідження за частотою

Крок 1.

$$X^1[0] = x[0] + x[2] = -2 + 0 = -2;$$

$$X^1[1] = x[1] + x[3] = -1 + 1 = 0;$$

$$X^1[2] = (x[0] - x[2])W_4^0 = (-2 - 0)e^{-j0} = -2;$$

$$X^1[3] = (x[1] - x[3])W_4^1 = (-1 - 1)e^{-j\frac{\pi}{2}} = 2j.$$

Крок 2.

$$X^2[0] = X^1[0] + X^1[1] = -2 + 0 = -2;$$

$$X^2[1] = (X^1[0] - X^1[1])W_2^0 = -2 - 0 = -2;$$

$$X^2[2] = X^1[2] + X^1[3] = -2 + 2j;$$

$$X^2[3] = (X^1[2] - X^1[3])W_2^0 = -2 - 2j.$$

Крок 3. Перестановки

$$X[0] = X^2[0] = -2; \quad X[1] = X^2[2] = -2 + 2j;$$

$$X[2] = X^2[1] = -2; \quad X[3] = X^2[3] = -2 - 2j.$$

Отримані двома алгоритмами спектри повністю співпадають.

Приклад №4.3. Розрахувати значення вихідного сигналу за допомогою ЗШПФ на базі отриманих у прикладі 2 спектрів.

Розв'язок.

Крок 1.

$$X^1[0] = X[0] = -2; \quad X^1[1] = X[2] = -2;$$

$$X^1[2] = X[1] = -2 + 2j; \quad X^1[3] = X[3] = -2 - 2j.$$

Крок 2.

$$x^2[0] = X^1[0] + W_2^0 X^1[1] = -2 - 2 = -4;$$

$$x^2[1] = X^1[0] - W_2^0 X^1[1] = -2 + 2 = 0;$$

$$x^2[2] = X^1[2] + W_2^0 X^1[3] = -2 + 2j + (-2 - 2j) = -4;$$

$$x^2[3] = X^1[2] - W_2^0 X^1[3] = -2 + 2j + 2 + 2j = 4j.$$

Крок 3.

$$x^3[0] = x^2[0] + W_4^0 x^2[2] = -4 - 4 = -8;$$

$$x^3[1] = x^2[1] + W_4^{-1} x^2[3] = 0 + 4j e^{j\frac{\pi}{2}} = 4j^2 = -4;$$

$$x^3[2] = x^2[0] - W_4^0 x^2[2] = -4 + 4 = 0 ;$$

$$x^3[3] = x^2[1] - W_4^{-1} x^2[3] = 0 - 4je^{j\frac{\pi}{2}} = -4j^2 = 4 .$$

Крок 4. Виконуємо нормування на N , тобто $x[n] = \frac{1}{4} x^3[n]$,
 $x[n] = (-2, -1, 0, 1)$.

4.3. Завдання до розділу 4 для самостійної роботи

1) Розрахувати ШПФ дискретного сигналу. Вхідні дані див. у табл. 3.1. та табл. 4.2 (N – номер студента за журналом).

2) Розрахувати ЗШПФ дискретного сигналу, заданого своїм спектром. Вхідні дані див. у табл. 3.2. та табл. 4.2.

Таблиця 4.2 – Варіанти алгоритмів ШПФ та ЗШПФ

N	Алгоритм ШПФ та ЗШПФ з прорідженням за ...	N	Алгоритм ШПФ та ЗШПФ з прорідженням за ...
1	часом	16	частотою
2	частотою	17	часом
3	часом	18	частотою
4	частотою	19	часом
5	часом	20	частотою
6	частотою	21	часом
7	часом	22	частотою
8	частотою	23	часом
9	часом	24	частотою
10	частотою	25	часом
11	часом	26	частотою
12	частотою	27	часом
13	часом	28	частотою
14	частотою	29	часом
15	часом	30	частотою

5. КРУГОВА І ЛІНІЙНА ЗГОРТКИ

5.1. Визначення згортки

За визначенням кругова згортка двох послідовностей $x[n]$ та $y[n]$ дорівнює

$$g[n] = \sum_{m=0}^{N-1} x[m]y[n-m], \quad n = \overline{0, N-1}. \quad (5.1)$$

Оскільки послідовності $x[n]$ та $y[n]$ скінченні, то вони розглядаються як періодичні з періодом повторення N (рис. 5.1).

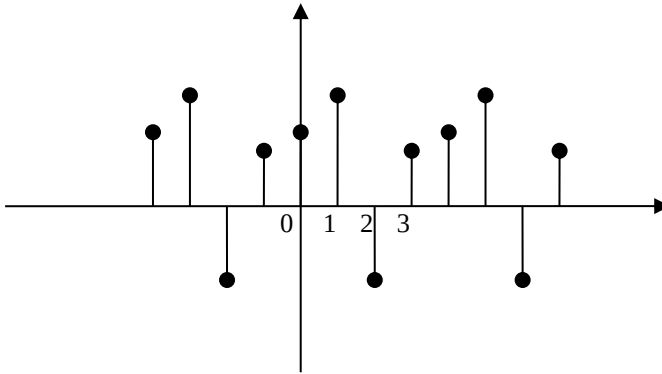


Рисунок 5.1. – Приклад періодичності скінченного сигналу

5.1.1. Кругова згортка

Розпишемо вираз (5.1):

$$g[0] = x[0]y[0] + x[1]y[N-1] + x[2]y[N-2] + \dots$$

$$g[1] = x[0]y[1] + x[1]y[0] + x[2]y[N-1] + \dots$$

...

$$g[N-1] = x[0]y[N-1] + x[1]y[N-2] + x[2]y[N-3] + \dots$$

Цей алгоритм можна записати в матричній формі, якщо послідовність $y[n]$ записати у вигляді матриці $\mathbf{Y}$:

$$\mathbf{Y} = \begin{bmatrix} y[0] & y[N-1] & y[N-2] & \dots & y[1] \\ y[1] & y[0] & y[N-1] & \dots & y[2] \\ y[2] & y[1] & y[0] & \dots & y[3] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y[N-1] & y[N-2] & y[N-3] & \dots & y[0] \end{bmatrix}.$$

Тоді згортка послідовностей в матричному вигляді розраховується шляхом множення матриці $\mathbf{Y}$ на вектор $\mathbf{x}$:

$$\bar{g} = \mathbf{Y}\bar{x}.$$

5.1.2. Лінійна згортка

Для отримання лінійної згортки скінчених послідовностей $x[n]$ і $y[n]$ необхідно додати в кінець послідовностей нулі так, щоб довжина кожної послідовності була $N = N_1 + N_2 - 1$, де N_1 , N_2 – довжини послідовностей $x[n]$ і $y[n]$ відповідно. Після чого згортка знаходиться за виразом (5.1).

5.2. Приклади до розділу 5

Приклад №5.1. Знайти кругову та лінійну згортки послідовностей $x[n] = (1, 2, -1, 1)$ та $y[n] = (1, 2, 3, 4)$.

Розв'язок.

Знайдемо кругову згортку. У часовій області (в матричному вигляді) матимемо:

$$\begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+8-3+2 \\ 2+2-4+3 \\ 3+4-1+4 \\ 4+6-3+1 \end{bmatrix} = \begin{bmatrix} 8 \\ 3 \\ 10 \\ 9 \end{bmatrix}.$$

В результаті отримали кругову згортку $g[n] = (8, 3, 10, 9)$.

Перевіримо результат за допомогою ДПФ (властивості ДПФ розглянуті в підрозділі 3.2). Знайдемо ДПФ послідовності $x = (1, 2, -1, 1)$:

$$X[0] = 1 + 2 - 1 + 1 = 3;$$

$$X[1] = 1e^0 + 2e^{-j\frac{2\pi}{4}} - 1e^{-j\frac{2\pi}{4}2} + 1e^{-j\frac{2\pi}{4}3} = 1 - 2j + 1 + j = 2 - j;$$

$$X[2] = 1e^0 + 2e^{-j\frac{2\pi}{4}2} - 1e^{-j\frac{2\pi}{4}4} + 1e^{-j\frac{2\pi}{4}6} = 1 - 2 - 1 - 1 = -3;$$

$$X[3] = 1e^0 + 2e^{-j\frac{2\pi}{4}3} - 1e^{-j\frac{2\pi}{4}6} + 1e^{-j\frac{2\pi}{4}9} = 1 + 2j + 1 - j = 2 + j.$$

Знайдемо ДПФ послідовності $y = (1, 2, 3, 4)$:

$$Y[0] = 1 + 2 + 3 + 4 = 10;$$

$$Y[1] = 1e^0 + 2e^{-j\frac{2\pi}{4}} + 3e^{-j\frac{2\pi}{4}2} + 4e^{-j\frac{2\pi}{4}3} = 1 - 2j - 3 + 4j = -2 + 2j;$$

$$Y[2] = 1e^0 + 2e^{-j\frac{2\pi}{4}2} + 3e^{-j\frac{2\pi}{4}4} + 4e^{-j\frac{2\pi}{4}6} = 1 - 2 + 3 - 4 = -2;$$

$$Y[3] = 1e^0 + 2e^{-j\frac{2\pi}{4}3} + 3e^{-j\frac{2\pi}{4}6} + 4e^{-j\frac{2\pi}{4}9} = 1 + 2j - 3 - 4j = -2 - 2j.$$

Знайдемо добуток $X[k]$ та $Y[k]$:

$$G[0] = X[0]Y[0] = 10 \times 3 = 30;$$

$$G[1] = X[1]Y[1] = (2 - j)(-2 + 2j) = -4 + 2j + 4j + 2 = -2 + 6j;$$

$$G[2] = X[2]Y[2] = (-3)(-2) = 6;$$

$$G[3] = X[3]Y[3] = (2 + j)(-2 - 2j) = -4 - 2j - 4j + 2 = -2 - 6j.$$

Обчислимо 3ДПФ послідовності $G[k]$:

$$g[0] = \frac{1}{4}(30 - 2 + 6j + 6 - 2 - 6j) = 32/4 = 8;$$

$$\begin{aligned} g[1] &= \frac{1}{4}(30e^0 + (-2 + 6j)e^{j\frac{2\pi}{4}1} + 6e^{j\frac{2\pi}{4}2} + (-2 - 6j)e^{j\frac{2\pi}{4}3}) = \\ &= \frac{1}{4}(30 + (-2 + 6j)j + 6(-1) + (-2 - 6j)(-j)) = 12/4 = 3; \end{aligned}$$

$$\begin{aligned} g[2] &= \frac{1}{4}(30e^0 + (-2 + 6j)e^{j\frac{2\pi}{4}2} + 6e^{j\frac{2\pi}{4}4} + (-2 - 6j)e^{j\frac{2\pi}{4}6}) = \\ &= \frac{1}{4}(30 + (-2 + 6j)(-1) + 6 + (-2 - 6j)(-j)) = 40/4 = 10; \end{aligned}$$

$$\begin{aligned} g[3] &= \frac{1}{4}(30e^0 + (-2 + 6j)e^{j\frac{2\pi}{4}3} + 6e^{j\frac{2\pi}{4}6} + (-2 - 6j)e^{j\frac{2\pi}{4}9}) = \\ &= \frac{1}{4}(30 + (-2 + 6j)(-j) + 6(-1) + (-2 - 6j)j) = 36/4 = 9. \end{aligned}$$

Результат співпав з результатом кругової згортки, розрахованої матричним способом.

Аналогічні результати можна отримати за допомогою наступної програми:

```
%Приклад №5.1
%початкові послідовності
x=[1,2,-1,1]
y=[1,2,3,4]
%розрахунок кругової згортки
```

```
g=conv(x,y,4) %вбудована функція
%розрахунок з використанням властивості ДПФ
gg=ifft(fft(x).*fft(y))
```

В наведеному прикладі демонструється властивість ДПФ, що була розглянута в підрозділі 3.2. Результати розрахунку різними способами повністю співпадають.

Тепер розрахуємо лінійну згортку. Доповнимо вхідні послідовності нулями так, щоб довжина кожної послідовності дорівнювала $N = N_1 + N_2 - 1 = 4 + 4 - 1 = 7$, тобто необхідно додати 3 нулі, щоб вихідна послідовність мала довжину рівну 7.

В часовій області (у матричному вигляді) матимемо:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 4 & 3 & 2 \\ 2 & 1 & 0 & 0 & 0 & 4 & 3 \\ 3 & 2 & 1 & 0 & 0 & 0 & 4 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 4 & 3 & 2 & 1 & 0 & 0 \\ 0 & 0 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 4 & 3 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2+2 \\ 3+4-1 \\ 4+6-2+1 \\ 6-3+2 \\ -4+3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 9 \\ 7 \\ -1 \\ 4 \end{bmatrix}.$$

В результаті отримали лінійну згортку $g[n] = (1, 4, 6, 9, 7, -1, 4)$. Очевидно, що лінійна згортка не співпадає з круговою.

Аналогічні результати можна отримати за допомогою наступної програми:

```
%Приклад №5.2
%початкові послідовності
x=[1,2,-1,1]
y=[1,2,3,4]
%розрахунок кругової згортки
g=conv(x,y,7) %вбудована функція
```

```
%розрахунок з використанням властивості ДПФ
gg=ifft (fft (x, 7) .* fft (y, 7) )
```

У наведеній програмі при розрахунку з використанням властивості ДПФ кожна послідовність доповнюється нулями.

Приклад №5.2. Знайти лінійну згортку двох послідовностей $x[n]=(3,-1,2,4)$ та $y[n]=(3,-2,-1)$.

Розв'язок.

Довжина лінійної згортки дорівнює $N = 4 + 3 - 1 = 6$. Значить, в послідовність $x[n]$ необхідно додати 2 нулі, а в $y[n]$ – 3 нулі.

У часовій області (у матричному вигляді) матимемо:

$$\begin{bmatrix} 3 & 0 & 0 & 0 & -1 & -2 \\ -2 & 3 & 0 & 0 & 0 & -1 \\ -1 & -2 & 3 & 0 & 0 & 0 \\ 0 & -1 & -2 & 3 & 0 & 0 \\ 0 & 0 & -1 & -2 & 3 & 0 \\ 0 & 0 & 0 & -1 & -2 & 3 \end{bmatrix} \times \begin{bmatrix} 3 \\ -1 \\ 2 \\ 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ -9 \\ 5 \\ 9 \\ -10 \\ -4 \end{bmatrix}$$

В результаті отримали лінійну згортку $g[n] = (9, -9, 5, 9, -10, -4)$.

5.3. Завдання до розділу 5 для самостійної роботи

Розрахувати кругову і лінійну згортки послідовностей (матричним способом). Вхідні дані див. в табл. 5.1 (N – номер студента за журналом).

Таблиця 5.1 – Варіанти дискретних сигналів для розрахунків

N	$x[n]$	$y[n]$	N	$x[n]$	$y[n]$
1	$(-2;-1;1;2)$	$(2;3;1;2)$	16	$(-2;-3;1;2)$	$(-1;2;1;4)$
2	$(-2;1;-1;2)$	$(2;-3;1;2)$	17	$(-2;1;-1;4)$	$(1;-2;1;4)$
3	$(-2;1;2;-1)$	$(2;3;-1;2)$	18	$(-4;1;2;-1)$	$(1;2;-1;4)$
4	$(-2;-1;2;1)$	$(2;3;1;-2)$	19	$(-2;-1;5;1)$	$(1;2;1;-4)$
5	$(1;2;3;4)$	$(1;3;5;2)$	20	$(1;2;3;4)$	$(2;0;-3;0)$
6	$(1;-2;3;-4)$	$(-1;3;5;2)$	21	$(1;-2;3;-4)$	$(-2;-1;5;1)$
7	$(1;-2;-3;4)$	$(1;-3;5;2)$	22	$(1;-2;-3;4)$	$(-2;1;-5;1)$
8	$(-3;-1;0;2)$	$(1;3;-5;2)$	23	$(-3;-1;0;2)$	$(2;1;5;1)$
9	$(-3;0;1;3)$	$(1;3;5;-2)$	24	$(-3;0;1;3)$	$(2;1;5;-1)$
10	$(-3;0;0;3)$	$(5;2;1;3)$	25	$(-3;0;0;3)$	$(-1;2;1;4)$
11	$(0;2;2;0)$	$(-5;2;1;3)$	26	$(0;2;2;0)$	$(5;2;1;-3)$
12	$(2;0;3;0)$	$(5;-2;1;3)$	27	$(2;0;3;0)$	$(5;-2;1;3)$
13	$(2;0;3;0)$	$(5;2;-1;3)$	28	$(2;0;3;0)$	$(1;2;-1;4)$
14	$(-2;0;3;0)$	$(5;2;1;-3)$	29	$(-2;0;3;0)$	$(1;2;1;4)$
15	$(2;0;-3;0)$	$(1;2;1;4)$	30	$(2;0;-3;0)$	$(1;2;1;-4)$

6. КОРЕЛЯЦІЙНА ФУНКЦІЯ

6.1. Автокореляційна функція та її властивості

Кореляційний аналіз грає велику роль в теорії сигналів. Його суть полягає в кількісному вимірі міри схожості різних сигналів. Для цього служать кореляційні функції.

Кореляційна функція $r_{xx}[n]$ сигналу $x[n]$ – це згортка інверсних сигналів $x[n]$ та $x[-n]$ (рис. 6.1):

$$r_{xx}[n] = \sum_{m=0}^{N-1} x[m]x[n - (-m)] = \sum_{m=0}^{N-1} x[m]x[n + m]. \quad (6.1)$$

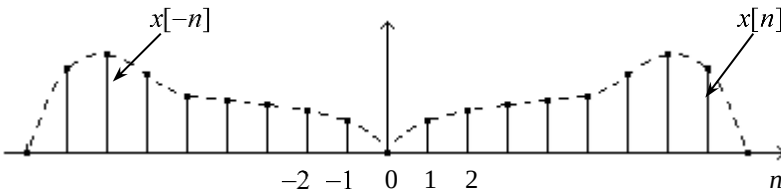


Рисунок 6.1 – Приклад інверсних сигналів $x[n]$ та $x[-n]$

Розпишемо вираз (6.1):

$$r_{xx}[0] = x[0]x[0] + x[1]x[1] + x[2]x[2] + \dots$$

$$r_{xx}[1] = x[0]x[1] + x[1]x[2] + x[2]x[3] + \dots$$

$$r_{xx}[2] = x[0]x[2] + x[1]x[3] + x[2]x[4] + \dots$$

...

$$r_{xx}[N-1] = x[0]x[N-1] + x[1]x[0] + x[2]x[1] + \dots$$

Цей алгоритм можна записати в матричній формі, якщо послідов-

ність $x[n]$ записати у вигляді матриці:

$$\mathbf{X} = \begin{bmatrix} x[0] & x[1] & x[2] & \dots & x[N-1] \\ x[1] & x[2] & x[3] & \dots & x[0] \\ x[2] & x[3] & x[4] & \dots & x[1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[N-1] & x[0] & x[1] & \dots & x[N-2] \end{bmatrix}.$$

Тоді кореляційна функція $r_{xx}[n]$ сигналу $x[n]$ в матричному [вигляді](#) розраховується шляхом множення матриці $\mathbf{X}$ на вектор $\vec{x}$:

$$\vec{r}_{xx} = \mathbf{X}\vec{x}.$$

Для періодичних дискретних сигналів кореляційна функція і енергетичний спектр пов'язані формулами ДПФ:

$$r_{xx}[n] = \frac{1}{N} \sum_{k=0}^{N-1} R_{xx}[k] W_N^{-kn} = \frac{1}{N} \sum_{k=0}^{N-1} X^2[k] W_N^{-kn}.$$

Кореляційну функцію також ще називають автокореляційною функцією (АКФ). АКФ показує міру схожості між сигналом і його зсуненою копією. Чим більше значення АКФ, тим ця схожість сильніша. Крім того, АКФ має наступні *властивості*:

- 1) Значення АКФ при $n = 0$ дорівнює енергії сигналу.
- 2) АКФ є парною функцією свого аргументу n , тобто $r_{xx}[n] = r_{xx}[N - n]$.
- 3) Значення АКФ при $n = 0$ є максимально можливим значенням, тобто $r_{xx}[n] \leq r_{xx}[0]$.

Багато що з фізичних явищ є періодичними процесами. За допомогою АКФ можна визначити наявність періоду досліджуваної функ-

ції і величину цього періоду (рис. 6.2). Функція має властивість періодичності, якщо на АКФ з'являться пікові значення через інтервали, що кратні цілим числам. Величина цього інтервалу дорівнює періоду досліджуваної функції.

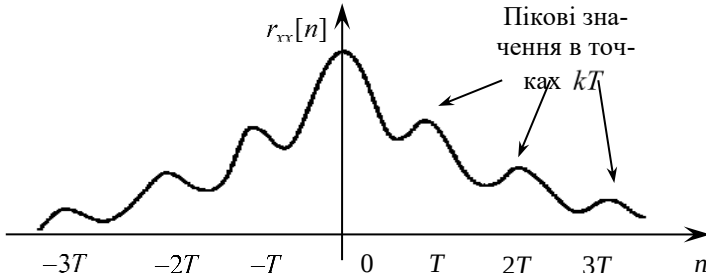


Рисунок 6.2 – АКФ для періодичної функції з періодом T

Варто зазначити, що АКФ не унікальні, тобто різні сигнали можуть мати однакові кореляційні функції, тому за знайденою АКФ не можна визначити форму сигналу.

6.2. Взаємнокореляційна функція та її властивості

Для визначення схожості зсунутих екземплярів двох різних сигналів використовується взаємнокореляційна функція.

Згортка сигналу з інверсною копією іншого сигналу називається взаємною кореляцією цих сигналів або взаємнокореляційною функцією (ВКФ):

$$r_{xy}[n] = \sum_{m=0}^{N-1} x[m]y[n+m].$$

де $N = N_1 + N_2 - 1$ – кількість відліків взаємнокореляційної функції;

N_1 , N_2 – кількість відліків скінчених сигналів $x[n]$ та $y[n]$ відповідно.

Розрахунок ВКФ також можна записати в матричному вигляді:

$$r_{xy} = \mathbf{Y}\mathbf{x},$$

$$\mathbf{Y} = \begin{bmatrix} y[0] & y[1] & y[2] & \dots & y[N-1] \\ y[1] & y[2] & y[3] & \dots & y[0] \\ y[2] & y[3] & y[4] & \dots & y[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y[N-1] & y[0] & x[1] & \dots & y[N-2] \end{bmatrix}.$$

Властивості ВКФ дещо відрізняються від властивостей АКФ:

- 1) ВКФ також скінченна для скінчених за енергією сигналів;
- 2) ВКФ властивості парності не має, проте $r_{xy}[n] = r_{yx}[N-n]$,

тобто зміна знаку n дорівнює взаємній перестановці сигналів;

- 3) значення ВКФ при $n=0$ нічим не виділяється, максимум може бути розташований у будь-якому місці осі абсцис.

АКФ і ВКФ можна розрахувати, використовуючи спектри сигналів. Оскільки АКФ є окремим випадком ВКФ, розглянемо загальний випадок розрахунку ВКФ через спектри сигналів.

Як зазначалось раніше, ВКФ обчислюються як згортки сигналу з інверсною копією іншого сигналу. При цьому інверсній копії сигналу відповідає комплексно зв'язаний спектр початкового сигналу. Тоді, використовуючи властивість згортки, спектр ВКФ дорівнюватиме:

$$R_{xy}[k] = X^*[k]Y[k]. \quad (6.2)$$

Виконавши зворотне ДПФ, легко отримати значення ВКФ.

6.3. Приклади до розділу 6

Приклад №6.1. Знайти кореляційну функцію (автокореляційну функцію – АКФ) послідовності $x = (1, 2, 3, 4)$.

Розв'язок.

Знайдемо АКФ. Для отримання правильного результату (не циклічної кореляційної функції) доповнимо вхідну послідовність нулями так, щоб довжина послідовності дорівнювала $N = 2 \cdot N_1 - 1 = 2 \cdot 4 - 1 = 7$, тобто необхідно додати 3 нулі.

Обчислення АКФ в часовій області (в матричному вигляді):

$$\vec{r}_{xx} = \begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 2 & 3 & 4 & 0 & 0 & 0 & 1 \\ 3 & 4 & 0 & 0 & 0 & 1 & 2 \\ 4 & 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 2 & 3 & 4 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+4+9+16 \\ 2+6+12 \\ 3+8 \\ 4 \\ 4 \\ 3+8 \\ 2+6+12 \end{bmatrix} = \begin{bmatrix} 30 \\ 20 \\ 11 \\ 4 \\ 4 \\ 11 \\ 20 \end{bmatrix}$$

Перевіримо результат за допомогою ДПФ.

Знайдемо ДПФ послідовності $x = (1, 2, 3, 4, 0, 0, 0)$:

$$X[0] = 1 + 2 + 3 + 4 + 0 + 0 + 0 = 10;$$

$$X[1] = 1e^0 + 2e^{-j\frac{2\pi}{7}} + 3e^{-j\frac{2\pi}{7}2} + 4e^{-j\frac{2\pi}{7}3} = 1 + 2e^{-j\frac{2\pi}{7}} + 3e^{-j\frac{4\pi}{7}} + 4e^{-j\frac{6\pi}{7}};$$

$$X[2] = 1e^0 + 2e^{-j\frac{2\pi}{7}2} + 3e^{-j\frac{2\pi}{7}4} + 4e^{-j\frac{2\pi}{7}6} = 1e^0 + 2e^{-j\frac{4\pi}{7}} - 3e^{-j\frac{\pi}{7}} - 4e^{-j\frac{5\pi}{7}};$$

$$X[3] = 1e^0 + 2e^{-j\frac{2\pi}{7}3} + 3e^{-j\frac{2\pi}{7}6} + 4e^{-j\frac{2\pi}{7}9} = 1 + 2e^{-j\frac{6\pi}{7}} - 3e^{-j\frac{5\pi}{7}} + 4e^{-j\frac{4\pi}{7}};$$

$$X[4] = 1e^0 + 2e^{-j\frac{2\pi}{7}4} + 3e^{-j\frac{2\pi}{7}8} + 4e^{-j\frac{2\pi}{7}12} = 1 - 2e^{-j\frac{\pi}{7}} + 3e^{-j\frac{2\pi}{7}} - 4e^{-j\frac{3\pi}{7}};$$

$$X[5] = 1e^0 + 2e^{-j\frac{2\pi}{7}5} + 3e^{-j\frac{2\pi}{7}10} + 4e^{-j\frac{2\pi}{7}15} = 1 - 2e^{-j\frac{3\pi}{7}} + 3e^{-j\frac{6\pi}{7}} + 4e^{-j\frac{2\pi}{7}};$$

$$X[6] = 1e^0 + 2e^{-j\frac{2\pi}{7}6} + 3e^{-j\frac{2\pi}{7}12} + 4e^{-j\frac{2\pi}{7}18} = 1 - 2e^{-j\frac{5\pi}{7}} - 3e^{-j\frac{3\pi}{7}} - 4e^{-j\frac{\pi}{7}}.$$

Найдем добуток $R_{xx}[k] = X[k]X^*[k]$:

$$R_{xx}[0] = X[0]X^*[0] = 10 \times 10 = 100;$$

$$\begin{aligned} R_{xx}[1] &= X[1]X^*[1] = (1 + 2e^{-j\frac{2\pi}{7}} + 3e^{-j\frac{4\pi}{7}} + 4e^{-j\frac{6\pi}{7}}) * (1 + 2e^{j\frac{2\pi}{7}} + 3e^{j\frac{4\pi}{7}} + \\ &+ 4e^{j\frac{6\pi}{7}}) = 1 + \underline{2e^{-j\frac{2\pi}{7}}} + \underline{3e^{-j\frac{4\pi}{7}}} + 4e^{-j\frac{6\pi}{7}} + \underline{2e^{j\frac{2\pi}{7}}} + 4 + \underline{6e^{-j\frac{4\pi}{7} + j\frac{2\pi}{7}}} + \\ &+ \underline{8e^{-j\frac{6\pi}{7} + j\frac{2\pi}{7}}} + \underline{3e^{j\frac{4\pi}{7}}} + \underline{6e^{-j\frac{2\pi}{7} + j\frac{4\pi}{7}}} + \underline{9 + 12e^{-j\frac{6\pi}{7} + j\frac{4\pi}{7}}} + 4e^{j\frac{6\pi}{7}} + \underline{8e^{-j\frac{2\pi}{7} + j\frac{6\pi}{7}}} + \\ &+ 12e^{-j\frac{4\pi}{7} + j\frac{6\pi}{7}} + 16 = 30 + 20e^{-j\frac{2\pi}{7}} + 11e^{-j\frac{4\pi}{7}} + 4e^{-j\frac{6\pi}{7}} + 20e^{j\frac{2\pi}{7}} + 11e^{j\frac{4\pi}{7}} + \\ &+ 4e^{j\frac{6\pi}{7}} = 30 + 40 \cos \frac{2\pi}{7} + 22 \cos \frac{4\pi}{7} + 8 \cos \frac{6\pi}{7} = 42.8364; \end{aligned}$$

$$\begin{aligned} R_{xx}[2] &= X[2]X^*[2] = (1e^0 + 2e^{-j\frac{4\pi}{7}} - 3e^{-j\frac{\pi}{7}} - 4e^{-j\frac{5\pi}{7}}) * \\ &*(1e^0 + 2e^{j\frac{4\pi}{7}} - 3e^{j\frac{\pi}{7}} - 4e^{j\frac{5\pi}{7}}) = 6.2658; \end{aligned}$$

$$\begin{aligned} R_{xx}[3] &= X[3]X^*[3] = (1 + 2e^{-j\frac{6\pi}{7}} - 3e^{-j\frac{5\pi}{7}} + 4e^{-j\frac{4\pi}{7}}) * \\ &*(1 + 2e^{j\frac{6\pi}{7}} - 3e^{j\frac{5\pi}{7}} + 4e^{j\frac{4\pi}{7}}) = 5.8979; \end{aligned}$$

$$\begin{aligned} R_{xx}[4] &= X[4]X^*[4] = (1 - 2e^{-j\frac{\pi}{7}} + 3e^{-j\frac{2\pi}{7}} - 4e^{-j\frac{3\pi}{7}}) * (1 - 2e^{j\frac{\pi}{7}} + 3e^{j\frac{2\pi}{7}} - 4e^{j\frac{3\pi}{7}}) = \\ &= 5.8979; \end{aligned}$$

$$\begin{aligned} R_{xx}[5] &= X[5]X^*[5] = (1 - 2e^{-j\frac{3\pi}{7}} + 3e^{-j\frac{6\pi}{7}} + 4e^{-j\frac{2\pi}{7}}) * \\ &*(1 - 2e^{j\frac{3\pi}{7}} + 3e^{j\frac{6\pi}{7}} + 4e^{j\frac{2\pi}{7}}) = 6.2658; \end{aligned}$$

$$R_{xx}[6] = X[6]X^*[6] = (1 - 2e^{-j\frac{5\pi}{7}} - 3e^{-j\frac{3\pi}{7}} - 4e^{-j\frac{\pi}{7}})^* \\ * (1 - 2e^{-j\frac{5\pi}{7}} - 3e^{-j\frac{3\pi}{7}} - 4e^{-j\frac{\pi}{7}}) = 42.8364.$$

ЗДПФ послідовності $R_{xx}[k]$:

$$r_{xx}[0] = \frac{1}{7}(100 + 42.8364 + 6.2658 + 5.8979 + 5.8979 + 6.2658 + 42.8364) = 30;$$

$$r_{xx}[1] = \frac{1}{7}(100e^0 + 42.8364e^{j\frac{2\pi}{7}} + 6.2658e^{j\frac{2\pi}{7}2} + 5.8979e^{j\frac{2\pi}{7}3} + \\ + 5.8979e^{j\frac{2\pi}{7}4} + 6.2658e^{j\frac{2\pi}{7}5} + 42.8364e^{j\frac{2\pi}{7}6}) = 20;$$

$$r_{xx}[2] = \frac{1}{7}(100e^0 + 42.8364e^{j\frac{2\pi}{7}2} + 6.2658e^{j\frac{2\pi}{7}4} + 5.8979e^{j\frac{2\pi}{7}6} + \\ + 5.8979e^{j\frac{2\pi}{7}8} + 6.2658e^{j\frac{2\pi}{7}10} + 42.8364e^{j\frac{2\pi}{7}12}) = 11;$$

$$r_{xx}[3] = \frac{1}{7}(100e^0 + 42.8364e^{j\frac{2\pi}{7}3} + 6.2658e^{j\frac{2\pi}{7}6} + 5.8979e^{j\frac{2\pi}{7}9} + \\ + 5.8979e^{j\frac{2\pi}{7}12} + 6.2658e^{j\frac{2\pi}{7}15} + 42.8364e^{j\frac{2\pi}{7}18}) = 4;$$

$$r_{xx}[4] = \frac{1}{7}(100e^0 + 42.8364e^{j\frac{2\pi}{7}4} + 6.2658e^{j\frac{2\pi}{7}8} + 5.8979e^{j\frac{2\pi}{7}12} + \\ + 5.8979e^{j\frac{2\pi}{7}16} + 6.2658e^{j\frac{2\pi}{7}20} + 42.8364e^{j\frac{2\pi}{7}24}) = 4;$$

$$r_{xx}[5] = \frac{1}{7}(100e^0 + 42.8364e^{j\frac{2\pi}{7}5} + 6.2658e^{j\frac{2\pi}{7}10} + 5.8979e^{j\frac{2\pi}{7}15} + \\ + 5.8979e^{j\frac{2\pi}{7}20} + 6.2658e^{j\frac{2\pi}{7}21} + 42.8364e^{j\frac{2\pi}{7}22}) = 11;$$

$$r_{xx}[6] = \frac{1}{7}(100e^0 + 42.8364e^{j\frac{2\pi}{7}6} + 6.2658e^{j\frac{2\pi}{7}12} + 5.8979e^{j\frac{2\pi}{7}18} + \\ + 5.8979e^{j\frac{2\pi}{7}24} + 6.2658e^{j\frac{2\pi}{7}30} + 42.8364e^{j\frac{2\pi}{7}36}) = 20.$$

Результат співпав з результатом розрахунку в часовій області.

Отримаємо результат за допомогою лінійної згортки. Для отримання лінійної згортки сигналу з його інверсною копією доповнимо вхідну послідовність нулями так, щоб довжина послідовності дорівнювала $N = 2N_1 - 1 = 2 \cdot 4 - 1 = 7$, тобто необхідно додати 3 нулі. Запишемо сигнал в інверсному вигляді: $x^*[n] = (1, 0, 0, 0, 4, 3, 2)$.

$$\vec{r}_{xx} = \begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 & 4 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 4 & 0 & 0 & 0 & 1 & 2 & 3 \\ 3 & 4 & 0 & 0 & 0 & 1 & 2 \\ 2 & 3 & 4 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+4+9+16 \\ 2+6+12 \\ 3+8 \\ 4 \\ 4 \\ 3+8 \\ 2+6+12 \end{bmatrix} = \begin{bmatrix} 30 \\ 20 \\ 11 \\ 4 \\ 4 \\ 11 \\ 20 \end{bmatrix}.$$

З розрахунків видно, що результати, отримані різними способами, співпали.

Аналогічні результати можна отримати за допомогою наступної програми (з урахуванням властивостей ДПФ):

```
%Приклад №6.1
%початкова послідовність
x=[1,2,3,4]
%розрахунок АКФ
c=circshift(xcorr(x,x),[0,4]) %вбудована функція
%розрахунок з використанням властивості ДПФ
XX=fft(x,7); % ДПФ
cc=fft(XX.*conj(XX))
```

Результати розрахунку різними способами повністю співпадають.

Приклад №6.2. Знайти взаємнокореляційну функцію (ВКФ) двох

послідовностей $x[n]=(3,-2,-1)$ та $y[n]=(3,-1,2,4)$.

Розв'язок.

Довжина ВКФ дорівнює $N = N_1 + N_2 - 1 = 3 + 4 - 1 = 6$. Значить, в послідовність $x[n]$ необхідно додати 3 нулі, а в послідовність $y[n]$ – 2 нулі.

1-й спосіб (безпосереднє обчислення ВКФ)

$$\vec{r}_{xy} = \begin{bmatrix} 3 & -1 & 2 & 4 & 0 & 0 \\ -1 & 2 & 4 & 0 & 0 & 3 \\ 2 & 4 & 0 & 0 & 3 & -1 \\ 4 & 0 & 0 & 3 & -1 & 2 \\ 0 & 0 & 3 & -1 & 2 & 4 \\ 0 & 3 & -1 & 2 & 4 & 0 \end{bmatrix} \times \begin{bmatrix} 3 \\ -2 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ -11 \\ -2 \\ 12 \\ -3 \\ -5 \end{bmatrix}.$$

2-й спосіб (з використанням згортки сигналу з інверсною копією 2-го сигналу).

Важливо, який сигнал необхідно інвертувати. Згідно з виразом (6.2) інвертуємо послідовність, додану нулями $x[n]$.

$$\vec{r}_{xy} = \begin{bmatrix} 3 & -2 & -1 & 0 & 0 & 0 \\ 0 & 3 & -2 & -1 & 0 & 0 \\ 0 & 0 & 3 & -2 & -1 & 0 \\ 0 & 0 & 0 & 3 & -2 & -1 \\ -1 & 0 & 0 & 0 & 3 & -2 \\ -2 & -1 & 0 & 0 & 0 & 3 \end{bmatrix} \times \begin{bmatrix} 3 \\ -1 \\ 2 \\ 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ -11 \\ -2 \\ 12 \\ -3 \\ -5 \end{bmatrix}.$$

Аналогічні результати можна отримати за допомогою наступної програми (з урахуванням властивостей ДПФ):

```

%Приклад №6.2
%початкова послідовність
x=[3,-2,-1]
y=[3,-1,2,4]
%розрахунок ВКФ з використанням властивості ДПФ
XX=fft(x,6);
YY=fft(y,6);
cc=ifft(conj(XX).*YY)

```

Якщо виконати інверсію іншого сигналу, то результат не співпадає, тому що це відповідає взаємній перестановці сигналів, тобто розрахунку $\vec{r}_{yx}$ (згідно з властивостями ВКФ). Нижче наведено результат розрахунку ВКФ при взаємній перестановці сигналів:

$$\vec{r}_{yx} = \begin{bmatrix} 3 & -1 & 2 & 4 & 0 & 0 \\ 0 & 3 & -1 & 2 & 4 & 0 \\ 0 & 0 & 3 & -1 & 2 & 4 \\ 4 & 0 & 0 & 3 & -1 & 2 \\ 2 & 4 & 0 & 0 & 3 & -1 \\ -1 & 2 & 4 & 0 & 0 & 3 \end{bmatrix} \times \begin{bmatrix} 3 \\ -2 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ -5 \\ -3 \\ 12 \\ -2 \\ -11 \end{bmatrix}.$$

Це пояснюється тим, що в частотній області ВКФ відповідає добутку спектру одного сигналу на комплексно зв'язаний спектр іншого сигналу.

6.4. Завдання до розділу 6 для самостійної роботи

Розрахувати ВКФ послідовностей $x[n]$ і $y[n]$ (в часовій області) 2-ма способами. Вхідні дані див. в табл. 5.1.

7. Z-ПЕРЕТВОРЕННЯ

7.1. Пряме z-перетворення

Визначимо z-перетворення дійсне для усіх n :

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad (7.1)$$

Якщо $x[n] = 0$ при $n < 0$, то вираз (7.1) зводиться до так званого одностороннього z-перетворення:

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

Зрозуміло, що z-перетворення – це степеневий ряд з нескінченною кількістю членів, тому він може не збігатися для усіх значень z . Область, в якій z-перетворення збігається, називається областю збіжності, і в цій області значення скінченні. Значення z , для яких $X(z) = \infty$, називаються полюсами функції $X(z)$. Значення z , для яких $X(z) = 0$, називаються нулями функції $X(z)$.

7.2. Зворотне z-перетворення

Зворотне z-перетворення дозволяє відновлювати дискретний сигнал по його z-образу. У символічному виді зворотне z-перетворення можна визначити наступним чином:

$$x[n] = Z^{-1}[X(z)].$$

На практиці $X(z)$ часто виражається через відношення двох багаточленів від z^{-n} , наприклад:

$$X(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}.$$

У цьому виді зворотнє z -перетворення можна знайти за допомогою одного з багатьох методів. Найбільш поширені з них:

- 1) метод розкладання в степеневий ряд;
- 2) метод розкладання на елементарні дробки;
- 3) метод лишків.

7.3. Властивості z -перетворення

Розглянемо властивості z -перетворення.

1) Лінійність.

Якщо послідовності $x_1[n]$ і $x_2[n]$ мають z -образи $X_1(z)$ і $X_2(z)$, то z -образ лінійної комбінації $y[n] = ax_1[n] + by_2[n]$, де a, b – довільні константи, дорівнює $Y(z) = aX_1(z) + bX_2(z)$.

2) Затримка.

Якщо z -образ послідовності $x[n]$ дорівнює $X(z)$, то z -образ послідовності із затримкою на m елементів буде дорівнювати $z^{-m}X(z)$.

3) Згортка.

Нехай $x[n]$ і $y[n]$ мають z -образи $X(z)$ і $Y(z)$ відповідно. За визначенням згортки двох дискретних послідовностей дорівнює

$$g[n] = \sum_{m=0}^{\infty} x[m]y[n-m], \quad n = 0, 1, \dots$$

Тоді z -образ послідовності $g[n]$ дорівнює $G(z) = X(z)Y(z)$.

4) Зв'язок z -перетворення зі спектром сигналу.

Спектр дискретного сигналу легко може бути отриманий з його z -перетворення:

$$X(j\omega) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n T_c} = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \bigg|_{z=e^{j\omega T_c}} = X(z) \bigg|_{z=e^{j\omega T_c}}.$$

Тобто спектр сигналу є значеннями його z -перетворення, узятими в точках $z = e^{j\omega T_c}$, які лежать на колі одиничного радіусу, до того ж параметр ωT_c є полярним кутом точки на колі. Зроблений висновок у черговий раз свідчить, що спектр дискретного сигналу зобов'язаний бути періодичним з періодом $2\pi/T_c$.

7.4. Приклади до розділу 7

Приклад №7.1. Знайти z -перетворення і область збіжності усіх послідовностей дискретного часу, наведених на рис. 7.1.

Розв'язок.

1) Послідовність на рис. 7.1а має скінченну довжину і не дорівнює 0 при $n < 0$. Значення послідовності $x[-6]=0$, $x[-5]=1$, $x[-4]=3$, $x[-3]=5$, $x[-2]=3$, $x[-1]=1$, $x[0]=0$. Тоді z -образ виглядає наступним чином:

$$X_1(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = z^5 + 3z^4 + 5z^3 + 3z^2 + z.$$

Очевидно, що $X(z) = \infty$ при $z = \infty$. Отже, областю збіжності буде уся площина z , окрім $z = \infty$.

2) Послідовність на рис. 7.1б двостороння і має скінченну довжи-

ну. Значення послідовності $x[-3]=0$, $x[-2]=1$, $x[-1]=3$, $x[0]=5$, $x[1]=3$, $x[2]=1$, $x[3]=0$. Тоді z -образ послідовності має наступний вигляд:

$$X_2(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = z^2 + 3z + 5 + 3z^{-1} + z^{-2}.$$

Очевидно, що $X(z) = \infty$ при $z=0$ або $z=\infty$. Отже, в область збіжності не входять тільки точки $z=0$ та $z=\infty$.

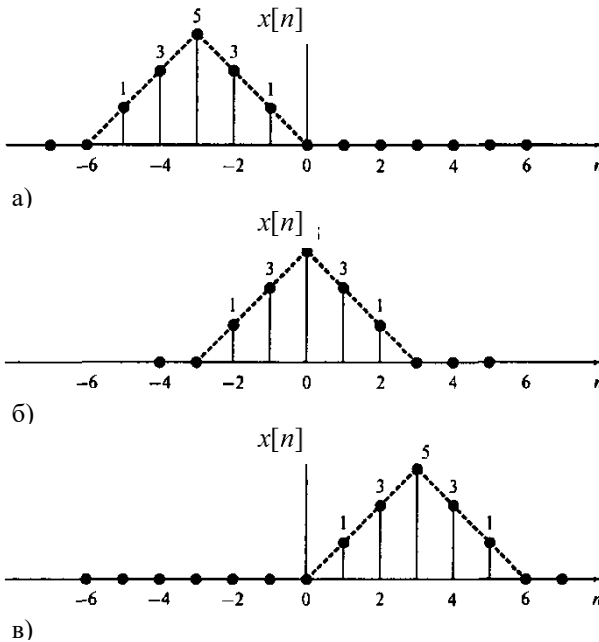


Рисунок 7.1 – Послідовності дискретного часу

3) Послідовність на рис. 7.1в має скінчену довжину і дорівнює 0 при $n < 0$, тобто сигнал каузальний. Величини послідовності

$$x[0] = 0, \quad x[1] = 1, \quad x[2] = 3, \quad x[3] = 5, \quad x[4] = 3, \quad x[5] = 1, \quad x[6] = 0.$$

Тоді z -образ матиме вигляд:

$$X_3(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} = z^{-1} + 3z^{-2} + 5z^{-3} + 3z^{-4} + z^{-5}.$$

У цьому випадку $X(z) = \infty$ при $z = 0$. Отже, в область збіжності не входить тільки точка $z = 0$.

Приклад №7.2. Дано наступний z -образ сигналу:

$$X(z) = \frac{1 + 3z^{-1} + z^{-2}}{1 - z^{-1} - 2z^{-2}}.$$

Необхідно знайти часову послідовність методом розкладання в степеневий ряд.

Розв'язок.

Вирішимо задачу методом степеневих рядів.

Якщо дано z -образ каузального сигналу, то його можна розкласти в нескінченний ряд відносно z^{-1} або z шляхом ділення в стовпчик (іноді таке ділення називають синтетичним діленням). У цьому методі чисельник і знаменник функції $X(z)$ спочатку виражають через зменшуваний показник степені z , або через збільшуваний показник степені z^{-1} , а потім, шляхом ділення в стовпчик, знаходиться частка.

Розглянемо метод на прикладі.

Чисельник і знаменник $X(z)$ задані у вигляді багаточленів зі зростаючим степенем z^{-1} , отже, для отримання оригіналу функції виконаємо звичайне ділення в стовпчик:

$$\begin{array}{r}
 1 + 3z^{-1} + z^{-2} \quad \left| \begin{array}{l} 1 - z^{-1} - 2z^{-2} \\ \hline 1 - z^{-1} - 2z^{-2} \\ \hline 4z^{-1} + 3z^{-2} \\ \hline 4z^{-1} - 4z^{-2} - 8z^{-3} \\ \hline 7z^{-2} + 8z^{-3} \\ \hline 7z^{-2} - 7z^{-3} - 14z^{-4} \\ \hline 15z^{-3} + 14z^{-4} \\ \hline 15z^{-3} - 15z^{-4} - 30z^{-5} \\ \hline 29z^{-4} + 30z^{-5} \\ \hline 29z^{-4} - 29z^{-5} + 58z^{-5} \\ \hline 59z^{-4} - 58z^{-5} \end{array} \right. \\
 1 - z^{-1} - 2z^{-2} \qquad 1 + 4z^{-1} + 7z^{-2} + 15z^{-3} + 29z^{-4} + \dots
 \end{array}$$

Тоді z -образ розкладається в степеневий ряд:

$$X(z) = 1 + 4z^{-1} + 7z^{-2} + 15z^{-3} + 29z^{-4} + \dots$$

Тепер можна безпосередньо записати зворотнє z -перетворення:

$$x[0] = 1,$$

$$x[1] = 4,$$

$$x[2] = 7,$$

$$x[3] = 15,$$

$$x[4] = 29 \dots$$

7.5. Завдання до розділу 7 для самостійної роботи

1) Знайти z -перетворення і область збіжності послідовності $x[n]$.

Вхідні дані див. в табл. 7.1 (N – номер студента за журналом).

Таблиця 7.1 – Варіанти часових послідовностей

N	$x[n]$	n	N	$x[n]$	n
1	$(-2; -1; 1; 2)$	$-1, 0, 1, 2$	16	$(-2; -3; 1; 2)$	$6, 7, 8, 9$
2	$(-2; 1; -1; 2)$	$-2, -1, 0, 1$	17	$(-2; 1; -1; 4)$	$4, 5, 6, 7$
3	$(-2; 1; 2; -1)$	$-5, -4, -3, -2$	18	$(-4; 1; 2; -1)$	$-7, -6, -5, -4$
4	$(-2; -1; 2; 1)$	$0, 1, 2, 3$	19	$(-2; -1; 5; 1)$	$13, 14, 15, 16$
5	$(1; 2; 3; 4)$	$-3, -2, -1, 0$	20	$(1; 2; 3; 4)$	$-9, -8, -7, -6$
6	$(1; -2; 3; -4)$	$1, 2, 3, 4$	21	$(1; -2; 3; -4)$	$8, 9, 10, 11$
7	$(1; -2; -3; 4)$	$-4, -3, -2, -1$	22	$(1; -2; -3; 4)$	$15, 16, 17, 18$
8	$(-3; -1; 0; 2)$	$2, 3, 4, 5$	23	$(-3; -1; 0; 2)$	$11, 12, 13, 14$
9	$(-3; 0; 1; 3)$	$5, 6, 7, 8$	24	$(-3; 0; 1; 3)$	$-11, -10, -9, -8$
10	$(-3; 0; 0; 3)$	$-6, -5, -4, -3$	25	$(-3; 0; 0; 3)$	$9, 10, 11, 12$
11	$(0; 2; 2; 0)$	$10, 11, 12, 13$	26	$(0; 2; 2; 0)$	$-13, -12, -11, -10$
12	$(2; 0; 3; 0)$	$-8, -7, -6, -5$	27	$(2; 0; 3; 0)$	$12, 13, 14, 15$
13	$(2; 0; 3; 0)$	$7, 8, 9, 10$	28	$(2; 0; 3; 0)$	$-12, -11, -10, -9$
14	$(-2; 0; 3; 0)$	$n = 3, 4, 5, 6$	29	$(-2; 0; 3; 0)$	$-14, -13, -12, -11$
15	$(2; 0; -3; 0)$	$-10, -9, -8, -7$	30	$(2; 0; -3; 0)$	$14, 15, 16, 17$

2) Розрахувати зворотнє z -перетворення для заданої функції $X(z)$ методом розкладання в степеневий ряд. Вхідні дані див. в табл. 7.2 (N – номер студента за журналом).

Таблиця 7.2 – Варіанти z-образів послідовностей

N	$X(z)$	N	$X(z)$	N	$X(z)$
1	$\frac{1-3z^{-1}-2z^{-2}}{1-3z^{-1}+2z^{-2}}$	11	$\frac{1-3z^{-1}-2z^{-2}}{1+5z^{-1}+6z^{-2}}$	21	$\frac{1-3z^{-1}-2z^{-2}}{1-3z^{-1}+2z^{-2}}$
2	$\frac{1-3z^{-1}-2z^{-2}}{1+z^{-1}-2z^{-2}}$	12	$\frac{1-3z^{-1}-2z^{-2}}{1-z^{-1}-6z^{-2}}$	22	$\frac{1-3z^{-1}-2z^{-2}}{1+z^{-1}-2z^{-2}}$
3	$\frac{1-3z^{-1}-2z^{-2}}{1-z^{-1}-2z^{-2}}$	13	$\frac{1+3z^{-1}-2z^{-2}}{1-5z^{-1}+6z^{-2}}$	23	$\frac{1-3z^{-1}-2z^{-2}}{1-z^{-1}-2z^{-2}}$
4	$\frac{1-3z^{-1}-2z^{-2}}{1+3z^{-1}+2z^{-2}}$	14	$\frac{1+3z^{-1}-2z^{-2}}{1+z^{-1}-6z^{-2}}$	24	$\frac{1-3z^{-1}-2z^{-2}}{1+3z^{-1}+2z^{-2}}$
5	$\frac{1+3z^{-1}-2z^{-2}}{1-3z^{-1}+2z^{-2}}$	15	$\frac{1+3z^{-1}-2z^{-2}}{1+5z^{-1}+6z^{-2}}$	25	$\frac{1+3z^{-1}-2z^{-2}}{1-3z^{-1}+2z^{-2}}$
6	$\frac{1+3z^{-1}-2z^{-2}}{1+z^{-1}-2z^{-2}}$	16	$\frac{1+3z^{-1}-2z^{-2}}{1-z^{-1}-6z^{-2}}$	26	$\frac{1+3z^{-1}-2z^{-2}}{1+z^{-1}-2z^{-2}}$
7	$\frac{1+3z^{-1}-2z^{-2}}{1-z^{-1}-2z^{-2}}$	17	$\frac{1-3z^{-1}-2z^{-2}}{1+5z^{-1}+6z^{-2}}$	27	$\frac{1+3z^{-1}-2z^{-2}}{1-z^{-1}-2z^{-2}}$
8	$\frac{1+3z^{-1}-2z^{-2}}{1+3z^{-1}+2z^{-2}}$	18	$\frac{1-3z^{-1}-2z^{-2}}{1-z^{-1}-6z^{-2}}$	28	$\frac{1+3z^{-1}-2z^{-2}}{1+3z^{-1}+2z^{-2}}$
9	$\frac{1-3z^{-1}-2z^{-2}}{1-5z^{-1}+6z^{-2}}$	19	$\frac{1+3z^{-1}-2z^{-2}}{1-5z^{-1}+6z^{-2}}$	29	$\frac{1-3z^{-1}-2z^{-2}}{1-5z^{-1}+6z^{-2}}$
10	$\frac{1-3z^{-1}-2z^{-2}}{1+z^{-1}-6z^{-2}}$	20	$\frac{1+3z^{-1}-2z^{-2}}{1+z^{-1}-6z^{-2}}$	30	$\frac{1-3z^{-1}-2z^{-2}}{1+z^{-1}-6z^{-2}}$

8. МЕТОДИ ЗВОРОТНОГО Z-ПЕРЕТВОРЕННЯ

8.1. Метод розкладання на елементарні дробі

У цьому методі z -образ спочатку розкладається на суму простих дробів. Потім за таблицями знаходиться зворотне z -перетворення кожного елементарного дробу. Ці образи підсумовуються, і виходить загальне зворотне z -перетворення.

Нехай z -образ заданий як відношення багаточленів виду:

$$X(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}. \quad (8.1)$$

Якщо полюси функції $X(z)$ – першого порядку і $N = M$, то $X(z)$ можна розкласти наступним чином:

$$\begin{aligned} X(z) &= B_0 + C_1 \frac{z}{z - p_1} + C_2 \frac{z}{z - p_2} + \dots + C_M \frac{z}{z - p_M} = \\ &= B_0 + \sum_{k=1}^M C_k \frac{z}{z - p_k}, \end{aligned} \quad (8.2)$$

де p_k – полюси функції $X(z)$; C_k – коефіцієнти елементарних дробів (лишки функції C_k), а коефіцієнт B_0 знаходиться так:

$$B_0 = \frac{b_N}{a_M}. \quad (8.3)$$

Якщо у виразі (8.1) $N < M$, то $B_0 = 0$. Якщо $N > M$, то $X(z)$ спочатку треба скоротити шляхом ділення в стовпчик багаточленів чисельника і знаменника, щоб вийшло $N \leq M$. Залишок можна знайти

за виразом (8.2).

Коефіцієнт C_k , пов'язаний з полюсом p_k , можна знайти, помноживши праву та ліву частину рівняння (8.2) на $\frac{(z-p_k)}{z}$, а потім зробивши заміну $z = p_k$:

$$C_k = \frac{X(z)}{z} (z - p_k) \Big|_{z=p_k}. \quad (8.4)$$

Якщо функція $X(z)$ має один або декілька полюсів більше 1-го порядку (тобто співпадаючих полюсів), то в рівнянні (8.2) треба додати додаткові члени.

Наприклад, якщо функція $X(z)$ має полюс m -го порядку в точці $z = p_k$, то в розкладанні на елементарні дробі повинні входити члени виду:

$$\sum_{i=1}^m \frac{D_i z}{(z - p_k)^i}. \quad (8.5)$$

Коефіцієнти D_i обчислюються за виразом:

$$D_i = \frac{1}{(m-i)!} \frac{d^{m-i}}{dz^{m-i}} \left[(z - p_k)^m \frac{X(z)}{z} \right]_{z=p_k}. \quad (8.6)$$

За таблицями можна знайти зворотне z -перетворення кожного елементарного дробу. У табл. 8.1 приведені деякі табличні функції.

Таблиця 8.1 – Приклади z-зображень деяких поширених послідовностей

Послідовність дискретного часу $x[n]$ для $n \geq 0$	z-образ $X(z)$	Область збіжності $X(z)$
$k\delta[n]$	K	всюди
k	$\frac{kz}{z-1}$	$ z > 1$
kn	$\frac{kz}{(z-1)^2}$	$ z > 1$
kn^2	$\frac{kz(z+1)}{(z-1)^3}$	$ z > 1$
ke^{-an}	$\frac{kz}{z-e^{-a}}$	$ z > e^{-a}$
kne^{-an}	$\frac{kze^{-a}}{(z-e^{-a})^2}$	$ z > e^{-a}$
ka^n	$\frac{kz}{z-a}$	$ z > a$
kna^n	$\frac{kaz}{(z-a)^2}$	$ z > a$

8.2. Метод лишків

У цьому методі зворотне z-перетворення знаходиться шляхом обчислення контурного інтеграла

$$x[n] = \frac{1}{2\pi j} \oint_C z^{n-1} X(z) dz ,$$

де C – контур інтегрування, що охоплює усі полюси функції $X(z)$.

Для раціональних багаточленів контурний інтеграл знаходиться згідно теореми Коши про лишки (або теорема лишків):

$$x[n] = \frac{1}{2\pi j} \oint_C z^{n-1} X(z) dz = \sum_{k=1}^M \operatorname{Res}[z^{n-1} X(z), p_k] = \sum_{k=1}^M \operatorname{Res}[F(z), p_k], \quad (8.7)$$

$$\operatorname{Res}[F(z), p_k] = (z - p_k) F(z) \Big|_{z=p_k} = (z - p_k) z^{n-1} X(z) \Big|_{z=p_k}, \quad (8.8)$$

$$F(z) = z^{n-1} X(z), \quad (8.9)$$

де $\operatorname{Res}[F(z), p_k]$ – лишки функції $F(z)$ в точці $z = p_k$; p_k – полюси функції $F(z)$.

Вираз (8.8) справедливий для простих полюсів.

Якщо функція $F(z)$ має полюси вищого першого порядку, тоді вираз (8.8) матиме вигляд:

$$\operatorname{Res}[F(z), p_k] = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} \left[(z - p_k)^m F(z) \right]_{z=p_k}.$$

де m – порядок полюса.

8.3. Порівняння методів зворотного z-перетворення

Обмеження методу розкладання в степеневий ряд полягає в тому, що він не дає рішення в аналітичному вигляді, але при цьому він простий і придатний для обчислень за допомогою комп'ютера. Проте через його рекурсивну природу треба уважно стежити за можливим наростанням числових помилок при великій кількості заданих точок зворотного z-перетворення.

Як метод розкладання на елементарні дробі, так і метод лишків

дають результат в аналітичному виді. Головний недолік – необхідність розкласти на множники багаточлен знаменника, тобто знаходити полюси функції $X(z)$. Якщо порядок функції $X(z)$ високий, то пошук полюсів – досить важка задача. Крім того, якщо функція $X(z)$ має полюси високого порядку, обидва методи можуть включати диференціювання високого порядку.

Очевидно, якщо треба знайти рішення в аналітичному виді, то краще вибрати метод лишків або розкладання на елементарні дробі. Метод розкладання на елементарні дробі особливо корисний при синтезі цифрових фільтрів.

8.4. Приклади до розділу 8

Приклад №8.1. Дано наступне z -зображення сигналу:

$$X(z) = \frac{1 + 3z^{-1} + z^{-2}}{1 - z^{-1} - 2z^{-2}}.$$

Необхідно знайти часову послідовність методом розкладання на елементарні дробі.

Розв'язок.

Для простоти виразимо z -образ через позитивні показники степеню z , помноживши чисельник і знаменник на z^2 :

$$X(z) = \frac{z^2 + 3z + 1}{z^2 - z - 2}.$$

Знайдемо полюси функції $X(z)$. Для цього знаменник прирівняємо до нуля і знайдемо корені рівняння: $z^2 - z - 2 = 0$. Корені квадратного рівняння обчислюються за наступним виразом:

$$p_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

$$\text{Отже, } p_1 = \frac{1 + \sqrt{1+8}}{2} = 2, \quad p_2 = \frac{1 - \sqrt{1+8}}{2} = -1.$$

Функція $X(z)$ має 2 полюси першого порядку в точках $z = 2$ та $z = -1$.

Тоді функцію $X(z)$ можна записати в наступному вигляді

$$X(z) = \frac{z^2 + 3z + 1}{(z - 2)(z + 1)}.$$

Згідно з виразом (8.3) $B_0 = 1/(-2) = -0.5$.

Знайдемо лишки функції $X(z)$ за виразом (8.4):

$$\begin{aligned} C_1 &= \left. \frac{X(z)}{z} (z - p_1) \right|_{z=p_1} = \left. \frac{z^2 + 3z + 1}{z(z - 2)(z + 1)} (z - 2) \right|_{z=2} = \\ &= \left. \frac{z^2 + 3z + 1}{z(z + 1)} \right|_{z=2} = \frac{4 + 6 + 1}{2(2 + 1)} = \frac{11}{6}; \end{aligned}$$

$$\begin{aligned} C_2 &= \left. \frac{X(z)}{z} (z - p_2) \right|_{z=p_2} = \left. \frac{z^2 + 3z + 1}{z(z - 2)(z + 1)} (z + 1) \right|_{z=-1} = \\ &= \left. \frac{z^2 + 3z + 1}{z(z - 2)} \right|_{z=-1} = \frac{1 - 3 + 1}{-1(-1 - 2)} = -\frac{1}{3}. \end{aligned}$$

Отже згідно з виразом (8.2) функція $X(z)$ має вигляд:

$$X(z) = B_0 + C_1 \frac{z}{z - p_1} + C_2 \frac{z}{z - p_2} = -0.5 + \frac{11}{6} \frac{z}{z - 2} - \frac{1}{3} \frac{z}{z + 1}.$$

Згідно табл. 8.1 маємо:

$$Z^{-1}[-0.5] = -0.5\delta[n];$$

$$Z^{-1}\left[\frac{11}{6} \frac{z}{z-2}\right] = \frac{11}{6} 2^n;$$

$$Z^{-1}\left[-\frac{1}{3} \frac{z}{z+1}\right] = -\frac{1}{3}(-1)^n.$$

Отже, оригінал функції має вигляд

$$x[n] = -0.5\delta[n] + \frac{11}{6} 2^n - \frac{1}{3}(-1)^n, \quad n \geq 0.$$

Порівняємо отриманий результат з результатом, отриманим методом розкладання в степеневий ряд (див. приклад №2 підрозділу 7.4). Обчислимо декілька перших значень функції $x[n]$:

$$x[0] = -0.5 + \frac{11}{6} - \frac{1}{3} = -0.5 + 1.5 = 1;$$

$$x[1] = \frac{11}{6} 2 - \frac{1}{3}(-1) = \frac{12}{3} = 4;$$

$$x[2] = \frac{11}{6} 2^2 - \frac{1}{3}(-1)^2 = \frac{21}{3} = 7;$$

$$x[3] = \frac{11}{6} 2^3 - \frac{1}{3}(-1)^3 = \frac{45}{3} = 15.$$

Приклад №8.2. Дано наступний z -образ сигналу:

$$X(z) = \frac{1 + 3z^{-1} + z^{-2}}{1 - z^{-1} - 2z^{-2}}.$$

Необхідно знайти часову послідовність методом лишків.

Розв'язок.

У попередньому прикладі ми показали, що початкова функція може бути представлена в наступному вигляді:

$$X(z) = \frac{z^2 + 3z + 1}{(z - 2)(z + 1)}.$$

Тоді згідно з виразом (8.7) функція $F(z)$ дорівнює:

$$F(z) = \frac{z^{n-1}(z^2 + 3z + 1)}{(z - 2)(z + 1)} = \frac{z^n(z^2 + 3z + 1)}{z(z - 2)(z + 1)}.$$

Полюси функції $F(z)$ знаходяться в точках $z = 2$, $z = -1$ (як і у функції $X(z)$) і при $n = 0$ в точці $z = 0$. Полюса в точці $z = 0$ при $n > 0$ не існують, тому необхідно розглянути 2 окремі випадки.

При $n = 0$ функція $F(z)$ зводиться до наступного виразу:

$$F(z) = \frac{z^{n-1}(z^2 + 3z + 1)}{(z - 2)(z + 1)} = \frac{(z^2 + 3z + 1)}{z(z - 2)(z + 1)}.$$

$$\text{Тоді } x[0] = \text{Res}[F(z), 0] + \text{Res}[F(z), 2] + \text{Res}[F(z), -1].$$

Отже,

$$\begin{aligned} \text{Res}[F(z), 0] &= zF(z) \Big|_{z=0} = \frac{z(z^2 + 3z + 1)}{z(z - 2)(z + 1)} \Big|_{z=0} = \\ &= \frac{(z^2 + 3z + 1)}{(z - 2)(z + 1)} \Big|_{z=0} = \frac{1}{-2} = -0.5; \end{aligned}$$

$$\begin{aligned}\operatorname{Res}[F(z), 2] &= (z-2)F(z)\Big|_{z=2} = \frac{(z-2)(z^2+3z+1)}{z(z-2)(z+1)}\Big|_{z=2} = \\ &= \frac{(z^2+3z+1)}{z(z+1)}\Big|_{z=2} = \frac{4+6+1}{2(2+1)} = \frac{11}{6};\end{aligned}$$

$$\begin{aligned}\operatorname{Res}[F(z), -1] &= (z+1)F(z)\Big|_{z=-1} = \frac{(z+1)(z^2+3z+1)}{z(z-2)(z+1)}\Big|_{z=-1} = \\ &= \frac{(z^2+3z+1)}{z(z-2)}\Big|_{z=-1} = \frac{1-3+1}{-1(-1-2)} = -\frac{1}{3};\end{aligned}$$

Таким чином, $\chi[0] = -0.5 + \frac{11}{6} - \frac{1}{3} = -0.5 + 1.5 = 1$.

При $n > 0$ полюс в точці $z = 0$ зникає, і функція $F(z)$ має вигляд

$$F(z) = \frac{z^n(z^2+3z+1)}{z(z-2)(z+1)}.$$

Тоді $\chi[n] = \operatorname{Res}[F(z), 2] + \operatorname{Res}[F(z), -1]$.

Знайдемо лишки:

$$\begin{aligned}\operatorname{Res}[F(z), 2] &= (z-2)F(z)\Big|_{z=2} = \frac{(z-2)z^n(z^2+3z+1)}{z(z-2)(z+1)}\Big|_{z=2} = \\ &= \frac{z^n(z^2+3z+1)}{z(z+1)}\Big|_{z=2} = \frac{2^n(4+6+1)}{2(2+1)} = \frac{11}{6}2^n;\end{aligned}$$

$$\begin{aligned}\operatorname{Res}[F(z), -1] &= (z+1)F(z)\Big|_{z=-1} = \frac{(z+1)z^n(z^2+3z+1)}{z(z-2)(z+1)}\Big|_{z=-1} = \\ &= \frac{z^n(z^2+3z+1)}{z(z-2)}\Big|_{z=-1} = \frac{(-1)^n(1-3+1)}{-1(-1-2)} = -\frac{1}{3}(-1)^n.\end{aligned}$$

Тому $x[n] = \frac{11}{6} 2^n - \frac{1}{3} (-1)^n$ для $n > 0$.

Отже, для $n \geq 0$ оригінал функції дорівнює $x[n] = -0.5\delta[n] + \frac{11}{6} 2^n - \frac{1}{3} (-1)^n$. Результат повністю співпадає з результатом, отриманим попередніми методами.

Приклад №8.3. Знайти дискретну в часі послідовність $x[n]$ з наступним z -образом:

$$X(z) = \frac{z^2}{(z - 0.5)(z - 1)^2}.$$

Функція $X(z)$ містить полюс 2-го порядку.

Розв'язок.

1) Розглянемо метод розкладання на елементарні дробі.

У функції є полюс 1-го порядку в точці $z = 0.5$ і полюс 2-го порядку в точці $z = 1$. В цьому випадку з урахуванням додаткових членів (8.5) розкладання на елементарні дробі виглядатиме наступним чином:

$$X(z) = \frac{Cz}{z - 0.5} + \frac{D_1 z}{z - 1} + \frac{D_2 z}{(z - 1)^2}.$$

Для розрахунку коефіцієнта діятимемо так само, як і раніше, використовуючи вираз (8.4):

$$C = \frac{X(z)}{z} (z - 0.5) \Big|_{z=0.5} = \frac{z^2 (z - 0.5)}{z (z - 0.5) (z - 1)^2} \Big|_{z=0.5} = \frac{0.5^2}{0.5(0.5 - 1)^2} = 2.$$

Щоб знайти D_1 , скористаємося рівнянням (8.6) при $i = 1$ і $m = 2$.

Отже,

$$\begin{aligned} D_1 &= \frac{d}{dz} \left[\frac{(z-1)^2 X(z)}{z} \right]_{z=1} = \frac{d}{dz} \left[\frac{(z-1)^2 z^2}{z(z-0.5)(z-1)^2} \right]_{z=1} = \\ &= \frac{d}{dz} \left(\frac{z}{z-0.5} \right)_{z=1}. \end{aligned}$$

Оскільки $\left[\frac{U(x)}{V(x)} \right]' = \frac{U'(x)V(x) - V'(x)U(x)}{V^2(x)}$, тоді

$$D_1 = \frac{d}{dz} \left(\frac{z}{z-0.5} \right)_{z=1} = \frac{z-0.5-z}{(z-0.5)^2} \Big|_{z=1} = \frac{-0.5}{(1-0.5)^2} = -2.$$

Аналогічно D_2 знаходиться з рівняння (8.6) при підстановці $i = 2$ і $m = 2$:

$$\begin{aligned} D_2 &= \frac{(z-1)^2 X(z)}{z} \Big|_{z=1} = \frac{(z-1)^2 z^2}{z(z-0.5)(z-1)^2} \Big|_{z=1} = \frac{z}{z-0.5} \Big|_{z=1} = \\ &= \frac{1}{1-0.5} = 2. \end{aligned}$$

Об'єднавши результати, отримаємо:

$$X(z) = \frac{2z}{z-0.5} - \frac{2z}{z-1} + \frac{2z}{(z-1)^2}.$$

Зворотне z -перетворення кожного члена з правої частини рівняння можна знайти за табл. 8.1, а потім підсумувати їх. Тоді,

$$x[n] = 2(0.5)^n - 2 + 2n = 2(n-1+0.5^n), \quad n \geq 0.$$

2) Розглянемо метод лишків.

Згідно з методом лишків послідовність дискретного часу задається як

$$x[n] = \sum_{k=1}^M \text{Res}[F(z), p_k],$$

$$\text{де } F(z) = z^{n-1} X(z) = z^{n-1} \frac{z^2}{(z-0.5)(z-1)^2} = \frac{z^{n+1}}{(z-0.5)(z-1)^2}.$$

Функція $F(z)$ має простий полюс в точці $z = 0.5$ і полюс 2-го порядку в точці $z = 1$. Отже, послідовність $x[n]$ задається як

$$x[n] = \text{Res}[F(z), p_1] + \text{Res}[F(z), p_2].$$

$$\begin{aligned} \text{Res}[F(z), 0.5] &= \left. \frac{(z-0.5)z^{n+1}}{(z-0.5)(z-1)^2} \right|_{z=0.5} = \left. \frac{z^{n+1}}{(z-1)^2} \right|_{z=0.5} = \\ &= \frac{0.5^{n+1}}{(0.5-1)^2} = 2(0.5)^n. \end{aligned}$$

$$\begin{aligned} \text{Res}[F(z), 1] &= \frac{d}{dz} \left[\frac{(z-1)^2 z^{n+1}}{(z-0.5)(z-1)^2} \right]_{z=1} = \left. \frac{(z-0.5)(n+1)z^n - z^{n+1}}{(z-0.5)^2} \right|_{z=1} = \\ &= \frac{(1-0.5)(n+1)-1}{(1-0.5)^2} = \frac{0.5(n+1)-1}{0.5^2} = \frac{0.5(n-1)}{0.5^2} = 2(n-1). \end{aligned}$$

Об'єднавши результати, отримаємо:

$$x[n] = 2(0.5)^n + 2(n-1) = 2(n-1+0.5^n), \quad n \geq 0,$$

що аналогічно результату, отриманому методом розкладання на елементарні дроби.

8.5. Завдання до розділу 8 для самостійної роботи

Вичислити зворотнє z-перетворення для заданої функції $X(z)$:

1) методом розкладання на елементарні дроби (обов'язкове завдання);

2) методом лишків (завдання на додаткові бали).

Вхідні дані див. в табл. 7.2.

9. ПРОЕКТУВАННЯ ФІЛЬТРІВ ЗІ СКІНЧЕННОЮ ІМПУЛЬСНОЮ ХАРАКТЕРИСТИКОЮ МЕТОДОМ ЗВАЖУВАННЯ

9.1. Введення в цифрові фільтри

По суті, фільтр – це система або ланцюг, що вибірково змінює форму сигналу (амплітудно-частотну або фазо-частотну характеристику). Основними цілями фільтрації є поліпшення якості сигналу (наприклад, усунення або зниження перешкод), добування із сигналів інформації або розподіл декількох сигналів, об'єднаних раніше для, наприклад, ефективного використання доступного каналу зв'язку.

Важливу роль у цифровій обробці сигналів відіграють цифрові фільтри (ЦФ). Наприклад, за допомогою ЦФ можуть виконуватись стиснення даних, біомедична обробка сигналів, обробка мовлення, обробка зображень, передача даних, цифрове аудіо, телефонне ехозаглушення [5].

Проектування цифрового фільтра містить у собі 5 етапів.

1) *Специфікація фільтра*. Задається тип фільтра, (наприклад, фільтр нижніх частот), амплітудна і/або фазова характеристика та дозволени допуски (якщо вони є), частота дискретизації й довжина слів, якими будуть подані вхідні дані.

2) *Обчислення коефіцієнтів*. Визначаються коефіцієнти передавальної функції $H(z)$, яка задовольняє специфікаціям.

3) *Вибір структури*. Виконується перетворення передавальної функції в придатну фільтруючу структуру або мережу.

4) *Аналіз наслідків скінченної розрядності*. Оцінюється вплив квантування на коефіцієнти фільтра та вхідні дані, а також вплив на продуктивність фільтра операції фільтрації зі словами скінченної довжини.

5) *Втілення*. Розробляється програмний код і/або апаратний блок і виконується власне фільтрація.

Названі п'ять етапів не завжди незалежні; крім того, вони не завжди розташовуються в указаному порядку. Фактично існують методи, які дозволяють об'єднати другий етап і деякі аспекти третього й четвертого. Щоб отримати ефективний фільтр, іноді доводиться проводити цей процес у кілька ітерацій, особливо, якщо специфікації не є зовсім визначеними (як зазвичай і буває), або ж розробник бажає досліджувати альтернативні структури [5].

Специфікація вимог включає специфікації:

1) характеристик сигналів (тобто джерела й одержувача сигналу, інтерфейс вводу-виводу, швидкість передачі даних і ширина смуги, найвища частота, що має практичний інтерес);

2) характеристик фільтра (бажана амплітудна і/або фазова характеристика й те, наскільки строгі ці вимоги, швидкість роботи та режими фільтрації (реальний або модельний час));

3) принципу реалізації (наприклад, у вигляді комп'ютерної програми мовою високого рівня або системи цифрової обробки сигналів на базі процесора, тут же виконується вибір процесора сигналів);

4) інших вимог до структури (наприклад, вартість фільтра). Розробник може не мати достатньо інформації, щоб повністю визначити фільтр на початкових етапах, але для спрощення процесу розробки варто сформулювати максимальну кількість вимог.

Хоча перераховані вище вимоги уточнюються залежно від конкретної задачі, деяким аспектам специфікації характеристик фільтра необхідно приділити особливу увагу.

Характеристики ЦФ часто задаються в частотній області [5]. Для частотно-вибіркових фільтрів, таких як фільтри нижніх частот (ФНЧ), фільтри високих частот (ФВЧ), смугові фільтри (СФ) або режекторні фільтри (РФ), специфікації часто формуються у вигляді схем допусків. Приклад подібних схем для ЦФ наведено на рис. 9.1 (для ФНЧ) і рис. 9.2 (для СФ). Горизонтальні лінії позначають межі допустимих відхилень. У смузі пропускання амплітудна характеристика має пікове відхилення δ_n , а в смузі заглушення – максимальне відхилення δ_s .

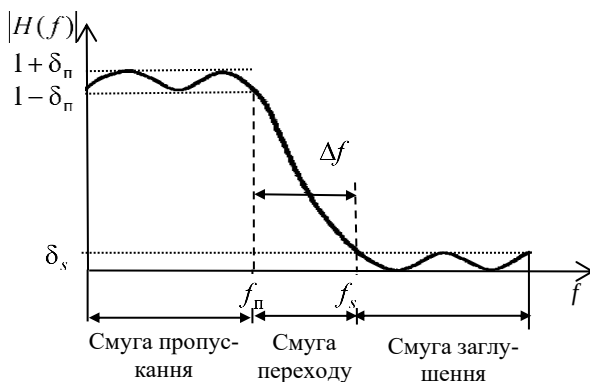


Рисунок 9.1 – Схема допусків для фільтра нижніх частот

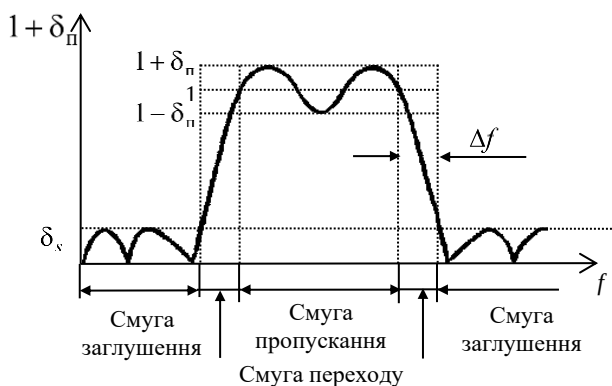


Рисунок 9.2 – Схема допусків для смугового фільтра

Ширина смуги переходу визначає, наскільки різкою є характеристика фільтра. У цій області амплітудна характеристика монотонно зменшується від смуги пропускання до смуги заглушення. Інтерес становлять такі ключові параметри:

$\delta_{\text{п}}$ – відхилення в смузі пропускання;

$\delta_{\text{с}}$ – відхилення в смузі заглушення;

f_{Π} – гранична частота смуги пропускання;

f_s – гранична частота смуги заглушення.

Граничні частоти іноді нормують до частоти дискретизації f / f_c . Але частіше в специфікації використовуються стандартні одиниці частоти, тобто Гц або кГц. Відхилення в смузі пропускання визначають нерівномірність у смузі пропускання, а відхилення в смузі заглушення – загасання в смузі заглушення. Відхилення в смузі пропускання і смузі заглушення можуть виражатися як звичайними числами, так і в децибелах. Отже, мінімальне загасання в смузі заглушення A_s і максимальна нерівномірність у смузі пропускання A_{Π} в децибелах для фільтрів з скінченною імпульсною характеристикою (СІХ-фільтрів) записуються так:

$$A_s = -20 \lg \delta_s; \quad (9.1)$$

$$A_{\Pi} = 20 \lg(1 + \delta_{\Pi}). \quad (9.2)$$

Величина $\Delta f = f_s - f_{\Pi}$ називається смугою переходу фільтра. Ще один важливий параметр – це порядок фільтра N , який визначає кількість коефіцієнтів фільтра.

При специфікації фазової характеристики цифрових фільтрів у багатьох випадках достатньо вказати, яке фазове спотворення суттєве або яка бажана лінійна фазова характеристика. У той же час, у деяких пристроях, де фільтри використовуються для вирівнювання або компенсації фазової характеристики системи або як фазообертачі, фазову характеристику задавати потрібно.

Оскільки у СІХ-фільтрів ФЧХ завжди лінійна, то в специфікації вимог для нерекурсивних фільтрів вимоги до ФЧХ не вказуються.

У більшості випадків указані параметри повністю визначають частотну характеристику СІХ-фільтра.

9.2. Обчислення коефіцієнтів методом зважування

Розглянемо фільтр нижніх частот. Оскільки частотна характеристика будь-якого ЦФ є періодичною функцією, то імпульсну характеристику (ІХ) фільтра можна обчислити за допомогою зворотного перетворення Фур'є заданої частотної характеристики:

$$H_3(\omega) = \begin{cases} 1, & |\omega| < \omega_{\pi}, \\ 0, & \omega_{\pi} < |\omega| < \pi/T_c. \end{cases}$$

Використовуючи зворотне перетворення Фур'є дискретних сигналів, визначимо ІХ фільтра: $h_3[n] = f_{\pi}^{\pi} \text{sinc}(f_{\pi}^{\pi} \pi n)$, де $f_N = 1/2T_c$ – частота Найквіста; $f_{\pi}^{\pi} = f_{\pi} / f_N$ – нормована частота зрізу.

Аналогічно можна отримати ІХ фільтрів верхніх частот, смугових і режкторних фільтрів. Для їхнього розрахунку можна скористатися такими фактами:

$$H_3^{\text{ФВЧ}}(\omega) = 1 - H_3^{\text{ФНЧ}}(\omega) \Rightarrow h_3^{\text{ФВЧ}}[n] = \delta[n] - h_3^{\text{ФНЧ}}[n];$$

$$H_3^{\text{СФ}}(\omega) = H_3^{\text{ФНЧ2}}(\omega) - H_3^{\text{ФНЧ1}}(\omega) \Rightarrow h_3^{\text{СФ}}[n] = h_3^{\text{ФНЧ2}}[n] - h_3^{\text{ФНЧ1}}[n];$$

$$H_3^{\text{РФ}}(\omega) = 1 - H_3^{\text{СФ}}(\omega) \Rightarrow h_3^{\text{РФ}}[n] = \delta[n] - h_3^{\text{СФ}}[n].$$

У табл. 9.1 наведено ідеальні ІХ всіх 4-х типів фільтрів.

Ідеальні ІХ визначені для всіх цілих значень n і є нескінченними, тому такі ЦФ фізично не можуть бути реалізовані. Для фізичної реалізації ЦФ необхідно, щоб ІХ мала каузальний характер, тому необхідно ІХ затримати на K відліків і обмежити значення n , тобто

$$h[n] = h_3[n - M], \quad 0 \leq n \leq N - 1.$$

Таблиця 9.1 – Ідеальні ІХ стандартних частотно-вибіркових фільтрів

рів (для нормованих частот)

Тип фільтра	$h_3[n]$	$h_3[0]$
ФНЧ	$f_n^H \operatorname{sinc}(f_n^H \pi n)$	f_n^H
ФВЧ	$\delta[n] - f_n^H \operatorname{sinc}(f_n^H \pi n)$	$1 - f_n^H$
СФ	$f_2^H \operatorname{sinc}(f_2^H \pi n) - f_1^H \operatorname{sinc}(f_1^H \pi n)$	$f_2^H - f_1^H$
РФ	$\delta[n] + f_1^H \operatorname{sinc}(f_1^H \pi n) - f_2^H \operatorname{sinc}(f_2^H \pi n)$	$1 + f_1^H - f_2^H$

Фактично обмеження IX рівносильне накладенню на безперервну IX прямокутного вікна [2, 5, 6]:

$$h[n] = h_3[n - M]w[n], \quad 0 \leq n \leq N - 1,$$

де $w[n]$ – прямокутне вагове вікно довжиною N відліків.

З метою забезпечення лінійності ФЧХ вагові вікна повинні бути симетричними $w[n] = w[N - k - 1]$ і визначені для $0 \leq n \leq N - 1$.

Синтезовані таким чином СІХ-фільтри мають значні пульсації в районі частоти зрізу. Ці пульсації називаються ефектом Гібса, які пов'язані з пульсуючою поведінкою частотної характеристики прямокутного вагового вікна. Для зменшення негативного впливу ефекту Гібса застосовують вагові вікна, відмінні від прямокутного [2, 5–8].

Основні характеристики найпоширеніших вагових функцій подано в табл. 9.2. Відзначимо, що перші чотири функції мають фіксовані характеристики, такі як ширина переходу і загасання в смузі заглушення. Отже, їхнє використання обмежує свободу розробника.

Відзначимо також, що фільтр, побудований за допомогою методу зважування, має рівні нерівномірності в смузі пропускання і смузі заглушення, тобто $\delta_n = \delta_s$ (рис. 9.1 і 9.2). На практиці це обмеження може дати фільтр, нерівномірність якого в смузі пропускання буде надмірно малою.

Таблиця 9.2 – Характеристики поширених вагових функцій

Функція	Δf (нормована на f_N)	A_n , дБ	$\gamma_{\max}$, дБ	A_s , дБ	Формула для $0 \leq n \leq N-1$
Прямокутна	$1,8 / N$	0,7416	-13	21	1
Ханна	$6,2 / N$	0,0546	-31,5	44	$0,5 - 0,5 \cos(\frac{2\pi n}{N-1})$
Хеммінга	$6,6 / N$	0,0194	-42	53	$0,54 - 0,46 \cos(\frac{2\pi n}{N-1})$
Блекмана	$11 / N$	0,0017	-58	75	$0,42 - 0,5 \cos(\frac{2\pi n}{N-1}) + 0,08 \cos(\frac{4\pi n}{N-1})$
Кайзера $\alpha = 4,54$	$5,86 / N$	0,0274	-35	50	$\frac{I_0 \left[\alpha \sqrt{1 - \left(\frac{2n}{N-1} \right)^2} \right]}{I_0(\alpha)}$
Кайзера $\alpha = 6,76$	$8,64 / N$	0,00275	-50,5	70	
Кайзера $\alpha = 8,96$	$11,42 / N$	0,000275	-65	90	

Вікно Кайзера трохи згладжує описані вище проблеми, оскільки має параметр, керуючий нерівномірністю, що дозволяє розробнику грати на компромісах між шириною переходу й нерівномірністю. Для вікна Кайзера є емпірична формула для знаходження параметра α :

$$\alpha = \begin{cases} 0, & A \leq 21, \\ 0,5842(A-21)^{0,4} + 0,07886(A-21), & 21 < A < 50, \\ 0,1102(A-8,7), & A \geq 50, \end{cases} \quad (9.3)$$

де $A = -20 \lg(\delta)$ – загасання в смузі заглушення, $\delta = \min(\delta_n, \delta_s)$.

Кількість коефіцієнтів фільтра N підпорядковується залежності

$$N \geq \frac{A - 7,95}{7,18\Delta f}, \quad (9.4)$$

де Δf – нормована на частоту Найквіста ширина смуги переходу.

З табл. 9.2 видно, що використання вагових функцій дозволяє досягти заглушення бічних пелюсток, що перевищують 90 дБ. Однак необхідно пам'ятати про те, що зменшення рівня бічних пелюсток неминуче призводить до розширення перехідної зони між смугами пропускання та затримування. Тому потрібно вибирати вагову функцію виходячи з вимог, що ставлять до параметрів фільтра.

Переваги методу: не потребує складних обчислень; точність апроксимації підвищується збільшенням порядку фільтра; дозволяє отримувати аналітичні вирази для визначення коефіцієнтів фільтрів.

Основним недоліком методу є складність визначення граничних частот і нерівномірний розподіл помилок апроксимації за частотним діапазоном.

9.3. Приклади до розділу 9

Приклад №9.1. За допомогою методу зважування визначити вагову функцію та її параметри (якщо вони є) й обчислити мінімальний порядок СІХ-фільтра нижніх частот, який задовольняє таким специфікаціям:

гранична частота смуги пропускання	10 кГц;
ширина смуги переходу	1 кГц;
загасання в смузі заглушення	>25 дБ;
частота дискретизації	60 кГц.

Розв'язок.

Частота Найквіста $f_N = f_c / 2 = 60 / 2 = 30$ кГц. Нормована шири-

на смуги переходу $\Delta f = 1/30$. З табл. 9.2 видно, що вимозі до загасання в смузі заглушення задовольняють функції Ханна, Хеммінга, Блекмана і Кайзера.

Для вікна Ханна $N \geq \frac{6,2}{\Delta f} = \frac{6,2}{1/30} = 186$, для вікна Хеммінга $N \geq \frac{6,6}{\Delta f} = \frac{6,6}{1/30} = 198$, для вікна Блекмана $N \geq \frac{11}{\Delta f} = \frac{11}{1/30} = 330$ і для вікна Кайзера згідно з (9.4) $N \geq \frac{A-7,95}{7,18\Delta f} = \frac{25-7,5}{7,18 \cdot 1/30} = 73$. Таким чином, вікно Кайзера дозволяє спроектувати фільтр із мінімальним порядком методом зважування. Отже, $N = 73$.

Оскільки $A_s = 25$ дБ і

$$\alpha = \begin{cases} 0, & A \leq 21, \\ 0,5842(A-21)^{0,4} + 0,07886(A-21), & 21 < A < 50, \\ 0,1102(A-8,7), & A \geq 50, \end{cases}$$

то $\alpha = 0,5842(25-21)^{0,4} + 0,07886(25-21) = 1,33$.

Приклад №9.2. За допомогою методу зважування визначити вагову функцію та її параметри (якщо вони є) й обчислити мінімальний порядок смугового СІХ-фільтра, який задовольняє таким специфікаціям:

смуга пропускання	150–250 Гц;
ширина смуги переходу	50 Гц;
нерівномірність у смузі пропускання	0,1 дБ;
загасання в смузі заглушення	>60 дБ;
частота дискретизації	1 кГц.

Розв'язок.

Частота Найквіста $f_N = f_c / 2 = 1000 / 2 = 500$ Гц. Нормована ши-

рина смуги переходу $\Delta f = 50/500 = 0.1$.

Вимоги до загасання можна задовольнити тільки при використанні функції Блекмана або Кайзера (див. табл. 9.2). Для функції Блекмана

$$N \geq \frac{11}{\Delta f} = \frac{11}{0.1} = 110 \text{ Для функції Кайзера згідно з (9.4) кількість коефі-}$$

цієнтів дорівнює $N \geq \frac{60 - 7,95}{7,18 \cdot 0.1} = 72,49$. Таким чином, вікно Кайзера

дозволяє спроектувати фільтр із мінімальним порядком методом зважування. Отже, $N = 73$. Для знаходження параметра нерівномірності α згідно з виразами (9.1) і (9.2) знайдемо нерівномірність у смузі пропускання та загасання в смузі заглушення:

$$A_s = -20 \lg \delta_s = 60, \text{ звідки } \delta_s = 10^{-60/20} = 0,001;$$

$$A_n = 20 \lg(1 + \delta_n) = 0,1, \text{ звідки } \delta_n = 10^{0,1/20} - 1 = 0,0116.$$

Отже, $\delta = \min(\delta_n, \delta_s) = 0,001$. Згідно з виразом (9.3) параметр нерівномірності дорівнює $\alpha = 0,1102(60 - 8,7) = 5,65$.

Приклад №9.3. За допомогою методу зважування визначити вагову функцію та її параметри (якщо вони є) й обчислити мінімальний порядок режекторного СІХ-фільтра, який задовольняє таким специфікаціям:

смуга затримування	3,5–5,5 кГц;
ширина смуги переходу	500 Гц;
нерівномірність у смузі пропускання	0,5 дБ;
загасання в смузі заглушення	>40 дБ;
частота дискретизації	20 кГц;

Розв'язок.

Частота Найквіста $f_N = f_c / 2 = 20 / 2 = 10$ кГц. Нормована шири-

на смуги переходу $\Delta f = 0,5/10 = 0,05$.

Вимоги до загасання можна задовольнити при використанні функцій Ханна, Хеммінга, Блекмана або Кайзера. Визначимо, при використанні якого вагового вікна порядок фільтра буде мінімальним. Очевидно, що серед функцій Ханна, Хеммінга й Блекмана мінімальний порядок фільтра може забезпечити вікно Ханна.

Для функції Ханна порядок фільтра $N = 6,2/\Delta f = 6,2/0,05 = 124$ (табл. 9.2). Оскільки для задання РФ порядок фільтра $N-1$ має бути величиною парною, то приймаємо кількість коефіцієнтів $N = 125$.

Для функції Кайзера згідно з (9.4) кількість коефіцієнтів дорівнює

$$N \geq \frac{40 - 7,95}{7,18 \cdot 0,05} = 89,28.$$

Отже, приймаємо $N = 91$. Очевидно, що при використанні вікна Кайзера порядок фільтра буде найменший, порівняно з використанням інших вікон. Для знаходження параметра нерівномірності α згідно з виразами (9.1) і (9.2) знайдемо нерівномірність у смузі пропускання та загасання в смузі заглушення:

$$\begin{aligned} -20\lg\delta_s &= 40, \text{ звідки } \delta_s = 10^{-40/20} = 0,01; \\ 20\lg(1+\delta_n) &= 0,5, \text{ звідки } \delta_n = 10^{0,5/20} - 1 = 0,0593. \end{aligned}$$

$$\text{Отже, } \delta = \min(\delta_n, \delta_s) = 0,01.$$

Для функції Кайзера відповідно до виразу (9.3) параметр нерівномірності дорівнює $\alpha = 0,5842(40-21)^{0,4} + 0,07886(40-21) = 3,4$.

9.4. Завдання до розділу 9 для самостійної роботи

Методом зважування визначити вагову функцію та її параметри

(якщо вони є) й обчислити мінімальний порядок СІХ-фільтра, який задовольняє специфікації в табл. 9.3 (N – номер студента за журналом).

Таблиця 9.3 – Специфікації СІХ-фільтрів

N	Тип ЦФ	Нижня частота зрізу, Гц	Верхня частота зрізу, Гц	Ширина смуги переходу Δf , Гц	Загасання в смузі заглушення A_s , дБ	Частота дискретизації f_c , кГц
1	ФНЧ	5500	–	400	30	30
2	ФВЧ	–	6000	300	35	30
3	СФ	3000	6000	350	40	30
4	РФ	3500	7500	450	45	30
5	ФНЧ	800	–	250	50	15
6	ФВЧ	–	1000	300	55	15
7	СФ	800	1000	280	60	15
8	РФ	600	1600	320	65	15
9	ФНЧ	1200	–	550	70	25
10	ФВЧ	–	3000	600	75	25
11	СФ	2000	4000	650	80	25
12	РФ	1500	5000	700	85	25
13	ФНЧ	3600	–	2750	30	35
14	ФВЧ	–	2350	350	35	35
15	СФ	1800	2500	400	40	35
16	РФ	1600	4500	500	45	35
17	ФНЧ	2500	–	200	50	10
18	ФВЧ	–	1500	300	55	10
19	СФ	1500	2000	250	60	10
20	РФ	1300	2300	300	65	10
21	ФНЧ	600	–	500	70	20
22	ФВЧ	–	2000	600	75	20
23	СФ	2000	6000	700	80	20
24	РФ	2000	3000	800	85	20
25	ФНЧ	1000	–	350	30	40
26	ФВЧ	–	3000	500	35	40
27	СФ	1500	3500	500	40	40
28	РФ	2500	5000	800	45	40

10. ПРОЕКТУВАННЯ ФІЛЬТРІВ ЗА ДОПОМОГОЮ ПАКЕТА ПРИКЛАДНИХ ПРОГРАМ MATLAB

10.1. Характеристика пакета програм FDATool

Спеціальний пакет програм (ПП) **FDATool** (Filter Design & Analysis Tool) містить зручний інтерфейс користувача і дозволяє здійснювати розрахунок передавальної функції рекурсивних і нерекурсивних фільтрів різними методами синтезу, переглядати характеристики фільтра, аналізувати зміну характеристик при квантуванні коефіцієнтів фільтра, відліків вхідного сигналу й результатів проміжних обчислень. Можлива робота з різними структурами фільтра. Виклик пакета здійснюється в командному вікні **MatLab** наступною командою (рис. 10.1):

>> fdatool

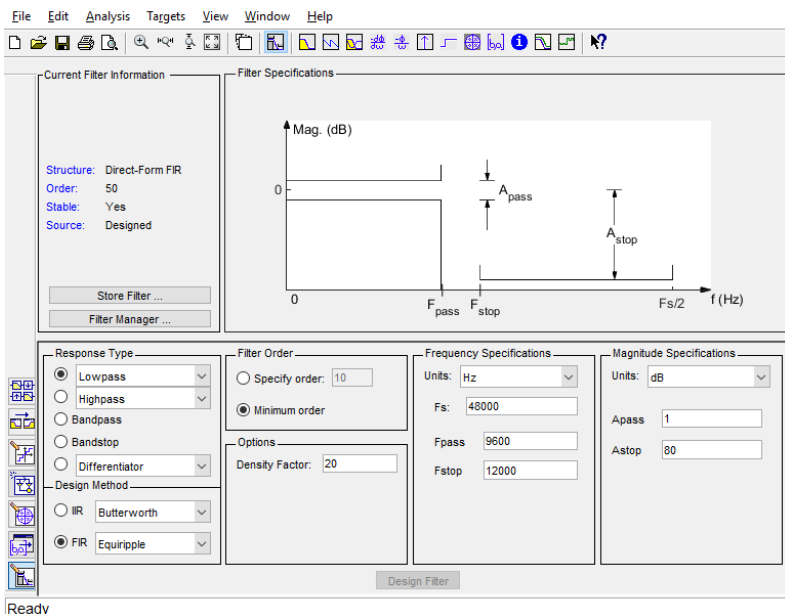


Рисунок 10.1 – Вікно ПП FDATool для проектування фільтра

Крім основного меню і панелі інструментів тут міститься поточна інформація про структуру фільтра, його порядок, стійкість (**Current Filter Information**); графік допусків для АЧХ (у дБ) (**Filter Specifications**), а також вкладка **Design Filter** для задання типу фільтра, його класу, методу синтезу, порядку фільтра, частоти дискретизації, граничних частот смуг пропускання та затримування і припустимих загасань у цих смугах [2].

10.1.1. Задання специфікації і розрахунок фільтра

На вкладці **Design Filter** у нижній частині вікна (рис. 10.1) можна встановити перемикач **Filter Type** в одне з таких положень:

- Lowpass (ФНЧ);
- Highpass (ФВЧ);
- Bandpass (СФ);
- Bandstop (РФ).

За допомогою перемикача **Design Method** вибирається рекурсивний (IIR – Infinite Impulse Response) або нерекурсивний (FIR – Finite Impulse Response) фільтр.

Якщо вибрати рекурсивний фільтр (фільтр з нескінченною імпульсною характеристикою – НІХ-фільтр), то далі в списку, що розкривається, потрібно вказати фільтр-прототип:

- Баттерворта (Batterworth);
- Чебишова (Chebyshev Type I);
- інверсний Чебишова (Chebyshev Type II);
- еліптичний (Elliptic).

При синтезі цих фільтрів використовується метод білінійного з-перетворення.

Можливі методи синтезу нерекурсивного фільтра (СІХ-фільтра):

- метод Ремеза, що забезпечує рівномірні пульсації АЧХ (Equiripple);
- забезпечення мінімуму середнього квадратичного відхилення АЧХ від заданої (Least-Squares);

- використання вікон як вагових функцій при синтезі фільтра (Window);

- інші.

У розділі **Filter Order** необхідно вказати необхідний порядок фільтра або встановити перемикач у положення Minimum order (найменший можливий порядок).

У розділах **Frequency Specifications** і **Magnitude Specifications** задаються безпосередньо параметри специфікації фільтра. Смысл параметрів можна зрозуміти з розташованого у верхній частині вікна графіка допусків (розділ **Filter Specifications**). Потрібно ввести частоту дискретизації F_s , граничні частоти смуги пропускання та смуги затримання (F_{pass} і F_{stop}), припустимі загасання в смузі пропускання і у смузі затримання (A_{pass} і A_{stop}).

Для запуску конструювання фільтра за заданими параметрами досить натиснути кнопку **Design Filter** (Конструювання фільтра). Після цього, використовуючи кнопки панелі інструментів, можна переглянути і, якщо необхідно, скорегувати отримані характеристики фільтра.

10.1.2. Перегляд характеристик фільтра

Вивести на екран частотні та часові характеристики фільтра, діаграму полюсів і нулів, коефіцієнти фільтра зручно в спеціальному вікні перегляду шляхом натискання кнопки **Full View Analysis** панелі інструментів (рис. 10.2).

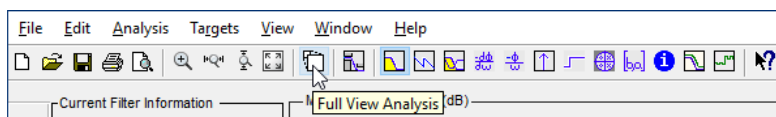


Рисунок 10.2 – Панель інструментів вікна Filter Design & Analysis Tool

У вікні **Filter Visualization Tool**, що з'явилося (рис. 10.3), за допомогою відповідних кнопок панелі інструментів (2-й ряд кнопок) можна

побачити АЧХ, ФЧХ, групову та фазову затримки, імпульсну характеристику, діаграму нулів і полюсів, значення коефіцієнтів передавальної функції та інші параметри спроектованого фільтра.

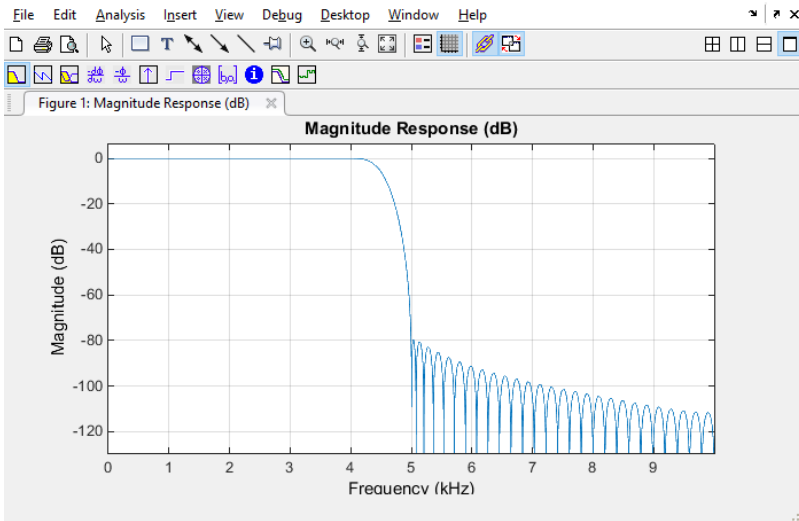


Рисунок 10.3 – Вікно Filter Visualization Tool

Аналогічні кнопки розташовані і на верхній панелі інструментів основного вікна ПП **FDATool** (рис. 10.1 та 10.2).

10.1.3. Експорт та імпорт опису фільтра

Спроекований раніше в робочому середовищі **MatLab** фільтр можна імпортувати в ПП **FDATool**. Для цього в меню **File** необхідно вибрати підменю **Import Filter from Workspace**. В результаті з'явиться вікно, зображене на рис. 10.4. Крім основного меню і панелі інструментів тут міститься поточна інформація про структуру фільтра, його порядок, стійкість (**Current Filter Information**), графік АЧХ (у дБ) (**Magnitude Response**), а також вкладка **Filter Coefficients** (коефіцієнти

фільтра) для задання параметрів фільтра, який необхідно імпортувати. Розглянемо більш докладно поля вкладки **Filter Coefficients**.

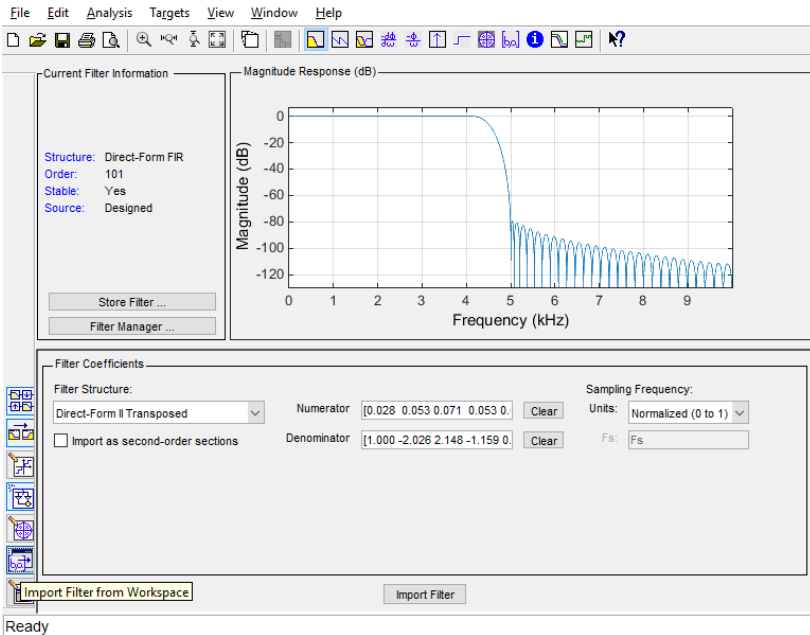


Рисунок 10.4 – Вікно ПП FDATool для імпорту фільтра

У полях вводу вказуються коефіцієнти чисельника (**Numerator**) і знаменника (**Denominator**) передавальної функції фільтра. Список береться у квадратні дужки, значення коефіцієнтів поділяються пробілами.

У поле **Sampling Frequency** необхідно ввести частоту дискретизації **F_s**, указавши одиницю виміру в полі **Units**.

Якщо вектори коефіцієнтів фільтра **b_z** та **a_z**, а також частота дискретизації **F_c** уже існують у робочій області MatLab, то замість введення числових значень у відповідних полях вводу потрібно просто

вказати імена змінних (рис. 10.4). Після натискання кнопки **Import Filter** можна переглядати характеристики фільтра.

Коефіцієнти будь-якого розробленого в ПП **FDATool** фільтра можна експортувати в блок ПП **Simulink** (ПП **Simulink** буде розглянутий нижче). Для цього в меню **File** необхідно вибрати підменю **Export to Simulink Model**. У вікні ПП **FDATool**, що з'явилося (рис. 10.5), необхідно встановити прапорець **Build model using basic elements**.

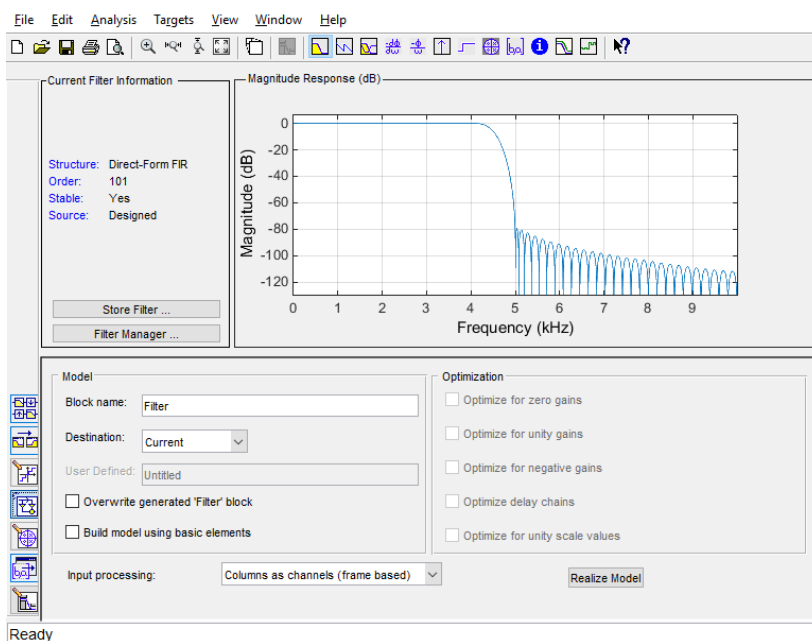


Рисунок 10.5 – Вікно ПП FDATool для експорту фільтра

Після натискання кнопки **Realize Model** буде створений блок фільтра (рис. 10.6), який можна використовувати в ПП **Simulink**.

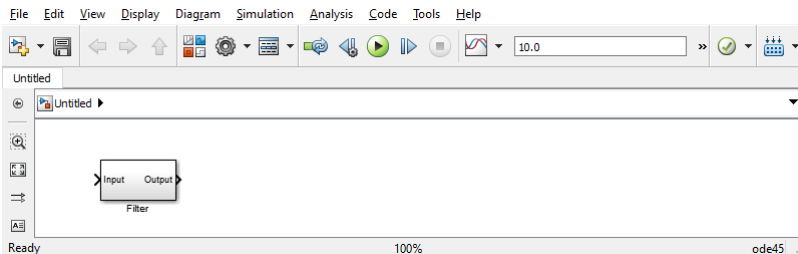


Рисунок 10.6 – Вікно редагування моделі

10.2. Характеристика пакета програм Simulink

Пакет програм **Simulink** є додатком до пакета **MatLab**. При моделюванні з використанням ПП **Simulink** реалізується принцип візуального програмування, у відповідності з яким користувач на екрані з бібліотеки стандартних блоків створює модель пристрою і здійснює розрахунки.

При роботі з ПП **Simulink** користувач має можливість модернізувати бібліотечні блоки, створювати свої власні, а також складати нові бібліотеки блоків. При моделюванні користувач може вибирати спосіб зміни модельного часу (з фіксованим або змінним кроком). Під час моделювання є можливість стежити за процесами, що відбуваються в системі. Для цього використовуються спеціальні пристрої спостереження, що входять до складу бібліотеки ПП **Simulink**. Результати моделювання можуть бути подані у вигляді графіків або таблиць.

Запуск ПП **Simulink** можна зробити одним із трьох способів:

- натиснути кнопку  (Simulink Library) на вкладці Home панелі інструментів **MatLab**;
- у рядку командного вікна **MatLab** виконати команду **simulink**;
- виконати команду **Open...** меню **File** і відкрити файл моделі (mdl-файл).

Останній спосіб кращий при запуску вже готової й налагодженої

моделі, коли потрібно лише провести моделювання і не потрібно додавати нові блоки в модель. При застосуванні двох перших способів відкривається вікно оглядача бібліотеки блоків (рис. 10.7).

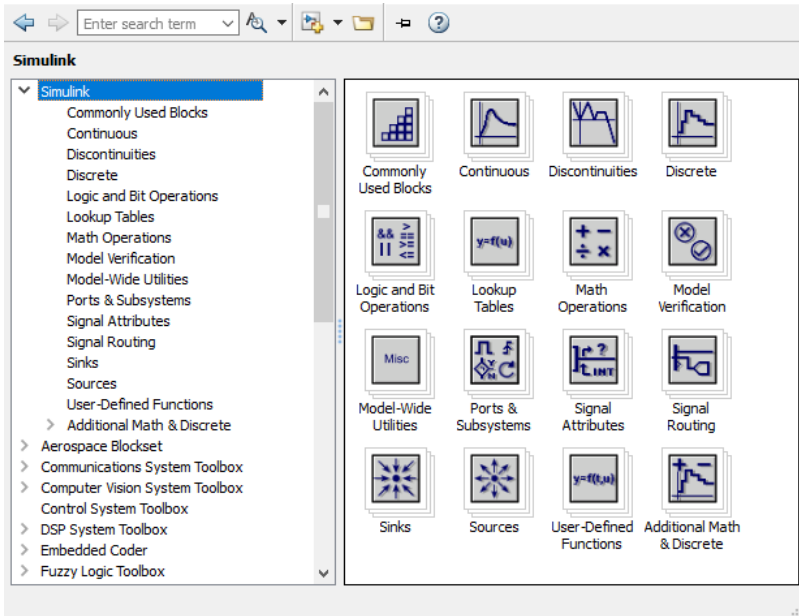


Рисунок 10.7 – Вікно оглядача бібліотеки блоків ПП Simulink

10.2.1. Оглядач бібліотеки блоків

Вікно оглядача бібліотеки блоків **Simulink Library Browser** містить такі елементи: заголовок із назвою вікна (Simulink Library Browser); панель меню (File, Edit, View, Help); панель інструментів із кнопками найбільш часто використовуваних опцій меню; вікно назви вибраного розділу бібліотеки; список розділів бібліотеки; вікно для виводу вмісту розділу бібліотеки (список підрозділів або блоків бібліотеки); рядок стану оглядача. На рис. 10.7 виділено бібліотеку ПП **Simulink** (у лівій частині вікна) і показано її розділи (у правій частині вікна). Бібліотека ПП **Simulink** містить такі розділи:

- Continuous – блоки аналогових елементів;
- Discontinuous – блоки нелінійних елементів;
- Discrete – блоки дискретних елементів;
- LookUp Tables – блоки таблиць;
- Math Operations – блоки елементів математичних операцій;
- Model Verification – блоки перевірки властивостей сигналів;
- Model-Wide Utilities – розділ додаткових утиліт;
- Ports&Subsystems – порти та підсистеми;
- Signal Attributes – блоки задання властивостей сигналів;
- Signal Routing – блоки маршрутизації сигналів;
- Sinks – блоки прийому та відображення сигналів;
- Sources – блоки джерел сигналу;
- User-Defined Function – функції, визначені користувачем.

10.2.2. Створення моделі

Для створення моделі в ПП **Simulink** необхідно послідовно виконати ряд дій.

1) *Створити новий файл моделі* (вибрати команду **File/New/Model** або натиснути відповідну кнопку на панелі інструментів).

2) *Розташувати блоки у вікні моделі*. Для цього необхідно відкрити відповідний розділ бібліотеки. Далі, указавши курсором на необхідний блок і натиснувши на ліву клавішу миші, «перетягнути» блок у створене вікно. Для видалення блока необхідно вибрати блок (вказати курсором на його зображення і натиснути ліву клавішу миші), а потім натиснути клавішу Delete на клавіатурі.

3) Для *зміни параметрів блока*, які встановлені за замовчуванням, необхідно двічі клацнути лівою клавішею миші, указавши курсором на зображення блока. Відкриється вікно параметрів цього блока. При введенні числового значення параметра як десяткового розділювача повинна використовуватися крапка, а не кома. Після внесення змін закрити вікно кнопкою ОК. Можна також, не закриваючи вікна, скористатися кнопкою Apply для збереження змін.

4) Після установки на схемі всіх блоків потрібно *виконати* їхнє *з'єднання*. Для цього необхідно вказати курсором на вихід блока, а потім натиснути і, не відпускаючи ліву клавішу миші, провести лінію до входу іншого блока та відпустити клавішу. Якщо з'єднання не створене, то лінія зв'язку буде пунктирною і мати червоний колір. У випадку правильного з'єднання лінія зв'язку буде суцільною. Для створення лінії зв'язку можна також виділити блок-джерело сигналу, потім натиснути клавішу Ctrl й виділити блок-приймач. Для створення точки розгалуження в сполучній лінії потрібно підвести курсор до передбачуваного вузла і, натиснувши праву клавішу миші, протягнути лінію. Для видалення лінії потрібно виділити її, а потім натиснути клавішу Delete.

5) Після складання розрахункової схеми необхідно *зберегти* її у вигляді файла на диску (пункт меню **File/Save As...** у вікні схеми). Ім'я файлу може бути довільної довжини, але має починатися з букви і містити тільки алфавітно-цифрові символи латиниці та знаки підкреслення. При наступному редагуванні схеми можна користуватися пунктом меню **File/Save**. При повторних запусках ПП **Simulink** завантаження схеми здійснюється за допомогою меню **File/Open...** у вікні оглядача бібліотеки блоків або з командного вікна системи **MatLab**.

На рис. 10.8 наведено приклад моделі.

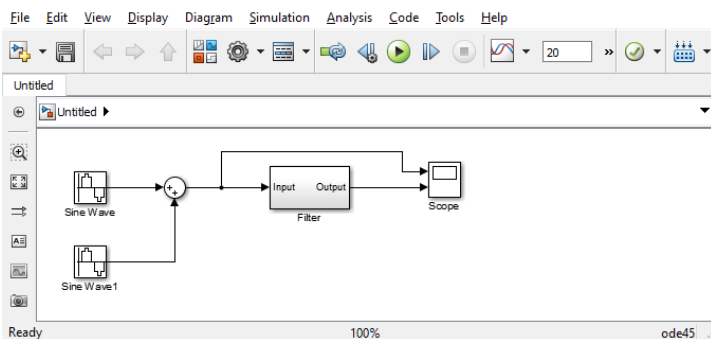


Рисунок 10.8 – Модель, створена зі стандартних блоків ПП Simulink і блока фільтра, експортованого із ПП DFATool

У цій моделі використовуються 2 генератори синусоїдальних сигналів. Один генератор задає синусоїду із частотою 0,2 Гц і амплітудою 5 (рис. 10.9), а інший – із частотою 7 Гц і амплітудою 3.

Sine Wave

Output a sine wave:

$$O(t) = \text{Amp} * \sin(\text{Freq} * t + \text{Phase}) + \text{Bias}$$

Sine type determines the computational technique used. The parameters in the two types are related through:

$$\text{Samples per period} = 2 * \pi / (\text{Frequency} * \text{Sample time})$$
$$\text{Number of offset samples} = \text{Phase} * \text{Samples per period} / (2 * \pi)$$

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

Parameters

Sine type: Time based

Time (t): Use simulation time

Amplitude: 5

Bias: 0

Frequency (rad/sec): $0.5 * 2 * \pi$

Phase (rad): 0

Sample time: $1/20$

☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Рисунок 10.9 – Вікно налаштування параметрів генератора синусоїдального сигналу

10.2.3. Установка параметрів моделювання

Перш ніж запустити завантажену модель, потрібно ознайомитися з установкою загальних параметрів моделювання. Для цього виконаємо команду **Model Configuration Parameters...** у меню **Simulation** вікна з

зображенням моделі. З'явиться вікно установки параметрів моделювання (рис. 10.10). Це вікно має ряд вкладок з великою кількістю параметрів. За замовчуванням відкривається вкладка **Solver**, яка дозволяє встановити параметри розв'язувального пристрою системи моделювання **Simulink**. До найважливіших параметрів розв'язувача належить час моделювання – **Simulation time**. Він задається початковим часом **Start time** (зазвичай 0) і скінченим часом **Stop time**. Рівність **Stop time** нескінченності (inf) означає, що моделювання буде відбуватися нескінченно довго, поки ми не перервемо його. Однак у цьому випадку важко отримати розрізнені осцилограми роботи пристрою, тому рекомендується задавати скінчені значення **Stop time**.

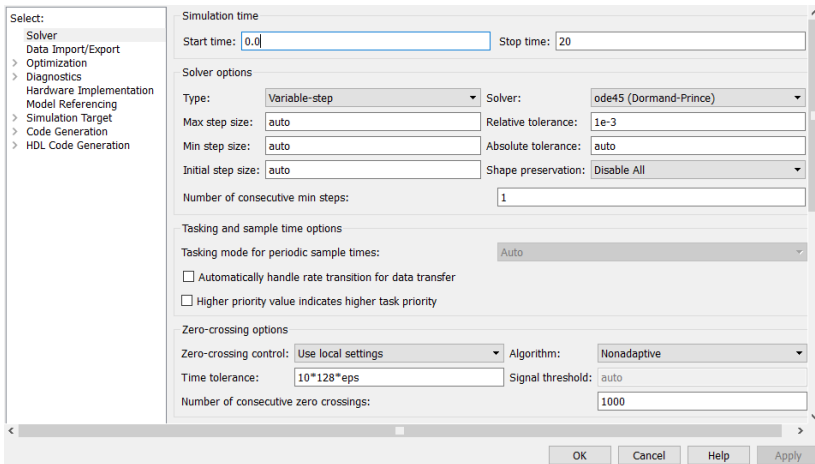


Рисунок 10.10 – Вікно установки параметрів моделювання

Необхідно відзначити, що час моделювання – величина достатньо умовна. Точної відповідності між часом моделювання в секундах і заданим значенням немає (**Stop time** = 50 не означає моделювання протягом 50 с). Реальний час моделювання сильно залежить від швидкодії комп'ютера, на якому виконується моделювання.

10.2.4. Запуск процесу моделювання

Наприкінці панелі інструментів ПП **Simulink** знаходяться дві кнопки керування моделюванням. Одна з них, у вигляді чорного трикутника (**Start/Pause Simulation**), запускає або припиняє розпочатий процес моделювання, а інша, у вигляді чорного квадрата (**Stop**), зупиняє його. Усе, що потрібно для запуску моделювання – це натиснути кнопку із зображенням трикутника. На рис. 10.11 показаний результат запуску моделі, зображеної на рис. 10.8. Замість кнопок можна використовувати команди **Start** і **Pause** у меню **Simulation** вікна моделі.

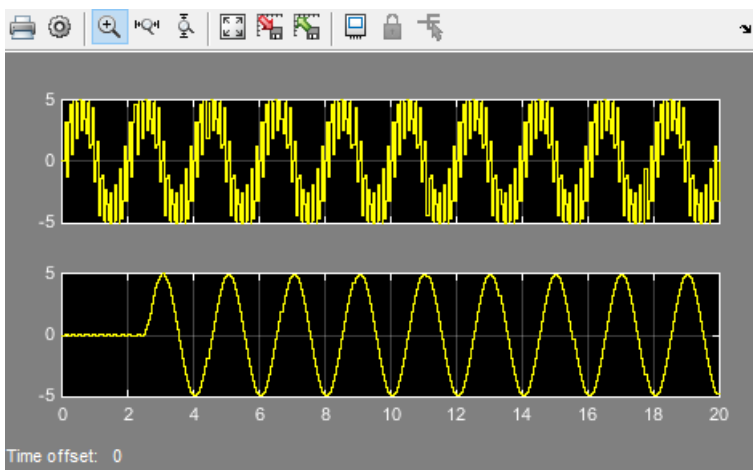


Рисунок 10.11 – Приклад результату роботи моделі

У цьому випадку результати моделювання подано у вигляді зашумленого сигналу і сигналу після фільтрації, побудованих за допомогою віртуального осцилографа (блок **Scope**). Потрібно зазначити, що у блока **Scope** за замовчуванням один вхід. Для додавання потрібної кількості входів на панелі інструментів вікна **Scope** необхідно натиснути кнопку **Parameters**. У вікні, що з'явилося, (рис. 10.12) в полі **Number of axes** необхідно вказати кількість входних сигналів.

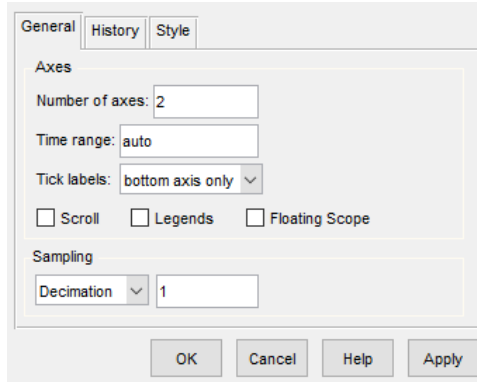


Рисунок 10.12 – Вікно налаштування параметрів віртуального осцилографа

10.3. Завдання до розділу 10 для самостійної роботи

1) Згідно з обраною специфікацією (табл. 9.3) спроекуйте методом зважування СІХ-фільтр із мінімальним порядком фільтра за допомогою ПП **FDATool** (для всіх типів фільтрів нерівномірність у смузі пропускання дорівнює 1 дБ). Побудуйте ІХ, АЧХ, ФЧХ отриманого СІХ-фільтра.

2) Згідно з обраною специфікацією (табл. 10.1) за допомогою програми **FDATool** спроекуйте НІХ-фільтр (для всіх типів фільтрів нерівномірність у смузі пропускання дорівнює 1 дБ). Як метод дискретизації використовуйте метод білінійного перетворення. Побудуйте ІХ, АЧХ, ФЧХ, групову та фазову затримки, діаграму нулів і полюсів отриманого НІХ-фільтра.

3) Виконайте експорт спроектованих фільтрів в ПП **Simulink**. У середовищі **Simulink** побудуйте схему, зображену на рис. 10.8. Задайте такі параметри вхідних сигналів, щоб можна було продемонструвати роботу спроектованого фільтра (частота одного сигналу має бути в смузі пропускання фільтра, а частота другого сигналу – поза смугою пропускання).

Таблиця 10.1 – Специфікації НІХ-фільтрів (N – номер студента за журналом)

N	Тип ЦФ	Нижня частота зрізу, Гц	Верхня частота зрізу, Гц	Δf , Гц	A_s , дБ	f_c , кГц	Аналоговий фільтр-прототип
1	ФНЧ	5500	–	400	30	30	Баттерворта
2	ФВЧ	–	6000	300	35	30	еліптичний
3	СФ	3000	6000	350	40	30	Чебишова 1
4	РФ	3500	7500	450	45	30	Чебишова 2
5	ФНЧ	800	–	250	50	15	еліптичний
6	ФВЧ	–	1000	300	55	15	Чебишова 1
7	СФ	800	1000	280	60	15	Чебишова 2
8	РФ	600	1600	320	65	15	Баттерворта
9	ФНЧ	1200	–	550	70	25	Чебишова 1
10	ФВЧ	–	3000	600	75	25	Чебишова 2
11	СФ	2000	4000	650	80	25	Баттерворта
12	РФ	1500	5000	700	85	25	еліптичний
13	ФНЧ	3600	–	2750	30	35	Чебишова 2
14	ФВЧ	–	2350	350	35	35	Баттерворта
15	СФ	1800	2500	400	40	35	еліптичний
16	РФ	1600	4500	500	45	35	Чебишова 1
17	ФНЧ	2500	–	200	50	10	Баттерворта
18	ФВЧ	–	1500	300	55	10	еліптичний
19	СФ	1500	2000	250	60	10	Чебишова 1
20	РФ	1300	2300	300	65	10	Чебишова 2
21	ФНЧ	600	–	500	70	20	еліптичний
22	ФВЧ	–	2000	600	75	20	Чебишова 1
23	СФ	2000	6000	700	80	20	Чебишова 2
24	РФ	2000	3000	800	85	20	Баттерворта
25	ФНЧ	1000	–	350	30	40	Чебишова 1
26	ФВЧ	–	3000	500	35	40	Чебишова 2
27	СФ	1500	3500	500	40	40	Баттерворта
28	РФ	2500	5000	800	45	40	еліптичний

11. ПЕРВИННА (ПОПЕРЕДНЯ) ОБРОБКА ЗОБРАЖЕНЬ

11.1. Дискретизація зображень

Перетворення зображень в комп'ютері і зберігання оброблених даних виконуються в дискретному виді. Для отримання дискретного відображення з безперервних аналогових зображень реального світу застосовується дискретизація (sampling). Практично її здійснюють пристрої вводу (цифровий фотоапарат, сканер або інші). Для візуального сприйняття оброблених зображень на пристроях виводу (дисплей, плоттер та ін.) здійснюється реконструкція аналогового зображення за його дискретизованим відображенням.

Дискретний сигнал представляється як функція, відмінна від нуля тільки в точках растру. Тому для розгляду зображень, визначених на прямокутнику, поступають двояко: або трактують як сигнал, рівний 0 скрізь окрім цього прямокутника (аналогова математика спектральних перетворень), або вважають довжину і ширину головним періодом сигналу і за межами растру простір заповнюють зсунутими на період копіями зображення (дискретна математика перетворень).

У найпростішому випадку чорно-білих зображень ми маємо двомірний аналоговий масив $s_a(x, y)$. Для кольорових зображень в моделі RGB, враховуючи властивість адитивності при додаванні кольорів, кожен шар R, G і B також може розглядатися і оброблятися, як двомірний масив, з подальшим підсумовуванням результатів.

Із способів узагальнення одновимірної періодичної дискретизації на двомірний випадок найбільш простою є періодична дискретизація в прямокутних координатах:

$$s[n, m] = s_a(n\Delta x, m\Delta y),$$

де Δx , Δy – горизонтальний і вертикальний інтервали дискретизації двомірного безперервного сигналу $s_a(x, y)$ з безперервними коорди-

татами x та y відповідно.

Зазвичай значення Δx і Δy , як і в одновимірному випадку, приймаються рівними 1.

Дискретизація двовимірного сигналу також призводить до періодизації його спектру і навпаки. Зберігається і умова інформаційної рівноцінності координатного і частотного представлень дискретного сигналу при рівній кількості точок дискретизації в головних діапазонах сигналу. Для прямокутної дискретизації пряме і зворотне перетворення Фур'є визначаються виразами:

$$S[k, l] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} s[n, m] e^{-j \frac{2\pi}{N} nk - j \frac{2\pi}{M} ml}; \quad (11.1)$$

$$S[k, l] = \sum_{n=0}^{N-1} e^{-j \frac{2\pi}{N} nk} \sum_{m=0}^{M-1} s[n, m] e^{-j \frac{2\pi}{M} ml}; \quad (11.2)$$

$$s[n, m] = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} S[k, l] e^{j \frac{2\pi}{N} nk + j \frac{2\pi}{M} ml}; \quad (11.3)$$

$$s[n, m] = \frac{1}{NM} \sum_{k=0}^{N-1} e^{j \frac{2\pi}{N} nk} \sum_{l=0}^{M-1} S[k, l] e^{j \frac{2\pi}{M} ml}. \quad (11.4)$$

За виразами (11.1) і (11.2) обчислюється пряме двовірне ДПФ, а за виразами (11.3) і (11.4) – зворотне двовірне ДПФ. Ці вирази показують, що двовірне ДПФ по прямокутному растру дискретизації даних може обчислюватися за допомогою одновірних послідовних ДПФ. Інші суми виразів (11.2) і (11.4) являються одновірними ДПФ перерізів функцій $s[n, m]$ і $S[k, l]$ по лініям n і k відповідно, а перші – одновірними ДПФ обчислених функцій в перерізах по m та l . Іншими словами, початкові матриці значень $s[n, m]$ і $S[k, l]$ перерахову-

ються спочатку в проміжні матриці з ДПФ по рядках (або по стовпцях), а проміжні – в остаточні з ДПФ по стовпцях (або відповідно по рядках).

Для того, щоб періодичне повторення спектру (рис. 11.1), викликане дискретизацією аналогового сигналу, з частотою $f_x = 1/\Delta x$ і $f_y = 1/\Delta y$, не змінювало спектр в головному частотному діапазоні (по відношенню до спектру початкового аналогового сигналу), необхідно і достатньо, щоб максимальні частотні складові $f_{\max}$, у спектрі аналогового сигналу, як по рядках, так і по стовпцях, не перевищували частоти Найквіста ($f_{\max x} \leq f_N = f_x/2$, $f_{\max y} \leq f_M = f_y/2$). Це означає, що частота дискретизації сигналу має бути мінімум в два рази вища за максимальну частоту складову в спектрі сигналу:

$$f_x \geq 2f_{\max x}, \quad f_y \geq 2f_{\max y},$$

що забезпечує вихід спектральних функцій на нульові значення на кінцях головного діапазону спектру.

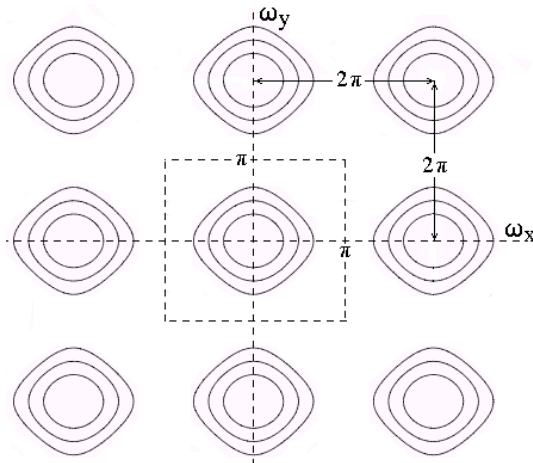


Рисунок 11.1 – Періодизація спектру

11.2. Робота із зображеннями в системі MATLAB

11.2.1. Характеристика пакета Image Processing Toolbox

Пакет розширення Image Processing Toolbox призначений для роботи у складі системи MATLAB. Він налічує понад 100 функцій, що забезпечують ефективну роботу із зображеннями з командного рядка системи MATLAB. Призначення пакету розширення Image Processing Toolbox полягає у вдосконаленні вже відомих і розробці нових математичних методів обробки зображень, і створенні програмних засобів. Відповідно до цього, переважна більшість функцій пакету призначена для використання їх в командному режимі і у складі m-файлів. Основне в цьому пакеті – можливість компілювати m-файли, що звертаються до цього пакету, разом з пакетом і, таким чином, включати його методи в оригінальне програмне забезпечення легальних користувачів MATLAB. Цьому програмному забезпеченню знадобиться власний графічний інтерфейс користувача, а GUI пакету тільки збільшив би розмір коду. Тому розробники пакету відмовилися від створення GUI варіантів утиліт обробки довільних зображень, як це реалізовано в інших пакетах розширення.

11.2.2. Растрова графіка в системі MATLAB

Машинна графіка підрозділяється на векторну і растрову. Поширенішою є растрова (точкова) графіка, при якій зображення будується з безлічі точок і, як в телебаченні, має структуру растру. При цьому зображення ліній виглядають ступінчастими, але ця ступінчастість стає мало помітною при високій роздільній здатності графіки. Растрова графіка реалізована в звичайних персональних комп'ютерах, її засоби досить дешеві і добре опрацьовані.

Основним поняттям растрової графіки є піксель – це маленький прямокутник (майже квадрат), що має координати і деякі ознаки (атрибути). Такими ознаками можуть бути колір пікселя, його прозорість і так далі. Для системи MATLAB колір пікселя характеризується його

глибиною d – вона задається числом бітів, потрібним для задання будь-якого кольору пікселя. Число пікселів по горизонталі і вертикалі екрану задає розширення графіки. Об’єм пам’яті у бітах, потрібний для зберігання зображення, дорівнює $c \times r \times d$, а в байтах – $c \times r \times d / 8$, де c – число стовпців (columns) зображення; r – число рядків (rows).

11.2.3. Типи растрових зображень

Растрові зображення, що застосовуються в системі MATLAB і в пакетах розширення Image Processing Toolbox, можуть бути наступного типу:

- бінарні;
- напівтонові;
- палітрові;
- повнокольорові.

Піксель *бінарних* зображень представлений всього двома кольорами – зазвичай білий (піксель не видно на білому фоні) і чорний (піксель видно). Кожен елемент матриці має значення 0 або 1 і може бути представлений у виді $I(r, c)$, де r – номер рядка; c – номер стовпця елемента, відповідного до заданого пікселя.

Напівтонові зображення (grayscale) можуть мати пікселі з відтінків сірого кольору. Ці зображення зберігаються у вигляді двовірних матриць. Індеси матриць задають положення кожного пікселя на екрані дисплея, а значення відповідного елемента матриці $I(r, c)$ задає його яскравість – відтінок сірого кольору.

У *кольорових палітрових* зображеннях використовуються дві матриці. Одна зберігає значення індексів, які задають звернення до рядка матриці палітра. Матриця палітра, що іменується колірною картою, має тип double і містить три групи стовпців – червоного R, зеленого G і синього B кольорів зі значеннями елементів від 0 до 1.0. Вони і задають колір відповідного пікселя. Колірна карта або просто палітра зображення є масивом дійсних чисел з плаваючою точкою в діапазоні

$[0;1.0]$ розміру $m \times 3$. У більшості зовнішніх файлів з графічним форматом колірна карта зберігається в цілочисельному вигляді, проте MATLAB такий формат представлення даних не підтримує, і при записі або читуванні зображення із зовнішнього файлу колірна карта конвертується автоматично.

Повнокольорові зображення, або RGB-зображення, будуються у форматі RGB (Red, Green, Blue). У цьому форматі зображення зберігаються в тривимірному масиві I , що фактично містить три матриці. Кожна з них містить індексовані змінні зі значенням, що відповідає відносній яскравості відповідно червоного (Red), зеленого (Green) і синього (Blue) кольорів. Елемент масиву $I(r, c, n)$, де r, c – номери рядка і стовпця, задають положення пікселя; n – номер матриці, задає доступ до тієї або іншої матриці тривимірного масиву: 1 – для R-матриці, 2 – для G-матриці і 3 – для B-матриці. Повнокольорове зображення не використовує колірну карту. Відзначимо, що властивість `CData` (колірні дані) повнокольорового зображення є тривимірним масивом розміру $n \times m \times 3$. Цей масив складається з трьох матриць розміру $n \times m$, що містять, відповідно, інтенсивності червоного, зеленого і синього кольорів кожного пікселя. Найчастіше використовуються кольорові зображення, що мають 16, 256, 65536 (High Color) і 10,7 мільйони (True Color) кольорів.

Необхідно помітити, що в MATLAB зображення можуть зберігатися у вигляді масивів цілих чисел, що мають 8 біт на піксель і 16 біт на піксель. Проте найбільш перспективний метод зберігання – у вигляді масиву чисел з плаваючою комою з подвійною точністю 64 біт/піксель.

11.2.4. Вивід на екран графічного об'єкту

Функція дескрипторної графіки **image()** створює зображення, представляючи кожен елемент вхідної матриці у вигляді індексу точної палітри або безпосередньо як модель RGB, залежно від форми вхідних даних. Функцію **image()** можна представити в двох формах:

- функція високого рівня, яка викликає базову функцію MATLAB **newplot()** для визначення місця розташування нового графічного об'єкту і установки властивостей осей;

- функція низького рівня, яка будує зображення без установки параметрів осей і виклику функції **newplot()**. У такому вигляді список аргументів цієї функції може бути поданий у вигляді багатовимірного масиву, що складається з пар чисел, що характеризують кожен піксель цього графічного об'єкту.

Є наступні форми запису функції **image()**:

- **image(C)** – повертає зображення, отримане з матриці C. Кожен елемент матриці Z точно вказує колір або прямокутний сегмент зображення;

- **image(x, y, C)** – повертає додатково два двоелементні вектори x і y, що містять діапазони зміни координат по осях X і Y;

- **image(x, y, C, 'PropertyName', PropertyValue...)** – як функція високого рівня викликає **newplot()** для побудови нового зображення і застосовує спеціальні параметри **PropertyName**, **PropertyValue**. При використанні функції низького рівня **image('PropertyName', PropertyValue...)** також можуть бути задані параметри **PropertyName**, **PropertyValue**;

- **handle = image(...)** – повертає дескриптор створюваного зображення.

У наступному прикладі

```
I=magic(10); image(I);
```

можна спостерігати побудову зображення для магічної матриці I розміру 10×10 .

11.2.5. Вивід зображення на екран

Функція **imshow()** служить для виводу зображення на екран і використовується у ряді форматів:

- **imshow(I, n)** – виводить на екран півтонове зображення **I** з використанням **n** дискретних рівнів сірого. Якщо параметр **n** опущений, то функція **imshow()** за умовчанням використовує 256 градацій сірого 24-бітового режиму або 64 градації сірого для інших режимів;

- **imshow(I, [low high])** – виводить на екран півтонове зображення **I** з підвищеною контрастністю. Усі елементи вхідного масиву, що мають значення **low** чи менше, відображаються чорним кольором, усі елементи зі значеннями більше або рівними **high** відображаються білим кольором. Значення між **low** і **high** відображаються проміжними відтінками сірого. Якщо в якості вхідного аргументу задана порожня матриця ([]) для **[low high]**, функція **imshow()** для **low** і **high** використовує наступні значення за умовчанням **[min(I(:)) max(I(:))]**;

- **imshow(BW)** – виводить на екран бінарне зображення **BW**. Нульові елементи вхідного масиву відображаються чорним кольором, 1 – білим;

- **imshow(X, map)** – виводить на екран палітрове зображення **X** з колірною картою **map**;

- **imshow(RGB)** – виводить на екран повнокольорове зображення **RGB**;

- **imshow(...display_option)** – виводить на екран зображення, використовуючи параметри **'truesize'** або **'notruesize'** в позиції **display_option**;

- **imshow(x, y, A)** – додатково використовує двоелементні вектори **x** і **y** для установки просторової системи координат з використанням параметрів **XData** та **YData**;

- **imshow filename** – виводить на екран зображення з файлу з ім'ям **filename**. При цьому автоматично викликається функція **imread()** для прочитування зображення із зовнішнього файлу;

- **h = imshow(...)** – повертає дескриптор цього зображення.

Нижче наведений приклад виклику функції **imshow()** (рис. 11.2):

```
imshow 'cameraman.tif'
```

Потрібно відмітити, що файл cameraman.tif, як і інші демонстраційні файли, входить в комплект постачання розширення Image Processing і розташовується в папці TOOLBOX\IMAGES\IMDEMOS. Файли, які знаходяться в цій папці, можуть вказуватися тільки своїм ім'ям і розширенням. Для файлів в довільній папці треба вказувати повний шлях доступу до них.



Рисунок 11.2 – Зображення, отримане за допомогою функції **imshow()**

11.2.6. Вивід на екран декількох зображень в одному вікні

Функція **subimage()** може бути використана для виводу на екран декількох зображень в одному вікні, навіть якщо ці зображення мають різні колірні палітри:

- **subimage(X, map)** – виводить на екран растрове зображення **X** з палітрою кольорів **map** в поточному масштабі;
- **subimage(I)** – виводить на екран півтонове зображення **I** в поточному масштабі;

- **subimage(BW)** – виводить на екран бінарне зображення **BW**;
- **subimage (RGB)** – виводить на екран повнокольорове зображення;
- **subimage(x,y...)** – виводить на екран зображення з використанням заданої системи координат;
- **h = subimage(...)** – повертає дескриптор графічного об'єкту.

Наступний приклад пояснює отримання в одному графічному вікні двох зображень – галактики m83.tif і дерев trees.tif (рис. 11.3)

```
[X1,map1] = imread('m83.tif');
[X2,map2] = imread('trees.tif');
subplot(1,2,1); subimage(X1,map1);
subplot(1,2,2); subimage(X2,map2)
```

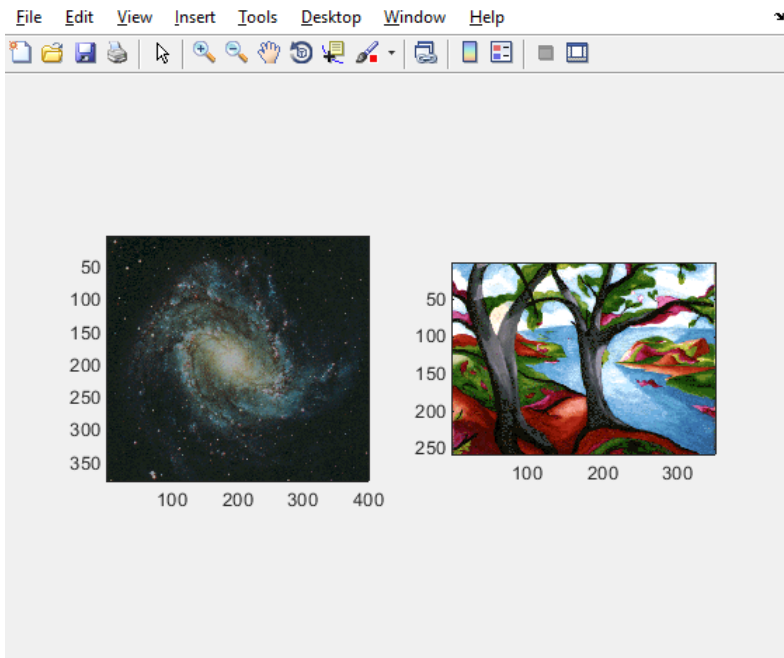


Рисунок 11.3 – Побудова двох рисунків в одному графічному вікні

11.2.7. Читання зображення з файлу

Функція **A=imread(filename, fmt)** читає з файлу з ім'ям **filename** півтонове або повнокольорове зображення і створює **A**. Якщо початкове зображення півтонове, то **A** – двовірний масив, якщо початкове зображення повнокольорове, то **A** – тривимірний масив розміру $n \times m \times 3$.

Інші форми цієї функції:

- **[X, map] = imread(filename, fmt)** – читає з файлу з ім'ям **filename** палітрове зображення у масив **A** з колірною картою **map**;

- **[...] = imread(filename)** – намагається визначити інформацію про формат файлу за його змістом. Параметри **filename** та **fmt** можна подивитись у довідковій системі функції **iminfo**;

- **[...] = imread(..., idx)** – читає одне зображення з TIFF-файлу, **idx** – ціле число, номер зображення по порядку.

Описувана функція має ряд особливостей для PNG-файлів, що містять прозорі пікселі (хоча і не завжди). Прозорі пікселі, якщо вони існують, ідентифікуються одним або двома компонентами: частина даних прозорості і альфа-канал. Частина даних прозорості визначає прозорі пікселі безпосередньо, наприклад, якщо частина даних прозорості 8-бітового зображення дорівнює 0.5020, то усі пікселі зображення з кольором 0.5020 будуть виведені на екран як прозорі. Помітимо, що PNG-файл може містити замість двох тільки один компонент – альфа-канал.

Альфа-канал є масивом з таким же числом пікселів, як і у початкового зображення, який визначає ознаку прозорості кожного пікселя (прозорий або непрозорий). І, нарешті, останній компонент PNG-файлу – це кольори фону, які визначають значення кольору, що «просвічується» з-під прозорих пікселів.

- **[...]=imread(..., 'BackgroundColor', bg)** – зчитує зображення з PNG-файлу і комбінує пікселі прозорості проти певного кольору. Форма параметра **bg** залежить від формату вхідного файлу. Якщо вхідне зображення палітрове, то параметр **bg** має бути цілим числом порядку $[1, P]$, де P – довжина масиву колірної карти. Якщо вхідне

зображення півтонове, то параметр **bg** має бути цілим числом порядку [0,1]. Якщо вхідне зображення повнокольорове, то параметр **bg** має бути трьохелементним вектором зі значеннями порядку [0,1];

- **[A, map, alpha] = imread(...)** – повертає **A**, де **A** – це шаблон способу, який використовується для визначення інформації про прозорість;

- **[A, map, alpha] = imread(filename)** або **[A, map, alpha] = imread(filename, fir)** читає зображення з PNG-файлу, якщо не застосовується комбінування і альфа-канал зберігається окремо від зображення;

- **[...] = imread(..., idx)** – читає одне зображення з CUR та ICO-файлів. Параметр **idx** – це ціле число, що визначає порядок зображення в файлі.

11.2.8. Двовірне дискретне перетворення Фур'є

Для двовірного прямого ДПФ використовується функція **fft2()**:

- **fft2(X)** повертає для масиву даних **X** двовірне дискретне перетворення Фур'є;

- **fft2(X, n, m)** усікає масив **X** або доповнює його нулями, щоб перед виконанням перетворення Фур'є створити матрицю розміром $n \times m$. Результат – матриця того ж розміру.

Нижче наведений приклад використання функції **fft2()** для обробки реального зображення планети Сатурн з кільцями (рис. 11.4):

```
%Програма №11.1
load imdemos saturn2;
subplot(1,2,1), imshow(saturn2);
B = fft2(saturn2); %двовірне ДПФ
subplot(1,2,2);
imshow(log(abs(fftshift(B))), []);
```

Для двовірного ЗДПФ використовується функція **ifft2()**:

- **ifft2(F)** робить зворотнє двовірне дискретне перетворення

Фур'є для матриці **F**;

- **ifft2(F, n, m)** робить зворотнє перетворення Фур'є розмірності $n \times m$ для матриці **F**.

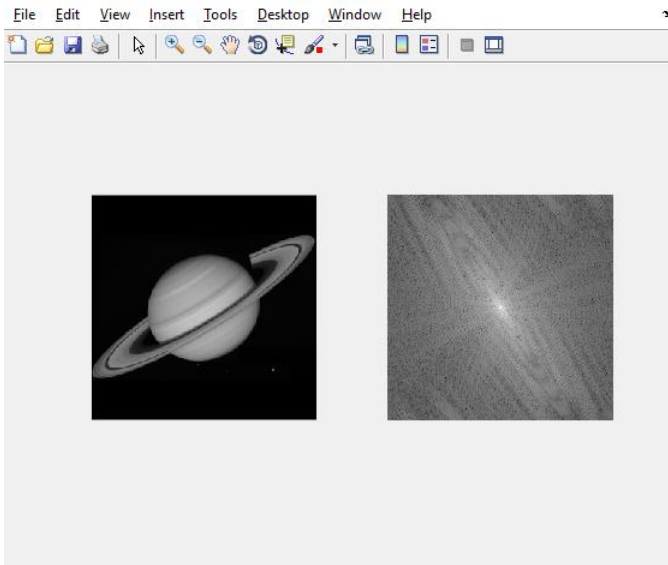


Рисунок 11.4 – Зображення Сатурна та його Фур'є-образу

Як було підкреслено в 11.1, двомірне ДПФ можна обчислити, використовуючи одновимірне ДПФ. Нижче наведено приклад, в якому продемонстрована ця властивість ДПФ:

```
%Програма №11.2
I=imread('coins.png');
F1=fft2(I); %двомірне ДПФ
X1=round(ifft2(F1)); %двомірне ЗДПФ
FF1=fft(I); %ДПФ стовпців
F2=fft(FF1.').'; %ДПФ рядків
XX2=ifft(F2); %ЗДПФ стовпців
X2=ifft(XX2.').'; %ЗДПФ рядків
```

```

%вивід вихідного зображення
subplot(3,2,2), imshow(I,[]);
%вивід Фур'є-образу
subplot(3,2,3), imshow(log(abs(fftshift(F1))),[]);
%отримане зображення за допомогою ЗДПФ
subplot(3,2,4), imshow(X1,[]);
%вивід Фур'є-образу
subplot(3,2,5), imshow(log(abs(fftshift(F2))),[]);
%отримане зображення за допомогою ЗДПФ
subplot(3,2,6), imshow(abs(X2),[]);

```

На рис. 11.5, в першому рядку, показано вхідне зображення; у другому рядку – Фур'є-образ, отриманий за допомогою функції двомірного ДПФ, і відновлене зображення, отримане за допомогою двомірного ЗДПФ; у третьому рядку – Фур'є-образ, отриманий за допомогою послідовного застосування функцій одновимірного ДПФ до рядків і стовпців, і відновлене зображення, отримане за допомогою послідовного застосування функцій ЗДПФ до стовпців і рядків.

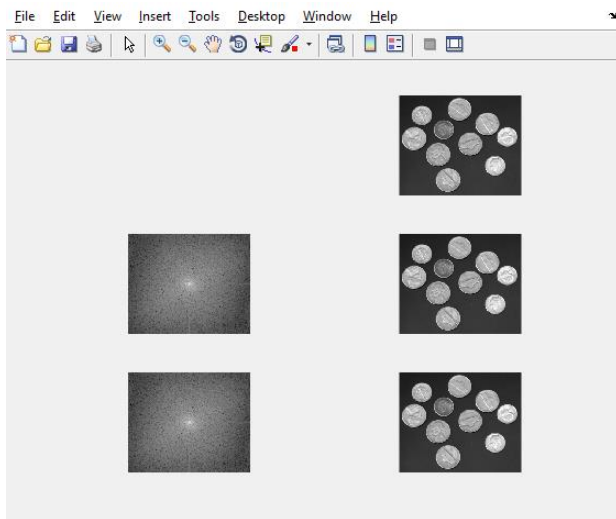


Рисунок 11.5 – Результати роботи програми «Приклад №11.2»

11.3. Приклади розрахунку двовірного ДПФ

Допустимо задано двовірний сигнал

$$\mathbf{s} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}.$$

Розрахуємо за виразом (11.1) перші декілька відліків спектру заданого сигналу:

$$S[0,0] = 1 + 1 + 1 + 1 + 2 + 2 + 2 + 2 + 3 + 3 + 3 + 3 + 4 + 4 + 4 + 4 = 40;$$

$$\begin{aligned} S[0,1] &= 1 + e^{-\frac{\pi}{2} \cdot 1 \cdot 1} + e^{-\frac{\pi}{2} \cdot 2 \cdot 1} + e^{-\frac{\pi}{2} \cdot 3 \cdot 1} + 2 + 2e^{-\frac{\pi}{2} \cdot 1 \cdot 1} + 2e^{-\frac{\pi}{2} \cdot 2 \cdot 1} + 2e^{-\frac{\pi}{2} \cdot 3 \cdot 1} + \\ &+ 3 + 3e^{-\frac{\pi}{2} \cdot 1 \cdot 1} + 3e^{-\frac{\pi}{2} \cdot 2 \cdot 1} + 3e^{-\frac{\pi}{2} \cdot 3 \cdot 1} + 4 + 4e^{-\frac{\pi}{2} \cdot 1 \cdot 1} + 4e^{-\frac{\pi}{2} \cdot 2 \cdot 1} + 4e^{-\frac{\pi}{2} \cdot 3 \cdot 1} = \\ &= 1 - j - 1 + j + 2(1 - j - 1 + j) + 3(1 - j - 1 + j) + 4(1 - j - 1 + j) = 0; \end{aligned}$$

$$\begin{aligned} S[0,2] &= 1 + e^{-\frac{\pi}{2} \cdot 1 \cdot 2} + e^{-\frac{\pi}{2} \cdot 2 \cdot 2} + e^{-\frac{\pi}{2} \cdot 3 \cdot 2} + 2 + 2e^{-\frac{\pi}{2} \cdot 1 \cdot 2} + 2e^{-\frac{\pi}{2} \cdot 2 \cdot 2} + 2e^{-\frac{\pi}{2} \cdot 3 \cdot 2} + \\ &+ 3 + 3e^{-\frac{\pi}{2} \cdot 1 \cdot 2} + 3e^{-\frac{\pi}{2} \cdot 2 \cdot 2} + 3e^{-\frac{\pi}{2} \cdot 3 \cdot 2} + 4 + 4e^{-\frac{\pi}{2} \cdot 1 \cdot 2} + 4e^{-\frac{\pi}{2} \cdot 2 \cdot 2} + 4e^{-\frac{\pi}{2} \cdot 3 \cdot 2} = \\ &= 1 - 1 + 1 - 1 + 2(1 - 1 + 1 - 1) + 3(1 - 1 + 1 - 1) + 4(1 - 1 + 1 - 1) = 0; \end{aligned}$$

$$\begin{aligned} S[0,3] &= 1 + e^{-\frac{\pi}{2} \cdot 1 \cdot 3} + e^{-\frac{\pi}{2} \cdot 2 \cdot 3} + e^{-\frac{\pi}{2} \cdot 3 \cdot 3} + 2 + 2e^{-\frac{\pi}{2} \cdot 1 \cdot 3} + 2e^{-\frac{\pi}{2} \cdot 2 \cdot 3} + 2e^{-\frac{\pi}{2} \cdot 3 \cdot 3} + \\ &+ 3 + 3e^{-\frac{\pi}{2} \cdot 1 \cdot 3} + 3e^{-\frac{\pi}{2} \cdot 2 \cdot 3} + 3e^{-\frac{\pi}{2} \cdot 3 \cdot 3} + 4 + 4e^{-\frac{\pi}{2} \cdot 1 \cdot 3} + 4e^{-\frac{\pi}{2} \cdot 2 \cdot 3} + 4e^{-\frac{\pi}{2} \cdot 3 \cdot 3} = \\ &= 1 + j - 1 - j + 2(1 + j - 1 - j) + 3(1 + j - 1 - j) + 4(1 + j - 1 - j) = 0; \end{aligned}$$

$$\begin{aligned}
S[1,0] &= 1 + 1 + 1 + 1 + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + \\
&+ 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1} + \\
&+ 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1} = 4(1 + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1}) = \\
&= 4(1 - 2j - 3 + 4j) = -8 + 8j;
\end{aligned}$$

$$\begin{aligned}
S[1,1] &= 1 + e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + e^{-j\frac{\pi}{2} \cdot 3 \cdot 1} + \\
&+ 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1} + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1 - j\frac{\pi}{2} \cdot 1 \cdot 1} + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1 - j\frac{\pi}{2} \cdot 2 \cdot 1} + 2e^{-j\frac{\pi}{2} \cdot 1 \cdot 1 - j\frac{\pi}{2} \cdot 3 \cdot 1} + \\
&+ 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1 - j\frac{\pi}{2} \cdot 1 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1 - j\frac{\pi}{2} \cdot 2 \cdot 1} + 3e^{-j\frac{\pi}{2} \cdot 2 \cdot 1 - j\frac{\pi}{2} \cdot 3 \cdot 1} + \\
&+ 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1 - j\frac{\pi}{2} \cdot 1 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1 - j\frac{\pi}{2} \cdot 2 \cdot 1} + 4e^{-j\frac{\pi}{2} \cdot 3 \cdot 1 - j\frac{\pi}{2} \cdot 3 \cdot 1} = \\
&= 1 - j - 1 + j + 2(-j - 1 + j + 1) + 3(-1 + j + 1 - j) + 4(j + 1 - j - 1) = 0.
\end{aligned}$$

Очевидно, що розрахунок двовірного ДПФ за виразом (11.1) – достатньо трудомісткий процес. Тому скористаємося одновірними ДПФ (11.2) для розрахунку двовірного спектру. Спочатку вичислимо одновірні ДПФ по рядках. Оскільки у кожному рядку 4 величини, то для розрахунку спектру можна скористатися алгоритмом ШПФ. Скористаємося алгоритмом ШПФ з проріджуванням за часом (рис. 4.2).

Розрахуємо ШПФ 1-го рядка, тобто $x[n] = (1, 1, 1, 1)$.

Крок 1. Оскільки усі значення дискретної послідовності однакові, то $x^1[n] = (1, 1, 1, 1)$.

Крок 2.

$$X^2[0] = x^1[0] + W_2^0 x^1[1] = 1 + 1 = 2;$$

$$X^2[1] = x^1[0] - W_2^0 x^1[1] = 1 - 1 = 0;$$

$$X^2[2] = x^1[2] + W_2^0 x^1[3] = 1 + 1 = 2;$$

$$X^2[3] = x^1[2] - W_2^0 x^1[3] = 1 - 1 = 0.$$

Крок 3.

$$X[0] = X^2[0] + W_4^0 X^2[2] = 2 + 2 = 4 ;$$

$$X[1] = X^2[1] + W_4^1 X^2[3] = 0 + 0e^{-j\frac{2\pi}{4}} = 0 ;$$

$$X[2] = X^2[0] - W_4^0 X^2[2] = 2 - 2 = 0 ;$$

$$X[3] = X^2[1] - W_4^1 X^2[3] = 0 - 0e^{-j\frac{2\pi}{4}} = 0 .$$

Розрахуємо ШПФ 2-го рядка, тобто $x[n] = (2, 2, 2, 2)$.

Крок 1. Оскільки усі значення дискретної послідовності однакові, то $x^1[n] = (2, 2, 2, 2)$.

Крок 2.

$$X^2[0] = x^1[0] + W_2^0 x^1[1] = 2 + 2 = 4 ;$$

$$X^2[1] = x^1[0] - W_2^0 x^1[1] = 2 - 2 = 0 ;$$

$$X^2[2] = x^1[2] + W_2^0 x^1[3] = 2 + 2 = 4 ;$$

$$X^2[3] = x^1[2] - W_2^0 x^1[3] = 2 - 2 = 0 .$$

Крок 3.

$$X[0] = X^2[0] + W_4^0 X^2[2] = 4 + 4 = 8 ;$$

$$X[1] = X^2[1] + W_4^1 X^2[3] = 0 + 0e^{-j\frac{2\pi}{4}} = 0 ;$$

$$X[2] = X^2[0] - W_4^0 X^2[2] = 4 - 4 = 0 ;$$

$$X[3] = X^2[1] - W_4^1 X^2[3] = 0 - 0e^{-j\frac{2\pi}{4}} = 0 .$$

Розрахуємо ШПФ 3-го рядка, тобто $x[n] = (3, 3, 3, 3)$.

Крок 1. Оскільки усі значення дискретної послідовності однакові, то $x^1[n] = (3, 3, 3, 3)$.

Крок 2.

$$X^2[0] = x^1[0] + W_2^0 x^1[1] = 3 + 3 = 6 ;$$

$$X^2[1] = x^1[0] - W_2^0 x^1[1] = 3 - 3 = 0 ;$$

$$X^2[2] = x^1[2] + W_2^0 x^1[3] = 3 + 3 = 6 ;$$

$$X^2[3] = x^1[2] - W_2^0 x^1[3] = 3 - 3 = 0 .$$

Крок 3.

$$X[0] = X^2[0] + W_4^0 X^2[2] = 6 + 6 = 12 ;$$

$$X[1] = X^2[1] + W_4^1 X^2[3] = 0 + 0e^{-j\frac{2\pi}{4}1} = 0 ;$$

$$X[2] = X^2[0] - W_4^0 X^2[2] = 6 - 6 = 0 ;$$

$$X[3] = X^2[1] - W_4^1 X^1[3] = 0 - 0e^{-j\frac{2\pi}{4}1} = 0 .$$

I, нарешті, розрахуємо ШПФ 4-го рядка, тобто $x[n] = (4, 4, 4, 4)$.

Крок 1. Оскільки усі значення дискретної послідовності однакові, то $x^1[n] = (4, 4, 4, 4)$.

Крок 2.

$$X^2[0] = x^1[0] + W_2^0 x^1[1] = 4 + 4 = 8 ;$$

$$X^2[1] = x^1[0] - W_2^0 x^1[1] = 4 - 4 = 0 ;$$

$$X^2[2] = x^1[2] + W_2^0 x^1[3] = 4 + 4 = 8 ;$$

$$X^2[3] = x^1[2] - W_2^0 x^1[3] = 4 - 4 = 0 .$$

Крок 3.

$$X[0] = X^2[0] + W_4^0 X^2[2] = 8 + 8 = 16 ;$$

$$X[1] = X^2[1] + W_4^1 X^2[3] = 0 + 0e^{-j\frac{2\pi}{4}1} = 0 ;$$

$$X[2] = X^2[0] - W_4^0 X^2[2] = 8 - 8 = 0 ;$$

$$X[3] = X^2[1] - W_4^1 X^1[3] = 0 - 0e^{-j\frac{2\pi}{4}1} = 0 .$$

В результаті отримаємо наступну проміжну матрицю, кожен ря-

док якої складений з отриманих одновимірних ДПФ рядків початкової матриці $\mathbf{s}$:

$$\mathbf{S}' = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 12 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix}.$$

Потім для матриці $\mathbf{S}'$ вичислимо одновимірні ДПФ по стовпцях. Оскільки 2-й, 3-й і 4-й стовпці проміжної матриці $\mathbf{S}'$ містять нульові елементи, то і відповідні спектри теж міститимуть тільки нульові значення. Тому необхідно розрахувати тільки спектр 1-го стовпця, тобто послідовності $x[n] = (4, 8, 12, 16)$. Також скористаємося алгоритмом ШПФ з проріджуванням за часом (рис. 4.2).

Крок 1. Виконаємо перестановку відліків $x^1[0] = x[0] = 4$, $x^1[1] = x[2] = 12$, $x^1[2] = x[1] = 8$, $x^1[3] = x[3] = 16$, тобто $x^1[n] = (4, 12, 8, 16)$.

Крок 2.

$$X^2[0] = x^1[0] + W_2^0 x^1[1] = 4 + 12 = 16;$$

$$X^2[1] = x^1[0] - W_2^0 x^1[1] = 4 - 12 = -8;$$

$$X^2[2] = x^1[2] + W_2^0 x^1[3] = 8 + 16 = 24;$$

$$X^2[3] = x^1[2] - W_2^0 x^1[3] = 8 - 16 = -8.$$

Крок 3.

$$X[0] = X^2[0] + W_4^0 X^2[2] = 16 + 24 = 40;$$

$$X[1] = X^2[1] + W_4^1 X^2[3] = -8 - 8e^{-j\frac{2\pi}{4}} = -8 + 8j;$$

$$X[2] = X^2[0] - W_4^0 X^2[2] = 16 - 24 = -8;$$

$$X[3] = X^2[1] - W_4^1 X^2[3] = -8 + 8e^{-j\frac{2\pi}{4}} = -8 - 8j.$$

В результаті отримуємо двовірний спектр вхідного сигналу:

$$\mathbf{S} = \begin{bmatrix} 40 & 0 & 0 & 0 \\ -8 + 8j & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 \\ -8 - 8j & 0 & 0 & 0 \end{bmatrix}.$$

Порівнявши результати, отримані за допомогою виразів (11.1) і (11.2), легко помітити, що значення відповідних відліків двовірного спектру повністю співпадають.

Виконаємо зворотне двовірне ДПФ отриманого спектру за виразом (11.4). Для розрахунку скористаємося алгоритмом ШПФ з провіджуванням за частотою (рис. 4.4).

Розрахуємо ЗШПФ 1-го рядка, тобто $X[k] = (40, 0, 0, 0)$.

Крок 1. Виконаємо перестановку відліків $X^1[0] = X[0] = 40$;
 $X^1[1] = X[2] = 0$; $X^1[2] = X[1] = 0$; $X^1[3] = X[3] = 0$.

Крок 2.

$$x^2[0] = X^1[0] + W_2^0 X^1[1] = 40 + 0 = 40 ;$$

$$x^2[1] = X^1[0] - W_2^0 X^1[1] = 40 - 0 = 40 ;$$

$$x^2[2] = X^1[2] + W_2^0 X^1[3] = 0 + 0 = 0 ;$$

$$x^2[3] = X^1[2] - W_2^0 X^1[3] = 0 - 0 = 0 .$$

Крок 3.

$$x^3[0] = x^2[0] + W_4^0 x^2[2] = 40 + 0 = 40 ;$$

$$x^3[1] = x^2[1] + W_4^{-1} x^2[3] = 40 + 0 j e^{j\frac{\pi}{2}} = 40 ;$$

$$x^3[2] = x^2[0] - W_4^0 x^2[2] = 40 - 0 = 40 ;$$

$$x^3[3] = x^2[1] - W_4^{-1} x^2[3] = 40 - 0je^{j\frac{\pi}{2}} = 40.$$

Крок 4. Нормування на N , тобто $x[n] = \frac{1}{4} x^3[n]$:

$$x[n] = (10, 10, 10, 10).$$

Розрахуємо ЗШПФ 2-го рядка, тобто $X[k] = (-8 + 8j, 0, 0, 0)$.

Крок 1. Виконаємо перестановку відліків $X^1[0] = X[0] = -8 + 8j$;

$$X^1[1] = X[2] = 0; \quad X^1[2] = X[1] = 0; \quad X^1[3] = X[3] = 0.$$

Крок 2.

$$x^2[0] = X^1[0] + W_2^0 X^1[1] = -8 + 8j + 0 = -8 + 8j;$$

$$x^2[1] = X^1[0] - W_2^0 X^1[1] = -8 + 8j - 0 = -8 + 8j;$$

$$x^2[2] = X^1[2] + W_2^0 X^1[3] = 0 + 0 = 0;$$

$$x^2[3] = X^1[2] - W_2^0 X^1[3] = 0 - 0 = 0.$$

Крок 3.

$$x^3[0] = x^2[0] + W_4^0 x^2[2] = -8 + 8j + 0 = -8 + 8j;$$

$$x^3[1] = x^2[1] + W_4^{-1} x^2[3] = -8 + 8j + 0je^{j\frac{\pi}{2}} = -8 + 8j;$$

$$x^3[2] = x^2[0] - W_4^0 x^2[2] = -8 + 8j - 0 = -8 + 8j;$$

$$x^3[3] = x^2[1] - W_4^{-1} x^2[3] = -8 + 8j - 0je^{j\frac{\pi}{2}} = -8 + 8j.$$

Крок 4. Нормування на N , тобто $x[n] = \frac{1}{4} x^3[n]$:

$$x[n] = (-2 + 2j, -2 + 2j, -2 + 2j, -2 + 2j).$$

Розрахуємо ЗШПФ 3-го рядка, тобто $X[k] = (-8, 0, 0, 0)$.

Крок 1. Виконаємо перестановку відліків $X^1[0] = X[0] = -8$;
 $X^1[1] = X[2] = 0$; $X^1[2] = X[1] = 0$; $X^1[3] = X[3] = 0$.

Крок 2.

$$x^2[0] = X^1[0] + W_2^0 X^1[1] = -8 + 0 = -8 ;$$

$$x^2[1] = X^1[0] - W_2^0 X^1[1] = -8 - 0 = -8 ;$$

$$x^2[2] = X^1[2] + W_2^0 X^1[3] = 0 + 0 = 0 ;$$

$$x^2[3] = X^1[2] - W_2^0 X^1[3] = 0 - 0 = 0 .$$

Крок 3.

$$x^3[0] = x^2[0] + W_4^0 x^2[2] = -8 + 0 = -8 ;$$

$$x^3[1] = x^2[1] + W_4^{-1} x^2[3] = -8 + 0 j e^{j \frac{\pi}{2}} = -8 ;$$

$$x^3[2] = x^2[0] - W_4^0 x^2[2] = -8 - 0 = -8 ;$$

$$x^3[3] = x^2[1] - W_4^{-1} x^2[3] = -8 - 0 j e^{j \frac{\pi}{2}} = -8 .$$

Крок 4. Нормування на N , тобто $x[n] = \frac{1}{4} x^3[n]$:

$$x[n] = (-2, -2, -2, -2) .$$

Розрахуємо ЗШПФ 4-го рядка, тобто $X[k] = (-8 - 8j, 0, 0, 0)$.

Крок 1. Виконаємо перестановку відліків $X^1[0] = X[0] = -8 - 8j$;
 $X^1[1] = X[2] = 0$; $X^1[2] = X[1] = 0$; $X^1[3] = X[3] = 0$.

Крок 2.

$$x^2[0] = X^1[0] + W_2^0 X^1[1] = -8 - 8j + 0 = -8 - 8j ;$$

$$x^2[1] = X^1[0] - W_2^0 X^1[1] = -8 - 8j - 0 = -8 - 8j ;$$

$$x^2[2] = X^1[2] + W_2^0 X^1[3] = 0 + 0 = 0 ;$$

$$x^2[3] = X^1[2] - W_2^0 X^1[3] = 0 - 0 = 0.$$

Крок 3.

$$x^3[0] = x^2[0] + W_4^0 x^2[2] = -8 - 8j + 0 = -8 - 8j;$$

$$x^3[1] = x^2[1] + W_4^{-1} x^2[3] = -8 - 8j + 0 j e^{j\frac{\pi}{2}} = -8 - 8j;$$

$$x^3[2] = x^2[0] - W_4^0 x^2[2] = -8 - 8j - 0 = -8 - 8j;$$

$$x^3[3] = x^2[1] - W_4^{-1} x^2[3] = -8 - 8j - 0 j e^{j\frac{\pi}{2}} = -8 - 8j.$$

Крок 4. Нормування на N , тобто $x[n] = \frac{1}{4} x^3[n]$:

$$x[n] = (-2 - 2j, -2 - 2j, -2 - 2j, -2 - 2j).$$

В результаті отримуємо проміжну матрицю

$$\mathbf{s}' = \begin{bmatrix} 10 & 10 & 10 & 10 \\ -2 + 2j & -2 + 2j & -2 + 2j & -2 + 2j \\ -2 & -2 & -2 & -2 \\ -2 - 2j & -2 - 2j & -2 - 2j & -2 - 2j \end{bmatrix}.$$

Тепер обчислимо одновимірні ДПФ по стовпцях проміжної матриці $\mathbf{s}'$. Виконаємо розрахунок для 1-го стовпця, тобто $X[k] = (10, -2 + 2j, -2, -2 - 2j)$.

Крок 1. Виконаємо перестановку відліків $X^1[0] = X[0] = 10$;
 $X^1[1] = X[2] = -2$; $X^1[2] = X[1] = -2 + 2j$; $X^1[3] = X[3] = -2 - 2j$.

Крок 2.

$$x^2[0] = X^1[0] + W_2^0 X^1[1] = 10 - 2 = 8;$$

$$x^2[1] = X^1[0] - W_2^0 X^1[1] = 10 + 2 = 12;$$

$$x^2[2] = X^1[2] + W_2^0 X^1[3] = -2 + 2j + (-2 - 2j) = -4;$$

$$x^2[3] = X^1[2] - W_2^0 X^1[3] = -2 + 2j - (-2 - 2j) = 4j.$$

Крок 3.

$$x^3[0] = x^2[0] + W_4^0 x^2[2] = 8 - 4 = 4;$$

$$x^3[1] = x^2[1] + W_4^{-1} x^2[3] = 12 + 4je^{j\frac{\pi}{2}} = 12 + 4j \cdot j = 12 - 4 = 8;$$

$$x^3[2] = x^2[0] - W_4^0 x^2[2] = 8 - (-4) = 12;$$

$$x^3[3] = x^2[1] - W_4^{-1} x^2[3] = 12 - 4je^{j\frac{\pi}{2}} = 12 - 4j \cdot j = 12 + 4 = 16.$$

Крок 4. Нормування на N , тобто $x[n] = \frac{1}{4} x^3[n]$:

$$x[n] = (1, 2, 3, 4).$$

Оскільки усі стовпці проміжної матриці рівні між собою, то отриманий двомірний сигнал має наступний вигляд

$$\mathbf{s} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix},$$

що повністю співпадає із заданим початковим сигналом.

11.4. Завдання до розділу 11 для самостійної роботи

Порахувати пряме ДПФ заданого двомірного сигналу $\mathbf{s}$. Вхідні дані дивись в табл. 11.1 (N – номер студента за журналом).

Таблиця 11.1 – Двомірні сигнали

N	$\mathbf{s}$	N	$\mathbf{s}$	N	$\mathbf{s}$
-----	--------------	-----	--------------	-----	--------------

1	$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	8	$\begin{bmatrix} 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 \end{bmatrix}$	15	$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \end{bmatrix}$
2	$\begin{bmatrix} -1 & -2 & -3 & -4 \\ 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	9	$\begin{bmatrix} 1 & 0 & 2 & 0 \\ 3 & 0 & 4 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 3 & 0 & 4 \end{bmatrix}$	16	$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix}$
3	$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 2 & 0 & -2 & 0 \\ 3 & 0 & -3 & 0 \\ 4 & 0 & -4 & 0 \end{bmatrix}$	10	$\begin{bmatrix} -1 & 0 & -2 & 0 \\ -3 & 0 & -4 & 0 \\ 0 & -1 & 0 & -2 \\ 0 & -3 & 0 & -4 \end{bmatrix}$	17	$\begin{bmatrix} -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & -2 & -2 \\ 0 & 0 & -2 & -2 \end{bmatrix}$
4	$\begin{bmatrix} -1 & 0 & 1 & 0 \\ -2 & 0 & 2 & 0 \\ -3 & 0 & 3 & 0 \\ -4 & 0 & 4 & 0 \end{bmatrix}$	11	$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 2 & 3 \\ 1 & 2 & 0 & 3 \\ 1 & 2 & 3 & 0 \end{bmatrix}$	18	$\begin{bmatrix} 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \\ -2 & -2 & 0 & 0 \\ -2 & -2 & 0 & 0 \end{bmatrix}$
5	$\begin{bmatrix} 1 & 2 & 1 & 2 \\ 0 & 4 & 0 & 4 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	12	$\begin{bmatrix} 0 & -1 & -2 & -3 \\ -1 & 0 & -2 & -3 \\ -1 & -2 & 0 & -3 \\ -1 & -2 & -3 & 0 \end{bmatrix}$	19	$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 2 \end{bmatrix}$
6	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \end{bmatrix}$	13	$\begin{bmatrix} -1 & 0 & -2 & 0 \\ -3 & 0 & -4 & 0 \\ 0 & -1 & 0 & -2 \\ 0 & -3 & 0 & -4 \end{bmatrix}$	20	$\begin{bmatrix} -1 & 0 & -1 & 0 \\ -2 & 0 & -2 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & -2 & 0 & -2 \end{bmatrix}$
7	$\begin{bmatrix} -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$	14	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 \\ -1 & -2 & -1 & -2 \\ -3 & -4 & -3 & -4 \end{bmatrix}$	21	$\begin{bmatrix} 1 & 2 & -1 & -2 \\ 1 & 2 & -1 & -2 \\ -1 & -2 & 1 & 2 \\ -1 & -2 & 1 & 2 \end{bmatrix}$

12. ПОЕЛЕМЕНТНІ ПЕРЕТВОРЕННЯ ЗОБРАЖЕНЬ

12.1. Поліпшення візуальної якості зображень шляхом поелементного перетворення

У великій кількості інформаційних систем, застосовується подання результатів обробки даних у вигляді зображення, що виводиться на екран для використання спостерігачем. Процедуру, що забезпечує таке подання, називають *візуалізацією*. Бажано за допомогою обробки надати зображенню, що виводиться, такі якості, завдяки яким його сприйняття людиною було б по можливості комфортним. Часто буває корисним підкреслити, посилити якісь риси, особливості, нюанси картини, за якою йде спостереження, з метою поліпшення її суб'єктивного сприйняття.

Останнє – суб'єктивність сприйняття – сильно ускладнює застосування формалізованого підходу в досягненні цих цілей. Тому при обробці зображень для візуалізації отримали поширення методи, в яких часто відсутні строгі математичні критерії оптимальності. Їх замінюють якісні представлення про доцільність тієї або іншої обробки, що спираються на суб'єктивні оцінки результатів.

Переважає більшість процедур обробки для отримання результату в кожній точці кадру, залучають вхідні дані з деякої безлічі точок початкового зображення, що оточують оброблювану точку. Проте є група процедур, де здійснюється, так звана, поелементна обробка. Тут результат обробки у будь-якій точці кадру залежить тільки від значення вхідного зображення в цій же точці. Очевидною перевагою таких процедур є їх гранична простота. В той же час, багато які з них призводять до очевидного суб'єктивного поліпшення візуальної якості. Цим визначається увага, яка приділяється *поелементним процедурам*. Не перебільшуючи їх ролі, відмітимо, що дуже часто поелементна обробка застосовується як завершальний етап при вирішенні складнішої задачі обробки зображення.

Суть поелементної обробки зображень зводиться до наступного.

Нехай x_{ij} , y_{ij} – значення яскравості початкового і отриманого, після обробки зображень відповідно в точці, кадру, що має декартові координати i (номер рядка) та j (номер стовпця).

Поелементна обробка означає, що існує функціональна однозначна залежність між цими яскравостями наступного виду:

$$y_{ij} = f_{ij}(x_{ij}), \quad (12.1)$$

що дозволяє за значенням початкового сигналу визначити значення вихідного продукту. У загальному випадку, як це враховано в (12.1), вид або параметри функції f_{ij} , що описує обробку, залежать від точних координат. При цьому обробка є неоднорідною. Проте у більшості практично застосовуваних процедур використовується однорідна поелементна обробка. В цьому випадку індекси i та j у виразі (12.1) можуть бути відсутніми. При цьому залежність між яскравостями початкового і обробленого зображень описується функцією, однаковою для усіх точок кадру:

$$y = f(x), \quad (12.2)$$

Розглянемо детальніше найбільш поширені процедури поелементної обробки.

12.2. Лінійне контрастування зображення

Задача *контрастування* пов'язана з поліпшенням узгодження динамічного діапазону зображення і екрану, на якому виконується візуалізація. Якщо для цифрового відображення кожного відліку зображення відводиться 1 байт (8 біт) запам'ятовуючого пристрою, то вхідний або вихідний сигнали можуть приймати одне з 256 значень. Зазвичай як робочий використовується діапазон 0...255; при цьому значення 0 відповідає при візуалізації рівню чорного, а значення 255 – рівню біло-

го. Припустимо, що мінімальна і максимальна яскравості початкового зображення дорівнюють $x_{\min}$ та $x_{\max}$ відповідно. Якщо ці параметри або один з них істотно відрізняються від граничних значень діапазону яскравості, то візуальна картина виглядає як ненасичена, незручна, що втомлює при спостереженні. Наприклад, така картина спостерігатиметься, якщо діапазон яскравостей має межі $x_{\min} = 180$, $x_{\max} = 240$.

При лінійному контрастуванні використовується лінійне поелементне перетворення виду:

$$y = ax + b, \quad (12.3)$$

де a , b – параметри, які визначаються бажаними значеннями мінімальної $y_{\min}$ і максимальної $y_{\max}$ вихідної яскравості. Розв’язавши систему рівнянь

$$\begin{cases} y_{\min} = ax_{\min} + b \\ y_{\max} = ax_{\max} + b \end{cases}$$

відносно параметрів перетворення a та b , неважко привести (12.3) до виду:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} (y_{\max} - y_{\min}) + y_{\min}. \quad (12.4)$$

12.3. Соляризація зображення

При цьому виді обробки перетворення (12.2) має вигляд:

$$y = kx(x_{\max} - x), \quad (12.5)$$

де $x_{\max}$ – максимальне значення початкового сигналу, k – константа,

що дозволяє управляти динамічним діапазоном перетвореного зображення.

Функція, що описує це перетворення, є квадратичною параболою, її графік при $k=1$ приведено на рис. 12.1. При $y_{\max} = x_{\max}$ динамічні діапазони зображень співпадають, що може бути досягнуто при $k = 4x_{\max}$.

Інший шлях нормалізації динамічного діапазону може полягати в застосуванні одного з методів контрастування, наприклад, описаного вище лінійного контрастування.

Сенс соляризації полягає в тому, що ділянки початкового зображення, що мають рівень білого або близький до нього рівень яскравості, після обробки мають рівень чорного. При цьому зберігають рівень чорного і ділянки, що мають його на початковому зображенні. Рівень же білого на виході набувають ділянки, що мають на вході середній рівень яскравості (рівень сірого).

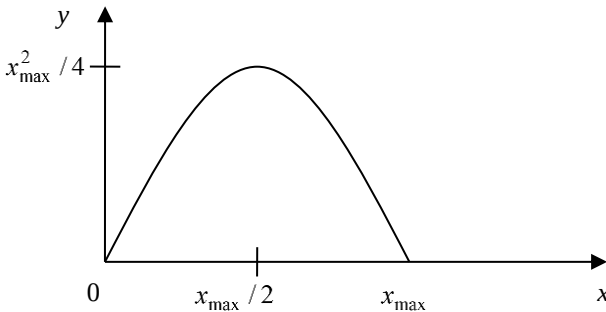


Рисунок 12.1 – Функція, що описує соляризацію

12.4. Препарування зображення

Препарування є цілим класом поелементних перетворень зображень. Характеристики застосовуваних на практиці процедур препарування наведені на рис. 12.2. Зупинимося на описі деяких з них.

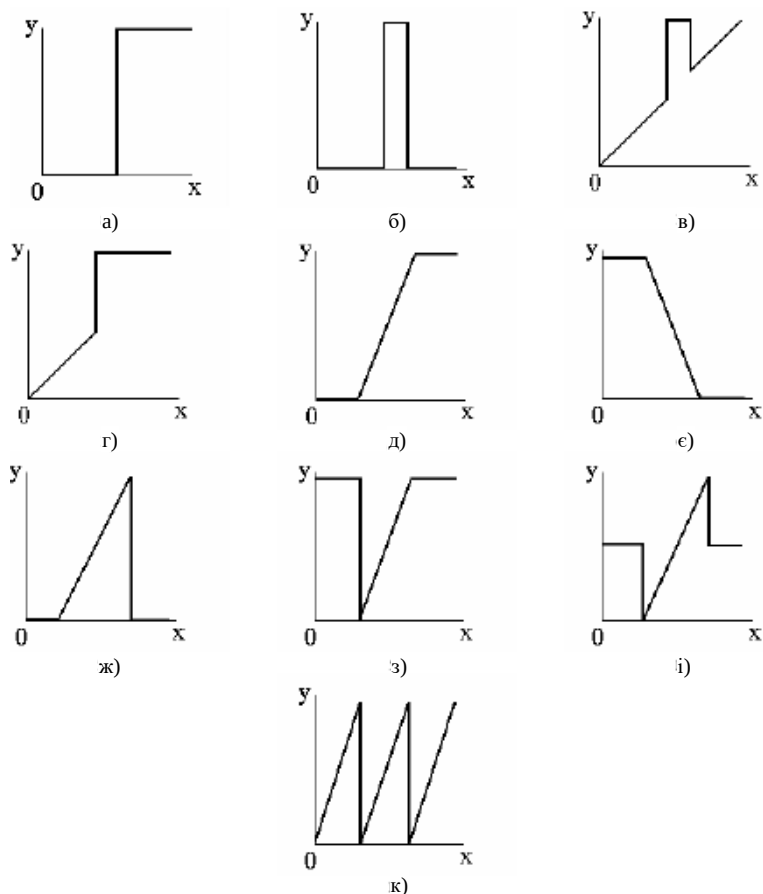


Рисунок 12.2 – Приклади перетворень, що застосовуються при препаруванні

Перетворення з пороговою характеристикою (рис. 12.2а) перетворює напівтонове зображення, що містить усі рівні яскравості, у бінарне, точки якого мають яскравості $y = 0$ або $y = y_{\max}$. Така операція, що називається іноді бінаризацією або бінарним квантуванням, може бути корисною, коли для спостерігача важливі контури об'єктів, присутніх на зображенні, а деталі, що містяться усередині об'єктів або усередині

фону, не представляють інтересу. Основною проблемою, при проведенні такої обробки, є визначення порогу x_0 , порівняння яскравості початкового зображення з яким дозволяє визначити значення вихідного зображення в кожній його точці. Найбільш виправданим, для математичного опису зображення, є застосування теорії ймовірностей, випадкових процесів і випадкових полів. При цьому, визначення оптимального порогу бінарного квантування є статистичною задачею.

Іноді при обробці доводиться мати справу зі зображеннями, що зберігаються як напівтонові, але за своїм змістом мало відрізняються від бінарних. До них відносяться текст, штрихові малюнки, креслення і таке інше. Якщо щільність вірогідності $w(x)$, що описує розподіл яскравості зображення, містить два добре розділених піки, то інтуїтивно зрозуміло, що поріг бінарного квантування слід вибирати посередині провалу між цими піками, як це показано на рис. 12.3.

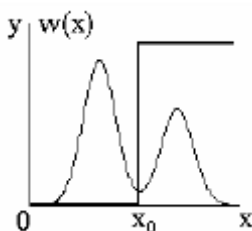


Рисунок 12.3 – Вибір порогу бінарного квантування

Заміна початкового напівтонового зображення бінарним препаратом вирішує дві основні задачі. По-перше, досягається велика наглядність при візуальному сприйнятті, на відміну від початкового зображення. По-друге, відчутно скорочується об'єм пам'яті для зберігання зображення, оскільки бінарний препарат для запису кожної точки бінарного зображення вимагає лише 1 біт пам'яті, тоді як півтонове зображення для вирішення тієї ж задачі при найчастіше застосовуваному форматі представлення – 8 біт.

Суть інших перетворень, представлених на рис. 12.2, неважко

зрозуміти, розглядаючи їх характеристики. Наприклад, перетворення рис. 12.2б виконує яскравісний зріз зображення, виділяючи ті його ділянки, де яскравість відповідає виділеному інтервалу. При цьому інші ділянки виявляються повністю «погашеними» (мають яскравість, що відповідає рівню чорного). Переміщаючи виділений інтервал за яскравісною шкалою і змінюючи його ширину, можна детально досліджувати зміст картини.

Перетворення, приведене на рис. 12.2ж, також дозволяє підвищити детальність картини, що спостерігається, у вибраному діапазоні яскравостей, проте, на відміну від попереднього, тут вихідне зображення використовує повний динамічний діапазон. По суті, це перетворення є лінійним контрастуванням, що застосовується до обраного діапазону вхідного зображення. Як і в попередньому варіанті, ділянки, що не потрапили в цей діапазон, утворюють після препарування чорний фон.

Іноді наглядність зображення підвищується застосуванням перетворення типу пілкоподібного контрастування (рис. 12.2к). При цьому різні діапазони яскравості одночасно піддаються локальному яскравісному контрастуванню. Проте, необхідно мати на увазі, що це перетворення, як і деякі інші, може супроводжуватися появою неправдивих контурів на отриманому препараті.

Аналогічно можна якісно розглянути і інші процедури препарування, що представлені на рис. 12.2.

12.5. Поелементна обробка зображень в MATLAB

12.5.1. Бінаризація відсіканням за порогом яскравості

Функція **BW = im2bw(I,level)** перетворює повнокольорове або палітрове зображення у бінарне шляхом перетворення його спочатку на напівтонове зображення, а потім застосування операції відсікання за порогом яскравості. Пікселі перетвореного зображення мають значення 0 (чорний колір), якщо рівень порогу **level** не перевищений і 1 (білий колір) в іншому випадку. Поріг задається в межах [0, 1.0]. Ці

операції називають бінаризацією зображення.

BW=im2bw(X,map,level) – здійснює те ж, але для зображення, заданого як початковим масивом **X**, так і палітрою **map**. Функція **BW=im2bw(RGB,level)** перетворює RGB-зображення в чорно-біле зображення. Вхідні аргументи можуть бути класів **uint8**, **uint16** або **double**, вихідний параметр **BW** – класу **uint8**.

У наведеному нижче прикладі завантажується і будується зображення фрагменту інтегральної мікросхеми, після чого воно перетворюється у бінарне зображення (рис. 12.4):

```
%Програма №12.1  
I=imread('circuit.tif');  
BW = im2bw(I, 0.2);  
subplot(1,2,1); imshow(I);  
subplot(1,2,2); imshow(BW)
```

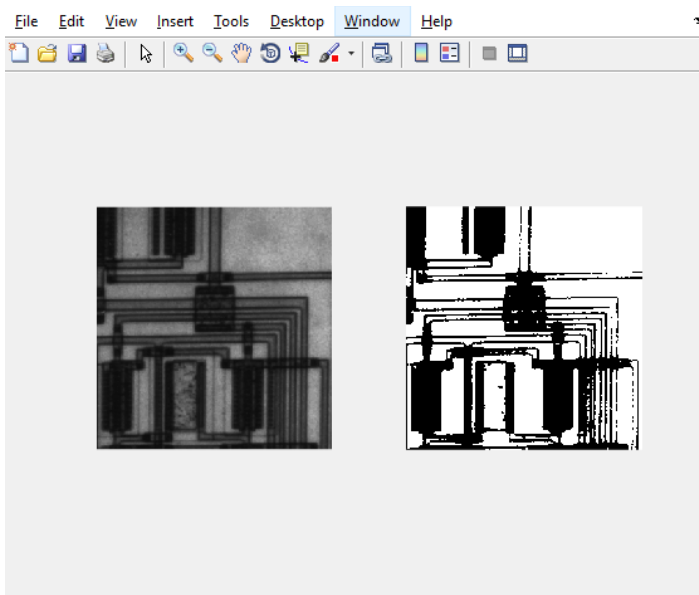


Рисунок 12.4 – Півтонове зображення мікросхеми (ліворуч) і результат його перетворення у бінарне зображення з порогом 0.2

Неважко помітити, що перетворення у бінарне зображення дозволяє різко виділити окремі фрагменти початкового зображення. Проте, у більшості випадків таке виділення виявляється занадто грубим. Виключенням, мабуть, є виділення зображень з текстовими написами. Наступний приклад (рис. 12.4) ілюструє перетворення зображення в чорно-біле з різним рівнем порогу:

```
%Програма №12.2
X = imread('rice.png');
BW1 = im2bw(X, 0.7);
BW2 = im2bw(X, 0.5);
subplot(1,3,1); imshow(X);
subplot(1,3,2); imshow(BW1);
subplot(1,3,3); imshow(BW2);
```

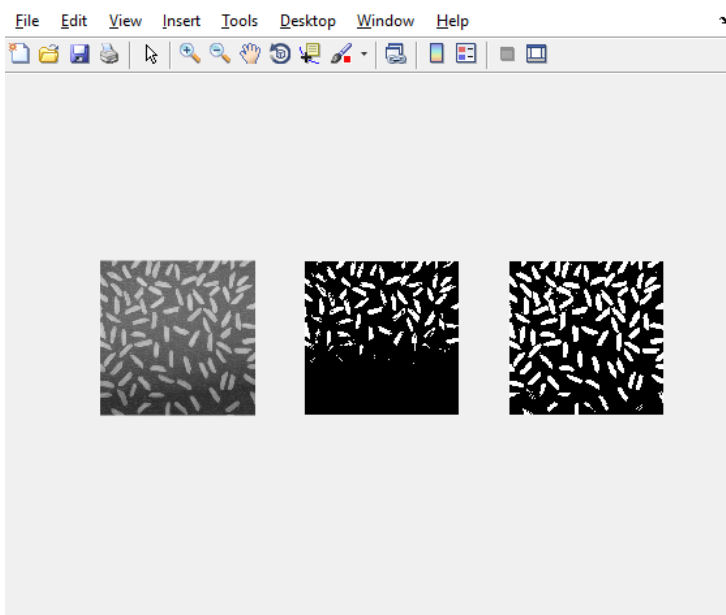


Рисунок 12.4 – Півтонове зображення рису і результати його перетворення у бінарне з різними значеннями порогів

12.5.2. Вирівнювання гістограми зображення

Одним із способів поліпшення зображення, наприклад, шляхом підвищення або пониження його контрастності, є вирівнювання його гістограми, що представляє розподіли яскравості пікселів зображення. Цей спосіб реалізується функцією **histeq**, що має ряд форм, розглянутих нижче.

Функція **J=histeq(I,hgram)** перетворить гістограму зображень, намагаючись підігнати її під зразок гістограми, заданий у векторі **hgram**. Число його елементів задає число стовпців гістограми, а значення елементів в межах від 0 до 1 – висоту стовпців гістограми.

Функція **histeq** автоматично масштабує гістограму **hgram** так, щоб виконувалася умова **sum(hgram) = prod(sized)**.

Функція **J=histeq(d,n)** дозволяє задати число стовпців гістограми з рівномірним їх розподілом (але за замовчуванням **n=64**). Воно має бути набагато менше, ніж число градацій яскравості початкового зображення.

Функція **[J,T] = histeq(I,...)** додатково повертає вектор значень **T** характеристики передачі яскравості зображення в ході його поліпшення.

newmap=histeq(X,map) перетворить палітру яскравостей початкового зображення **X** у **newmap** так, щоб забезпечити приблизно рівномірну яскравість пікселів.

newmap=histeq(X,map,hgram) перетворить палітру яскравостей **map** початкового зображення **X** так, щоб забезпечити відповідність до гістограми зразка **hgram**.

[newmap,T]=histeq(X,...) додатково повертає вектор відтінків сірого кольору.

Матриця **I** може бути класів **uint8**, **uint16** або **double**. Вихідна матриця **J** має той самий клас, що й **I**. Матриця **X** палітрового зображення може бути класів **uint8** або **double**; палітра кольорів на виході завжди має клас **double** (той самий клас має і вектор **T**).

Наступний приклад показує побудову початкового зображення хлопчика з файлу **rout.tif**, гістограму його, а потім поліпшене функці-

сю **histeq** зображення і відповідну гістограму (рис. 12.5):

```
%Програма №12.3  
I = imread('pout.tif');  
J = histeq(I);  
subplot(2,2,1);  
imshow(I);  
subplot(2,2,2);  
imhist(I);  
subplot(2,2,3);  
imshow(J);  
subplot(2,2,4);  
imhist(J);
```

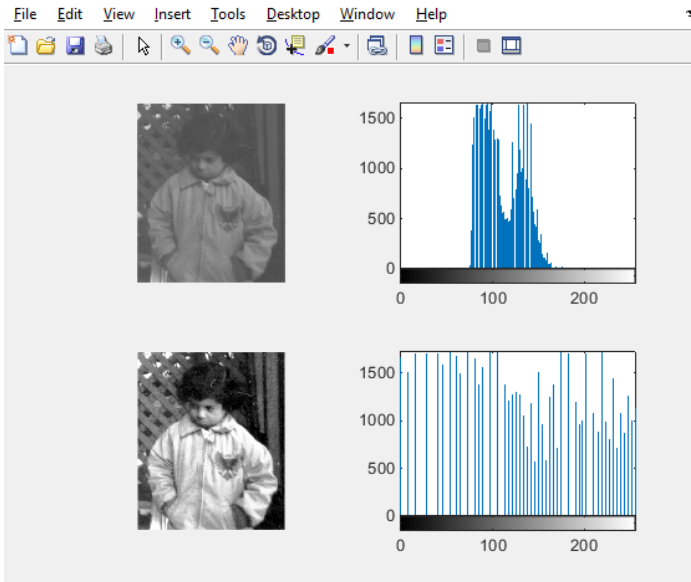


Рисунок 12.5 – Початкове зображення хлопчика, його гістограма (зверху) і поліпшене зображення і його гістограма (знизу)

На рис. 12.5 представлено початкове зображення хлопчика. Це явно перетримана стара фотографія дуже низької якості. Контрастність

знімка явно недостатня. Гістограма знімка показує скупченість ліній по її центру. Після обробки функцією **histeq** виразно видно поліпшення якості перетвореного зображення, що підтверджує і його гістограма.

12.5.3. Зміна яскравості палітри

Нерідко зображення виявляються занадто світлими (недотримання у фотографії) або занадто темними (перетримання). Ці дефекти обумовлені тим або іншим видом передаточної характеристики за яскравістю – залежності яскравості пікселя від його значення – рис. 12.6. При ідеальній апаратурі ця залежність виключно лінійна, принаймні, в інтервалі зміни значення пікселя від **low** до **high**. Зазвичай ця залежність має ділянки обмеження за яскравістю знизу (**bottom**) і зверху (**top**). У проміжку вона пропорційна величині γ .

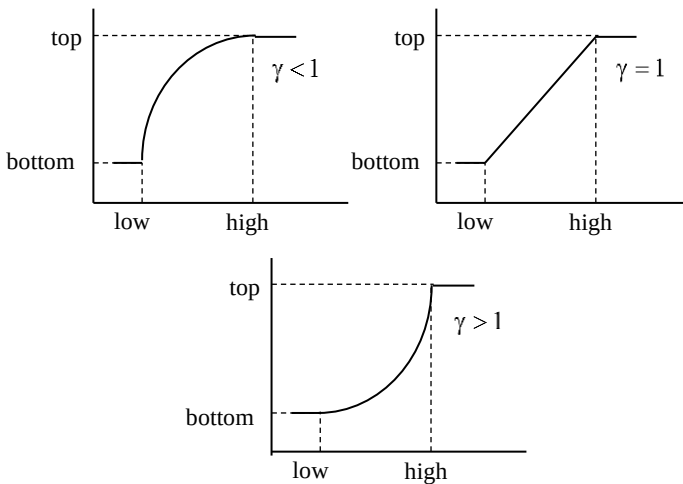


Рисунок 12.6 – Передаточна характеристика яскравості пікселя при різних значеннях γ

При $\gamma = 1$ ми маємо лінійну передаточну характеристику (лінійне

контрастування). Якщо $\gamma < 1$, передаточна характеристика буде опуклою, і перетворене зображення буде світліше, ніж оригінальне. При $\gamma > 1$ передаточна характеристика буде увігнутою, і перетворене зображення буде темнішим, ніж оригінальне. Зміну показника γ називають гамма-корекцією зображення.

Для зображень низької якості (недотримані або перетримані фотографії) введення гамма-корекції нерідко істотно покращує їх якість – помітно вище, ніж просто підвищення або пониження яскравості зображення.

Функція

J=imadjust(I, [low_in high_in], [low_out high_out], gamma)

створює зображення **J** з початкового зображення **I** при заданому параметрі гамма-корекції **gamma** з порогами обмеження по входу (значенню пікселя) **low_in** і **high_in** та виходу **low_out** і **high_out**. Діапазони **[low_in high_in]** і **[low_out high_out]** задаються в межах від 0 до 1. Замість вхідних та/або вихідних порогів може задаватися пуста матриця **[]**, що означає задання вектору **[low_in high_in]** та/або **[low_out high_out]** у максимально можливому діапазоні.

RGB2=imadjust(RGB1...) задає перетворення для RGB-зображення.

Якщо **high_out < low_out**, то матиме місце негативний вид зображення.

Вхід і вихід функцій можуть відноситися до класів **uint8**, **uint16** або **double**, а карта вихідних кольорів до класу **double**.

У наведеному нижче прикладі виконане лінійне контрастування зображення хлопчика двома способами (рис. 12.7) – за допомогою функції **imadjust** і за виразом (12.4):

```
%Програма №12.4
I = imread('pout.tif');
mx=double(max(max(I))); %максимальне значення
mn=double(min(min(I))); %мінімальне значення
%лінійне контрастування за допомогою imadjust
```

```

J = imadjust(I,[mn/255 mx/255],[],1);
subplot(3,2,1);
imshow(I);
subplot(3,2,2);
imhist(I);
subplot(3,2,3);
imshow(J);
subplot(3,2,4);
imhist(J);
%лінійне контрастування за виразом (12.4)
J2=uint8((double(I)-mn)/(mx-mn)*255);
subplot(3,2,5);
imshow(J2);
subplot(3,2,6);
imhist(J2);

```

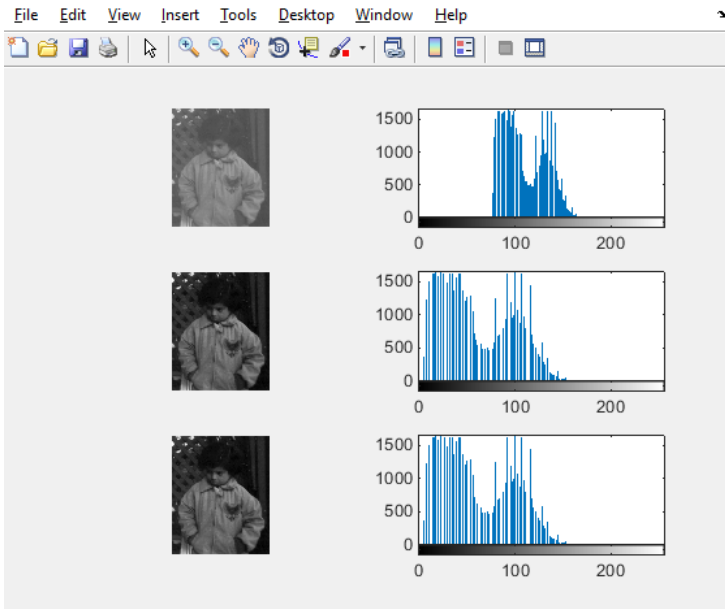


Рисунок 12.7 – Результати лінійного контрастування

З другого і третього зображень (див. рис. 12.7) видно, що результати, які отримані двома способами, ідентичні.

У наступному прикладі вже розглянуте зображення хлопчика поліпшене за допомогою гамма-корекції при $\gamma < 1$ (рис. 12.8):

```
%Програма №12.5
I = imread('pout.tif');
J = imadjust(I,[0.3 0.7],[],0.8); %гамма 0.8
subplot(2,2,1); imshow(I);
subplot(2,2,2); imhist(I);
subplot(2,2,3); imshow(J);
subplot(2,2,4); imhist(J);
```

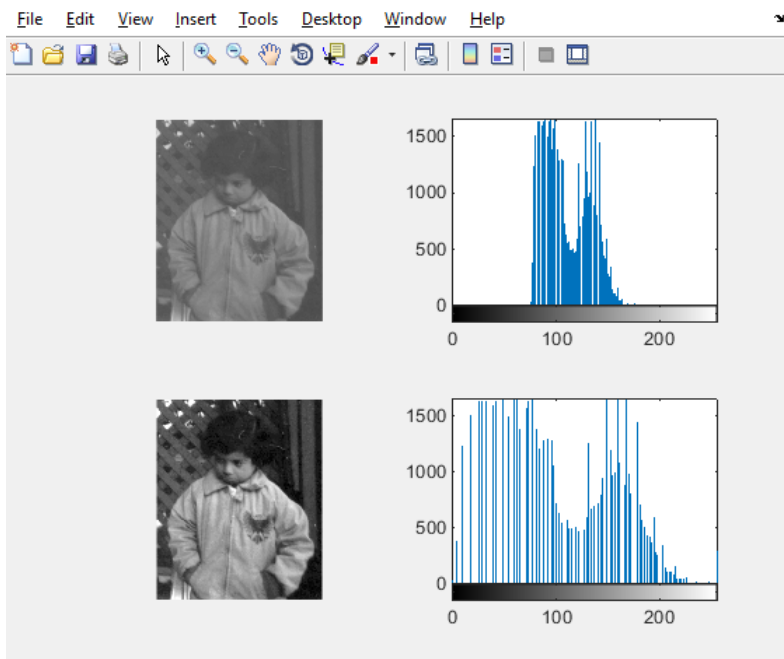


Рисунок 12.8 – Початкове зображення хлопчика і його гістограма (зверху) і поліпшене після гамма-корекції зображення і його гістограма (знизу)

Порівняння результатів двох останніх прикладів (див. рис. 12.7, 12.8) показує, що з допомогою гамма-корекції можна отримати кращі результати, ніж за допомогою лінійного контрастування.

12.5.4. Соляризація

У наведеному нижче прикладі завантажується і будується зображення фрагменту інтегральної мікросхеми, після чого виконується соляризація зображення з $k = 1$ та лінійне контрастування (рис. 12.9):

```
%Програма №12.6
I = imread('circuit.tif');
mx=255; %максимальне значення
mn=min(min(I)); %мінімальне значення
subplot(2,2,1);
imshow(I);
subplot(2,2,2);
imhist(I);
k=1;
%соляризація за виразом (12.5)
J1=double(k*double(I).*(mx-double(I)));
mxJ=max(max(J1)); %максимальне значення
mnJ=min(min(J1)); %мінімальне значення
%лінійне контрастування за виразом (12.4)
J2=uint8((double(J1)-mnJ)/(mxJ-mnJ)*255);
subplot(2,2,3);
imshow(J2);
subplot(2,2,4);
imhist(J2);
```

В результаті обробки вдалося виділити елементи схеми за рахунок висвітлення сірого фону, який в результаті соляризації став білим, при тому темні деталі осталися темними. Для досягнення кращого результату як максимальне значення для виразу (12.5) було обране значення 255 (максимальне можливе), а не реальне значення 191.

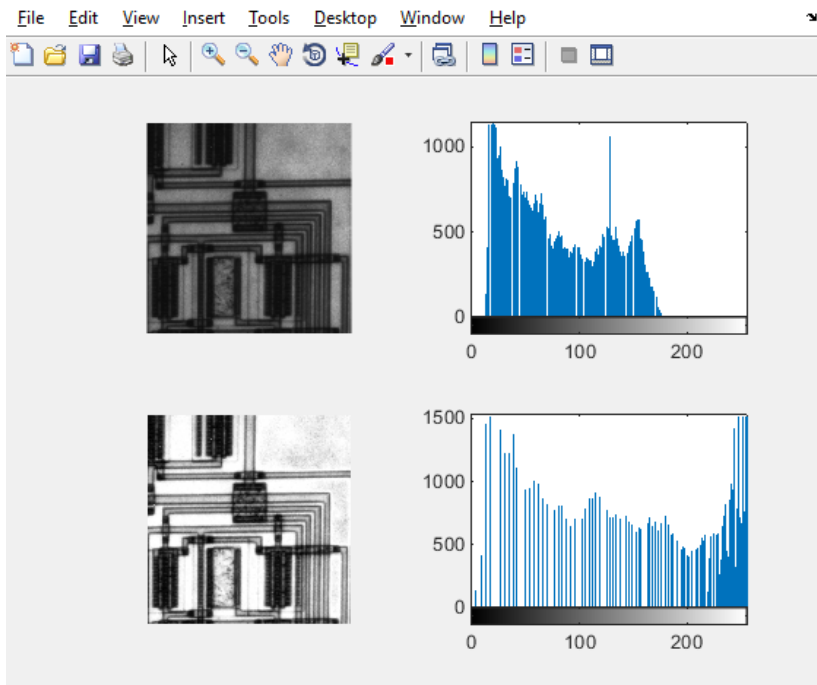


Рисунок 12.9 – Результати соляризації ($k = 1$)

12.6. Приклади до розділу 12

Приклад №12.1. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати лінійне контрастування з параметрами $y_{\max} = 30$, $y_{\min} = 1$.

Розв'язок.

Для розрахунку скористаємося вираженням (12.4). Знайдемо максимальне і мінімальне значення двомірного сигналу X : $x_{\max} = 45$, $x_{\min} = 0$. Підставивши параметри контрастування в (12.4), отримаємо

$$y = \frac{x}{45} (30 - 1) + 1 = \frac{x}{45} 29 + 1.$$

Тоді для першого рядка маємо:

$$y_{00} = \frac{x_{00}}{45} 29 + 1 = \frac{34}{45} 29 + 1 = 22.911;$$

$$y_{01} = \frac{x_{01}}{45} 29 + 1 = \frac{9}{45} 29 + 1 = 6.8;$$

$$y_{02} = \frac{x_{02}}{45} 29 + 1 = \frac{40}{45} 29 + 1 = 26.778;$$

$$y_{03} = \frac{x_{03}}{45} 29 + 1 = \frac{20}{45} 29 + 1 = 13.889.$$

Отже, після розрахунку усіх відліків отримаємо наступний двомірний сигнал:

$$Y = \begin{bmatrix} 22.911 & 6.800 & 26.778 & 13.889 \\ 11.956 & 14.533 & 15.822 & 30.000 \\ 6.156 & 28.067 & 15.178 & 1.000 \\ 5.511 & 16.467 & 15.178 & 10.022 \end{bmatrix}.$$

Тепер необхідно округлити отримані значення. В результаті отримуємо наступне зображення:

$$I = \begin{bmatrix} 23 & 7 & 27 & 14 \\ 12 & 15 & 16 & 30 \\ 6 & 28 & 15 & 1 \\ 61 & 16 & 15 & 10 \end{bmatrix}.$$

Приклад №12.2. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати соляризацію при $k = 1$.

Розв'язок.

Для розрахунку скористаємося виразом (12.5). Знайдемо максимальне значення двовірного сигналу X : $x_{\max} = 45$. Підставивши параметри в (12.5), отримаємо $y = x(45 - x)$.

Тоді для першого рядка маємо:

$$y_{00} = x_{00}(45 - x_{00}) = 34(45 - 34) = 374;$$

$$y_{01} = x_{01}(45 - x_{01}) = 9(45 - 9) = 324;$$

$$y_{02} = x_{02}(45 - x_{02}) = 40(45 - 40) = 200;$$

$$y_{03} = x_{03}(45 - x_{03}) = 20(45 - 20) = 500.$$

$$Y = \begin{bmatrix} 374 & 324 & 200 & 500 \\ 476 & 504 & 506 & 0 \\ 296 & 126 & 506 & 0 \\ 266 & 504 & 506 & 434 \end{bmatrix}.$$

Очевидно, що після соляризації потрібний етап лінійного контрастування, щоб привести отримане зображення до заданого динамічного діапазону яскравості. Виконаємо лінійне контрастування з тими ж параметрами, що і в попередньому прикладі. В результаті отримаємо

$$I = \begin{bmatrix} 22.4348 & 19.5692 & 12.4625 & 29.6561 \\ 28.2806 & 29.8854 & 30.0000 & 1.0000 \\ 17.9644 & 8.2213 & 30.0000 & 1.0000 \\ 16.2451 & 29.8854 & 30.0000 & 25.8735 \end{bmatrix}.$$

Після округлення отримуємо наступне результуюче зображення:

$$J = \begin{bmatrix} 22 & 20 & 12 & 30 \\ 28 & 30 & 30 & 1 \\ 18 & 8 & 30 & 1 \\ 16 & 30 & 30 & 26 \end{bmatrix}.$$

Легко помітити, що пікселі з середнім значення яскравості початкового зображення (значення від 20 до 24), отримали максимальні значення після соляризації. При цьому пікселі як з максимальним значенням яскравості ($x_{13} = 45$), так і з мінімальним значенням яскравості ($x_{23} = 0$) після соляризації отримали мінімальне значення ($j_{13} = 1$, $j_{23} = 1$).

Приклад №12.3. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати препарування згідно із законом, зображеному на рис. 12.10.

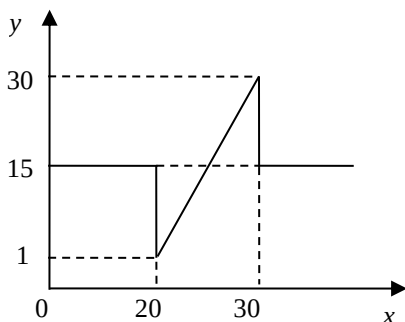


Рисунок 12.10 – Перетворення для препарування зображення

Розв'язок.

Згідно із законом для препарування (рис. 12.10) запишемо правило перетворення

$$y = \begin{cases} 15, & \text{якщо } x < 20 \text{ або } x > 30; \\ ax + b, & \text{якщо } 20 \leq x \leq 30. \end{cases}$$

З закону видно, що для $20 < x < 30$ треба виконати лінійне контрастування (12.4), при цьому $x_{\min} = 20$, $x_{\max} = 30$, $y_{\min} = 1$, $y_{\max} = 30$.

Тоді

$$y = \begin{cases} 15, & \text{якщо } x < 20 \text{ або } x > 30; \\ \frac{x - 20}{30 - 20} (30 - 1) + 1, & \text{якщо } 20 \leq x \leq 30, \end{cases}$$

$$y = \begin{cases} 15, & \text{якщо } x < 20 \text{ або } x > 30; \\ 2.9(x - 20) + 1, & \text{якщо } 20 \leq x \leq 30, \end{cases}$$

Отже, після розрахунку отримаємо наступний двовірний сигнал:

$$Y = \begin{bmatrix} 15 & 15 & 15 & 1 \\ 15 & 3.9 & 9.7 & 15 \\ 15 & 15 & 6.8 & 15 \\ 15 & 12.6 & 6.8 & 15 \end{bmatrix}.$$

Тепер необхідно округлити отримані значення. В результаті отримуємо наступне зображення:

$$Y = \begin{bmatrix} 15 & 15 & 15 & 1 \\ 15 & 4 & 10 & 15 \\ 15 & 15 & 7 & 15 \\ 15 & 13 & 7 & 15 \end{bmatrix}.$$

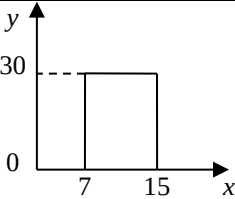
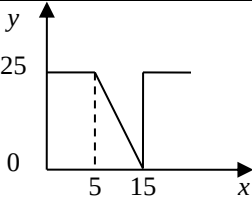
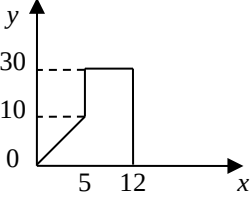
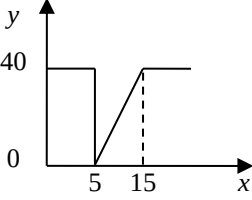
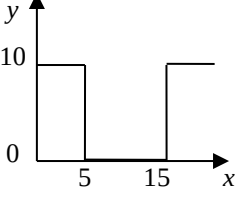
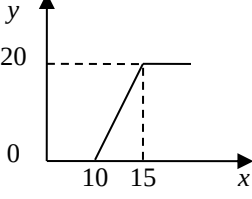
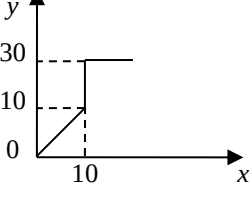
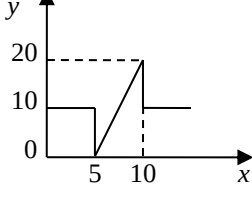
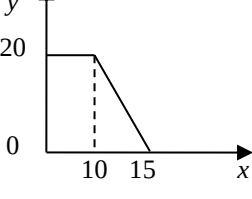
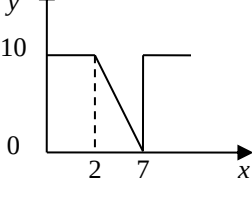
12.7. Завдання до розділу 12 для самостійної роботи

Дано двовірний сигнал

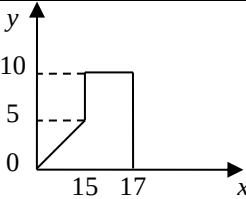
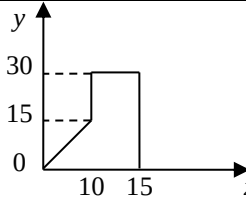
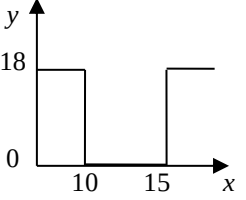
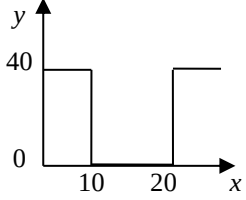
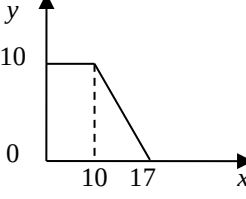
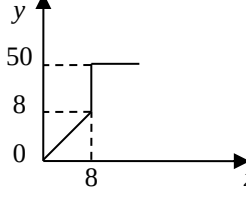
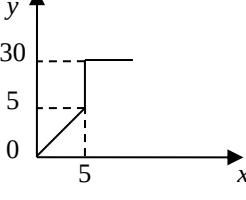
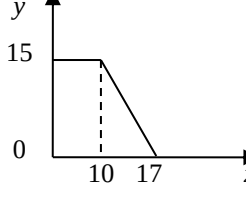
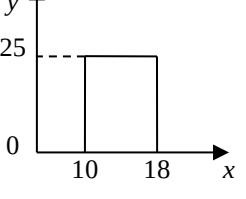
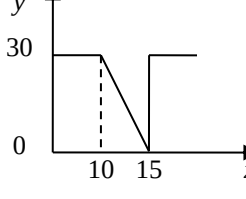
$$x = \begin{bmatrix} 14 & 16 & 20 & 3 & 13 \\ 11 & 19 & 5 & 0 & 6 \\ 9 & 10 & 5 & 18 & 9 \\ 14 & 18 & 18 & 4 & 1 \\ 12 & 3 & 15 & 6 & 20 \end{bmatrix}.$$

Виконати препарування двовірного сигналу за заданим законом.
Варіанти завдань в табл. 12.1 (N – номер студента за журналом).

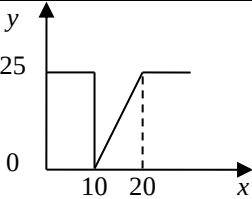
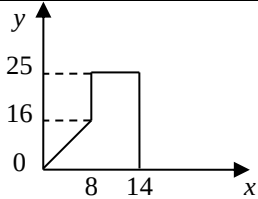
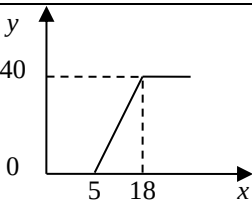
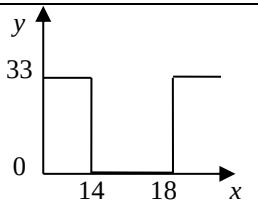
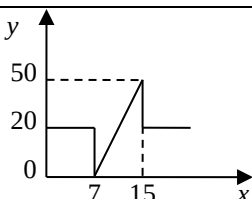
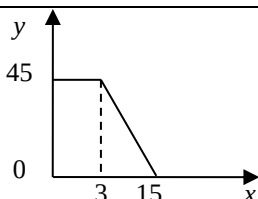
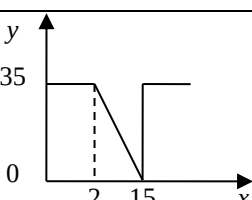
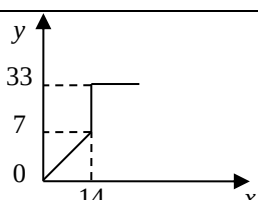
Таблиця 12.1 – Варіанти законів для препарування

N	Функція препарування	N	Функція препарування
1		6	
2		7	
3		8	
4		9	
5		10	

Продовження табл. 12.1

N	Функція препарування	N	Функція препарування
11		16	
12		17	
13		18	
14		19	
15		20	

Продовження табл. 12.1

N	Функція препарування	N	Функція препарування
21		25	
22		26	
23		27	
24		28	

13. ЛІНІЙНА ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ

13.1. Основні поняття

Зазвичай зображення, сформовані різними інформаційними системами, спотворюються дією завад. Це ускладнює як їх візуальний аналіз людиною-оператором, так і автоматичну обробку в ЕОМ. При рішенні деяких задач обробки зображень в ролі завад можуть виступати ті або інші компоненти самого зображення. Послаблення дії завад досягається *фільтрацією*. При фільтрації яскравість (сигнал) кожної точки початкового зображення, спотвореного завадою, замінюється деяким іншим значенням яскравості, яке визнається в найменшій мірі спотвореним завадою.

Зображення часто є двомірною функцією просторових координат, що змінюється по цих координатах повільніше (іноді значно повільніше), ніж завада, яка також є двомірною функцією. Це дозволяє при оцінці корисного сигналу в кожній точці кадру взяти до уваги деяку безліч сусідніх точок, скориставшись певною схожістю сигналу в цих точках. У інших випадках, навпаки, ознакою корисного сигналу є різкі перепади яскравості. Проте, як правило, частота цих перепадів відносно невелика, так що на значних проміжках між ними сигнал або постійний, або змінюється повільно. І в цьому випадку властивості сигналу проявляються при спостереженні його не лише в локальній точці, але й при аналізі її околу.

Звернемо увагу, що поняття околу є досить умовним. Він може бути утворений лише найближчими по кадру сусідами, але можуть бути околи, що містять досить багато і досить сильно видалених точок кадру. У цьому останньому випадку, звичайно, міра впливу далеких і близьких точок на рішення, що приймаються фільтром в цій точці кадру, буде абсолютно різною. Отже, *ідеологія фільтрації* ґрунтується на раціональному використанні даних як з робочої точки, так і з її околу. У цьому проявляється істотна відмінність фільтрації від розглянутих вище поелементних процедур: фільтрація не може бути поелементною

процедурою обробки зображень.

13.2. Лінійні фільтри

Лінійні фільтри мають дуже простий математичний опис. Вважатимемо, що задано початкове півтонове зображення A , і позначимо інтенсивності його пікселів $A[x, y]$. Лінійний фільтр визначається дійсною функцією h (ядром фільтру), заданою на растрі. Сама фільтрація робиться за допомогою операції дискретної згортки (зваженого підсумовування):

$$B[x, y] = h[i, j] ** A[x, y] = \sum_i \sum_j h[i, j] A[x - i, y - j]. \quad (13.1)$$

Результатом служить зображення B . Звичайне ядро фільтру відмінно від нуля тільки в деякому околі N точки $(0,0)$. За межами цього околу $h[i, j]$ дорівнює нулю, або дуже близько до нього і можна їм нехтувати. Підсумовування робиться по $(i, j) \in N$, і значення кожного пікселя $B[x, y]$ визначається пікселями зображення A , які лежать у вікні N , що центрується в точці (x, y) (позначення – множина $N(x, y)$). Ядро фільтру, задане на прямокутному околі N , може розглядатися як матриця $n \times t$, де довжини сторін є непарними числами. Задаючи ядро матриці її слід центрувати. Якщо піксель знаходиться в околі країв зображення, то координати $A[x - i, y - j]$ для визначених (i, j) можуть відповідати неіснуючим пікселям A за межами зображення. Цю проблему можна вирішити декількома способами:

- не проводити фільтрацію для таких пікселів, обрізавши зображення B по краях, або застосувавши для їх значень початкові значення зображення A ;
- не включати відсутній піксель в підсумовування, розподіливши його вагу $h[i, j]$ рівномірно серед інших пікселів околу $N(x, y)$;

- довизначити значення пікселів за межами зображення за допомогою екстраполяції;
- довизначити значення пікселів за межами зображення, за допомогою дзеркального продовження зображення.

Вибір способу відбувається враховуючи конкретний фільтр та особливості зображення.

13.2.1. Фільтри згладжування

Простий прямокутний фільтр згладжування радіусу r задається за допомогою матриці розміру $(2r+1) \times (2r+1)$, усі значення якої дорівнюють $1/(2r+1)^2$, а сума значень дорівнює одиниці. Це двовірний аналог низькочастотного одновірного П-подібного фільтра ковзного середнього. При фільтрації з таким ядром значення пікселя замінюється усередненим значенням пікселів в квадраті із стороною $2r+1$ навколо нього. Приклад маски фільтру 3×3 :

$$M_1^{low} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Одним із застосувань фільтрів є шумозаглушення. Шум міняється незалежно від пікселя до пікселя i , за умови, що математичне очікування значення шуму дорівнює нулю, шуми сусідніх пікселів при підсумовуванні компенсуватимуть один одного. Чим більше вікна фільтрації, тим менше буде усереднена інтенсивність шуму, проте при цьому відбуватиметься і відповідне розмиття значущих деталей зображення. Образом білої точки на чорному фоні при фільтрації (реакція на одиничний імпульс) буде рівномірно сірий квадрат.

Шумозаглушення за допомогою прямокутного фільтру має істотний недолік: усі пікселі в масці фільтру, на будь-якій відстані від оброблюваного, впливають на результат з однаковим ефектом. Дещо

кращий результат виходить при модифікації фільтру зі збільшенням ваги центральної точки:

$$M_2^{low} = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ або } M_3^{low} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

Ефективніше шумозаглушення можна здійснити, якщо вплив пікселів на результат зменшуватиметься зі збільшенням відстані від оброблюваного пікселя. Цю властивість має фільтр Гауса з ядром:

$$h[i, j] = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}.$$

Фільтр Гауса має ненульове ядро нескінченного розміру. Проте значення ядра фільтру дуже швидко спадає до нуля при віддаленні від точки (0,0), і тому на практиці можна обмежитися згортокою з вікном невеликого розміру навколо (0,0), наприклад, узявши радіус вікна рівним 3σ .

Фільтрація Гауса також є згладжуючою. Проте, на відміну від прямокутного фільтру, образом точки при фільтрації Гауса буде симетрична розмита пляма, зі зменшенням яскравості від середини до країв. Міра розмиття зображень визначається параметром σ .

13.2.2. Фільтри підвищення контрастності

Якщо фільтри згладжування знижують локальну контрастність зображення, розмиваючи його, то фільтри, що підвищують контрастність, здійснюють зворотне враження і, по суті, є фільтрами високих просторових частот. Ядро фільтру, що підвищує контрастність, в точці (0,0) має значення, більше 1, при загальній сумі значень, що дорівнює 1. Наприклад, фільтрами, що підвищують контрастність, є фільтри з

ядром, що задається матрицями:

$$M_1^{contr} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad M_2^{contr} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix},$$

$$M_3^{contr} = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}.$$

Ефект підвищення контрасту досягається за рахунок того, що фільтр підкреслює різницю між інтенсивностями сусідніх пікселів, віддаляючи ці інтенсивності один від одного. Цей ефект буде тим сильнішим, чим більшим є значення центрального члена ядра. Характерним артефактом лінійної фільтрації, що підвищує контрастність, є помітні світлі і менш помітні темні ореоли навколо меж.

13.2.3. Різницеві фільтри

Різницеві фільтри – це лінійні фільтри, що задаються дискретними апроксимаціями диференціальних операторів (за методом скінченних різниць). Ці фільтри грають найважливішу роль у багатьох додатках, наприклад, для задач пошуку меж на зображенні.

Простим диференціальним оператором є взяття похідної по x -координаті d/dx , яка визначена для безперервних функцій. Поширеними варіантами аналогічних операторів для дискретних зображень є фільтри Прюїтта (Prewitt) і Собеля (Sobel):

$$M_1^{prewitt} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad M_1^{sobel} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}.$$

Фільтри, що наближають оператор похідної по y -координаті d/dy , отримують шляхом транспонування матриць.

Простий алгоритм обчислення норми градієнта по трьох суміжних точках:

$$G[x, y] = \sqrt{(A[x+1, y] - A[x, y])^2 + (A[x, y+1] - A[x, y])^2}.$$

Застосовується також спрощена формула обчислень:

$$G[x, y] \cong |A[x+1, y] - A[x, y]| + |A[x, y+1] - A[x, y]|.$$

Обчислення норми градієнта по чотирьох суміжних точках (оператор Робертса):

$$G[x, y] = \sqrt{(A[x+1, y+1] - A[x, y])^2 + (A[x+1, y] - A[x, y+1])^2},$$

$$G[x, y] \cong |A[x+1, y+1] - A[x, y]| + |A[x+1, y] - A[x, y+1]|.$$

У алгоритмі Собеля використовується вісім відліків яскравості в околі центральної точки:

$$G[x, y] = \sqrt{G_x^2[x, y] + G_y^2[x, y]}, \quad G[x, y] = |G_x[x, y] + G_y[x, y]|,$$

$$G_x[x, y] = (A[x-1, y-1] + 2A[x-1, y] + A[x-1, y+1]) - \\ (A[x+1, y-1] + 2A[x+1, y] + A[x+1, y+1]),$$

$$G_y[x, y] = (A[x-1, y-1] + 2A[x, y-1] + A[x+1, y-1]) - \\ (A[x-1, y+1] + 2A[x, y+1] + A[x+1, y+1]).$$

Разом з точнішим визначенням норми градієнта алгоритм Собеля дозволяє визначати і напрям вектору градієнта в площині аналізу зображення у вигляді кута ϕ між вектором градієнта і напрямом рядків матриці:

$$\phi[x, y] = \arctg (G_y[x, y] / G_x[x, y]) .$$

На відміну від фільтрів згладження та контрастопідвищення, що не міняють середню інтенсивність зображення, в результаті застосування різницевих операторів, виходить, як правило, зображення з середнім значенням пікселя близьким до нуля. Вертикальним перепадам (межам) початкового зображення відповідають пікселі з великими по модулю значеннями на результуючому зображенні. Тому різницеві фільтри називають також *фільтрами виділення границь* об'єктів.

Аналогічно наведеним вище фільтрам, по методу скінченних різниць можна скласти фільтри для інших диференціальних операторів. Зокрема, важливий для багатьох додатків диференціальний оператор Лапласа (лапласіан) $\Delta = \partial^2 / \partial x^2 + \partial^2 / \partial y^2$ можна наблизити для дискретних зображень фільтром з матрицею (один з варіантів):

$$M_{\Delta} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} .$$

В результаті застосування дискретного лапласіану великі по модулю значення відповідають як вертикальним, так і горизонтальним перепадам яскравості. Фільтр являється, таким чином, фільтром, що знаходить межі будь-якої орієнтації. Знаходження меж на зображенні може робитися шляхом застосування цього фільтру і узяття усіх пікселів, модуль значення яких перевершує деякий поріг.

Проте такий алгоритм має істотні недоліки. Головний з них – не-

визначеність у виборі величини порогу. Для різних частин зображення прийнятний результат зазвичай виходить при істотно різних порогових значеннях. Крім того, різницеві фільтри дуже чутливі до шумів зображення.

13.3. Лінійна фільтрація в MATLAB

Функція **h=fspecial(type)** створює маску двовірного фільтру заданого типу **type** у вигляді матриці **h**, що зберігає коефіцієнти фільтру, які задають його передатну характеристику і властивості. Можливі наступні типи фільтрів (масок): **'average'**, **'gaussian'**, **'laplacian'**, **'log'**, **'sobel'**, **'prewitt'** і **'unsharp'**.

Функція **h=fspecial(type,parameters)** дозволяє вказати додаткові параметри. Наприклад, **h=fspecial('gaussian',n,sigma)** створює симетричний ФНЧ обертання гаусіану із стандартним відхиленням в пікселях.

Для фільтрації зображення з заданої користувачем апертурою фільтру використовується функція **Y=filter2(h,X)** – функція двовірної фільтрації (на основі КІХ-фільтру), де **h** – маска фільтру; **X**, **Y** – вхідне і вихідне зображення відповідно.

13.3.1. Усереднюючий фільтр

Усереднюючий фільтр відноситься до класу фільтрів низьких частот. Він може використовуватися для зменшення впливу шуму на зображення, проте, його застосування, призводить до розмиття зображення і погіршення його чіткості. Кожен елемент маски такого фільтру дорівнює $1/(n \times m)$, де n , m – розміри маски (число рядків и стовбців).

У наведеному нижче прикладі демонструється робота фільтру типу **'average'** з двома розмірами масок, що виконує лінійне згладжування зашумованого зображення рису з файлу `rice.png` (рис. 13.1):


```

%Програма №13.1
I=imread('rice.png'); %вхідне зображення
subplot(2,2,1);imshow(I);
%накладання шуму
J_g = imnoise(I,'gaussian',0,0.005);
subplot(2,2,2);imshow(J_g);
%фільтр 1
h = fspecial('average'); %маска фільтру 3*3
I1=filter2(h,J_g); %фільтрація (фільтр 1)
%зображення після фільтрації (фільтр 1)
subplot(2,2,3); imshow(uint8(I1));
%фільтр 2
h = fspecial('average',5); %маска фільтру 5*5
I2=filter2(h,J_g); %фільтрація (фільтр 2)
%зображення після фільтрації (фільтр 2)
subplot(2,2,4); imshow(uint8(I2));

```

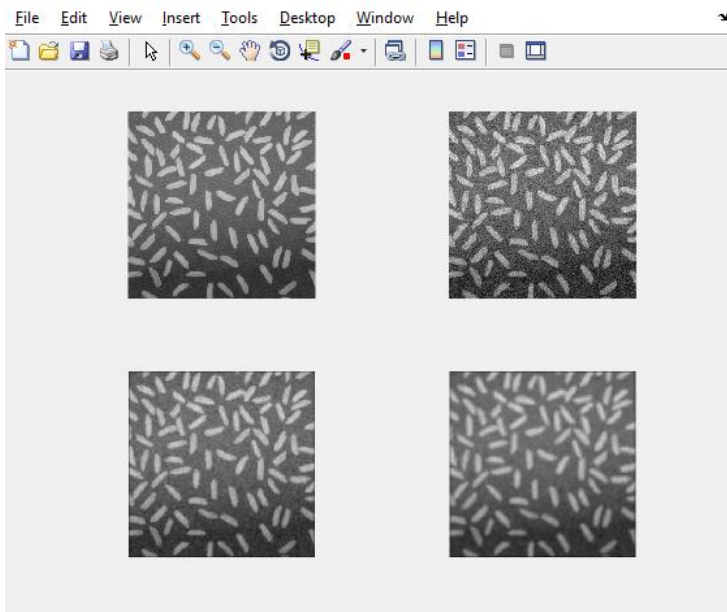


Рисунок 13.1 – Результати лінійної фільтрації

На рис. 13.1 в верхньому рядку розташовані вхідне (ліворуч) та зашумоване (праворуч) зображення, а в нижньому – відфільтроване зображення за допомогою усереднюючого фільтра з лінійними розмірами апертури 3×3 (ліворуч) та 5×5 (праворуч). З наведеного прикладу можна побачити, що зі збільшенням лінійних розмірів апертури шум видаляється краще, але корисний сигнал розмивається.

13.3.2. Фільтр Гауса

Це також фільтр нижніх частот, але, в порівнянні з усереднюючим фільтром, він менше розмиває зображення. Центральний елемент маски такого фільтра має максимальне значення, що відповідає піку розподілу Гауса. Ядро фільтра Гауса описується виразом:

$$h(i, j) = \frac{h_g(i, j)}{\sum_{i=1}^n \sum_{j=1}^m h_g(i, j)},$$

де $h_g(i, j) = e^{-\frac{i^2 + j^2}{2\sigma^2}}$ – двомірний розподіл Гауса; n, m – розміри маски; σ – середньоквадратичне відхилення розподілу.

У наведеному нижче прикладі демонструється робота фільтра типу **'gaussian'** з лінійними розмірами апертури 3×3 та із різними значеннями стандартного відхилення, що виконує лінійне згладжування зашумованого зображення рису з файлу rice.png (рис. 13.2):

```
%Програма №13.2
I=imread('rice.png'); %вхідне зображення
subplot(2,2,1); imshow(I);
%накладання шуму
J_g = imnoise(I,'gaussian',0,0.005);
subplot(2,2,2); imshow(J_g);
%фільтр 1 (сигма 0.5)
```

```

h = fspecial('gaussian'); %маска фільтру 1
I1=filter2(h,J_g); %фільтрація (фільтр 1)
%зображення після фільтрації (фільтр 1)
subplot(2,2,3); imshow(uint8(I1));
%фільтр 2 (сигма 1)
h = fspecial('gaussian',3,1); %маска фільтру 2
I2=filter2(h,J_g); %фільтрація (фільтр 2)
%зображення після фільтрації (фільтр 2)
subplot(2,2,4); imshow(uint8(I2));

```

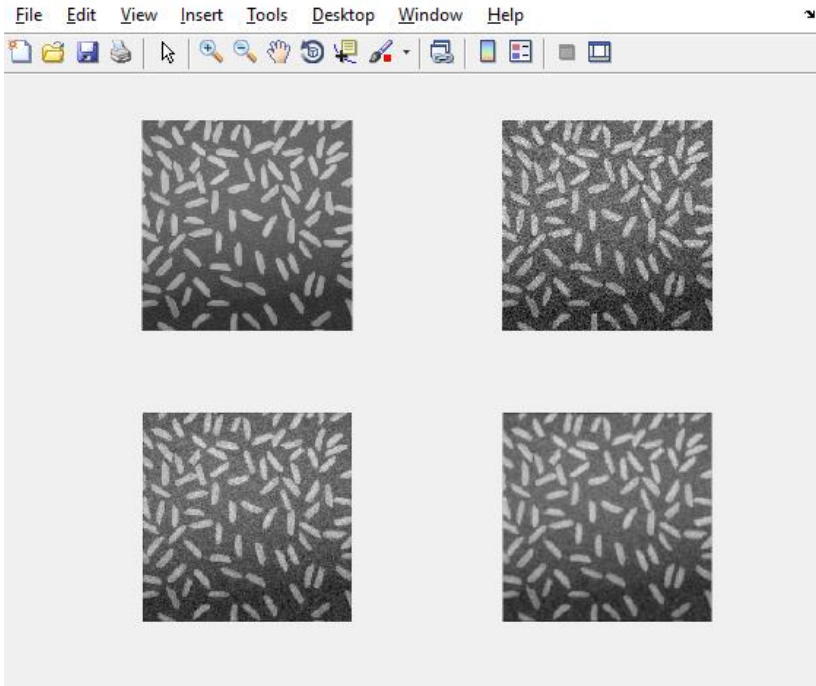


Рисунок 13.2 – Результати фільтрації фільтром Гауса

На рис. 13.2 (як і у попередньому прикладі) в верхньому рядку розташовані вхідне (ліворуч) та зашумоване (праворуч) зображення, а в нижньому – відфільтроване зображення за допомогою фільтру Гауса

з лінійними розмірами апертури 3×3 та $\sigma = 0.5$ (ліворуч) і $\sigma = 1$ (праворуч). З наведеного прикладу можна побачити, що збільшення значення стандартного відхилення дозволило отримати кращий результат без необхідності збільшення лінійних розмірів апертури, при цьому корисний сигнал практично не розмивається.

13.3.3. Фільтр підвищення різкості

Цей фільтр, що підвищує різкість зображення, має маску, яка визначається наступним виразом:

$$h = \frac{1}{a+1} \begin{bmatrix} -a & a-1 & -a \\ a-1 & a+5 & a-1 \\ -a & a-1 & -a \end{bmatrix},$$

де a – параметр, що обирається в межах від 0 до 1.

У наведеному нижче прикладі демонструється робота фільтру типу **'unsharp'** з лінійними розмірами апертури 3×3 та із різними значеннями параметру a , а також робота фільтрів з масками, наведеними у 13.2.2, які виконують підвищення різкості фотографії хлопчика з файлу `pout.tif` (рис. 13.3):

```
%Програма №13.3
I=imread('pout.tif'); %вхідне зображення
subplot(2,3,1);imshow(I);
h=fspecial('unsharp'); %маска фільтра 1: a=0.2
I1=filter2(h,I); %фільтрація (фільтр 1)
%зображення після фільтрації (фільтр 1)
subplot(2,3,2); imshow(uint8(I1));
h=fspecial('unsharp',1); %маска фільтра 2: a=1
I2=filter2(h,I); %фільтрація (фільтр 2)
%зображення після фільтрації (фільтр 2)
subplot(2,3,3); imshow(uint8(I2));
h=[1 -2 1; -2 5 -2; 1 -2 1]; %маска фільтра 3
```

```

I2=filter2(h,I); %фільтрація (фільтр 3)
%зображення після фільтрації (фільтр 3)
subplot(2,3,4); imshow(uint8(I2))
h=[-1 -1 -1; -1 9 -1; -1 -1 -1]; %маска фільтра 4
I2=filter2(h,I); %фільтрація (фільтр 4)
%зображення після фільтрації (фільтр 4)
subplot(2,3,5); imshow(uint8(I2));
h=[0 -1 0; -1 5 -1; 0 -1 0]; %маска фільтра 5
I2=filter2(h,I); %фільтрація (фільтр 5)
%зображення після фільтрації (фільтр 5)
subplot(2,3,6); imshow(uint8(I2));

```

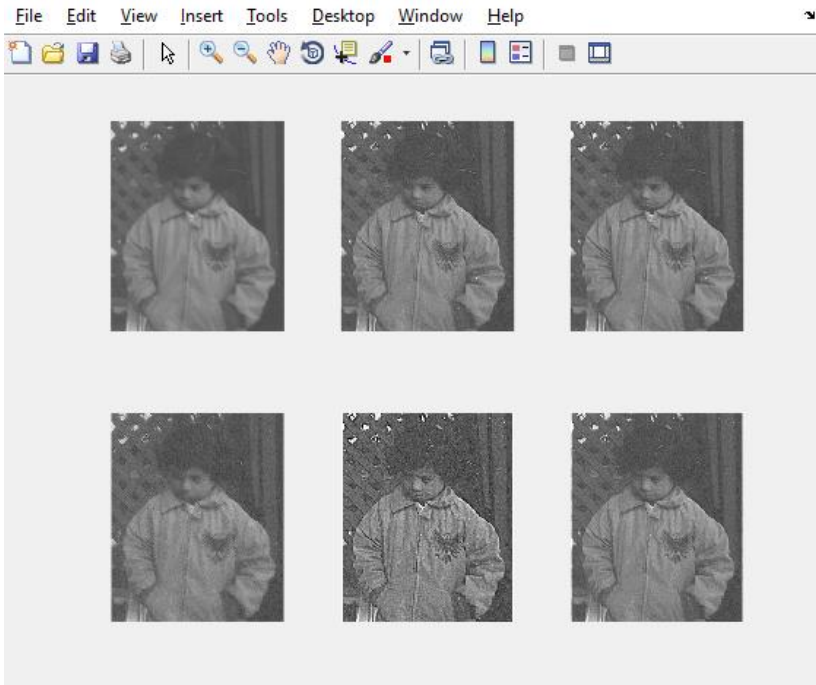


Рисунок 13.3 – Результати роботи фільтрів підвищення контрасту

13.3.4. Фільтр Собеля

Фільтр Собеля служить для виділення горизонтальних меж об'єктів зображення за допомогою маски, що задається виразом:

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Для виділення вертикальних меж цим фільтром досить транспонувати маску h .

У наведеному нижче прикладі демонструється робота фільтру типу **'sobel'**, що виконує виділення горизонтальних та вертикальних меж зображення рису з файлу `rice.png` (рис. 13.4):

```
%Програма №13.4
I=imread('rice.png'); %вхідне зображення
subplot(2,2,1);imshow(I);
h = fspecial('sobel'); %маска Фільтра Собеля
J1=filter2(h,I); %виділення горизонтальних меж
subplot(2,2,3); imshow(uint8(J1));
h=h'; %транспонування маски
J2=filter2(h,I); %виділення вертикальних меж
subplot(2,2,4); imshow(uint8(J2));
J3=(J1.^2+J2.^2).^0.5; %норми градієнтів
%виділення контурів
subplot(2,2,2); imshow(uint8(J3));
```

На рис. 13.4 в верхньому рядку розташовані вхідне зображення (ліворуч) та зображення з виділенням контурів об'єктів (праворуч), а в нижньому – зображення з виділеними горизонтальними (ліворуч) і вертикальними (праворуч) межами за допомогою фільтру Собеля.

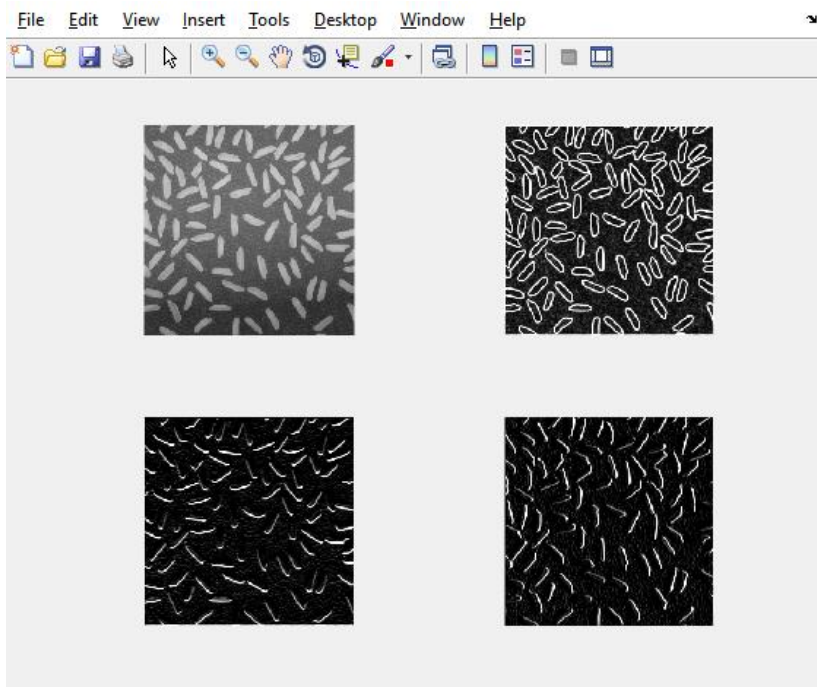


Рисунок 13.4 – Результати роботи фільтру Собеля

13.3.5. Фільтр Прюїтта

Цей фільтр також використовується для виділення горизонтальних меж зображень за допомогою маски (для виділення вертикальних меж зображення також можна транспонувати маску h цього фільтру):

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}.$$

У наведеному нижче прикладі демонструється робота фільтру типу '**prewitt**', що виконує виділення горизонтальних та вертикальних меж зображення рису з файлу `rice.png` (рис. 13.5):

```

%Програма №13.5
I=imread('rice.png'); %вхідне зображення
subplot(2,2,1);imshow(I);
h = fspecial('prewitt'); %маска Фільтра Прюїтта
J1=filter2(h,I); %виділення горизонтальних меж
subplot(2,2,3); imshow(uint8(J1));
h=h'; %транспонування маски
J2=filter2(h,I); %виділення вертикальних меж
subplot(2,2,4); imshow(uint8(J2));
J3=(J1.^2+J2.^2).^0.5; %норми градієнтів
%виділення контурів
subplot(2,2,2); imshow(uint8(J3));

```

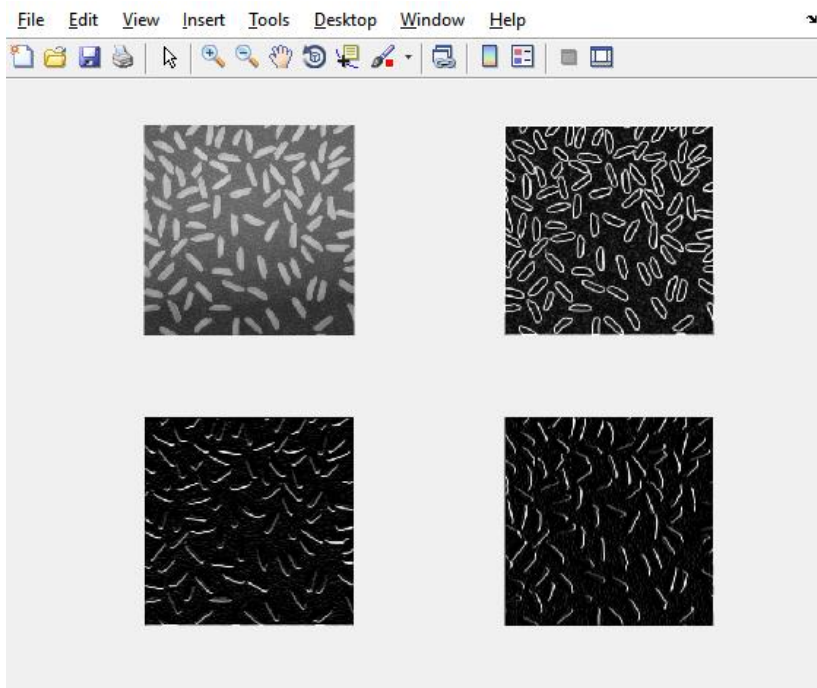


Рисунок 13.5 – Результати роботи фільтра Прюїтта

На рис. 13.5 в верхньому рядку розташовані вхідне зображення (ліворуч) та зображення з виділенням контурів об'єктів (праворуч), а в нижньому – зображення з виділеними горизонтальними (ліворуч) і вертикальними (праворуч) межами за допомогою фільтра Прюїтта.

13.3.6. Фільтр Лапласа

Фільтр Лапласа – це фільтр для виділення меж (перепадів) зображень по усіх напрямках. Маска фільтру задається у виді:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = h = \frac{4}{a+1} \begin{bmatrix} \frac{a}{4} & \frac{1-a}{4} & \frac{a}{4} \\ \frac{1-a}{4} & -1 & \frac{1-a}{4} \\ \frac{a}{4} & \frac{1-a}{4} & \frac{a}{4} \end{bmatrix},$$

де a – параметр, що обирається в межах від 0 до 1.

У наведеному нижче прикладі демонструється робота фільтру типу **'laplacian'** з лінійними розмірами апертури 3×3 та із різними значеннями параметру a , що виконує виділення контурів зображення рису з файлу `rice.png` (рис. 13.6):

```
%Програма №13.6
I=imread('rice.png'); %вхідне зображення
subplot(2,2,1);
imshow(I);
%фільтр 1
h = fspecial('laplacian'); %маска фільтра 1: a=0.2
J1=filter2(h,I); %зображення після фільтрації
subplot(2,2,2);
imshow(uint8(J1));
%фільтр 2
h = fspecial('laplacian',0); %маска фільтра 2: a=0
J1=filter2(h,I); %зображення після фільтрації
```

```

subplot(2,2,3);
imshow(uint8(J1));
%фільтр 3
h = fspecial('laplacian',1); %маска фільтра 3: a=1
J1=filter2(h,I); %зображення після фільтрації
subplot(2,2,4);
imshow(uint8(J1));

```

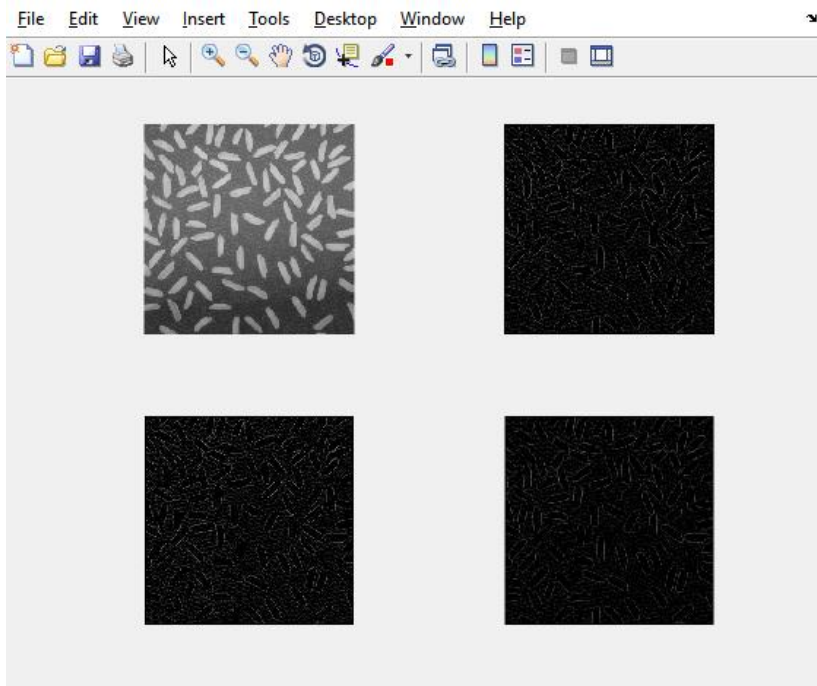


Рисунок 13.6 – Результати роботи фільтру Лапласа

13.3.7. Фільтр Лапласа-Гауса

Цей комбінований фільтр високих частот (послідовно включені фільтри Лапласа і Гауса) здатний виділяти різкіші переходи зображення, ніж фільтр Лапласа. Маска фільтра визначається виразом:

$$h(i, j) = \frac{(i^2 + j^2 - 2\sigma^2)h_g(i, j)}{2\pi\sigma^6 \sum_{i=1}^n \sum_{j=1}^m h_g(i, j)},$$

де n, m – розміри маски; σ – стандартне відхилення розподілу.

У наведеному нижче прикладі демонструється робота фільтру типу '**log**' із різними значеннями лінійних розмірів апертури та стандартного відхилення, що виконує виділення контурів зображення рису з файлу `rice.png` (рис. 13.7):

```
%Програма №13.7
I=imread('rice.png'); %вхідне зображення
subplot(2,2,1);
imshow(I);

%Фільтр Лапласа-Гауса 1: 5x5, sigma=0.5
h=fspecial('log'); %маска фільтра
J=filter2(h,I); %зображення після фільтрації
subplot(2,2,2);
imshow(uint8(J));

%Фільтр Лапласа-Гауса 2: 3x3, sigma=0.5
h=fspecial('log',3,0.5); %маска фільтра
J=filter2(h,I); %зображення після фільтрації
subplot(2,2,3);
imshow(uint8(J));

%Фільтр Лапласа-Гауса 2: 5x5, sigma=0.6
h=fspecial('log',5,0.6); %маска фільтра
J=filter2(h,I); %зображення після фільтрації
subplot(2,2,4);
imshow(uint8(J));
```

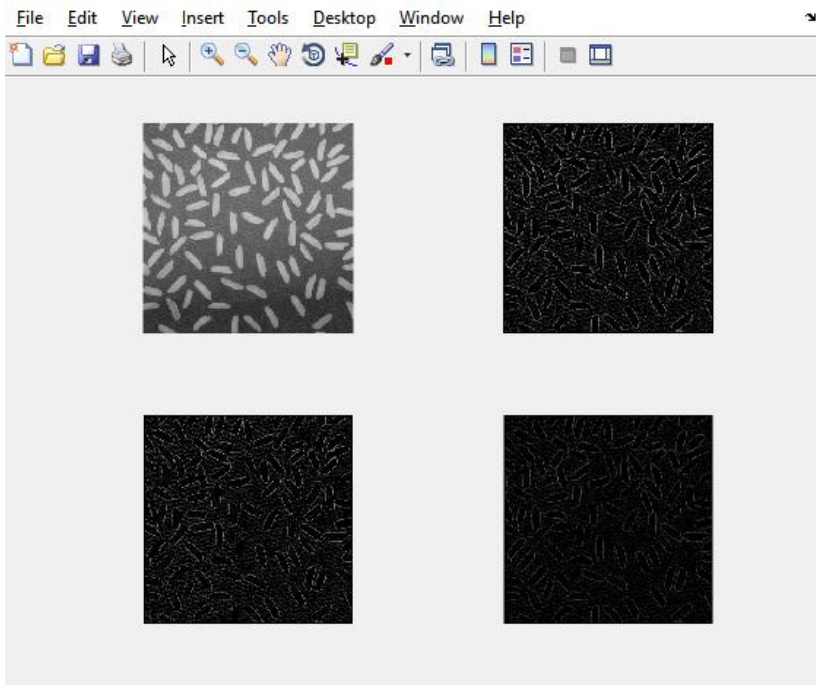


Рисунок 13.7 – Результати роботи фільтру Лапласа-Гауса

13.4. Приклади до розділу 13

Приклад №13.1. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати лінійне згладжування з наступними масками:

$$\text{а) } h = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix};$$

$$\text{б) } h = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Розв'язок.

Введемо позначення:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix}, Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \end{bmatrix},$$

де X – початкове зображення, Y – відфільтроване зображення.

Перед початком розрахунку необхідно вибрати метод обробки крайових точок. Нехай неіснуючі пікселі за межами зображення дорівнюють нулю.

а) Розрахунок перших двох рядків зображення з урахуванням за-

даної маски фільтру $h = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ виконується таким чином:

$$y_{11} = \frac{x_{11} + x_{12} + x_{21} + x_{22}}{9} = (34 + 9 + 17 + 21) / 9 = 9;$$

$$y_{12} = \frac{x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23}}{9} = (34 + 9 + 40 + 17 + 21 + 23) / 9 = 16;$$

$$y_{13} = \frac{x_{12} + x_{13} + x_{14} + x_{22} + x_{23} + x_{24}}{9} = (9 + 40 + 20 + 21 + 23 + 45) / 9 = 17.56;$$

$$y_{14} = \frac{x_{13} + x_{14} + x_{23} + x_{24}}{9} = (40 + 20 + 23 + 45)/9 = 14.22;$$

$$y_{21} = \frac{x_{11} + x_{12} + x_{21} + x_{22} + x_{31} + x_{32}}{9} = (34 + 9 + 17 + 21 + 8 + 42)/9 = 14.56;$$

$$y_{22} = \frac{x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33}}{9} =$$

$$= (34 + 9 + 40 + 17 + 21 + 23 + 8 + 42 + 22)/9 = 24;$$

$$y_{23} = \frac{x_{12} + x_{13} + x_{14} + x_{22} + x_{23} + x_{24} + x_{32} + x_{33} + x_{34}}{9} =$$

$$= (9 + 40 + 20 + 21 + 23 + 45 + 42 + 22 + 0)/9 = 24.67;$$

$$y_{24} = \frac{x_{13} + x_{14} + x_{23} + x_{24} + x_{33} + x_{34}}{9} = (40 + 20 + 23 + 45 + 22 + 0)/9 = 16.67.$$

Аналогічно виконується розрахунок для 2-х рядків, що залишилися. В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 9 & 16 & 17.56 & 14.22 \\ 14.56 & 24 & 24.67 & 16.67 \\ 13.22 & 20.67 & 23.67 & 14 \\ 9 & 13.89 & 13.78 & 6.44 \end{bmatrix}.$$

б) Розрахунок перших двох рядків зображення, з урахуванням за-

даної маски фільтру $h = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ виконується таким чином:

$$y_{11} = (2x_{11} + x_{12} + x_{21} + x_{22})/10 = (2 \cdot 34 + 9 + 17 + 21)/10 = 11.5;$$

$$y_{12} = (x_{11} + 2x_{12} + x_{13} + x_{21} + x_{22} + x_{23})/10 =$$

$$= (34 + 2 \cdot 9 + 40 + 17 + 21 + 23)/10 = 15.3;$$

$$y_{13} = (x_{12} + 2x_{13} + x_{14} + x_{22} + x_{23} + x_{24})/10 = \\ = (9 + 2 \cdot 40 + 20 + 21 + 23 + 45)/10 = 19.8;$$

$$y_{14} = (x_{13} + 2x_{14} + x_{23} + x_{24})/10 = (40 + 2 \cdot 20 + 23 + 45)/10 = 14.8;$$

$$y_{21} = (x_{11} + x_{12} + 2x_{21} + x_{22} + x_{31} + x_{32})/10 = \\ = (34 + 9 + 2 \cdot 17 + 21 + 8 + 42)/10 = 14.8;$$

$$y_{22} = (x_{11} + x_{12} + x_{13} + x_{21} + 2x_{22} + x_{23} + x_{31} + x_{32} + x_{33})/10 = \\ = (34 + 9 + 40 + 17 + 2 \cdot 21 + 23 + 8 + 42 + 22)/10 = 23.7;$$

$$y_{23} = (x_{12} + x_{13} + x_{14} + x_{22} + 2x_{23} + x_{24} + x_{32} + x_{33} + x_{34})/10 = \\ = (9 + 40 + 20 + 21 + 2 \cdot 23 + 45 + 42 + 22 + 0)/10 = 24.5;$$

$$y_{24} = (x_{13} + x_{14} + x_{23} + 2x_{24} + x_{33} + x_{34})/10 = \\ = (40 + 20 + 2 \cdot 23 + 45 + 22 + 0)/10 = 19.5.$$

Аналогічно виконується розрахунок для 2-х останніх рядків. В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 11.5 & 15.3 & 19.8 & 14.8 \\ 14.8 & 23.7 & 24.5 & 19.5 \\ 12.7 & 22.8 & 23.5 & 12.6 \\ 8.8 & 14.9 & 14.6 & 7.2 \end{bmatrix}.$$

Приклад №13.2. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати лінійне згладжування фільтром Гауса із ядром з $\sigma = 0.5$.

Розв'язок.

Розрахуємо маску фільтра Гауса:

$$\text{Для } i=-1 \text{ та } j=-1 \text{ маємо } h_g(-1,-1) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{(-1)^2 + (-1)^2}{2 \cdot 0.5^2}} = 0.0117.$$

$$\text{Для } i=-1 \text{ та } j=0 \text{ маємо } h_g(-1,0) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{(-1)^2}{2 \cdot 0.5^2}} = 0.0862.$$

$$\text{Для } i=-1 \text{ та } j=1 \text{ маємо } h_g(-1,1) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{(-1)^2 + 1^2}{2 \cdot 0.5^2}} = 0.0117.$$

$$\text{Для } i=0 \text{ та } j=-1 \text{ маємо } h_g(0,-1) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{(-1)^2}{2 \cdot 0.5^2}} = 0.0862.$$

$$\text{Для } i=0 \text{ та } j=0 \text{ маємо } h_g(0,0) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{0}{2 \cdot 0.5^2}} = 0.6366.$$

$$\text{Для } i=0 \text{ та } j=1 \text{ маємо } h_g(0,1) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{1^2}{2 \cdot 0.5^2}} = 0.0862.$$

$$\text{Для } i=1 \text{ та } j=-1 \text{ маємо } h_g(1,-1) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{1^2 + (-1)^2}{2 \cdot 0.5^2}} = 0.0117.$$

$$\text{Для } i=1 \text{ та } j=0 \text{ маємо } h_g(1,0) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{1^2}{2 \cdot 0.5^2}} = 0.0862.$$

$$\text{Для } i=1 \text{ та } j=1 \text{ маємо } h_g(1,1) = \frac{1}{2\pi \cdot 0.5^2} e^{-\frac{1^2 + 1^2}{2 \cdot 0.5^2}} = 0.0117.$$

В результаті отримуємо наступну матрицю:

$$h_g = \begin{bmatrix} 0.0117 & 0.0862 & 0.0117 \\ 0.0862 & 0.6366 & 0.0862 \\ 0.0117 & 0.0862 & 0.0117 \end{bmatrix}.$$

Для усіх фільтрів згладжування повинна виконуватися умова, що сума усіх елементів маски повинна дорівнювати 1. Сума елементів розрахованої матриці h_g дорівнює 1.0279, що більше 1, тому необхідно нормувати елементи матриці h_g .

$$h = \begin{bmatrix} \frac{0.0117}{1.0279} & \frac{0.0862}{1.0279} & \frac{0.0117}{1.0279} \\ \frac{0.0862}{1.0279} & \frac{0.6366}{1.0279} & \frac{0.0862}{1.0279} \\ \frac{0.0117}{1.0279} & \frac{0.0862}{1.0279} & \frac{0.0117}{1.0279} \end{bmatrix} = \begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}.$$

Тоді розрахунок перших двох рядків зображення з урахуванням заданої маски фільтра h виконується наступним чином:

$$\begin{aligned} y_{11} &= 0.6193x_{11} + 0.0838x_{12} + 0.0838x_{21} + 0.0113x_{22} = \\ &= 0.6193 \cdot 34 + 0.0838 \cdot 9 + 0.0838 \cdot 17 + 0.0113 \cdot 21 = 23.48; \\ y_{12} &= 0.0838x_{11} + 0.6193x_{12} + 0.0838x_{13} + 0.0113x_{21} + 0.0838x_{22} + \\ &+ 0.0113x_{23} = 0.0838 \cdot 34 + 0.6193 \cdot 9 + 0.0838 \cdot 40 + 0.0113 \cdot 17 + \\ &+ 0.0838 \cdot 21 + 0.0113 \cdot 23 = 13.99; \\ y_{13} &= 0.0838x_{12} + 0.6193x_{13} + 0.0838x_{14} + 0.0113x_{22} + 0.0838x_{23} + \\ &+ 0.0113x_{24} = 0.0838 \cdot 9 + 0.6193 \cdot 40 + 0.0838 \cdot 20 + 0.0113 \cdot 21 + \\ &+ 0.0838 \cdot 23 + 0.0113 \cdot 45 = 29.88; \\ y_{14} &= 0.0838x_{13} + 0.6193x_{14} + 0.0113x_{23} + 0.0838x_{24} = \\ &= 0.0838 \cdot 40 + 0.6193 \cdot 20 + 0.0113 \cdot 23 + 0.0838 \cdot 45 = 19.77; \\ y_{21} &= 0.0838x_{11} + 0.0113x_{12} + 0.6193x_{21} + 0.0838x_{22} + 0.0838x_{31} + \\ &+ 0.0113x_{32} = 0.0838 \cdot 34 + 0.0113 \cdot 9 + 0.6193 \cdot 17 + 0.0838 \cdot 21 + \\ &+ 0.0838 \cdot 8 + 0.0113 \cdot 42 = 16.39; \end{aligned}$$

$$\begin{aligned}
y_{22} &= 0.0113x_{11} + 0.0838x_{12} + 0.0113x_{13} + 0.0838x_{21} + 0.6193x_{22} + \\
&+ 0.0838x_{23} + 0.0113x_{31} + 0.0838x_{32} + 0.0113x_{33} = 0.0113 \cdot 34 + \\
&+ 0.0838 \cdot 9 + 0.0113 \cdot 40 + 0.0838 \cdot 17 + 0.6193 \cdot 21 + 0.0838 \cdot 23 + \\
&+ 0.0113 \cdot 8 + 0.0838 \cdot 42 + 0.0113 \cdot 22 = 21.81; \\
y_{23} &= 0.0113x_{12} + 0.0838x_{13} + 0.0113x_{14} + 0.0838x_{22} + 0.6193x_{23} + \\
&+ 0.0838x_{24} + 0.0113x_{32} + 0.0838x_{33} + 0.0113x_{34} = \\
&= 0.0113 \cdot 9 + 0.0838 \cdot 40 + 0.0113 \cdot 20 + 0.0838 \cdot 21 + 0.6193 \cdot 23 + \\
&+ 0.0838 \cdot 45 + 0.0113 \cdot 42 + 0.0838 \cdot 22 + 0 = 25.78; \\
y_{24} &= 0.0113x_{13} + 0.0838x_{14} + 0.0838x_{23} + 0.6193x_{24} + 0.0113x_{33} + \\
&+ 0.0838x_{34} = 0.0113 \cdot 40 + 0.0838 \cdot 20 + 0.0838 \cdot 23 + 0.6193 \cdot 45 + \\
&+ 0.0113 \cdot 22 + 0 = 32.18.
\end{aligned}$$

Аналогічно виконується розрахунок для 2-х рядків, що залишилися. В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 24.48 & 13.99 & 29.88 & 19.77 \\ 16.39 & 21.81 & 25.78 & 32.18 \\ 11 & 33.08 & 22.1 & 7.3 \\ 7.49 & 21.16 & 19.13 & 10.75 \end{bmatrix}.$$

Приклад №13.3. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати високочастотну фільтрацію з маскою фільтру

$$h = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}.$$

Розв'язок.

Введемо позначення:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix}, Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \end{bmatrix},$$

де X – початкове зображення, Y – відфільтроване зображення.

Перед початком розрахунку необхідно вибрати метод обробки крайових точок. Нехай неіснуючі пікселі за межами зображення дорівнюють нулю.

Розрахунок перших двох рядків зображення, з урахуванням заданої маски фільтра, виконується таким чином:

$$y_{11} = 5x_{11} - x_{12} - x_{21} = 5 \cdot 34 - 9 - 17 = 144;$$

$$y_{12} = -x_{11} + 5x_{12} - x_{13} - x_{22} = -34 + 5 \cdot 9 - 40 - 21 = -50;$$

$$y_{13} = -x_{12} + 5x_{13} - x_{14} - x_{23} = -9 + 5 \cdot 40 - 20 - 23 = 148;$$

$$y_{14} = -x_{13} + 5x_{14} - x_{24} = -40 + 5 \cdot 20 - 45 = 15;$$

$$y_{21} = -x_{11} + 5x_{21} - x_{22} - x_{31} = -34 + 5 \cdot 17 - 21 - 8 = 22;$$

$$y_{22} = -x_{12} - x_{21} + 5x_{22} - x_{23} - x_{32} = -9 - 17 + 5 \cdot 21 - 23 - 42 = 14;$$

$$y_{23} = -x_{13} - x_{22} + 5x_{23} - x_{24} - x_{33} = -40 - 21 + 5 \cdot 23 - 45 - 22 = -13;$$

$$y_{24} = -x_{14} - x_{23} + 5x_{24} - x_{34} = -20 - 23 + 5 \cdot 45 + 0 = 182.$$

Аналогічно виконується розрахунок для 2-х рядків, що залишили-

ся. В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 144 & -50 & 148 & 15 \\ 22 & 14 & -13 & 182 \\ -26 & 135 & 23 & -81 \\ 3 & 49 & 50 & 48 \end{bmatrix}.$$

Примітка: оскільки значення рівня яскравості (при зберіганні значення в 1 байті) не можуть виходити за діапазон $[0, 255]$, то негативні значення обнуляються, а значення, що перевищують 255, задаються рівними 255.

Приклад №13.4. Дано зображення:

$$X = \begin{bmatrix} 34 & 9 & 40 & 20 \\ 17 & 21 & 23 & 45 \\ 8 & 42 & 22 & 0 \\ 7 & 24 & 22 & 14 \end{bmatrix}.$$

Необхідно виконати підкреслення вертикальних і горизонтальних меж з використанням оператора Собеля.

Розв'язок.

Введемо позначення:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix}, Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \end{bmatrix},$$

де X – початкове зображення, Y – відфільтроване зображення.

Перед початком розрахунку необхідно вибрати метод обробки

крайових точок. Нехай неіснуючі пікселі за межами зображення дорівнюють нулю.

Виконаємо підкреслення вертикальних меж. Розрахунок перших двох рядків зображення з урахуванням маски фільтру

$$h_v^{sobel} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \text{ виконується таким чином:}$$

$$y_{v11} = -2x_{12} - x_{22} = -2 \cdot 9 - 21 = -39;$$

$$y_{v12} = 2x_{11} - 2x_{13} + x_{21} - x_{23} = 2 \cdot 34 - 2 \cdot 40 + 17 - 23 = -18;$$

$$y_{v13} = 2x_{12} - 2x_{14} + x_{22} - x_{24} = 2 \cdot 9 + -2 \cdot 20 + 21 - 45 = -46;$$

$$y_{v14} = 2x_{13} + x_{23} = 2 \cdot 40 + 23 = 103;$$

$$y_{v21} = -x_{12} - 2x_{22} - x_{32} = -9 - 2 \cdot 21 - 42 = -93;$$

$$\begin{aligned} y_{v22} &= x_{11} - x_{13} + 2x_{21} - 2x_{23} + x_{31} - x_{33} = \\ &= 34 - 40 + 2 \cdot 17 - 2 \cdot 23 + 8 - 22 = -32; \end{aligned}$$

$$\begin{aligned} y_{v23} &= x_{12} - x_{14} + 2x_{22} - 2x_{24} + x_{32} - x_{34} = \\ &= 9 - 20 + 2 \cdot 21 - 2 \cdot 45 + 42 - 0 = -17; \end{aligned}$$

$$y_{v24} = -x_{14} - 2x_{24} - x_{34} = -20 - 2 \cdot 45 - 0 = 108.$$

Аналогічно виконується розрахунок для 2-х рядків, що залишилися. В результаті отримаємо наступне відфільтроване зображення:

$$Y_v = \begin{bmatrix} -39 & -18 & -46 & 103 \\ -93 & -32 & -17 & 108 \\ -129 & -49 & 70 & 89 \\ -90 & -44 & 62 & 66 \end{bmatrix}.$$

Аналогічно виконується розрахунок для підкреслення горизон-

тальних меж з використанням маски $h_h^{sobel} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$. В резуль-

таті отримаємо наступне відфільтроване зображення:

$$Y_h = \begin{bmatrix} -55 & -82 & -112 & -113 \\ 19 & -22 & 23 & 58 \\ 17 & 5 & 30 & 63 \\ 58 & 114 & 86 & 22 \end{bmatrix}.$$

Матриці Y_v і Y_h містять складові градієнтів зміни яскравості пікселів по відповідних просторових координатах.

Обчислимо норми градієнтів. Для пікселя з координатами (1,1) норма градієнта розраховується таким чином:

$$y_{11} = \sqrt{y_{v11}^2 + y_{h11}^2} = \sqrt{(-39)^2 + (-55)^2} = 67.42.$$

В результаті розрахунку норми градієнтів для усіх пікселів:

$$Y = \begin{bmatrix} 67.42 & 83.95 & 121.08 & 152.9 \\ 94.92 & 38.83 & 28.6 & 122.59 \\ 130.12 & 49.25 & 76.16 & 109.04 \\ 107.07 & 122.2 & 106.02 & 69.57 \end{bmatrix}.$$

13.5. Завдання до розділу 13 для самостійної роботи

Виконати фільтрацію двовірного сигналу. Маска фільтру

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \text{ Вказати спосіб обробки крайових точок. Вхідні дані}$$

дивись в табл. 13.1 (N – номер студента за журналом).

Таблиця 13.1 – Варіанти двомірних сигналів

N	X	N	X	N	X
1	$\begin{bmatrix} 4 & 0 & 8 & 16 & 10 \\ 3 & 14 & 16 & 0 & 14 \\ 12 & 8 & 10 & 13 & 8 \\ 5 & 18 & 4 & 7 & 6 \\ 3 & 9 & 13 & 16 & 3 \end{bmatrix}$	6	$\begin{bmatrix} 9 & 7 & 12 & 1 & 1 \\ 0 & 13 & 0 & 7 & 9 \\ 0 & 1 & 0 & 12 & 8 \\ 6 & 0 & 3 & 14 & 7 \\ 0 & 12 & 11 & 13 & 3 \end{bmatrix}$	11	$\begin{bmatrix} 16 & 14 & 14 & 2 & 16 \\ 7 & 10 & 19 & 13 & 13 \\ 8 & 15 & 16 & 7 & 19 \\ 11 & 9 & 14 & 2 & 19 \\ 11 & 3 & 9 & 11 & 1 \end{bmatrix}$
2	$\begin{bmatrix} 3 & 13 & 9 & 13 & 14 \\ 13 & 7 & 17 & 6 & 6 \\ 6 & 17 & 16 & 5 & 16 \\ 10 & 17 & 12 & 6 & 11 \\ 3 & 11 & 16 & 10 & 7 \end{bmatrix}$	7	$\begin{bmatrix} 13 & 2 & 5 & 4 & 3 \\ 13 & 9 & 17 & 4 & 16 \\ 14 & 14 & 4 & 0 & 3 \\ 9 & 17 & 16 & 1 & 3 \\ 11 & 5 & 18 & 12 & 19 \end{bmatrix}$	12	$\begin{bmatrix} 7 & 11 & 8 & 18 & 14 \\ 10 & 14 & 12 & 12 & 8 \\ 5 & 19 & 16 & 5 & 19 \\ 11 & 15 & 1 & 17 & 1 \\ 0 & 14 & 18 & 10 & 11 \end{bmatrix}$
3	$\begin{bmatrix} 14 & 15 & 19 & 2 & 13 \\ 10 & 19 & 5 & 0 & 5 \\ 8 & 10 & 5 & 17 & 9 \\ 13 & 17 & 17 & 3 & 1 \\ 12 & 3 & 14 & 5 & 19 \end{bmatrix}$	8	$\begin{bmatrix} 8 & 11 & 18 & 12 & 2 \\ 6 & 2 & 5 & 19 & 16 \\ 6 & 0 & 3 & 13 & 8 \\ 7 & 9 & 17 & 17 & 17 \\ 7 & 17 & 4 & 0 & 14 \end{bmatrix}$	13	$\begin{bmatrix} 5 & 7 & 2 & 13 & 1 \\ 17 & 9 & 17 & 4 & 1 \\ 6 & 8 & 15 & 5 & 5 \\ 13 & 11 & 15 & 12 & 8 \\ 1 & 12 & 16 & 10 & 9 \end{bmatrix}$
4	$\begin{bmatrix} 11 & 4 & 4 & 11 & 8 \\ 8 & 11 & 7 & 15 & 6 \\ 10 & 15 & 15 & 1 & 17 \\ 6 & 10 & 13 & 12 & 0 \\ 8 & 12 & 9 & 1 & 15 \end{bmatrix}$	9	$\begin{bmatrix} 13 & 8 & 9 & 5 & 11 \\ 6 & 17 & 9 & 0 & 8 \\ 3 & 9 & 8 & 13 & 3 \\ 3 & 16 & 18 & 13 & 12 \\ 3 & 9 & 0 & 19 & 14 \end{bmatrix}$	14	$\begin{bmatrix} 18 & 3 & 16 & 7 & 19 \\ 11 & 16 & 12 & 3 & 2 \\ 6 & 19 & 14 & 16 & 17 \\ 9 & 11 & 1 & 16 & 15 \\ 11 & 0 & 8 & 9 & 8 \end{bmatrix}$
5	$\begin{bmatrix} 19 & 4 & 8 & 13 & 4 \\ 19 & 12 & 14 & 4 & 12 \\ 15 & 6 & 5 & 16 & 12 \\ 8 & 19 & 8 & 12 & 7 \\ 9 & 14 & 18 & 2 & 11 \end{bmatrix}$	10	$\begin{bmatrix} 7 & 18 & 3 & 10 & 12 \\ 0 & 16 & 18 & 18 & 13 \\ 8 & 7 & 11 & 6 & 7 \\ 15 & 12 & 12 & 13 & 8 \\ 15 & 14 & 4 & 7 & 13 \end{bmatrix}$	15	$\begin{bmatrix} 9 & 7 & 12 & 11 & 10 \\ 14 & 10 & 6 & 17 & 11 \\ 13 & 6 & 12 & 13 & 6 \\ 12 & 7 & 6 & 15 & 5 \\ 7 & 19 & 5 & 6 & 17 \end{bmatrix}$

Продовження табл. 13.1

N	X	N	X	N	X
16	$\begin{bmatrix} 12 & 19 & 6 & 19 & 8 \\ 9 & 13 & 12 & 9 & 6 \\ 16 & 19 & 10 & 8 & 11 \\ 13 & 18 & 8 & 11 & 12 \\ 7 & 9 & 14 & 15 & 14 \end{bmatrix}$	20	$\begin{bmatrix} 12 & 11 & 8 & 10 & 14 \\ 10 & 11 & 12 & 4 & 17 \\ 16 & 13 & 11 & 20 & 7 \\ 10 & 13 & 8 & 13 & 12 \\ 3 & 17 & 2 & 5 & 10 \end{bmatrix}$	24	$\begin{bmatrix} 5 & 7 & 17 & 10 & 9 \\ 10 & 12 & 11 & 3 & 6 \\ 10 & 8 & 18 & 6 & 15 \\ 5 & 14 & 10 & 12 & 6 \\ 3 & 16 & 3 & 5 & 9 \end{bmatrix}$
17	$\begin{bmatrix} 9 & 9 & 9 & 13 & 9 \\ 11 & 5 & 15 & 20 & 7 \\ 10 & 14 & 15 & 17 & 6 \\ 12 & 7 & 11 & 4 & 13 \\ 20 & 12 & 2 & 12 & 10 \end{bmatrix}$	21	$\begin{bmatrix} 5 & 1 & 6 & 6 & 5 \\ 15 & 16 & 9 & 5 & 10 \\ 11 & 6 & 8 & 20 & 10 \\ 7 & 8 & 19 & 2 & 13 \\ 11 & 18 & 11 & 19 & 8 \end{bmatrix}$	25	$\begin{bmatrix} 17 & 11 & 13 & 17 & 14 \\ 18 & 15 & 11 & 8 & 5 \\ 12 & 12 & 8 & 12 & 14 \\ 18 & 1 & 19 & 18 & 2 \\ 19 & 9 & 17 & 19 & 9 \end{bmatrix}$
18	$\begin{bmatrix} 7 & 10 & 17 & 9 & 12 \\ 12 & 18 & 11 & 8 & 18 \\ 2 & 9 & 14 & 7 & 9 \\ 10 & 0 & 17 & 11 & 12 \\ 14 & 6 & 1 & 20 & 8 \end{bmatrix}$	22	$\begin{bmatrix} 13 & 9 & 7 & 13 & 9 \\ 11 & 4 & 5 & 12 & 17 \\ 10 & 14 & 15 & 17 & 6 \\ 12 & 7 & 20 & 4 & 13 \\ 20 & 16 & 2 & 12 & 11 \end{bmatrix}$	26	$\begin{bmatrix} 14 & 9 & 10 & 12 & 5 \\ 19 & 20 & 5 & 12 & 18 \\ 17 & 20 & 16 & 3 & 16 \\ 10 & 18 & 18 & 18 & 18 \\ 16 & 8 & 19 & 10 & 6 \end{bmatrix}$
19	$\begin{bmatrix} 17 & 7 & 7 & 10 & 9 \\ 10 & 12 & 11 & 3 & 6 \\ 10 & 8 & 11 & 6 & 5 \\ 5 & 4 & 10 & 12 & 6 \\ 3 & 16 & 13 & 5 & 9 \end{bmatrix}$	23	$\begin{bmatrix} 17 & 10 & 8 & 9 & 2 \\ 12 & 18 & 11 & 8 & 18 \\ 2 & 9 & 16 & 20 & 19 \\ 0 & 10 & 17 & 11 & 12 \\ 14 & 16 & 1 & 12 & 8 \end{bmatrix}$	27	$\begin{bmatrix} 14 & 15 & 10 & 7 & 1 \\ 14 & 6 & 14 & 10 & 14 \\ 3 & 18 & 17 & 15 & 9 \\ 9 & 17 & 13 & 18 & 9 \\ 6 & 8 & 12 & 15 & 3 \end{bmatrix}$

14. НЕЛІНІЙНА ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ

14.1. Основні поняття

В цифровій обробці зображень широко застосовуються нелінійні алгоритми на основі рангової статистики для відновлення зображень, пошкоджених різними моделями шумів. Вони дозволяють уникнути додаткового спотворення зображення при видаленні шуму, а також значно поліпшити результати роботи фільтрів на зображеннях з великою мірою зашумленості.

Введемо поняття M -околу елементу зображення $A[x, y]$, який є для цього околу центральним. У простому випадку M -окіл містить N пікселів – точки, що потрапляють в маску фільтру, включаючи (або не включаючи) центральний. Значення цих N елементів можна розташувати у варіаційному ряду $V(r)$, ранжируваному за зростанням (або спаданням), і обчислити певні моменти цього ряду, наприклад, середнє значення яскравості m_N та дисперсії d_N . Обчислення вихідного значення фільтру, яким замінюється центральний відлік, виконується за наступною формулою:

$$B[x, y] = \alpha A[x, y] + (1 - \alpha)m_N.$$

Значення коефіцієнта $\alpha = [0, 1]$ зв'язується певною залежністю із статистикою відліків у вікні фільтру, наприклад:

$$\alpha = \frac{d_N}{d_N + kd_S},$$

де d_S — дисперсія шумів по зображенню в цілому або по S -околу при $S > M$ та $M \in S$; k – константа довіри дисперсії S -околів.

Як випливає з цієї формули, при $k=1$ та $d_N \approx d_S$ має місце $\alpha \approx 0.5$, а значення $B[x, y] = (A[x, y] + m_N) / 2$. Таким чином, додаються в рівній степені значення центрального відліку та середнього значення пікселів його M -околу. При збільшенні значень d_N відбувається збільшення вкладу в результат центрального відліку, при зменшенні – значення m_N . Вага вкладу середніх значень по M -околу можна змінювати значенням коефіцієнта k .

Вибір статистичної функції і характер залежності від неї коефіцієнта α може бути досить різноманітним (наприклад, по дисперсіях різниць відліків в M -околі з центральним відліком), і залежить як від розмірів апертури фільтру, так і від характеру зображень і шумів. По суті, значення коефіцієнта α повинне задавати міру пошкодження центрального відліку і, відповідно, функцію запозичення для його виправлення відліків з M -околу.

Найбільш простими і поширеними типами нелінійних фільтрів для обробки зображень є порогові і медіанні фільтри.

14.2. Порогова фільтрація

Порогова фільтрація задається, наприклад, наступним чином:

$$B[x, y] = \begin{cases} A[x, y], & A[x, y] - m_N \leq p; \\ m_N, & A[x, y] - m_N > p. \end{cases}$$

Величина p є порогом фільтрації. Якщо величина центральної точки фільтру перевищує середнє значення відліків m_N в її M -околі на величину порога, то вона замінюється середнім значенням. Значення порогу може бути як константою, так і функціонально залежним від величини центральної точки.

14.3. Медіанна фільтрація

Медіанна фільтрація визначається наступним чином:

$$B[x, y] = \text{med}\{M[x, y]\},$$

тобто результат фільтрації є медіанне значення пікселів околу, форма яких визначається маскою фільтру. Медіанна фільтрація здатна ефективно видаляти із зображення перешкоди, що незалежно впливають на окремі пікселі. Наприклад, такими перешкодами є «биті» пікселі при цифровій зйомці, «сніговий» шум, коли частина пікселів замінюється на пікселі з максимальною інтенсивністю, і тому подібне. Перевага медіанної фільтрації полягає в тому, що «гарячий» піксель на темному фоні буде замінений темним, а не «розмазаним» по околу.

Медіанна фільтрація має виражену вибірковість по відношенню до елементів масиву, що є немонотонною складовою послідовності чисел в межах апертури фільтру. В той же час монотонну складову послідовності медіанний фільтр залишає без змін. Завдяки цій особливості, медіанні фільтри при оптимально вибраній апертурі зберігають без спотворень різкі межі об'єктів, пригнічуючи некорельовані або слабо корельовані завади та малорозмірні деталі.

14.4. Фільтри екстремумів

Разом з медіанними фільтрами широко застосовується метод мінімаксної фільтрації, що використовує для обробки значення мінімального і максимального елементів варіаційного ряду, побудованого з відліків вікна фільтру. Фільтри екстремумів визначаються за такими правилами:

$$B_{\min}[x, y] = \min\{M[x, y]\},$$

$$B_{\max}[x, y] = \max\{M[x, y]\},$$

тобто результат фільтрації є мінімальне і максимальне значення пікселів в масці фільтра. Застосовуються такі фільтри, як правило, для бінарних зображень.

За наявності уніполярного імпульсного шуму, що характеризується або позитивними, або негативними викидами з середнього рівня фонові складові, медіанний фільтр може виявитися недостатньо надійним, коли щільність шуму висока і більше половини пікселів вікна обробки складають викиди однакової полярності. Очевидний вихід з цієї ситуації – використати елемент мінімального рангу для викидів позитивної полярності і елемент максимального рангу для викидів негативної полярності. В цьому випадку шумові імпульси віддаляються навіть при дуже сильному рівні зашумленості. В той же час окреме застосування мінімального і максимального фільтру призводить до спотворення форми сигналу об'єкта. Тому, з метою збереження форми корисного сигналу, доцільна послідовна схема мінімаксної фільтрації, що складається з двох проходів по зображенню і обробки спочатку мінімальним (максимальним), а потім максимальним (мінімальним) рангом локальної статистики, що збільшує ефективність фільтрації і у разі біполярного імпульсного шуму. Оптимальна послідовність, в якій слід вибирати мінімальну (максимальну) процедуру, визначається характеристиками вхідного зображення: якщо неспотворене зображення складається з яскравих об'єктів на темному фоні, то правильна послідовність min-max. Зворотна процедура справедлива для негативного зображення.

14.5. Нелінійна фільтрація зображень в MATLAB

14.5.1. Медіанна фільтрація зображення

Ефективним засобом фільтрації шуму типу «сіль та перець» (від англ. salt and pepper) є медіанна фільтрація. Для цього класу шуму ця

нелінійна фільтрація зазвичай дає кращі результати, ніж фільтрація на основі операції згортки. Вона є окремим випадком рангової фільтрації, описаної нижче.

Функція **B=medfilt2(A,[n m])** фільтрує матрицю початкового зображення, використовуючи маску фільтру розміру $n \times m$. Центральний пік маски отримують усередненням усіх її пікселів. Маска застосовується нерекурсивно до усього зображення. Якщо зображення **A** не задовольняє умовам застосування маски, воно доповнюється нульовими елементами при класі **uint8** або одиницями при класі **double**.

B = medfilt2(A) задає медіанну фільтрацію при масці, що містить 3x3 пікселі.

B = medfilt2(A, 'indexed'...) здійснює медіанну фільтрацію для палітрового зображення.

Вхідне зображення представляється матрицею **A**, що належить до класів **uint8**, **uint16** або **double** (якщо використовується індексоване зображення, то **A** не може бути класу **uint16**). Клас вихідної матриці **B** є той же, що клас вхідної матриці **A**.

Розглянемо побудову початкового зображення монет, накладення на нього сильної шумової компоненти і здійснення медіанної фільтрації для маски 3x3 та 8x8 пікселів (рис. 14.1):

```
%Програма №14.1
I = imread('eight.tif');
J = imnoise(I,'salt & pepper',0.02);
subplot(2,2,1); subimage(I);
title('Вхідне зображення');
subplot(2,2,2); subimage(J);
title('Зашумлене зображення');
K_med1 = medfilt2(J);
subplot(2,2,3);subimage(K_med1);
title('Фільтроване зображення 3*3');
K_med2 = medfilt2(J,[5 5]);
subplot(2,2,4);subimage(K_med2);
title('Фільтроване зображення 5*5');
```

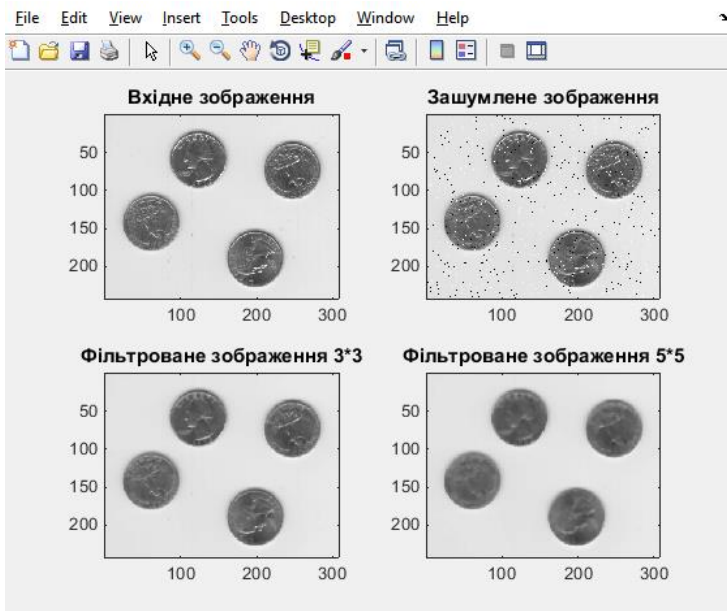


Рисунок 14.1 – Результати медіанної фільтрації для різних розмірів маски

14.5.2. Рангова фільтрація

Для здійснення рангової фільтрації півтонових зображень служить функція **B = ordfilt2(A,order,domain)**.

Пікселі початкового зображення, маски фільтру **domain**, що відповідають ненульовим елементам, сортуються в порядку зростання. Пікселю зображення **A**, що відповідає центральному елементу маски, присвоюється значення з порядковим номером **order**. Це робиться не рекурсивно. Щоб розміри **A** та **B** були однакові, матриця **A** тимчасово може доповнюватися елементами за правилом, описаним в 14.5.1 для функції **medfilt2**.

Функція **B = ordfilt2(A,order,domain,S)** працює аналогічно, але дозволяє задати матрицю **S**, з якої беруться відсутні для реалізації методу ненульові елементи.

B = ordfilt2(...padopt) дозволяє задати додатковий граничний па-

раметр **padopt**='zeros' (за замовчуванням) або 'symmetric'.

Матриця **A** може бути класів **uint8**, **uint16** або **double**. Матриця **B** має той же клас, що і матриця **A**, за винятком використання адитивної форми функції (з матрицею **S**), коли **B** – **double**.

Найцікавішою властивістю цього алгоритму є можливість задання ефектів розфокусування (ерозії) і фокусування (уточнення, нарощування фону і так далі) зображення. У першому випадку центральний піксель маски потрібно вибрати як піксель мінімальний, а в другому випадку – як піксель максимальний (за порядком в масці). Змінюючи положення центрального пікселя маски, властивості рангової фільтрації можна міняти в широких межах. Ці ефекти для зображення монет наочно ілюструє наступний приклад (рис. 14.2):

```
%Програма №14.2
I=imread('eight.tif');
subplot(1,3,1); imshow(I);
I2=ordfilt2(I,1,ones(3,3));
subplot(1,3,2); imshow(I2);
I3=ordfilt2(I,9,ones(3,3));
subplot(1,3,3); imshow(I3);
```

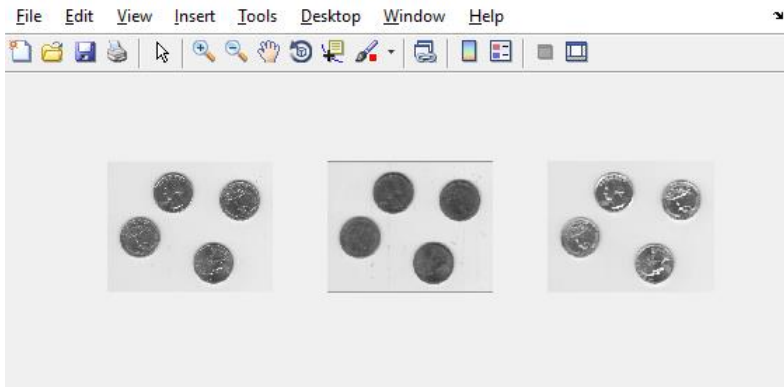


Рисунок 14.2 – Вихідне зображення монет (зліва), зображення із розфокусуванням (центр) та з фокусуванням (справа)

14.6. Приклади до розділу 14

Приклад №14.1. Дано зображення:

$$X = \begin{bmatrix} 16 & 19 & 1 & 18 & 3 \\ 0 & 19 & 1 & 6 & 11 \\ 11 & 9 & 6 & 7 & 17 \\ 7 & 9 & 1 & 15 & 8 \\ 4 & 13 & 0 & 14 & 15 \end{bmatrix}.$$

Необхідно виконати медіанну фільтрацію з лінійними розмірами апертури 3×3 .

Розв'язок.

Перед початком розрахунку необхідно вибрати метод обробки крайових точок. Нехай неіснуючі пікселі за межами зображення дорівнюють нулю.

Тоді для розрахунку значення пікселя вихідного зображення y_{ij} необхідно виконати наступні дії:

1) записати в окремий масив $\bar{z}$ значення пікселів, що потрапили в апертуру фільтру;

2) відсортувати отриманий масив $\bar{z}$ за зростанням;

3) $y_{ij} = z_5$.

Для відліку y_{11} отримаємо:

1) $\bar{z} = (0,0,0,0,16,19,0,0,19)$;

2) відсортований масив $\bar{z} = (0,0,0,0,0,16,19,19)$;

3) $y_{11} = 0$.

Для відліку y_{12} отримаємо:

1) $\bar{z} = (0,0,0,16,19,1,0,19,1)$;

2) відсортований масив $\bar{z} = (0,0,0,0,1,16,19,19)$;

3) $y_{12}=1$.

В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 9 & 9 & 7 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 \\ 7 & 7 & 9 & 8 & 8 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Аналіз результату показує, що вибраний невдалий спосіб обробки крайових відліків. Так, наприклад, скупчення великих значень у верхньому лівому кутку і в нижньому правому кутку замінюються на малі.

Скористаємося іншим способом обробки крайових відліків. Продублюємо крайові відліки, записав їх замість відсутніх.

Тоді для відліку y_{11} отримаємо:

1) $\bar{z} = (16, 16, 19, 16, 16, 19, 0, 0, 19)$;

2) відсортований масив $\bar{z} = (0, 0, 16, 16, \mathbf{16}, 16, 19, 19, 19)$;

3) $y_{11} = 16$.

Для відліку y_{12} отримаємо:

1) $\bar{z} = (16, 19, 1, 16, 19, 1, 0, 19, 1)$;

2) відсортований масив $\bar{z} = (0, 1, 1, 1, \mathbf{16}, 16, 19, 19, 19)$;

3) $y_{12} = 16$.

В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 16 & 16 & 18 & 3 & 6 \\ 11 & 9 & 7 & 6 & 11 \\ 9 & 7 & 7 & 7 & 11 \\ 9 & 7 & 9 & 8 & 15 \\ 7 & 4 & 13 & 14 & 15 \end{bmatrix}.$$

Приклад №14.2. Дано зображення:

$$X = \begin{bmatrix} 16 & 19 & 1 & 18 & 3 \\ 0 & 19 & 1 & 6 & 11 \\ 11 & 9 & 6 & 7 & 17 \\ 7 & 9 & 1 & 15 & 8 \\ 4 & 13 & 0 & 14 & 15 \end{bmatrix}.$$

Необхідно виконати мінімаксну фільтрацію з лінійними розмірами апертури 3×3 .

Розв'язок.

Перед початком розрахунку необхідно вибрати метод обробки крайових точок. Нехай неіснуючі пікселі за межами зображення дорівнюють нулю.

Виконаємо спочатку фільтрацію фільтром мінімумів. Тоді $y_{11} = \min\{0,0,0,0,16,19,0,0,19\} = 0$; $y_{12} = \min\{0,0,0,16,19,1,0,19,1\} = 0$ і так далі. В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Скористаємося іншим способом обробки крайових відліків. Продублюємо крайові відліки, записав їх замість відсутніх. Тоді $y_{11} = \min\{16,16,19,16,16,19,0,0,19\} = 0$;

$y_{12} = \min\{16,19,1,16,19,1,0,19,1\} = 0$ і так далі.

В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 0 & 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 1 & 6 \\ 4 & 0 & 0 & 0 & 7 \\ 4 & 0 & 0 & 0 & 8 \end{bmatrix}.$$

Тепер виконаємо фільтрацію фільтром максимумів. Тоді $y_{11} = \max\{0,0,0,0,16,19,0,0,19\} = 19$; $y_{12} = \max\{0,0,0,16,19,1,0,19,1\} = 19$ і так далі. В результаті отримаємо наступне відфільтроване зображення:

$$Y = \begin{bmatrix} 19 & 19 & 19 & 18 & 18 \\ 19 & 19 & 19 & 18 & 18 \\ 19 & 19 & 19 & 17 & 17 \\ 13 & 13 & 15 & 17 & 17 \\ 13 & 13 & 15 & 15 & 15 \end{bmatrix}.$$

Інший спосіб обробки крайових відліків: продублюємо крайові відліки, записав їх замість відсутніх. Тоді $y_{11} = \max\{16,16,19,16,16,19,0,0,19\} = 19$; $y_{12} = \max\{16,19,1,16,19,1,0,19,1\} = 19$ і так далі.

Порівнюючи результати фільтрації фільтром максимумів з різними способами обробки крайових точок легко помітити, що результати повністю співпадають. Це пояснюється тим, що незалежно від способу заповнення неіснуючих відліків в околі фільтрованого пікселя не може потрапити значення більше, ніж значення існуючих відліків.

14.7. Завдання до розділу 14 для самостійної роботи

Виконати медіанну фільтрацію двовірного сигналу з лінійними розмірами апертури 3×3 . Вказати спосіб обробки крайових точок. Вхідні дані див. в табл. 13.1.

15. СТИСНЕННЯ ЗОБРАЖЕНЬ

15.1. Дискретне косинусне перетворення

Дискретне косинусне перетворення (ДКП) є різновидом перетворення Фур'є. По суті, воно є дійсною частиною ДПФ. Нагадаємо, що двовірне ДПФ зображення **A** обчислюється за наступним виразом:

$$S[k, l] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} A[n, m] e^{-j \frac{2\pi}{N} nk - j \frac{2\pi}{M} ml} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} A[n, m] e^{-j \frac{2\pi}{N} nk} e^{-j \frac{2\pi}{M} ml};$$

$$0 \leq k \leq N-1; 0 \leq l \leq M-1.$$

Визначивши ДКП $S_c[k, l]$ ($0 \leq k \leq N-1; 0 \leq l \leq M-1$) як дійсну частину цього перетворення, отримаємо:

$$S_c[k, l] = \text{Re}(S[k, l]) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} A[n, m] \cos\left(\frac{2\pi}{N} nk\right) \cos\left(\frac{2\pi}{M} ml\right).$$

Це одна з форм ДКП. У загальному вигляді двовірне дискретне косинусне перетворення зображення **A** з розмірами $N \times M$ реалізується згідно з наступним виразом:

$$S_c[k, l] = \alpha_k \alpha_l \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} A[n, m] \cos \frac{\pi(2n+1)k}{2N} \cos \frac{\pi(2m+1)l}{2M}; \quad (15.1)$$

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{N}}, & \text{якщо } k = 0; \\ \sqrt{\frac{2}{N}}, & \text{якщо } 1 \leq k \leq N-1; \end{cases} \quad (15.2)$$

$$\alpha_l = \begin{cases} \frac{1}{\sqrt{M}}, & \text{якщо } l = 0; \\ \sqrt{\frac{2}{M}}, & \text{якщо } 1 \leq l \leq M-1, \end{cases} \quad (15.3)$$

де $S_c[k, l]$ – коефіцієнти двовірного дискретного косинусного перетворення ($0 \leq k \leq N-1$; $0 \leq l \leq M-1$).

Зворотнє дискретне косинусне перетворення (ЗДКП) реалізується згідно з виразами:

$$A[n, m] = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \alpha_k \alpha_l S_c[k, l] \cos \frac{\pi(2n+1)k}{2N} \cos \frac{\pi(2m+1)l}{2M}, \quad (15.4)$$

де α_k , α_l обчислюються за виразами (15.2) та (15.3) відповідно; $0 \leq n \leq N-1$; $0 \leq m \leq M-1$.

Існує два різні шляхи реалізації дискретних косинусних перетворень. Перший метод базується на безпосередньому обчисленні ДКП за виразами (15.1) – (15.3). Другий метод використовує матрицю дискретних косинусних перетворень $\mathbf{T}$, яка формується згідно наступного виразу:

$$T[k, l] = \begin{cases} \frac{1}{\sqrt{N}}, & \text{якщо } k = 0, 0 \leq l \leq N-1; \\ \sqrt{\frac{2}{N}} \cos \frac{\pi(2l+1)k}{2N}, & \text{якщо } 1 \leq k \leq N-1, 0 \leq l \leq N-1. \end{cases} \quad (15.5)$$

Для матриці $\mathbf{A}$ з розмірами $N \times N$ добуток $\mathbf{T} \times \mathbf{A}$ є матрицею з розмірами $N \times N$, де кожен стовпець містить одновимірне дискретне косинусне перетворення початкової матриці $\mathbf{A}$. Двовірне дискретне косинусне перетворення матриці $\mathbf{A}$ обчислюється як

$$\mathbf{B} = \mathbf{T} \times \mathbf{A} \times \mathbf{T}' . \quad (15.6)$$

Зворотнє двовірне дискретне косинусне перетворення $\mathbf{B}$ обчислюється як $\mathbf{A} = \mathbf{T}' \times \mathbf{B} \times \mathbf{T}$.

У наведеному нижче прикладі виконується розрахунок матрицю дискретних косинусних перетворень $\mathbf{T}$ двома способами та розрахунок ДКП трьома способами (рис. 15.1):

```
%Програма №15.1
%розрахунок матриці дискретних косинусних
%перетворень T за виразом (15.5)
N=4;
k=0:N-1;
l=k';
for i=1:N
    a(i)=2^0.5;
end
a(1)=1;
koef= repmat(a',1,N);
TT=(1/N^0.5*cos(pi*(2*l+1)*k/2/N))' .* koef
%розрахунок матриці дискретних косинусних
%перетворень T за допомогою вбудованої функції
T = dctmtx(N)
%Початкове зображення
X=[1 1 1 1; 2 2 2 2; 3 3 3 3; 4 4 4 4];
%1-й спосіб: розрахунок ДКП за виразом (15.6)
Y1=T*X*T'
%2-й спосіб:
%розрахунок за допомогою одновимірного ДКП
Y2=dct(dct(X))'
%3-й спосіб: розрахунок за допомогою двовимірного ДКП
Y3=dct2(X)
```

Аналіз результатів показує, що розрахунки різними способами дають однакові результати.

```

TT =

    0.5000    0.5000    0.5000    0.5000
    0.6533    0.2706   -0.2706   -0.6533
    0.5000   -0.5000   -0.5000    0.5000
    0.2706   -0.6533    0.6533   -0.2706

T =

    0.5000    0.5000    0.5000    0.5000
    0.6533    0.2706   -0.2706   -0.6533
    0.5000   -0.5000   -0.5000    0.5000
    0.2706   -0.6533    0.6533   -0.2706

Y1 =

    10.0000         0   -0.0000   -0.0000
    -4.4609   -0.0000    0.0000    0.0000
    -0.0000         0    0.0000    0.0000
    -0.3170   -0.0000    0.0000    0.0000

Y2 =

    10.0000         0         0         0
    -4.4609         0         0         0
         0         0         0         0
    -0.3170         0         0         0

Y3 =

    10.0000         0         0         0
    -4.4609         0         0         0
         0         0         0         0
    -0.3170         0         0         0

```

Рисунок 15.1 – Результати розрахунків за допомогою програми «Приклад №15.1»

15.2. Застосування дискретного косинусного перетворення для стиснення зображень

15.2.1. Передумови появи стандарту стиснення зображень

Кожне зображення складається з великої кількості даних. Наприклад, півтонове зображення розміром 320×240 , яке описується 8 бітами на піксель, займатиме 75 Кбайт, що еквівалентно приблизно 25 сторінкам тексту. Отже, для зберігання зображень досить часто потрібне їх стискування. Стиснуті зображення, як правило, кодуються для передачі, що ще більше їх стискає.

Раніше різні організації розробляли свої власні підходи і стандарти, поки в 1986 році група фахівців не організувала Об'єднану групу експертів по обробці фотозображень (Joint Photographic Experts Group – JPEG), яка спробувала стандартизувати стискування і передачу нерухомих чорно-білих і кольорових зображень. Комітет JPEG був одним з відділів Міжнародної організації по стандартизації (International Telegraph and Telephone Consultative Committee – CCITT) і МЕК (Міжнародна електротехнічна комісія, International Electronic Commission – IEC). Стандартизація потрібна для того, щоб дозволити взаємний обмін зображеннями між різними пристроями такими як ЕОМ, локальні мережі, зовнішні носії, цифрові камери тощо.

Стандарт JPEG встановлює набір базових функцій стискання зображень, які можуть відрізнятися деталями. Є основна структура, якої повинні дотримуватися усі системи, але окрім цього існують розширені та ієрархічні структури. Успіх JPEG сприяв появі MPEG (Moving Pictures Expert Group – Експертна група з питань рухомого зображення) та JBIG (Joint Bi-level Image Expert Group – Об'єднана група з питань дворівневих зображень). Розглянемо деякі основні поняття стандарту JPEG.

15.2.2. Дискретне косинусне перетворення в стандарті JPEG

Будь-яке прямокутне зображення можна представити як масив

числових значень, які передають такі ознаки зображення як інтенсивність, тон і колір. Кожне з цих значень називають пікселем, і зображення найчастіше складається з дуже великої кількості пікселів. Статистика зображення в різних областях може бути зовсім різною, тому для перетворення прийнятніше розділити зображення на ряд прилеглих одна до однієї менших ділянок з більш менш схожими статистичними характеристиками. Крім того, менші ділянки дозволяють отримати більше стискування через більшу кореляцію між сусідніми пікселями. Більше того, перетворення менших ділянок вчислити набагато легше. Тому базовий блок стандарту JPEG складається з квадрата 8×8 пікселів, що безпосередньо є сусідами один з одним.

Для перетворення цих блоків розміром 8×8 пікселів використовується двовірне ДКП. В результаті найнижчі частоти знаходяться у верхній лівій частині отриманої матриці, а найвищі частоти – в нижній правій. Оскільки, амплітуда низькочастотних компонентів, як правило, більше, ніж у високочастотних, то у верхній лівій частині відображення містяться відносно великі значення, а в нижній правій – малі.

У стандарті JPEG двовірне ДКП задається таким чином:

$$S_c[k, l] = \frac{1}{4} \alpha_k \alpha_l \sum_{n=0}^7 \sum_{m=0}^7 A[n, m] \cos \frac{\pi(2n+1)k}{16} \cos \frac{\pi(2m+1)l}{16}; \quad (15.7)$$

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } k = 0; \\ 1, & \text{якщо } k > 0; \end{cases} \quad (15.8)$$

$$\alpha_l = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } l = 0; \\ 1, & \text{якщо } l > 0, \end{cases} \quad (15.9)$$

де $S_c[k, l]$ – коефіцієнти двовірного ДКП; $A[n, m]$ – значення пікселя з блоку 8×8 пікселів; $0 \leq k \leq 7$; $0 \leq l \leq 7$.

Зворотне двомірне ДКП, необхідне для відновлення зображення, має вигляд:

$$A[n, m] = \frac{1}{4} \sum_{k=0}^7 \sum_{l=0}^7 \alpha_k \alpha_l S_c[k, l] \cos \frac{\pi(2n+1)k}{16} \cos \frac{\pi(2m+1)l}{16}, \quad (15.10)$$

де α_k , α_l обчислюються за виразами (15.8) і (15.9) відповідно; $0 \leq n \leq 7$; $0 \leq m \leq 7$.

15.2.3. Кодування зображення

В результаті двомірного ДКП верхній лівий коефіцієнт в масиві відповідає середньому рівню сигналу для блоку 8×8 пікселів, і його значення зазвичай не дуже швидко змінюється при переході до сусіднього блоку. Отже, цей коефіцієнт розглядається окремо від інших коефіцієнтів. Він кодується за допомогою диференціальної імпульсно-кової модуляції і йому привласнюється значення різниці між його значенням і відповідним значенням коефіцієнта попереднього блоку пікселів. В результаті, як правило, виходять відносно невеликі значення. Інші коефіцієнти розташовані послідовно в зигзагоподібному порядку в масиві 8×8 пікселів (рис. 15.2) Послідовність будується згідно з номерами в масиві з розташуванням двомірних коефіцієнтів в низхідному порядку.

Після цього обидва набори коефіцієнтів кодуються, внаслідок чого дані стискаються ще більше. Якщо попадаються довгі серії нулів, їх можна стиснути за допомогою серійного кодування. При такому кодуванні просто показується скільки нулів в серії. Якщо подібна серія зустрічається у кінці зигзагу або блоку, використовується кодове слово. Інші коефіцієнти кодуються за допомогою коду Хаффмана за стандартною послідовною схемою або за допомогою арифметичного коду за схемою розширеного ДКП. Обидва коди містять кодові слова різної довжини, причому частіше зустрічаються найкоротші кодові слова. Це зменшує число бітів, які передаватимуться, або, іншими словами,

збільшується міра стиснення.

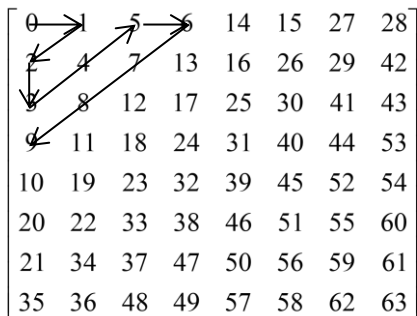


Рисунок 15.2 – Зигзагоподібна послідовність двомірного ДКП

Ці два коди найефективніші, тобто вони передають максимум інформації за допомогою найменшої кількості бітів. У коді Хаффмана використовується оптимальний набір кодових слів, які містять ціле число інформаційних бітів. При використанні арифметичного кодування стискування збільшується приблизно на 10%. Використовується один з видів однопрохідного адаптивного кодування, при якому список кодових слів динамічно адаптується до даних, які потрібно закодувати.

У наведеному нижче прикладі демонструється здібність відновлювати зображення за допомогою ДКП після проріджування коефіцієнтів у кожному блоці 8×8 пікселів (рис. 15.3):

```
%Програма №15.2
I = imread('coins.png'); I = im2double(I);
subplot(2,2,1); imshow(I);
subplot(2,2,2); imhist(I);
T = dctmtx(8); %матриця ДКП
%пряме ДКП блоків
B = blkproc(I,[8 8],'P1*x*P2',T,T');
%маска для проріджування коефіцієнтів в блоках
```

```

mask = [1  1  1  1  0  0  0  0
        1  1  1  0  0  0  0  0
        1  1  0  0  0  0  0  0
        1  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0];

%проріджування частот в блоках
B2 = blkproc(B,[8 8],'P1.*x',mask);
I2 = blkproc(B2,[8 8],'P1*x*P2','T',T);%ЗДКП блоків
subplot(2,2,3);imshow(I2);
subplot(2,2,4); imhist(I2);

```

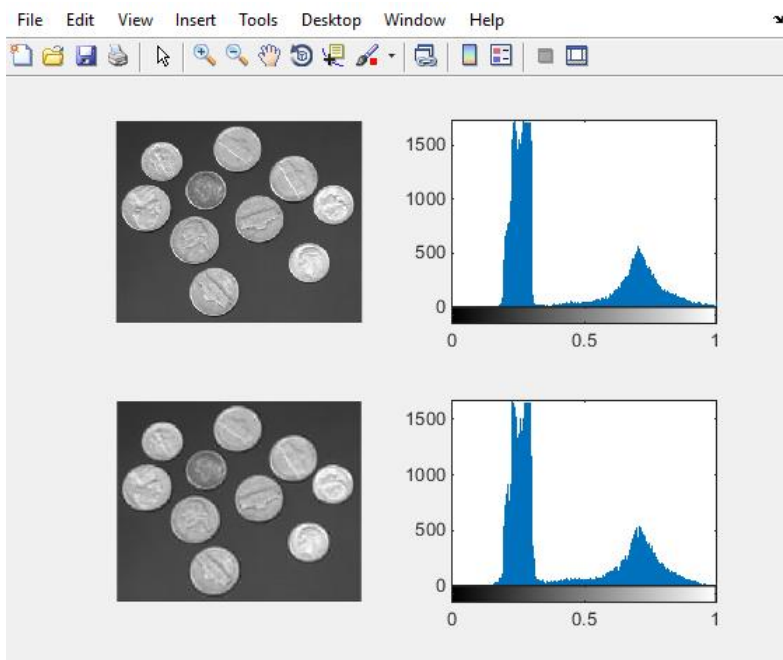


Рисунок 15.3 – Результати роботи програми «Приклад №15.2»

15.3. Основи вейвлет-перетворень

Ідея вейвлет-стискування зображень, як і інших методів з перетворенням, досить проста. Спочатку до зображення застосовується вейвлет-перетворення, а потім з даних перетвореного зображення видаляються деякі коефіцієнти. До коефіцієнтів, що залишилися, може бути застосоване кодування. Стисле зображення відновлюється шляхом декодування коефіцієнтів, якщо це необхідно, і застосуванням зворотного перетворення до результату. Передбачається, що в процесі видалення частини коефіцієнтів перетворення втрачається не надто багато інформації.

Процес стискування включає застосування до зображення вейвлет-перетворення, за яким йде певний вид децимації і/або квантування, а потім, при необхідності, кодування отриманих вейвлет-коефіцієнтів. Відновлюється зображення шляхом декодування коефіцієнтів, якщо воно потрібне, і застосуванням зворотного вейвлет-перетворення.

Найпростіше стиснути зображення, якщо замінити це зображення середнім значенням його пікселів. Цей підхід дає дуже велику степінь стискування (одно число представляє усе зображення), але небагато дає негативу в плані якості зображення. Щоб відновити початкове зображення з цього середнього значення, нам треба знати, які деталі були видалені для отримання цього значення. Розглянемо зображення, що складається з двох точок (x_1, x_2) . Ці значення можуть бути замінені середнім значенням a і напіврізницею d :

$$a = (x_1 + x_2)/2, \quad d = (x_1 - x_2)/2,$$

(коефіцієнт $1/2$ вводиться у визначення d для зручності позначень).

Помітимо, що ми можемо виразити (x_1, x_2) через (a, d) :

$$x_1 = a + d, \quad x_2 = a - d.$$

Вейвлет-перетворення початкової послідовності (x_1, x_2) – це (a, d) . У цьому представленні інформація не додається і не втрачається. Тоді виникає питання, яка користь від заміни (x_1, x_2) на (a, d) . Особливої користі немає, якщо тільки два значення (x_1, x_2) не виявляються близькими один до одного. В цьому випадку різниця d мала, і зображення (x_1, x_2) можна замінити його наближенням (a) .

Помітимо, що цей новий вид має менше відліків, ніж його наближення. Ми отримали стискування зображення. Відновленим зображенням буде зображення (a, a) з помилкою із зображення (error image) $(|x_1 - a|, |x_2 - a|) = (|d|, |d|)$. Оскільки d мале, то і помилка буде мала.

Розглянемо дещо більше зображення (x_1, x_2, x_3, x_4) . Обчислимо середні значення $a_{1,0} = (x_1 + x_2)/2$, $a_{1,1} = (x_3 + x_4)/2$ і різниці $d_{1,0} = (x_1 - x_2)/2$, $d_{1,1} = (x_3 - x_4)/2$.

Подвійні нижні індекси тут показують, що ми починаємо багатокроковий процес, і це його перший крок. Як і раніше, ми отримали новий вид $(a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1})$ початкового зображення, яке містить рівно стільки ж відліків, скільки і початкове. Якщо ми захочемо стиснути це зображення, ми звернемо увагу на величину значень $d_{1,0}, d_{1,1}$ і вирішимо, чи можуть вони без збитку бути видалені. При цьому ми отримаємо стисле зображення $(a_{1,0}, a_{1,1})$. Припустимо, проте, що нас не задовольняє така міра стискування, і ми хочемо стиснути зображення сильніше. Тоді ми можемо застосувати ту ж процедуру до залишених відліків $(a_{1,0}, a_{1,1})$ і знову обчислити середнє значення і різницю:

$$a_{0,0} = (a_{1,0} + a_{1,1})/2, \quad d_{0,0} = (a_{1,0} - a_{1,1})/2.$$

Якщо різниця $d_{0,0}$ досить мала, ми можемо замінити усе початкове зображення (x_1, x_2, x_3, x_4) зображенням, що складається з одного єдиного пікселя $(a_{0,0})$. Подивимося, чим же є це $(a_{0,0})$:

$$a_{0,0} = (a_{1,0} + a_{1,1}) / 2 = ((x_1 + x_2) / 2 + (x_3 + x_4) / 2) / 2 = (x_1 + x_2 + x_3 + x_4) / 4.$$

Величина $a_{0,0}$ – це усього лише середнє значення усіх відліків початкового зображення. Якщо початкове зображення однорідне сіре (тобто усі x , дорівнюють одному і тому ж значенню), то ми можемо стиснути його, замінивши єдиним значенням, рівним цій величині градації сірого. Значення $a_{0,0}$ представляє дуже приблизний рівень інформації про це зображення, тобто інформацію при найнижчому розділенні або при найгрубішому масштабі. Значення $a_{1,0}$, $a_{1,1}$, разом узяті, представляють інформацію на наступному, більш високому рівні розділення, або при наступному, кращому масштабі. Помітимо, що ми можемо відновити $(a_{1,0}, a_{1,1})$ із $(a_{0,0}, d_{0,0})$, використовуючи наведену вище процедуру. Початкові значення відліків (x_1, x_2, x_3, x_4) представляють найвище розділення або найкращий масштаб, можливий для цього зображення. Ці значення можуть бути відновлені з $(a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1})$. Але оскільки ми можемо отримати $a_{1,0}$, $a_{1,1}$ з $a_{0,0}$, $d_{0,0}$, то ми можемо відновити значення пікселів початкового зображення із загального середнього значення $a_{0,0}$ і різниць $d_{0,0}$, $d_{1,0}$, $d_{1,1}$. Таким чином, послідовність

$$(a_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}) \tag{15.10}$$

є альтернативним видом початкового зображення і складається із загального середнього і значень різниць, що виражають два різні рівні

деталізації.

Послідовність (15.10) – це вейвлет-перетворення початкової послідовності (x_1, x_2, x_3, x_4) . Помітимо, що тепер ми маємо більше варіантів для стискування. Якщо $d_{1,0}$ та $d_{1,1}$ занадто великі, щоб їх ігнорувати, то, можливо, ми можемо виключити наступний рівень деталізації, а саме $d_{0,0}$. Якщо ми маємо справу із зображеннями більшого розміру, то продовжимо цей процес усереднювання і виділення деталей на грубішому рівні.

15.4. Двомірне вейвлет-перетворення

Раніше були розглянуті вейвлет-перетворення тільки для одновимірних послідовностей. Існує можливість поширити ідею вейвлет-перетворення на послідовності більшої розмірності. Перший спосіб полягає в тому, щоб спочатку перетворювати рядки зображення, а потім перетворювати стовпці зображення з перетвореними рядками.

Розглянемо двомірний аналог простої 4-елементної послідовності. Припустимо, є зображення 4×4

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{bmatrix}.$$

Один із способів отримання двомірного вейвлет-перетворення зображення розміром $2^n \times 2^n$, полягає в тому, щоб спочатку застосувати одновимірне вейвлет-перетворення до кожного з 2^n рядків, а потім застосувати одновимірне вейвлет-перетворення до кожного з 2^n стовпців.

Ми можемо представити собі операції усереднення і виділення

деталей за допомогою вейвлетів Хаара, як застосування низькочастотних і високочастотних фільтрів. Низькочастотний фільтр пропускає інформацію, передану на низьких частотах (тобто малу кількість деталей), блокуючи в той же час інформацію, передану на високих частотах (тобто що містить велику кількість деталей), а високочастотний фільтр – навпаки. Наше двовірне вейвлет-перетворення застосовує одновірне вейвлет-перетворення до кожного рядка, а потім одновірне вейвлет-перетворення до кожного стовпця результуючої інформації. Схема на рис. 15.4 показує, як виглядає трансформанта двовірного вейвлет-перетворення.

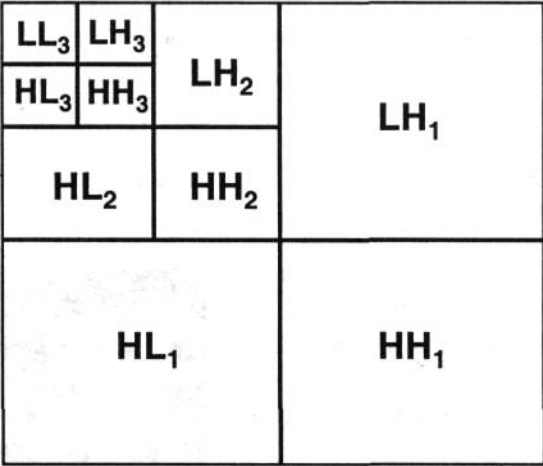


Рисунок 15.4 – Схема двовірного вейвлет-перетворення

Таким чином, чверть коефіцієнтів двовірного перетворення є результатом дії високочастотного фільтра (H) на рядки, а потім його дії на результуючі стовпці інформації. Це блок коефіцієнтів в правому нижньому кутку схеми, позначений HH1. Інша чверть коефіцієнтів двовірного перетворення є результатом дії низькочастотного фільтра

(L) на стовпці інформації, які вже пройшли через високочастотний фільтр. Цей блок, позначений LH1, знаходиться в правому верхньому кутку схеми. Аналогічно, блок HL1, розташований в лівому нижньому кутку схеми, є результатом низькочастотної обробки рядків з подальшою високочастотною обробкою стовпців. Лівий верхній кут схеми поділений на менші блоки, показуючи, що операція вейвлет-перетворення послідовно впливає на все меншу кількість коефіцієнтів, аж до останнього коефіцієнта в лівому верхньому кутку, до якого застосовувався тільки низькочастотний фільтр.

15.5. Приклади до розділу 15

Приклад №15.1. Дана послідовність $x[n] = [1, 3, 5, 7]$. Розрахуємо ненормоване вейвлет-перетворення Хаара.

Розв'язок.

Так як довжина послідовності дорівнює 4 відліки, то вейвлет-перетворення можна розрахувати за два кроки.

Крок 1.

Розрахуємо напівсуми $a_{1,0}$, $a_{1,1}$ і напіврізниці $d_{1,0}$, $d_{1,1}$ відповідних пар значень початкової послідовності:

$$a_{1,0} = (x_1 + x_2) / 2 = (1 + 3) / 2 = 2 ;$$

$$a_{1,1} = (x_3 + x_4) / 2 = (5 + 7) / 2 = 6 ;$$

$$d_{1,0} = (x_1 - x_2) / 2 = (1 - 3) / 2 = -1 ;$$

$$d_{1,1} = (x_3 - x_4) / 2 = (5 - 7) / 2 = -1 .$$

Запишемо спочатку отримані напівсуми $a_{1,0}$ та $a_{1,1}$, а потім на піврізниці $d_{1,0}$ та $d_{1,1}$, тобто $[a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1}]$. Таким чином, отримуємо

$[2, 6, -1, -1]$.

Крок 2.

Розрахуємо напівсуму $a_{0,0}$ і напіврізницю $d_{0,0}$ відповідних пар значень першої половини послідовності, що була отримана на попередньому кроці:

$$a_{0,0} = (a_{1,0} + a_{1,1}) / 2 = (2 + 6) / 2 = 4 ;$$

$$d_{0,0} = (a_{1,0} - a_{1,1}) / 2 = (2 - 6) / 2 = -2 .$$

Запишемо спочатку отримані напівсуму $a_{0,0}$ і напіврізницю $d_{0,0}$, потім напіврізниці $d_{1,0}$ та $d_{1,1}$, що були отримані на попередньому кроці, тобто $[a_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}]$. Таким чином, отримуємо $y[n] = [4, -2, -1, -1]$.

Приклад №15.2. Дано ненормоване вейвлет-перетворення Хаара $y[n] = [4, -2, -1, -1]$. Розрахуємо початкову послідовність.

Розв'язок.

Так як довжина вейвлет-перетворення дорівнює 4 відліки, то початкову послідовність можна розрахувати за два кроки.

Крок 1.

Розрахуємо суму і різницю між першою парою значень, які згідно (15.10) відображають загальне середнє значення $a_{0,0}$ і різницю $d_{0,0}$:

$$a_{1,0} = a_{0,0} + d_{0,0} = 4 - 2 = 2 ;$$

$$a_{1,1} = a_{0,0} - d_{0,0} = 4 + 2 = 6 .$$

Запишемо спочатку отримані значення $a_{1,0}$ та $a_{1,1}$, а потім пере-

пишемо значення $d_{1,0}$ та $d_{1,1}$ зі заданою послідовністю u , тобто $[a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1}]$.

Таким чином, отримуємо $[2, 6, -1, -1]$.

Крок 2.

Розрахунок відліків початкової послідовності виконується за наступними виразами:

$$x_1 = a_{1,0} + d_{1,0} = 2 - 1 = 1;$$

$$x_2 = a_{1,0} - d_{1,0} = 2 + 1 = 3;$$

$$x_3 = a_{1,1} + d_{1,1} = 6 - 1 = 5;$$

$$x_4 = a_{1,1} - d_{1,1} = 6 + 1 = 7.$$

Таким чином, отримуємо початкову послідовність $x[n] = [1, 3, 5, 7]$.

Приклад №15.3. Дана послідовність $x[n] = [1, 2, 3, 4, 5, 6, 7, 8]$. Розрахуємо ненормоване вейвлет-перетворення Хаара.

Розв'язок.

Так як довжина послідовності дорівнює 8 відлікам, то вейвлет-перетворення можна розрахувати за три кроки.

Крок 1.

Розрахуємо напівсуми $a_{2,0}$, $a_{2,1}$, $a_{2,2}$, $a_{2,3}$ і напіврізниці $d_{2,0}$, $d_{2,1}$, $d_{2,2}$, $d_{2,3}$ відповідних пар значень початкової послідовності:

$$a_{2,0} = (x_1 + x_2) / 2 = (1 + 2) / 2 = 1.5 ;$$

$$a_{2,1} = (x_3 + x_4) / 2 = (3 + 4) / 2 = 3.5 ;$$

$$a_{2,2} = (x_5 + x_6) / 2 = (5 + 6) / 2 = 5.5 ;$$

$$a_{2,3} = (x_7 + x_8) / 2 = (7 + 8) / 2 = 7.5 ;$$

$$d_{2,0} = (x_1 - x_2) / 2 = (1 - 2) / 2 = -0.5;$$

$$d_{2,1} = (x_3 - x_4) / 2 = (3 - 4) / 2 = -0.5;$$

$$d_{2,2} = (x_5 - x_6) / 2 = (5 - 6) / 2 = -0.5;$$

$$d_{2,3} = (x_7 - x_8) / 2 = (7 - 8) / 2 = -0.5.$$

Запишемо спочатку отримані напівсуми $a_{2,0}$, $a_{2,1}$, $a_{2,2}$ та $a_{2,3}$, а потім отримані напіврізниці $d_{2,0}$, $d_{2,1}$, $d_{2,2}$ та $d_{2,3}$, тобто $[a_{2,0}, a_{2,1}, a_{2,2}, a_{2,3}, d_{2,0}, d_{2,1}, d_{2,2}, d_{2,3}]$.

Таким чином, отримуємо $[1.5, 3.5, 5.5, 7.5, -0.5, -0.5, -0.5, -0.5]$.

Крок 2.

Розрахуємо напівсуми $a_{1,0}$, $a_{1,1}$ і напіврізниці $d_{1,0}$, $d_{1,1}$ відповідних пар значень першої половини послідовності, що була отримана на кроці 1:

$$a_{1,0} = (a_{2,0} + a_{2,1}) / 2 = (1.5 + 3.5) / 2 = 2.5;$$

$$a_{1,1} = (a_{2,2} + a_{2,3}) / 2 = (5.5 + 7.5) / 2 = 6.5;$$

$$d_{1,0} = (a_{2,0} + a_{2,1}) / 2 = (1.5 - 3.5) / 2 = -1;$$

$$d_{1,1} = (a_{2,2} + a_{2,3}) / 2 = (5.5 - 7.5) / 2 = -1.$$

Запишемо спочатку отримані напівсуми $a_{1,0}$, $a_{1,1}$ та напіврізниці $d_{1,0}$, а потім перепишемо напіврізниці $d_{2,0}$, $d_{2,1}$, $d_{2,2}$, $d_{2,3}$, що були отримані на кроці 1, тобто $[a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1}, d_{2,0}, d_{2,1}, d_{2,2}, d_{2,3}]$.

Таким чином, отримуємо $[2.5, 6.5, -1, -1, -0.5, -0.5, -0.5, -0.5]$.

Крок 3.

Розрахуємо напівсуму $a_{0,0}$ і напіврізницю $d_{0,0}$ відповідних пар значень першої половини послідовності, що була отримана на кроці 2:

$$a_{0,0} = (a_{1,0} + a_{1,1}) / 2 = (2.5 + 6.5) / 2 = 4.5;$$

$$d_{0,0} = (a_{1,0} - a_{1,1}) / 2 = (2 - 6) / 2 = -2.$$

Запишемо спочатку отримані напівсуму $a_{0,0}$ і напіврізницю $d_{0,0}$, потім перепишемо напіврізницю $d_{1,0}$ та $d_{1,1}$, що були отримані на кроці 2, а потім перепишемо напіврізницю $d_{2,0}$, $d_{2,1}$, $d_{2,2}$, $d_{2,3}$, що були отримані на кроці 1, тобто $[a_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}, d_{2,0}, d_{2,1}, d_{2,2}, d_{2,3}]$. Таким чином, отримуємо $y[n] = [4.5, -2, -1, -1, -0.5, -0.5, -0.5, -0.5]$.

Приклад №15.4. Дано ненормоване вейвлет-перетворення Хаара $y[n] = [4.5, -2, -1, -1, -0.5, -0.5, -0.5, -0.5]$. Розрахуємо початкову послідовність.

Розв'язок.

Так як довжина вейвлет-перетворення дорівнює 8 відлікам, то початкову послідовність можна розрахувати за три кроки.

Крок 1.

Розрахуємо суму і різницю між першою парою значень, які згідно (15.10) відображають загальне середнє значення $a_{0,0}$ і різницю $d_{0,0}$:

$$a_{1,0} = a_{0,0} + d_{0,0} = 4.5 - 2 = 2.5;$$

$$a_{1,1} = a_{0,0} - d_{0,0} = 4.5 + 2 = 6.5.$$

Запишемо спочатку отримані значення $a_{1,0}$ та $a_{1,1}$, а потім перепишемо значення $d_{1,0}$, $d_{1,1}$, $d_{2,0}$, $d_{2,1}$, $d_{2,2}$, $d_{2,3}$ зі заданою послідовністю y , тобто $[a_{1,0}, a_{1,1}, d_{1,0}, d_{1,1}, d_{2,0}, d_{2,1}, d_{2,2}, d_{2,3}]$.

Таким чином, отримуємо $[2.5, 6.5, -1, -1, -0.5, -0.5, -0.5, -0.5]$.

Крок 2.

Розрахуємо суми і різниці між відповідними парами першої половини послідовності, отриманої на кроці 1:

$$a_{2,0} = a_{1,0} + d_{1,0} = 2.5 - 1 = 1.5;$$

$$a_{2,1} = a_{1,0} - d_{1,0} = 2.5 + 1 = 3.5;$$

$$a_{2,2} = a_{1,1} + d_{1,1} = 6.5 - 1 = 5.5;$$

$$a_{2,3} = a_{1,1} - d_{1,1} = 6.5 + 1 = 7.5.$$

Запишемо спочатку отримані значення $a_{2,0}$, $a_{2,1}$, $a_{2,2}$ та $a_{2,3}$, а потім перепишемо значення $d_{2,0}$, $d_{2,1}$, $d_{2,2}$, $d_{2,3}$ зі заданою послідовністю у, тобто $[a_{2,0}, a_{2,1}, a_{2,2}, a_{2,3}, d_{2,0}, d_{2,1}, d_{2,2}, d_{2,3}]$.

Таким чином, отримуємо $[1.5, 3.5, 5.5, 7.5, -0.5, -0.5, -0.5, -0.5]$.

Крок 3.

Розрахунок відліків початкової послідовності виконується за наступними виразами:

$$x_1 = a_{2,0} + d_{2,0} = 1.5 - 0.5 = 1;$$

$$x_2 = a_{2,0} - d_{2,0} = 1.5 + 0.5 = 2;$$

$$x_3 = a_{2,1} + d_{2,1} = 3.5 - 0.5 = 3$$

$$x_4 = a_{2,1} - d_{2,1} = 3.5 + 0.5 = 4$$

$$x_5 = a_{2,2} + d_{2,2} = 5.5 - 0.5 = 5;$$

$$x_6 = a_{2,2} - d_{2,2} = 5.5 + 0.5 = 6$$

$$x_7 = a_{2,3} + d_{2,3} = 7.5 - 0.5 = 7;$$

$$x_8 = a_{2,3} - d_{2,3} = 7.5 + 0.5 = 8$$

Таким чином, отримуємо початкову послідовність

$$x[n] = [1, 2, 3, 4, 5, 6, 7, 8] .$$

Приклад №15.5. Дано зображення

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} .$$

Розрахуємо двомірне ненормоване вейвлет-перетворення Хаара.

Розв'язок.

Так як лінійні розміри зображення становлять 4×4 пікселів, то вейвлет-перетворення можна розрахувати за два етапи за допомогою одновимірного вейвлет-перетворення.

Етап 1.

Спочатку для кожного рядка розрахуємо тільки перший крок ненормованого вейвлет-перетворення Хаара (див. приклад №1).

В результаті отримуємо наступну матрицю

$$I_1 = \begin{bmatrix} 1.5 & 3.5 & -0.5 & -0.5 \\ 1.5 & 3.5 & -0.5 & -0.5 \\ 1.5 & 3.5 & -0.5 & -0.5 \\ 1.5 & 3.5 & -0.5 & -0.5 \end{bmatrix} .$$

Потім для кожного стовпця матриці I_1 також розрахуємо тільки перший крок ненормованого вейвлет-перетворення Хаара (див. приклад №1).

В результаті отримуємо наступну матрицю

$$I_2 = \begin{bmatrix} 1.5 & 3.5 & -0.5 & -0.5 \\ 1.5 & 3.5 & -0.5 & -0.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Етап 2.

Спочатку для кожного рядка лівого верхнього кута матриці I_2 розрахуємо другий крок ненормованого вейвлет-перетворення Хаара (див. приклад №1). В результаті отримуємо наступну матрицю

$$I_3 = \left[\begin{array}{cc|cc} 2.5 & -1 & -0.5 & -0.5 \\ 2.5 & -1 & -0.5 & -0.5 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right].$$

Потім для кожного стовпця матриці I_3 також розрахуємо другий крок ненормованого вейвлет-перетворення Хаара (див. приклад №1).

Таким чином на другому етапі розрахунок виконується тільки для лівого верхнього кута матриці, що отримана на першому етапі.

В результаті отримуємо шукану матрицю

$$J = \left[\begin{array}{cc|cc} 2.5 & -1 & -0.5 & -0.5 \\ 0 & 0 & -0.5 & -0.5 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right].$$

15.6. Завдання до розділу 15 для самостійної роботи

Обчислити вейвлет-перетворення Хаара (ненормоване):

1) Для одновимірної послідовності вхідні дані дивись в табл. 15.1 (N – номер студента за журналом).

2) Для двовимірного сигналу вхідні дані дивись в табл. 15.2 (N – номер студента за журналом).

Таблиця 15.1 – Варіанти завдань

N	Дискретний сигнал (для завдання 1)	N	Дискретний сигнал (для завдання 1)
1	$x[n] = (8; 9; 4; 1; 2; 0; 1; 4)$	16	$x[n] = (6; 5; 6; 1; 0; 0; 1; 3)$
2	$x[n] = (0; 4; 6; 7; 2; 4; 4; 6)$	17	$x[n] = (8; 0; 2; 2; 1; 5; 2; 9)$
3	$x[n] = (1; 2; 4; 8; 6; 3; 5; 9)$	18	$x[n] = (6; 0; 7; 2; 4; 1; 6; 5)$
4	$x[n] = (7; 4; 3; 9; 1; 4; 4; 5)$	19	$x[n] = (8; 7; 4; 1; 2; 8; 7; 9)$
5	$x[n] = (7; 0; 1; 2; 8; 9; 2; 6)$	20	$x[n] = (4; 8; 9; 4; 6; 0; 6; 4)$
6	$x[n] = (8; 1; 5; 4; 2; 8; 8; 4)$	21	$x[n] = (7; 4; 8; 5; 5; 7; 1; 4)$
7	$x[n] = (8; 5; 0; 6; 3; 3; 9; 6)$	22	$x[n] = (9; 8; 0; 1; 4; 7; 3; 3)$
8	$x[n] = (2; 9; 5; 9; 5; 6; 1; 8)$	23	$x[n] = (8; 7; 6; 7; 1; 4; 5; 3)$
9	$x[n] = (0; 1; 3; 3; 6; 1; 0; 4)$	24	$x[n] = (8; 9; 3; 8; 2; 9; 2; 6)$
10	$x[n] = (4; 0; 8; 1; 8; 5; 4; 2)$	25	$x[n] = (4; 9; 6; 5; 9; 4; 5; 0)$
11	$x[n] = (1; 7; 2; 1; 3; 3; 2; 9)$	26	$x[n] = (1; 0; 2; 7; 6; 7; 4; 7)$
12	$x[n] = (1; 2; 4; 6; 2; 9; 8; 3)$	27	$x[n] = (3; 3; 4; 2; 4; 2; 9; 1)$
13	$x[n] = (0; 6; 3; 6; 3; 8; 8; 1)$	28	$x[n] = (1; 4; 1; 4; 8; 8; 6; 7)$
14	$x[n] = (6; 9; 3; 5; 2; 3; 5; 8)$	29	$x[n] = (9; 1; 4; 1; 2; 8; 5; 9)$
15	$x[n] = (4; 7; 1; 2; 9; 1; 5; 5)$	30	$x[n] = (6; 1; 0; 9; 3; 6; 0; 8)$

Таблиця 15.2 – Варіанти двомірних сигналів

N	X	N	X	N	X
1	$\begin{bmatrix} 4 & 0 & 8 & 16 \\ 3 & 14 & 16 & 0 \\ 12 & 8 & 10 & 13 \\ 5 & 19 & 4 & 7 \end{bmatrix}$	8	$\begin{bmatrix} 8 & 11 & 18 & 12 \\ 6 & 2 & 5 & 19 \\ 6 & 0 & 3 & 13 \\ 7 & 9 & 17 & 17 \end{bmatrix}$	15	$\begin{bmatrix} 9 & 7 & 12 & 11 \\ 14 & 10 & 6 & 17 \\ 13 & 6 & 12 & 13 \\ 12 & 7 & 6 & 15 \end{bmatrix}$
2	$\begin{bmatrix} 3 & 13 & 9 & 13 \\ 13 & 7 & 17 & 6 \\ 6 & 17 & 16 & 5 \\ 10 & 17 & 12 & 6 \end{bmatrix}$	9	$\begin{bmatrix} 13 & 8 & 9 & 5 \\ 6 & 17 & 9 & 0 \\ 3 & 9 & 8 & 13 \\ 3 & 16 & 18 & 13 \end{bmatrix}$	16	$\begin{bmatrix} 12 & 19 & 6 & 19 \\ 9 & 13 & 12 & 9 \\ 16 & 19 & 10 & 8 \\ 13 & 18 & 8 & 11 \end{bmatrix}$
3	$\begin{bmatrix} 14 & 15 & 19 & 2 \\ 10 & 19 & 5 & 0 \\ 8 & 10 & 5 & 17 \\ 13 & 17 & 17 & 3 \end{bmatrix}$	10	$\begin{bmatrix} 7 & 18 & 3 & 10 \\ 0 & 16 & 18 & 18 \\ 8 & 7 & 11 & 6 \\ 15 & 12 & 12 & 13 \end{bmatrix}$	17	$\begin{bmatrix} 9 & 9 & 9 & 13 \\ 11 & 5 & 15 & 20 \\ 10 & 14 & 15 & 17 \\ 12 & 7 & 11 & 4 \end{bmatrix}$
4	$\begin{bmatrix} 11 & 4 & 4 & 11 \\ 8 & 11 & 7 & 15 \\ 10 & 15 & 15 & 1 \\ 6 & 10 & 13 & 12 \end{bmatrix}$	11	$\begin{bmatrix} 16 & 14 & 14 & 2 \\ 7 & 10 & 19 & 13 \\ 8 & 15 & 16 & 7 \\ 11 & 9 & 14 & 2 \end{bmatrix}$	18	$\begin{bmatrix} 7 & 10 & 17 & 9 \\ 12 & 18 & 11 & 8 \\ 2 & 9 & 14 & 7 \\ 10 & 0 & 17 & 11 \end{bmatrix}$
5	$\begin{bmatrix} 19 & 4 & 8 & 13 \\ 19 & 12 & 14 & 4 \\ 15 & 6 & 5 & 16 \\ 8 & 19 & 8 & 12 \end{bmatrix}$	12	$\begin{bmatrix} 7 & 11 & 8 & 18 \\ 10 & 14 & 12 & 12 \\ 5 & 19 & 16 & 5 \\ 11 & 15 & 1 & 17 \end{bmatrix}$	19	$\begin{bmatrix} 17 & 7 & 7 & 10 \\ 10 & 12 & 11 & 3 \\ 10 & 8 & 11 & 6 \\ 5 & 4 & 10 & 12 \end{bmatrix}$
6	$\begin{bmatrix} 9 & 7 & 12 & 1 \\ 0 & 13 & 0 & 7 \\ 0 & 1 & 0 & 12 \\ 6 & 0 & 3 & 14 \end{bmatrix}$	13	$\begin{bmatrix} 5 & 7 & 2 & 13 \\ 17 & 9 & 17 & 4 \\ 6 & 8 & 15 & 5 \\ 13 & 11 & 15 & 12 \end{bmatrix}$	20	$\begin{bmatrix} 12 & 11 & 8 & 10 \\ 10 & 11 & 12 & 4 \\ 16 & 13 & 11 & 20 \\ 10 & 13 & 8 & 13 \end{bmatrix}$
7	$\begin{bmatrix} 13 & 2 & 5 & 4 \\ 13 & 9 & 17 & 4 \\ 14 & 14 & 4 & 0 \\ 9 & 17 & 16 & 1 \end{bmatrix}$	14	$\begin{bmatrix} 18 & 3 & 16 & 7 \\ 11 & 16 & 12 & 3 \\ 6 & 19 & 14 & 16 \\ 9 & 11 & 1 & 16 \end{bmatrix}$	21	$\begin{bmatrix} 5 & 1 & 6 & 6 \\ 15 & 16 & 9 & 5 \\ 11 & 6 & 8 & 20 \\ 7 & 8 & 19 & 2 \end{bmatrix}$

СПИСОК ЛІТЕРАТУРИ

1. Обробка сигналів : навч. посіб. [для студентів спец. 122 "Комп'ютер. науки та інформ. технології", 123 "Комп'ютер. інженерія", 151 "Автоматизація та комп'ютер.-інтегр. технології", 172 "Телекомунікації та радіотехніка" / А. Й. Наконечний, Р. І. Стахів, Р. А. Наконечний ; Нац. ун-т "Львів. політехніка". – Львів : Растр-7, 2017. – 217 с.

2. Основи та методи цифрової обробки сигналів: від теорії до практики: навч. посібник / уклад. : Ю.О. Ушенко, М.С. Гавриляк, М.В. Талах, В.В. Дворжак. – Чернівці : Чернівецький нац. ун-т ім. Ю. Федьковича, 2021. – 308 с.

3. Цифрова обробка сигналів в телекомунікаційних системах : підручник / Г. Г. Бортник, В. М. Кичак. – Вінниця : ВНТУ, 2014. – 232 с.

4. Алгоритми та засоби обробки сигналів : навч. посібн. / Ваврук С., Лашко О., Попович Р. – Львів : СПОЛЮМ, 2021. – 240 с.

5. Digital Signal Processing: A Practical Approach», Emmanuel C. Ifeachor, Barry W. Jervis, 2nd Edition, 2001. ISBN 0-201-59619-9

6. Digital Image Processing Using MATLAB, 3rd Ed. Gonzalez, Woods, and Eddins, 2020. ISBN: 9780982085417

7. Тотосько О.В., Стухляк П.Д. Цифрова обробка сигналів та зображень / Навчальний посібник. – Тернопіль: ТНТУ, 2016 – 132с.

8. Digital Signal, Image and Video Processing for Emerging Multimedia Technology / Byung-Gyu Kim. – 2021. ISBN 978-3-03943-857-0

9. MATLAB. The Language of Technical Computing [Ел. ресурс]. – Режим доступу : <https://uk.mathworks.com/help/matlab/index.html>

10. Simulink. Simulation and Model-Based Design [Ел. ресурс]. – Режим доступу : <https://uk.mathworks.com/help/simulink/index.html>

Навчальне видання

ФІЛАТОВА Ганна Євгенівна

МАТЕМАТИЧНІ ОСНОВИ ОБРОБКИ СИГНАЛІВ ТА ЗОБРАЖЕНЬ:
ТЕОРІЯ ТА ПРАКТИКА

Навчальний посібник
за курсом «Обробка сигналів і зображень»

Відповідальний за випуск проф. С.Г. Семенов

Роботу до видання рекомендував проф. В.Д. Дмитрієнко
