

## ЛАБОРАТОРНА РОБОТА №1

## РОЗРАХУНОК НЕРЕКУРСИВНИХ ФІЛЬТРІВ НИЗЬКОЇ ЧАСТОТИ ТА НИЗЬКОЧАСТОТНА ФІЛЬТРАЦІЯ МОВНИХ СИГНАЛІВ

**Мета роботи:** Здобуття навичок розрахунку та застосування нерекурсивних фільтрів в середовищі Matlab.

**Тривалість виконання роботи:** 4 академічних години (2 заняття по 2 години).

**Робоче завдання:**

1. Проведіть розрахунок коефіцієнтів нерекурсивного фільтра 6-го порядку вручну та в середовищі Matlab за допомогою команди **fir1** (Додаток 1). Частоту дискретизації та частоту зрізу прийняти 100 Гц та 25 Гц, відповідно. Удоскональте програму `fir_low_koef.m`, додавши «легенду», яка дозволяла б розрізняти результати ручних розрахунків та розрахунків в середовищі Matlab.
2. Проведіть обчислення та побудову графіків АЧХ та ФЧХ в середовищі Matlab (Додаток 2). Спробуйте всі варіанти програми обчислення, що наведені в Додатку 2, оберіть найкращий та вкажіть причини вашого вибору. Додайте до графіків «легенду» (використайте команду **legend()**), яка дозволяла б розрізняти результати розрахунків за «саморобною» програмою та за командою **freqz()**. Обчисліть «вручну» фазову характеристику вашого фільтра, коефіцієнти якого обчислено за формулою (3). Також за допомогою команди **phasez()** обчисліть фазову характеристику фільтра, коефіцієнти якого обчислено в Matlab за допомогою команди **fir1**. Побудуйте та поясніть графік фазової характеристики фільтра.
3. Збільште порядок фільтра в два рази і зробіть висновки про зміни форми АЧХ та ФЧХ фільтра.
4. Використовуючи віконну функцію (див. табл. 1) для фільтра 12 порядку, проаналізуйте, як змінюються АЧХ та ФЧХ.
5. Синтезуйте ФНЧ (тобто розрахуйте коефіцієнти фільтра) із верхньою частотою зрізу 1 кГц та профільтруйте цим фільтром мовний сигнал тривалістю 2-3 с (приклад мовного сигналу: «Моє прізвище Іванов»). Частоту дискретизації мовного сигналу прийняти 22050 Гц. В якості зразка програми фільтрації можете використати програму, наведену в Додатку 3. Порівняйте на слух не фільтрований та фільтрований сигнали, та оцініть суб'єктивно якість фільтрованого сигналу за 5-бальною шкалою (від 1 до 5, дробові оцінки дозволяються).
6. Повторіть дії за п.5 для частот зрізу від 2 до 8 кГц, із кроком 1 кГц. Побудуйте графік залежності якості фільтрованого сигналу від смуги частот ФНЧ. Пов'яжіть отриману залежність із тим фактом, що в телефонії широко розповсюджено зв'язок, де мовний сигнал обмежено смугою частот 3,5 кГц (2G), рідше – смугою 7 кГц (3G).
7. Оформіть звіт за результатами лабораторної роботи (**припустимо оформляти один звіт на бригаду**).

Таблиця 1. Варіанти віконних функцій

№ вар.	Вікно	Формула
1	Прямокутне (Діріхле)	$w(n) = 1 \quad \text{для} \quad n = -N, \dots, -1, 0, 1, \dots, N; \quad 0 \quad \text{для} \quad \text{інш. } n$
2	Бартлета (Фейера)	$w(n) = 1 - \frac{ n }{N} \quad \text{для} \quad n = -N, \dots, -1, 0, 1, \dots, N; \quad 0 \quad \text{для} \quad \text{інш. } n$
3	Хенінга (Ханна)	$w(n) = \cos^2\left(\frac{n}{2N}\pi\right) = 0,5 \left[ 1 + \cos\left(\frac{n}{N}\pi\right) \right] \quad \text{для} \quad n = -N, \dots, -1, 0, 1, \dots, N; \quad 0 \quad \text{для} \quad \text{інш. } n$
4	Хемінга	$w(n) = 0,54 + 0,46 \cos\left(\frac{n}{N}\pi\right) \quad \text{для} \quad n = -N, \dots, -1, 0, 1, \dots, N; \quad 0 \quad \text{для} \quad \text{інш. } n$
5	Блекмена	$w(n) = 0,42 + 0,5 \cos\left(\frac{n}{N}\pi\right) + 0,08 \cos\left(\frac{2n}{N}\pi\right) \quad \text{для} \quad n = -N, \dots, -1, 0, 1, \dots, N; \quad 0 \quad \text{для} \quad \text{інш. } n$
6	Ланцоша	$w(n) = \frac{\sin\left(\frac{n}{N}\pi\right)}{\frac{n}{N}\pi} \quad \text{для} \quad n = -N, \dots, -1, 0, 1, \dots, N; \quad 0 \quad \text{для} \quad \text{інш. } n$

7	Бартлета (Фейера)	$w(n) = 1 - \frac{ n }{N}$ для $n = -N, \dots, -1, 0, 1, \dots, N$ ; 0 для інших $n$
8	Хенінга (Ханна)	$w(n) = \cos^2\left(\frac{n}{2N}\pi\right) = 0,5 \left[1 + \cos\left(\frac{n}{N}\pi\right)\right]$ для $n = -N, \dots, -1, 0, 1, \dots, N$ ; 0 для інших $n$
9	Хемінга	$w(n) = 0,54 + 0,46 \cos\left(\frac{n}{N}\pi\right)$ для $n = -N, \dots, -1, 0, 1, \dots, N$ ; 0 для інших $n$

## Додаток 1

### Обчислення коефіцієнтів НЧ КИХ-фільтра

Коефіцієнти  $a_k$  КИХ-фільтра, що обчислюються методом оберненого перетворення Фур'є, визначаються співвідношенням:

$$a_k = \frac{T_d}{2\pi} \int_{-\pi/T_d}^{\pi/T_d} H_d(j\omega) \exp(j\omega k T_d) d\omega, \quad (1)$$

де  $T_d$  - період дискретизації ( $T_d = \frac{1}{f_d}$ ,  $f_d$  - частота дискретизації).

Підставляючи в (1)

$$H_d(\omega) = \begin{cases} 1 & \text{для } |\omega| \leq \omega_c \\ 0 & \text{для } |\omega| > \omega_c \end{cases} \quad (2)$$

де  $\omega_c$  - частота зрізу, й приймаючи  $a_k = a_{-k}$ , одержимо

$$a_k = a_{-k} = \frac{\Omega_c}{\pi} \frac{\sin k\Omega_c}{k\Omega_c} = \frac{\Omega_c}{\pi} Sa(k\Omega_c), \quad (3)$$

де  $Sa(x) = \sin x/x$ ;  $\Omega_c = \frac{\omega_c}{\omega_d} = 2\pi \frac{f_c}{f_d}$ .

Лістинг програми **fir\_low\_koef.m** для обчислень, у відповідності із (3), коефіцієнтів  $a_k$ , наведено нижче.

```
% === fir_low_koef.m ===
% === коєффициенты НЧ КИХ-фильтра ===
%
N = 3;           % половина порядка фильтра
fs = 100;        % частота дискретизации, Гц
fc = 25;         % частота среза, Гц
%
% === вычисление коєфф-в фильтра ===
omc = 2*pi*fc/fs; % относит. частота среза
ak0 = omc/pi;
k = 1:N;
sinxx = ak0*sin(k*omc)./(k*omc);
ak = [fliplr(sinxx) ak0 sinxx];
%
% === стандартная команда вычисления коєфф-в фильтра ===
ak1 = fir1(6, 2*fc/fs, boxcar(2*N+1), 'noscale');
figure;
plot(ak, 'r');
hold on
stem(ak1);
```

Коментуючи програму **fir\_low\_koef.m**, зазначимо:

- 1) змінна  $k$ , що фігурує у формулі (3), задається як вектор-строка, який приймає значення 1, 2, 3;
- 2) частина коефіцієнтів  $a_k$  обчислюється для такого вектора  $k$  за формулою (3);
- 3) повний набір коефіцієнтів  $a_k$  komponується із трьох частин: в центрі знаходиться число  $a_0$ , до якого справа приєднано коефіцієнти, обчислені за п.2), а зліва – ці ж коефіцієнти, відображені дзеркально, тобто «зліва-направо».

Для контролю за правильністю обчислень, в кінці програми виконується обчислення коефіцієнтів  $a_k$  з допомогою бібліотечної програми **fir1.m**. Співпадіння графіків свідчить про правильність «нашої» програми обчислень, а також показує, які вимоги потрібно виконати для такого співпадіння (прямокутне окно та опція «noscale» при використанні програми **fir1.m**).

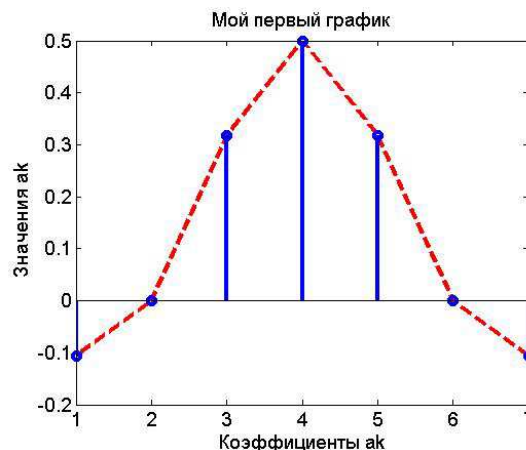


Рис. 1. Коефіцієнти  $a_k$ , обчислені за формулою (3) (червона пунктирна лінія) та за допомогою програми **fir1.m** (сині лінії)

## Додаток 2

### Обчислення частотних характеристик НЧ КІХ-фільтра

Наведемо чотири варіанти програми обчислення ЧХ НЧ КІХ-фільтра. У всіх варіантах обчислення виконуються відповідно до формули:

$$H_d(\omega) = a_0 + 2 \sum_{k=1}^N a_k \cos \omega k T_d. \quad (4)$$

**Варіант 1.** Відрізняється тим, що ЧХ обчислюється за допомогою подвійного циклу – зовнішній цикл забезпечує перебір значень частоти  $f$ , а внутрішній цикл забезпечує підсумовування за індексом  $k$ .

```
% === filter_1.m ===
% === программа построения частотной характеристики НЧ фильтра ===
%
% === задание параметров модели ===
a0=0.5; % нулевой коэффициент фильтра
ak=[0.3183 0 -0.1061]; % коэффициенты фильтра с 1 по 3
fd=100; % частота дискретизации
dt=1/fd; % период (шаг) дискретизации
N=3; % половина порядка фильтра
df=0.02*fd; % шаг по частоте
% === выделение областей ===
sum = zeros(1,100); % выделение области для функции
f = zeros(1,100); % выделение области для аргумента
% === вычисление значений функции ===
tic % начало контроля времени вычислений
for n=1:100 % перебор номеров значений аргумента
    f(n) = n*df-fd/2; % вычисление значений аргумента
    for k=1:N % перебор номеров частотных гармоник
```

```

sum(n)=sum(n)+ak(k)*cos(2*pi*f(n)*k*dt); % значення функції
end
end
toc % кінець контролю часу обчислень
H=a0+2*sum; % частотна характеристика фільтра
figure; % основа для графіка
plot(f,H) % побудова графіка
grid on % сітка на графіку
xlabel('Частота, Гц'); ylabel('Уровень');
title('Частотная хар-ка НЧ фильтра 6-го порядка')

```

**Варіант 2.** Відрізняється тим, що в згаданому подвійному циклі змінено порядок перебору значень змінних  $f$  і  $k$ .

```

% === filter_1b.m ===
% === програма побудови частотної характеристики НЧ фільтра ===
%
% === задання параметрів моделі ===
a0=0.5; % нульовий коефіцієнт фільтра
ak=[0.3183 0 -0.1061]; % коефіцієнти фільтра з 1 по 3
fd=100; % частота дискретизації
dt=1/fd; % період (крок) дискретизації
N=3; % половина порядку фільтра
df=0.02*fd; % крок по частоті
% === виділення областей ===
sum = zeros(1,100); % виділення області для функції
f = zeros(1,100); % виділення області для аргумента
% === обчислення значень функції ===
tic % початок контролю часу обчислень
for k=1:N % перебір номерів частотних гармонік
    for n=1:100 % перебір номерів частот
        f(n) = n*df-fd/2; % обчислення значень аргумента
        sum(n)=sum(n)+ak(k)*cos(2*pi*f(n)*k*dt); % значення функції
    end
end
toc % кінець контролю часу обчислень
H=a0+2*sum; % частотна характеристика фільтра
figure; % основа для графіка
plot(f,H,'r') % побудова графіка
grid on % сітка на графіку
xlabel('Частота, Гц', 'FontSize', 14); ylabel('Уровень', 'FontSize', 14);
title('Частотная хар-ка НЧ фильтра 6-го порядка', 'FontSize', 14);
hold on
% === стандартна програма побудови АЧХ ===
ak1 = fir1(2*N,0.5,boxcar(2*N+1),'noscale');
[H1,f1] = freqz(ak1,1,25,100);
stem(f1,abs(H1));

```

В цьому варіанті зіставляються результати обчислень ЧХ: «наші» (мал. 2, червона лінія) і отримані за допомогою функції **freqz.m** (мал. 2, сині лінії). Результати збігаються, однак є й відмінності: 1) функція **freqz.m** не обчислює «зайвих» значень ЧХ; 2) при побудові графіка ЧХ, отриманої за допомогою функції **freqz.m**, доводиться брати модуль – у протилежному випадку на графік виведеться тільки дійсна частина комплексної ЧХ. Ці відмінності обумовлені тим, що «наша» ЧХ будувалася в припущенні парності функції  $a_k$ , тоді як функція **freqz.m** здійснює обчислення для зрушених за часом коефіцієнтів  $a_k$ , нумерація яких починається з нуля.

Якщо будувати модуль «нашої» ЧХ, тоді збіг графіків (по вертикалі) буде повним.

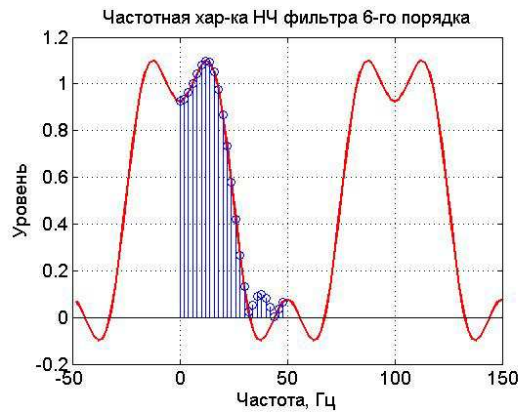


Рис. 2. ЧХ, обчислені по формулі (4) (червона пунктирна лінія) і за допомогою програми freqz.m (сині лінії)

**Варіант 3.** Відрізняється тим, що вектор значень частоти  $f$  задається попередньо – у результаті вдається позбутися від циклу для перебору значень  $f$ .

```
% === filter_2.m === Частотная хар-ка ФНЧ ===
a0=0.5; % нулевой коэффициент фильтра
ak=[0.3183 0 -0.1061]; % коэффициенты фильтра с 1 по 3
fd=100; % частота дискретизации
dt=1/fd; % период (шаг) дискретизации
N=3; % половина порядка фильтра
df=0.02*fd; % шаг по частоте
f=-0.5*fd:df:1.5*fd; % диапазон частот
% === расчет суммы ===
sum=zeros(1,length(f));
tic % начало контроля времени вычислений
for k=1:N,
    sum=sum+ak(k)*cos(2*pi*f*k*dt);
end;
toc % конец контроля времени вычислений
H=a0+2*sum; % частотная хар-ка
figure;
plot(f,H,'-r') % построение графика
grid on % построение сетки
hold on
% === стандартная программа построения АЧХ ===
ak1 = fir1(2*N,0.5,boxcar(2*N+1),'noscale');
[H1,f1] = freqz(ak1,1,25,100);
stem(f1,abs(H1));
```

**Варіант 4.** Відрізняється тим, що вектори значень частоти  $f$  й значень  $k$  задаються попередньо – у результаті вдається взагалі позбутися від циклів. Тим самим демонструється важливе достоїнство векторної алгебри, яка використовується в Matlab та дозволяє помітно спрощувати обчислювальні програми.

```
% === freq_har.m === Частотная хар-ка ФНЧ ===
a0=0.5; % нулевой коэффициент фильтра
ak=[0.3183 0 -0.1061]; % коэффициенты фильтра с 1 по 3
fd=100; % частота дискретизации
dt=1/fd; % период (шаг) дискретизации
N=3; % половина порядка фильтра
df=0.02*fd; % шаг по частоте
f=-0.5*fd:df:1.5*fd; % диапазон частот
k = 1:N;
% === расчет суммы ===
tic % начало контроля времени вычислений
cosm = cos(2*pi*k'*f*dt); % набор гармоник по частоте
sum = ak*cosm;
toc % конец контроля времени вычислений
```

```
H=a0+2*sum; % частотная хар-ка
figure;
plot(f,H,'-r') % построение графика
grid on % построение сетки
hold on
% === стандартная программа построения АЧХ ===
ak1 = fir1(2*N,0.5,boxcar(2*N+1),'noscale');
[H1,f1] = freqz(ak1,1,25,100);
stem(f1,abs(H1));
```

### Додаток 3

#### 3. Фільтрація штучного сигналу

```
% === filtering_fir.m ===
% === программа фильтрации процесса x(n) ===
%
% === модель процесса x(n) ===
x = [1 2 3 -3 -2 -1 0 1 2 3 -3 -2 -1 0 1 2 3 -3 -2 -1];
% === ИХ фильтра ===
ak=[-0.1061 0 0.3183 0.5 0.3183 0 -0.1061];
%
% === дописывание нулей спереди и сзади массива x(n) ===
lak = length(ak);
nz = lak-1;
zrs = zeros(1,nz);
xz = [zrs x zrs];
%
% === деление x(n) на фреймы ===
nfrm = length(xz)-lak+1;
xfrm = zeros(lak,nfrm); % выделение области
for k = 1:nfrm
    xfrm(:,k) = xz(k:k+lak-1);
end
%
% === фильтрация ===
y = ak*xfrm;
%
% === стандартная команда фильтрации ===
y1 = filter(ak,1,x);

% === построение графиков ===
tx = 0:(length(x)-1);
ty = 0:(length(y)-1);
ty1 = 0:(length(y1)-1);
figure; plot(tx,x,'-o');
xlabel('Номера выборок'); ylabel('Уровень'); title('Исходный сигнал');
figure; plot(ty,y);
xlabel('Номера выборок'); ylabel('Уровень'); title('Результат фильтрации');
hold on;
stem(ty1,y1,'r');
legend('наша фильтрация','фильтрация Matlab');

% === другой способ построения графиков ===
figure % new figure
ax1 = subplot(2,1,1); % top subplot
ax2 = subplot(2,1,2); % bottom subplot

plot(ax1,tx,x,'-o')
xlabel(ax1,'Номера выборок'); ylabel(ax1,'Уровень');
title(ax1,'Исходный сигнал');

plot(ax2,ty,y,'-x'); hold on; stem(ax2, ty1, y1, 'r');
```

```
xlabel(ax2,'Номера выборок'); ylabel(ax2,'Уровень');  
title(ax2,'Результат фильтрации');  
legend(ax2, 'наша фильтрация', 'фильтрация Matlab');
```

Результати виконання даної програми представлені нижче у вигляді графіків.

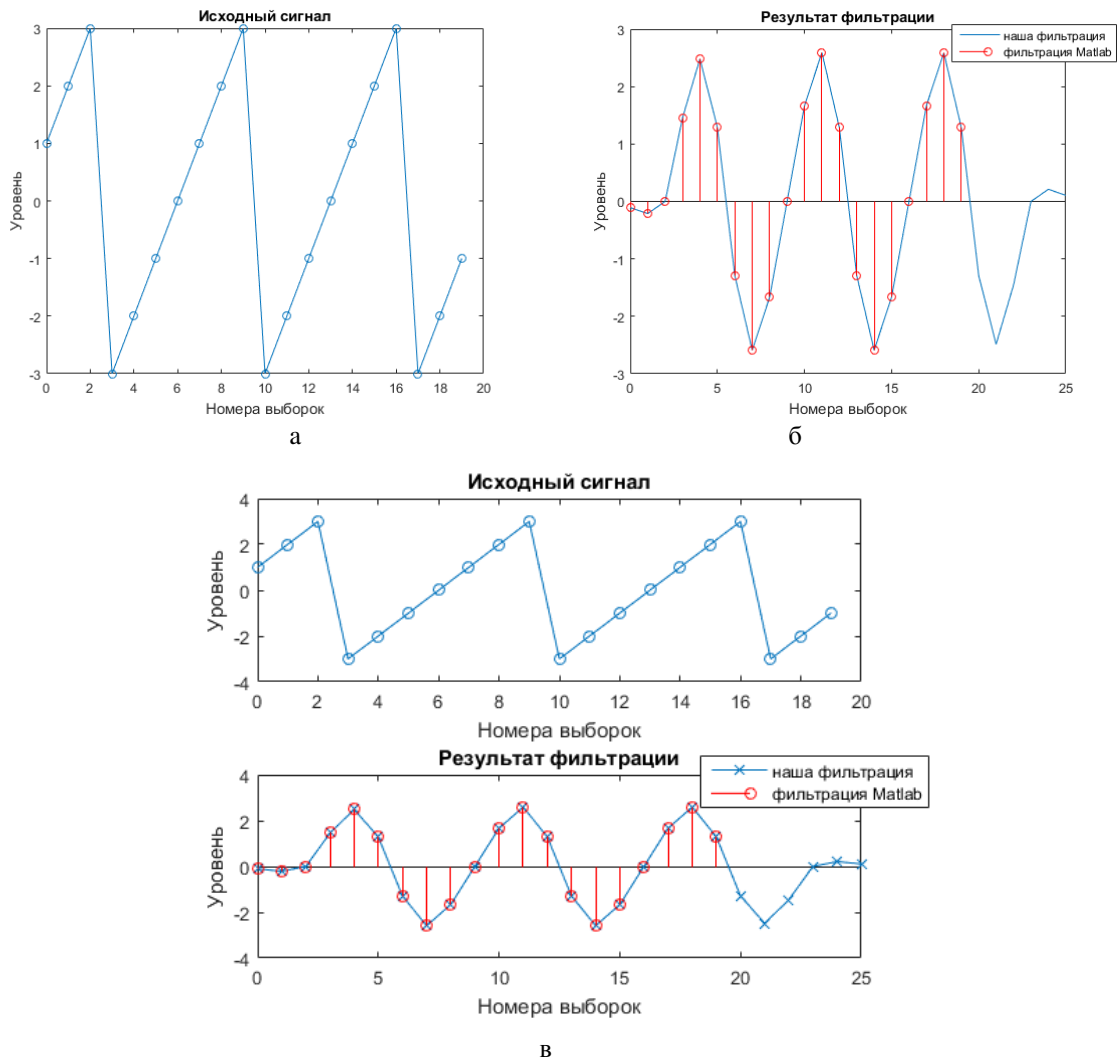


Рис.3. Графіки вихідного й фільтрованого сигналів, представлені роздільно (а, б) і спільно (в)

Зверніть увагу на специфіку програмної частини, де ми обчислюємо дискретну згортку - тут істотно використано особливість Matlab, що полягає в оперуванні із матрицями та векторами. Масив вхідного сигналу (одномірний) був перетворений у двовимірний масив, після чого здійснено векторно-матричний добуток коефіцієнтів фільтра та матриці вхідного сигналу - і в результаті було отримано вектор значень вихідного сигналу.

Зрозуміло, це далеко не єдиний спосіб написання програми фільтрації - ви можете самостійно потренуватися в написанні альтернативних варіантів.

Однак важливо звернути увагу ось на що: у той час як стандартна програма **filter.m** дає вихідний сигнал тієї ж довжини, що й вхідний сигнал (19 вибірок), наша програма дала на 6 вибірок більше. Збіг форми вихідних сигналів свідчить, що наша програма вірна. З іншого боку, виникає питання - звідки взялися додаткові вибірки в нашому результаті? Або інакше запитаємо – як трапилося, що в Matlab'і відсутні ці 6 вибірок, якщо вони насправді мають бути?

От над цими питаннями і пропонуємо вам подумати самостійно...