Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до лабораторної роботи №3

з дисципліни

**“Основи цифрової обробки сигналів”**

на тему: **“ ЦОС у часовій області, розробка і використання фільтрів із скінченною імпульсною характеристикою в Matlab”**

Варіант 7

Виконав: студент групи ОІ-36

**Лабунський Я.А.**

Прийняв: старший викладач кафедри АСУ

**Баран Р.Д.**

Львів – 2025

Лабораторна робота №3

Тема: ЦОС у часовій області, розробка і використання фільтрів із скінченною імпульсною характеристикою в Matlab.

Мета: ознайомитись обробкою сигналів у часовій області з використанням пакету MATLAB; навчитись створювати КІХ-фільтри; відпрацювати принципи фільтрації сигналів у пакеті MATLAB на рівні, достатньому для практичного використання; провести моделювання і проаналізувати результати виконання у пакеті MATLAB.

Короткі теоретичні відомості:

**MATLAB –** пакет прикладних програм символьної математики, статистики, оптимізації, аналізу і синтезу та обробки сигналів, зображень тощо. Для виконання лабораторних можна використовувати безкоштовну он-лайн версію пакету від розробника MathWorks, яка знаходиться за посиланням <https://matlab.mathworks.com/>. Перед використанням достатньо зареєструватися, використовуючи адресу власної електронної пошти.

Терміном цифровий фільтр називають апаратну або програмну реалізацію математичного алгоритму, входом якого є цифровий сигнал, а виходом – інший цифровий сигнал, форма якого і/або амплітудна та фазова характеристики спеціальним чином модифіковані.

В аналогових системах під фільтром розуміють деякий лінійний пристрій зі спеціальною частотною характеристикою , який перетворює вхідний сигнал  у вихідний (рис. 1), придушуючи або, навпаки, підсилюючи при цьому певні частоти в спектрі вхідного сигналу. Вихідний сигнал знаходиться як згортка вхідного сигналу і імпульсної характеристики фільтра :



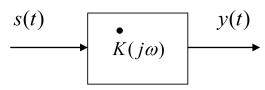


Рисунок 1 – Цифровий фільтр

За аналогією з аналоговим фільтром, цифровий фільтр (ЦФ, digital filter) перетворює послідовність відліків вхідного сигналу у числову послідовність вихідного сигналу . Для ЦФ також вводять поняття імпульсної характеристики , що є реакцією ЦФ на «одиничний імпульс (скачок)» тобто

.

Імпульсну характеристику (pulse response characteristic)  ЦФ можна трактувати як результат дискретизації безперервної імпульсної характеристики  відповідного аналогового фільтра- прототипу (рис. 2).

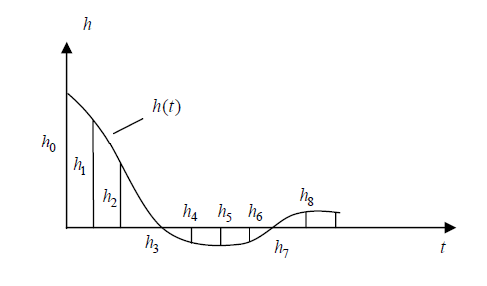


Рисунок 2 – Дискретизація імпульсної характеристики

Якщо взяти кінцеве число відліків , тоді отримаємо ЦФ із кінцево-імпульсною характеристикою (КІХ-фільтр, finite impulse response filtering). Якщо взяти нескінченне число відліків , отримаємо ЦФ із безкінечно-імпульсною характеристикою (БІХ-фільтр, infinite impulse response filtering).

У літературі вживається як термін «фільтр з кінцево-імпульсною характеристикою», так і термін «фільтр з скінченною імпульсною характеристикою» у випадку КІХ фільтрів.

Під фільтром зазвичай розуміють систему, яка одні частоти вхідного сигналу пропускає на вихід, а інші затримує. Однак у техніці цифрової обробки сигналів поняття фільтра трактується більш широко. Дискретним фільтром називають довільну систему обробки дискретного сигналу, що має властивості лінійності й стаціонарності. Існують також фільтри зі змінними параметрами, наприклад, адаптивні фільтри, що змінюють свої параметри залежно від статистичних властивостей вхідного сигналу.

У загальному випадку, фільтр змінює в спектрі сигналу і амплітуди гармонік, і їх фази. Однак фільтри можна проектувати так, щоб вони не змінювали фазу сигналу. Такі фільтри називаються фільтрами з лінійною фазою. Це означає, що якщо вони і змінюють фазу сигналу, то роблять це так, що всі гармоніки сигналу зсуваються за часом на одну й ту ж величину. Таким чином, фільтри з лінійною фазою не спотворюють фазу сигналу, а лише зсувають весь сигнал в часі. Ядро згортки такого фільтра симетричне щодо своєї центральної точки.

Основна властивість будь-якого фільтру – це його частотна і фазова характеристики. Вони показують, як фільтр впливає на амплітуду і фазу різних гармонік оброблюваного сигналу. Якщо фільтр має лінійну фазу, то розглядається лише частотна характеристика фільтру. Зазвичай частотна характеристика зображується у вигляді графіка залежності амплітуди від частоти (в децибелах). Наприклад, якщо фільтр пропускає всі сигнали в смузі 0 ... 10 кГц без зміни, а всі сигнали в смузі вище 10 кГц подавляє в 2 рази (на 6 дБ), то частотна характеристика буде мати такий вигляд:

.

Частотна характеристика в 0 дБ показує, що вказані частоти фільтр пропускає без зміни. Частоти, амплітуда яких послаблюється фільтром в 2 рази, повинні мати амплітуду на 6 дБ менше. Тому їх амплітуда становить -6 дБ. Якщо фільтр посилює частоти, то його частотна характеристика на цих частотах є позитивною.

Вихідний сигнал y(k) фільтра, що має складну частотну характеристику, залежить від декількох відліків вхідного сигналу x(k). У загальному випадку при обчисленні вихідного відліку використовується також деяка кількість попередніх відліків вихідного сигналу. Для фільтрів, що не використовують вихідні відліки, рівняння фільтрації має вигляд:

.

Такі фільтри називаються нерекурсивними (nonrecursive filter). Кількість відліків m називається порядком фільтра. Структурна схема нерекурсивного фільтра показана на рис.3.

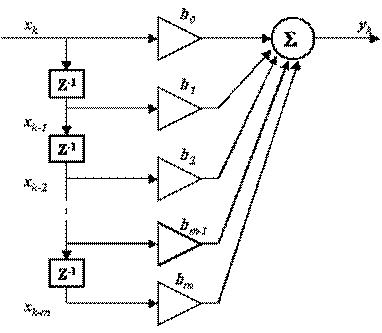


Рисунок 3 – Нерекурсивний фільтр

Імпульсна характеристика нерекурсивного фільтра визначається його коефіцієнтами h(k) = bk. Оскільки в реальному пристрої кількість ліній затримки обмежена, а отже, і кількість коефіцієнтів також, нерекурсивні фільтри відносять до класу КІХ-фільтрів.

Як правило КІХ-фільтри у ЦОС використовуються у якості фільтрів нижніх частот, верхніх частот або смугових. Має важливе значення чи фільтр у цьому діапазоні вказані частоти «пропускає» (Pass) на вихід, або «затримує» (Stop). У вітчизняній літературі нажаль немає чітких відповідників цих термінів!

У випадку фільтра нижніх частот (ФНЧ, Lowpass) він має пропускати на вихід низькі частоти і обмежувати пропускання на вихід високих. У випадку фільтра верхніх частот (ФВЧ, Highpass) він має пропускати на вихід високі частоти і обмежувати низькі. У випадку «пропускаючого» смугового фільтру (ФС, Bandpass) він повинен пропускати на вихід частоти у певному визначеному діапазоні і обмежувати пропускання на вихід визначених низьких та високих частот. У випадку «затримуючого» смугового фільтру (ФС, Bandstop) він має затримувати передавання на вихід частот у певному визначеному діапазоні і дозволяти пропускання на вихід визначених низьких та високих частот.

Обробка сигналів у часовій області надзвичайно широко використовується у системах ЦОС.

Розглянемо синтез КІХ фільтра за допомогою пакету MatLab.

У найпростішому випадку КІХ фільтр у MatLab можна створити, використавши функцію fir1:

%створити фільтр

h = fir1(48,[0.45 0.55]);

%показати частотний відгук створеного фільтра

figure; freqz(h,1,512);

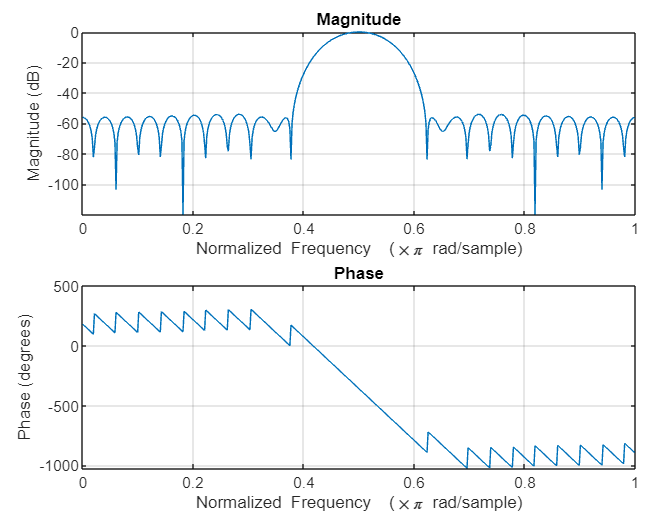


Рисунок 4 – Амплітудна та фазова характеристики фільтра

Така послідовність команд створить смуговий фільтр FIR 48-го порядку зі смугою пропускання 0,35π≤ω≤0,65π рад/вимір та візуалізує його амплітуду та фазові характеристики (рис.4). Другий параметр вказує нормовані стосовно половини частоти дискретизації мінімальну і максимальну частоти пропускання створюваного фільтра.

Вказана функція може використовувати низку параметрів, детальніше –

<https://www.mathworks.com/help/signal/ref/fir1.html>

Застосувати створений фільтр до обробки сигналу можна командою:

signal\_filtered = filter(h,1,signal);

Однак, розробка фільтра, особливо смугового вимагає детального підбору цілого переліку параметрів. Для спрощення цієї процедури пакет MatLab дозволяє створювати фільтри з заданими характеристиками за допомогою спеціалізованої утиліти **filterDesigner**. Вказану утиліту можна завантажити набравши у Командному вікні її назву – filterDesigner. Детальна інформація щодо використання наведена у описі –

<https://www.mathworks.com/help/signal/ug/introduction-to-filter-designer.html>

Основне вікно утиліти показано на рис. 5.

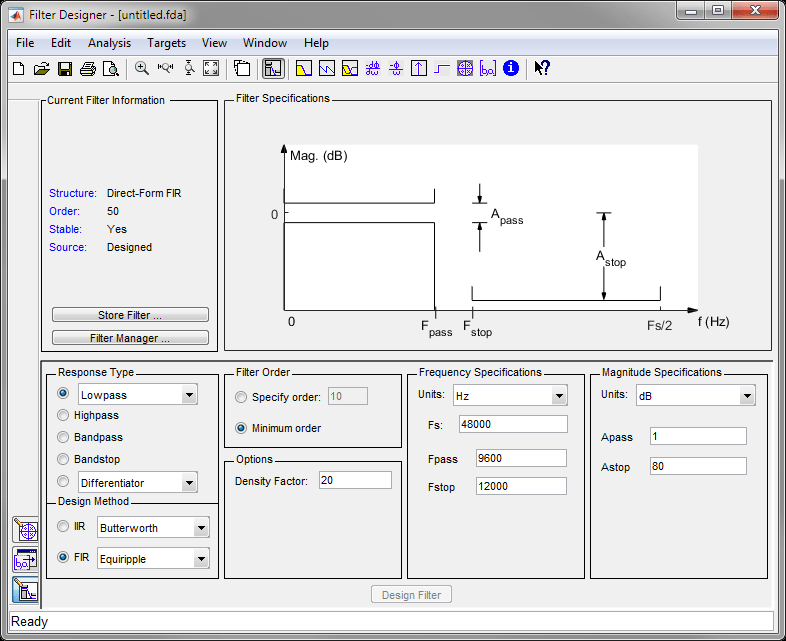


Рисунок 5 – Вікно filterDesigner

Проектування фільтра

Розробимо фільтр низьких частот, який пропускає всі частоти, менші або рівні 20% частоти Найквіста (половина частоти дискретизації), і послаблює частоти, більші або рівні 50% частоти Найквіста. Будемо використовувати фільтр FIR Equiripple із такими характеристиками:

• Затухання в смузі пропускання 1 дБ

• Смуга затухання 80 дБ

• Частота смуги пропускання 0,2 [нормалізована (від 0 до 1)]

• Частота смуги зупинки 0,5 [нормалізована (від 0 до 1)]

Для реалізації цього дизайну будемо використовувати такі специфікації:

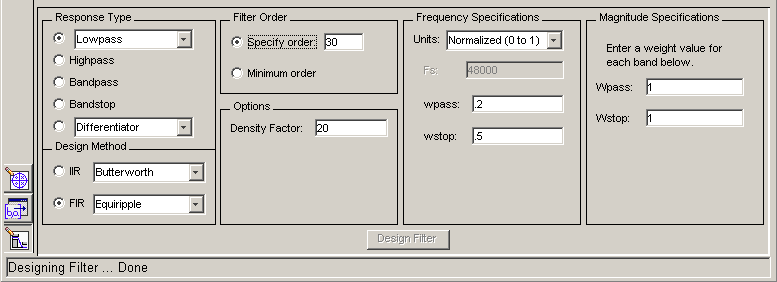


Рисунок 6 – Вікно параметрів filterDesigner

1. Виберемо **Lowpass** зі спадного меню в розділі **Response Type і Equiripple** у FIR Design Method. Загалом, коли змінюється тип відповіді або метод проектування, параметри фільтра та область відображення фільтра оновлюються автоматично.

2. Виберемо «**Specify order** » в області «**Filter Order** » та введемо 30.

3. Фільтр FIR Equiripple має параметр **Density Factor**, який контролює щільність сітки частот. Збільшення значення створює фільтр, який ближче наближається до ідеального фільтра рівних пульсацій, але потрібно більше часу, оскільки обчислення збільшується. Залишаємо це значення на 20.

4. Виберемо **Normalized** **(0 to 1)** у випадаючому меню **Units** в області **Frequency Specifications**.

5. Введемо 0,2 для **wpass** і 0,5 для **wstop** в області **Frequency Specifications**.

6. **Wpass** і **Wstop** в області **Specifications Magnitude** є додатними ваговими коефіцієнтами, по одному на діапазон, які використовуються під час оптимізації в фільтрі FIR Equiripple. Залишаємо ці значення 1.

7. Після встановлення специфікацій дизайну натикаємо кнопку «**Design Filter** » внизу графічного інтерфейсу користувача, щоб спроектувати фільтр.

Амплітудна характеристика фільтра відображається в області аналізу фільтра після обчислення коефіцієнтів.

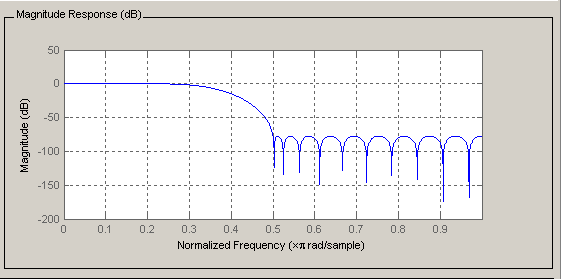


Рисунок 6 – Амплітудна характеристика фільтра

Перегляд інших характеристик **Analyzes.**



Після того як створено фільтр можна переглянути наступні параметри фільтра у вікні, натиснувши будь-яку з кнопок на панелі інструментів:

У порядку зліва направо розташовані кнопки

• Амплітудна характеристика

• Фазова характеристика

• Амплітуда і фаза відгуків

• Групова затримка реакції

• Реакція затримки фази

• Імпульсна характеристика

• Відповідь на крок

• Графік полюс-нуль

• Коефіцієнти фільтра

• Фільтр інформації

Зверніть увагу на опції **Magnitude Response (dB) and Phase Response, Impulse response та Step response.**

При створенні фільтру має значення **Частота дискретизації**, для якої розраховувався вказаний фільтр. Загалом, такий фільтр НЕ МОЖНА застосовувати для іншої частоти дискретизації сигналу!

**Наприклад**, створимо «пропускаючий» смуговий фільтр (Bandpass) з параметрами:

Частота дискретизації сигналу – 8 кГц;

Мінімальна обмежуюча частота Fstop1 – 300 Гц;

Мінімальна пропускаюча частота Fpass1 – 600 Гц;

Максимальна пропускаюча частота Fpass2 – 800 Гц;

Максимальна обмежуюча частота Fstop2 – 1000 Гц.

Оберемо мінімально можливий порядок фільтра.

Для створення фільтра натиснемо кнопку «**Design Filter**».

Отримаємо наступні його характеристики (рис. 7 – 9).

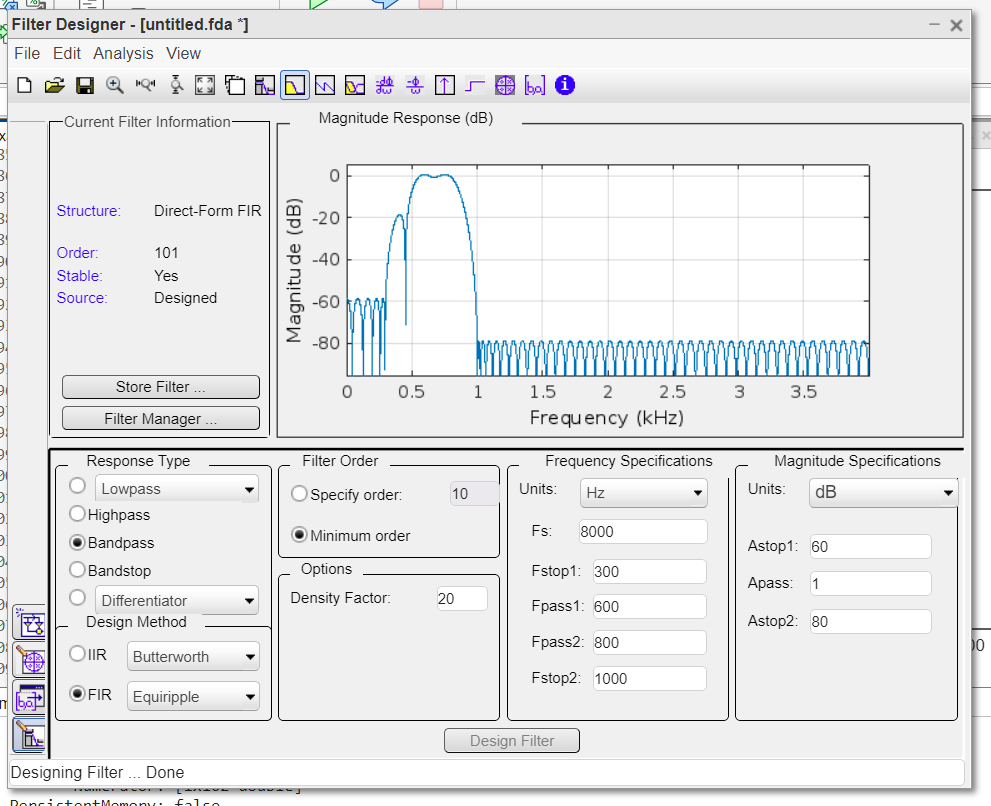


Рисунок 7 – Графік **Magnitude Response**

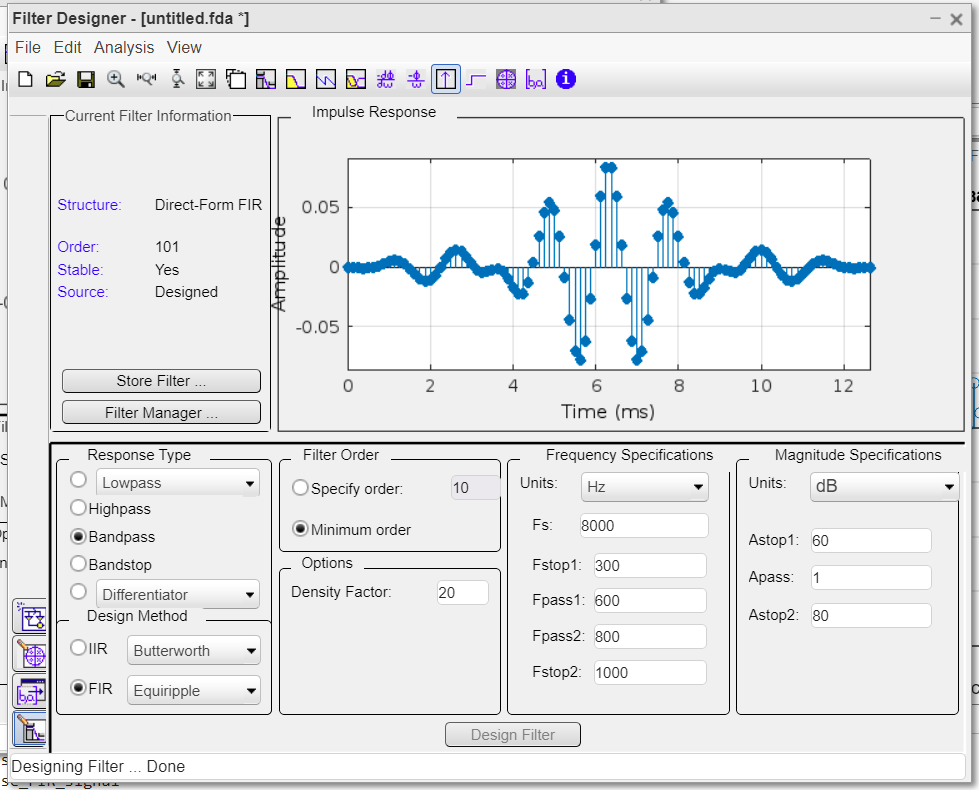


Рисунок 8 – Графік **Impulse response**

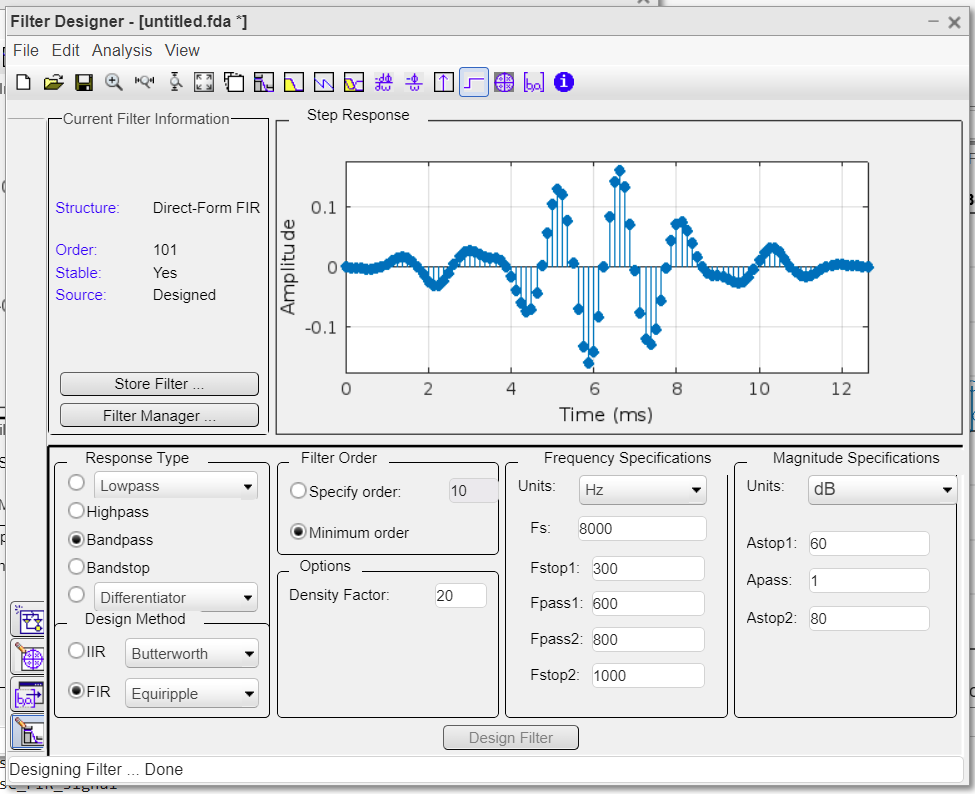


Рисунок 9 – Графік **Step response**

Щоб зберегти опис створеного фільтру у m файлі натиснемо меню «File», далі «Generate MATLAB Code» і «Filter Design Function».

**Увага!** Назву відповідного фільтру рекомендується створити у вигляді «MyFilter\_XXX\_XXX\_XXX\_XXX», де вказати його базові частоти. Далі у MatLab цей фільтр буде вказуватися по назві згенерованого m файлу!

Звертаємо увагу на графіки **Magnitude Response (dB) and Phase Response, Impulse response та Step response.**

Далі згенеруємо складний сигнал на базі функції sin або cos з параметрами амплітуди А1, A2 та А3 і частоти F1, F2 та F3. Початкова фаза рівна нулю. Відобразимо його (рис. 10).

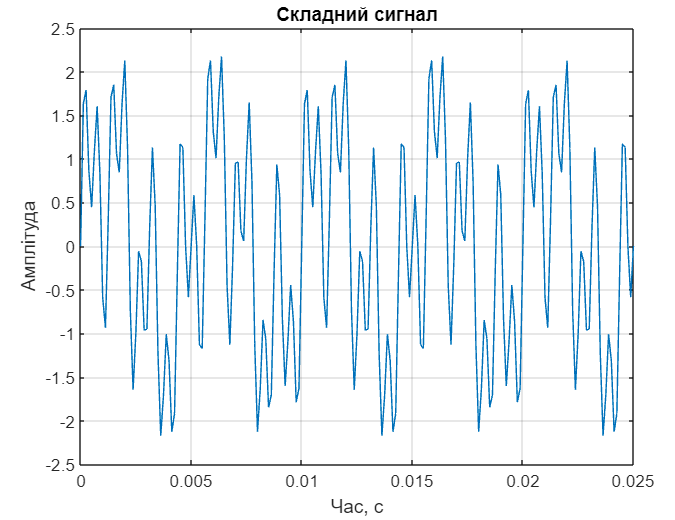


Рисунок 10 – Графік вхідного сигналу

За допомогою команди xlim відображаємо початкові виміри сигналу, щоб детальніше їх візуально проаналізувати.

figure; plot(t,signal); grid on; title('Складний сигнал');

xlim([0 0.025]);

xlabel('Час, с'); ylabel('Амплітуда');

Отримаємо спектр сигналу, щоб проаналізувати співвідношення амплітуд складових (рис. 11).

%сигнал->спектр

spectrum=fft(signal,N);

%нормалізація

spectrum\_n=zeros(N);

for i=1:N

spectrum\_n(i)=2\*spectrum(i)/N;

end

spectrum\_n(1)=spectrum\_n(1)/2;

freq=(0:N/2-1)\*Fs/N;

figure; bar(freq,abs(spectrum\_n(1:N/2))); title('Спектр сигналу');

xlabel('Частота, Гц'); ylabel('Амплітуда гармонік');

Як бачимо, амплітуди складових сигналу однакові.

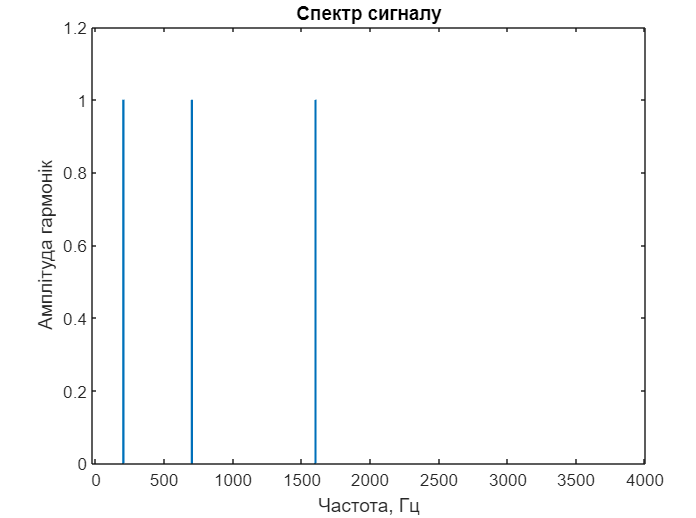


Рисунок 11 – Спектр складного сигналу

Далі розглянемо обробку сигналу у часовій області з використанням пакету MatLab і згенерованого на попередньому етапі смугового фільтра.

%завантажуємо фільтр

Hd = My\_filter\_300\_600\_800\_1000;

%figure; freqz(Hd,4,512);

%застосовуємо фільтр

signal\_filtered = filter(Hd,signal);

%відфільтрований сигнал

figure; plot(t,signal, t,signal\_filtered); grid on; title('Відфільтрований сигнал');

xlim([0 0.025]);

xlabel('Час, с'); ylabel('Амплітуда');

Результат обробки показано на рис. 12.

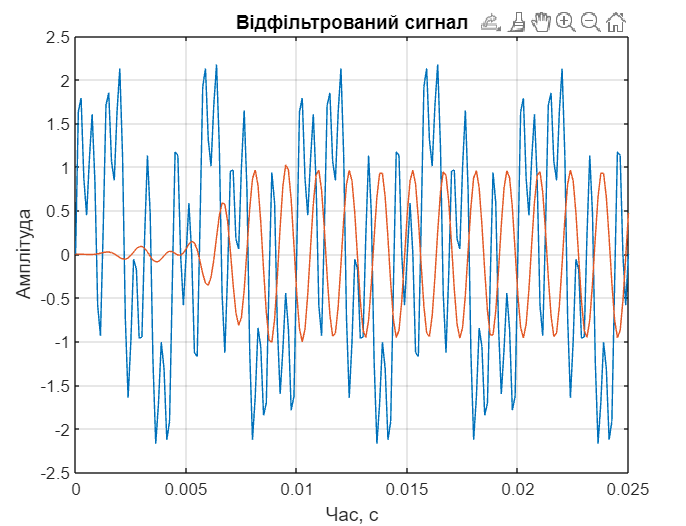


Рисунок 12 – Відфільтрований сигнал

Звертаємо увагу на початкові відліки відфільтрованого сигналу (червоний колір на рис.12). Оскільки порядок фільтра доволі високий, значення перших відліків не є достовірними.

Аналіз наступних відліків показує, що на виході смугового фільтру отримуємо відфільтроване значення певної (центральної) частоти. Спектр відфільтрованого сигналу показує, що на виході фільтру дійсно міститься тільки одна складова (рис. 13).

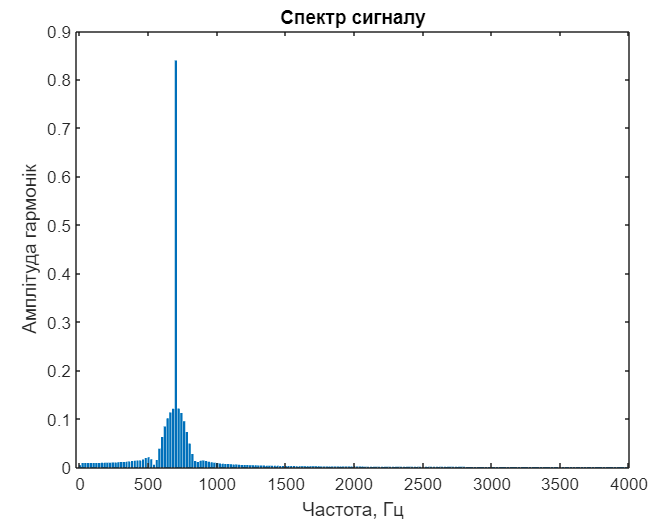


Рисунок 13 – Спектр відфільтрованого сигналу після обробки

Отже, розробка фільтру відповідно до завдання виконана успівшно, що і підтверджують отримані результати.

Далі, спробуємо замінити наш складний сигнал шумоподібним виду

noise=1\*(randn(size(t))-0.5);

signal=noise;

та виконати фільтрацію аналогічним чином (рис.14 – 17).



Рисунок 14 –Шумоподібний сигнал

Спектр сигналу. Звертаємо увагу на складові у спектрі у повній смузі частот, які відповідають шумоподібному сигналу (рис. 15).

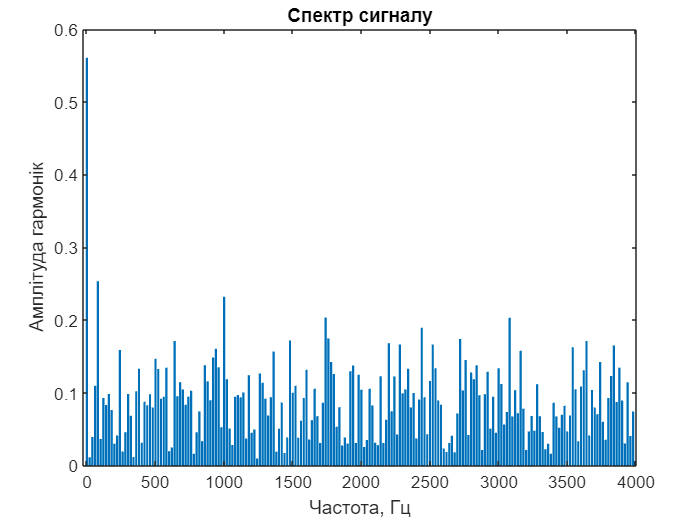


Рисунок 15 – Спектр шумоподібного сигналу

Виконаємо фільтрацію сигналу з використанням створеного попередньо фільтра (рис.16).

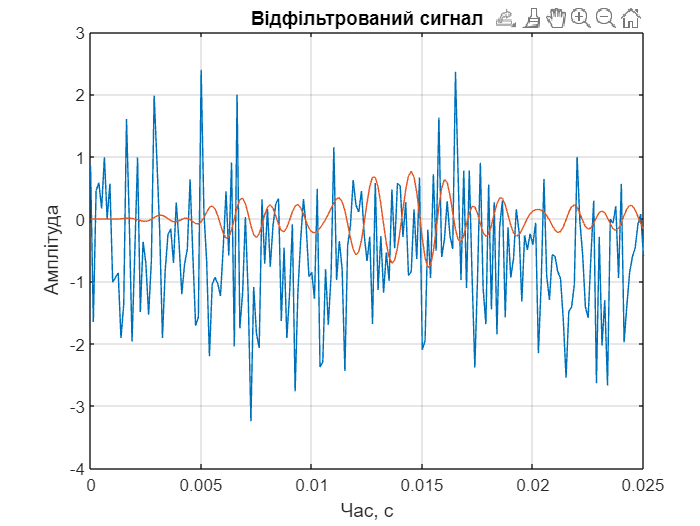


Рисунок 16 – Сигнал після фільтрації

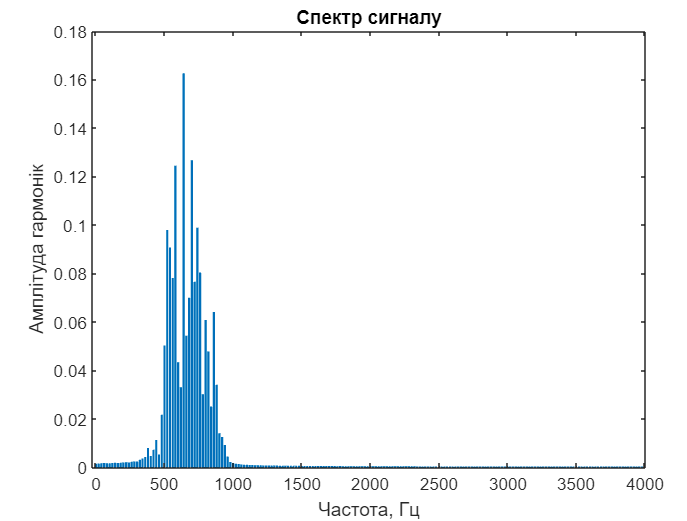


Рисунок 17 – Спектр сигналу після обробки

Звертаємо увагу на складові у спектрі у смузі частот, які залишилися після фільтрації (рис. 17). Вони відповідають проектованій нами **Амплітудній характеристиці фільтру**.

Далі згенеруємо **дельта-імпульс** з амплітудою рівною 1 та виконаємо аналогічні кроки по обробці. Порівняємо графіки **Impulse response** з **filterDesigner** та отриманого сигналу після обробки спроектованим фільтром. Очевидно, вони аналогічні.

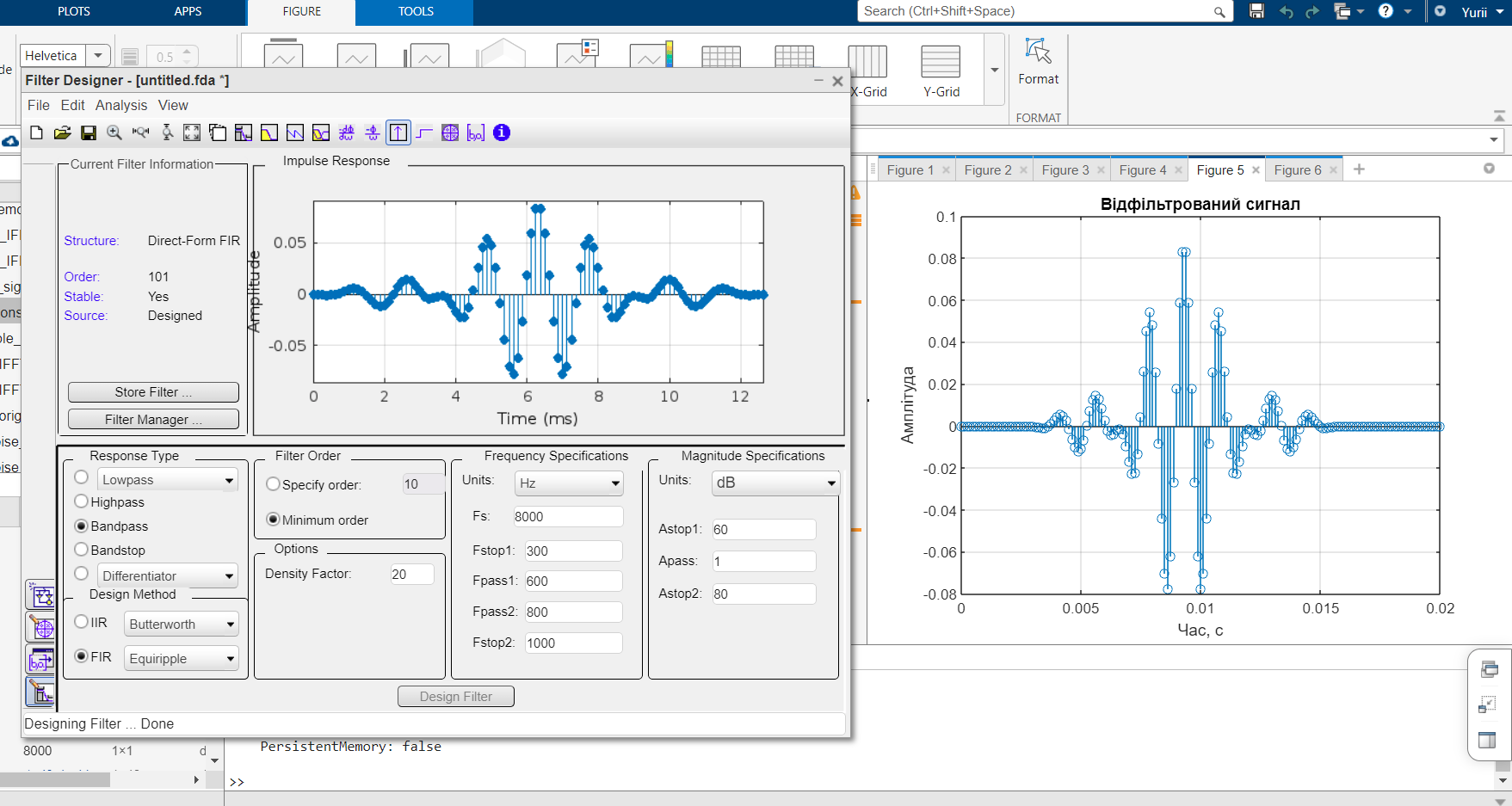


Рисунок 18 – Порівняння графіків **Impulse response** та отриманого сигналу після обробки

Порівняємо амплітудну характеристику (Magnitude Response), отриману при виконанні Завдання 1 з **filterDesigner** і отриманий спектр вихідного сигналу, як реакцію на дельта-імпульс (рис.19). Очікувано, отриманий спектр відповідає смузі частот спроектованого фільтру. Дельта-імпульс має «необмежений» спектр, отже на вихід фільтру пройшли тільки ті спектральні складові, які «пропустив» фільтр.

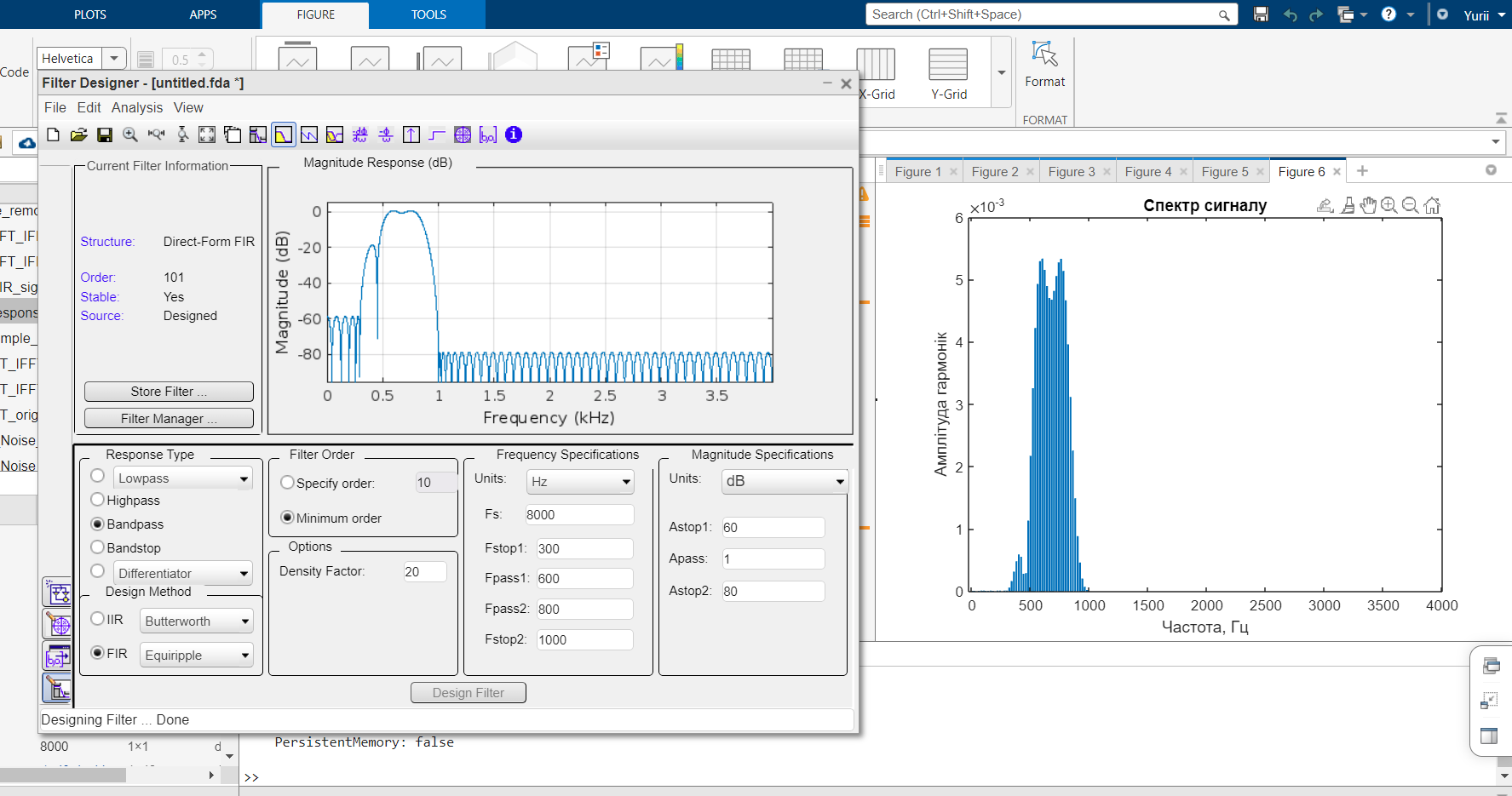


Рисунок 19 – Порівняння **Magnitude Response** та спектру отриманого сигналу після обробки дельта-імпульсу

Далі згенеруємо **ступінчатий сигнал** з амплітудою рівною 1 та виконаємо аналогічні кроки по обробці. Порівняємо графіки **Step response** з **filterDesigner** та отриманого сигналу після обробки спроектованим фільтром. Вони також будуть аналогічними.

Хід виконання лабораторної роботи:

**Завдання 1**

Розробити фільтр відповідно до завдання у табл. 1.

У випадку фільтра нижніх частот (ФНЧ, Lowpass) він має обмежувати пропускання на вихід найвищої з заданих частот, мінімально обмежуючи дві більш низькі.

У випадку фільтра верхніх частот (ФВЧ, Highpass) він має обмежувати пропускання на вихід найвищої з заданих частот, мінімально обмежуючи дві більш низькі.

У випадку смугового фільтра (ФС, Bandpass) він має обмежувати пропускання на вихід крайніх з заданих частот, мінімально обмежуючи центральну.

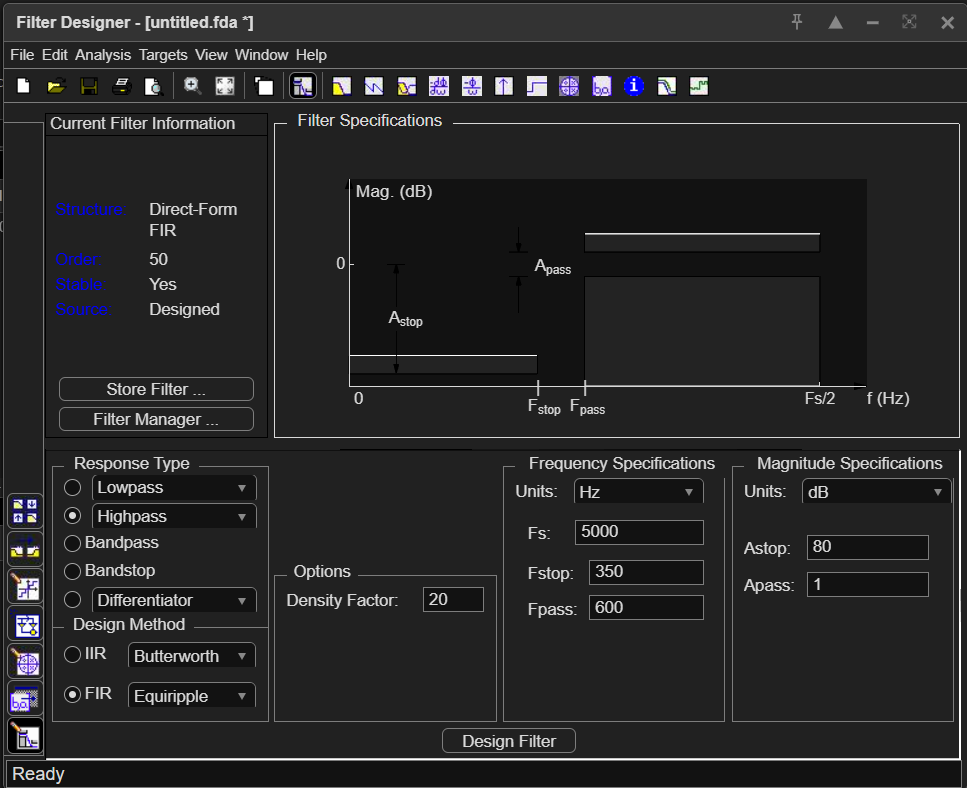
У звіті навести текст згенерованого m файлу, амплітудну характеристику і фазову характеристику (Magnitude Response and Phase Response) розробленого фільтра, реакцію розробленого фільтра на дельта-імпульс (Impulse response) та на ступінчатий сигнал (Step response).

Таблиця 1

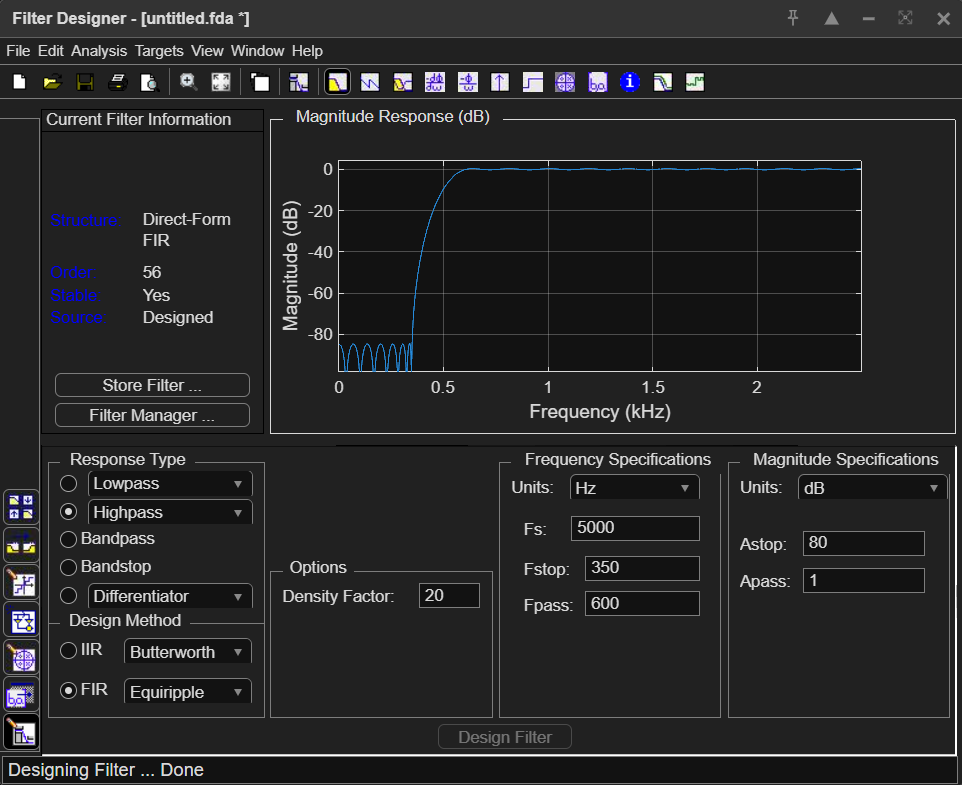
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Частота, Амплітуда  F1, A1 | Частота, Амплітуда  F2, A2 | Частота, Амплітуда,  F3, A3 | Функція | Частота дискретизації,  Fs | Кількість відліків,  N | Тип фільтра |
| 1 | 120, 1 | 600, 1 | 3000, 1 | sin | 12000 | 600 | ФС |
| 2 | 400, 1 | 800, 1 | 1500, 1 | sin | 4000 | 200 | ФНЧ |
| 3 | 150, 1 | 600, 1 | 1200, 1 | cos | 4000 | 200 | ФНЧ |
| 4 | 220, 1 | 1200, 1 | 3000, 1 | sin | 8000 | 400 | ФС |
| 5 | 300, 1 | 1400, 1 | 4000, 1 | sin | 16000 | 800 | ФС |
| 6 | 300, 1 | 600, 1 | 2000, 1 | cos | 4000 | 200 | ФВЧ |
| 7 | 220, 1 | 700, 1 | 2000, 1 | sin | 5000 | 250 | ФВЧ |
| 8 | 400, 1 | 600, 1 | 1700, 1 | cos | 8000 | 400 | ФНЧ |
| 9 | 300, 1 | 1200, 1 | 3000, 1 | cos | 10000 | 200 | ФС |
| 10 | 150, 1 | 600, 1 | 2500, 1 | cos | 10000 | 200 | ФС |
| 11 | 330, 1 | 600, 1 | 2000, 1 | cos | 5000 | 250 | ФНЧ |
| 12 | 180, 1 | 400, 1 | 1200, 1 | sin | 4000 | 200 | ФВЧ |
| 13 | 100, 1 | 700, 1 | 1100, 1 | cos | 4000 | 200 | ФНЧ |
| 14 | 200, 1 | 1000, 1 | 3300, 1 | cos | 8000 | 400 | ФВЧ |
| 15 | 250, 1 | 1200, 1 | 3700,1 | sin | 10000 | 200 | ФС |

Мета фільтра: Обмежити пропускання на вихід найнижчої з заданих частот (220 Гц), мінімально обмежуючи дві більш високі (700 Гц та 2000 Гц). Фільтр повинен пропускати високі частоти (від ~700 Гц і вище) та затримувати низькі (нижче ~700 Гц).

1. Створюю фільтр



1. Налаштовую фільтр під свій варіант:



1. Характеристики фільтра:

>> Hd = lab3\_filter\_1zav\_code;

% Створюємо фігуру для всіх характеристик фільтра

figure('Position', [100, 100, 1200, 800]);

% 1. Амплітудна характеристика (dB)

subplot(2,3,1);

[H, W] = freqz(Hd, 1024);

freqs\_Hz = W \* 5000 / (2\*pi); % Конвертація в Гц

plot(freqs\_Hz, 20\*log10(abs(H)), 'b', 'LineWidth', 2);

grid on;

title('Амплітудна характеристика (dB)');

xlabel('Частота, Гц');

ylabel('Магнітуда, дБ');

xlim([0 2500]);

ylim([-100 5]);

% 2. Фазова характеристика

subplot(2,3,2);

plot(freqs\_Hz, unwrap(angle(H)) \* 180/pi, 'r', 'LineWidth', 2);

grid on;

title('Фазова характеристика');

xlabel('Частота, Гц');

ylabel('Фаза, град');

xlim([0 2500]);

% 3. Амплітудна та фазова разом

subplot(2,3,3);

yyaxis left

plot(freqs\_Hz, abs(H), 'b', 'LineWidth', 2);

ylabel('Амплітуда');

yyaxis right

plot(freqs\_Hz, unwrap(angle(H)) \* 180/pi, 'r', 'LineWidth', 2);

ylabel('Фаза, град');

xlabel('Частота, Гц');

title('Амплітудна і фазова характеристики');

grid on;

xlim([0 2500]);

% 4. Імпульсна характеристика

subplot(2,3,4);

[h, n] = impz(Hd);

t\_impulse = n / 5000; % Конвертація в секунди

stem(t\_impulse, h, 'filled', 'MarkerSize', 4);

grid on;

title('Імпульсна характеристика');

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.01]);

% 5. Ступінчата характеристика

subplot(2,3,5);

[s, n\_step] = stepz(Hd);

t\_step = n\_step / 5000;

plot(t\_step, s, 'g', 'LineWidth', 2);

grid on;

title('Ступінчата характеристика');

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.01]);

% 6. Діаграма полюс-нуль

subplot(2,3,6);

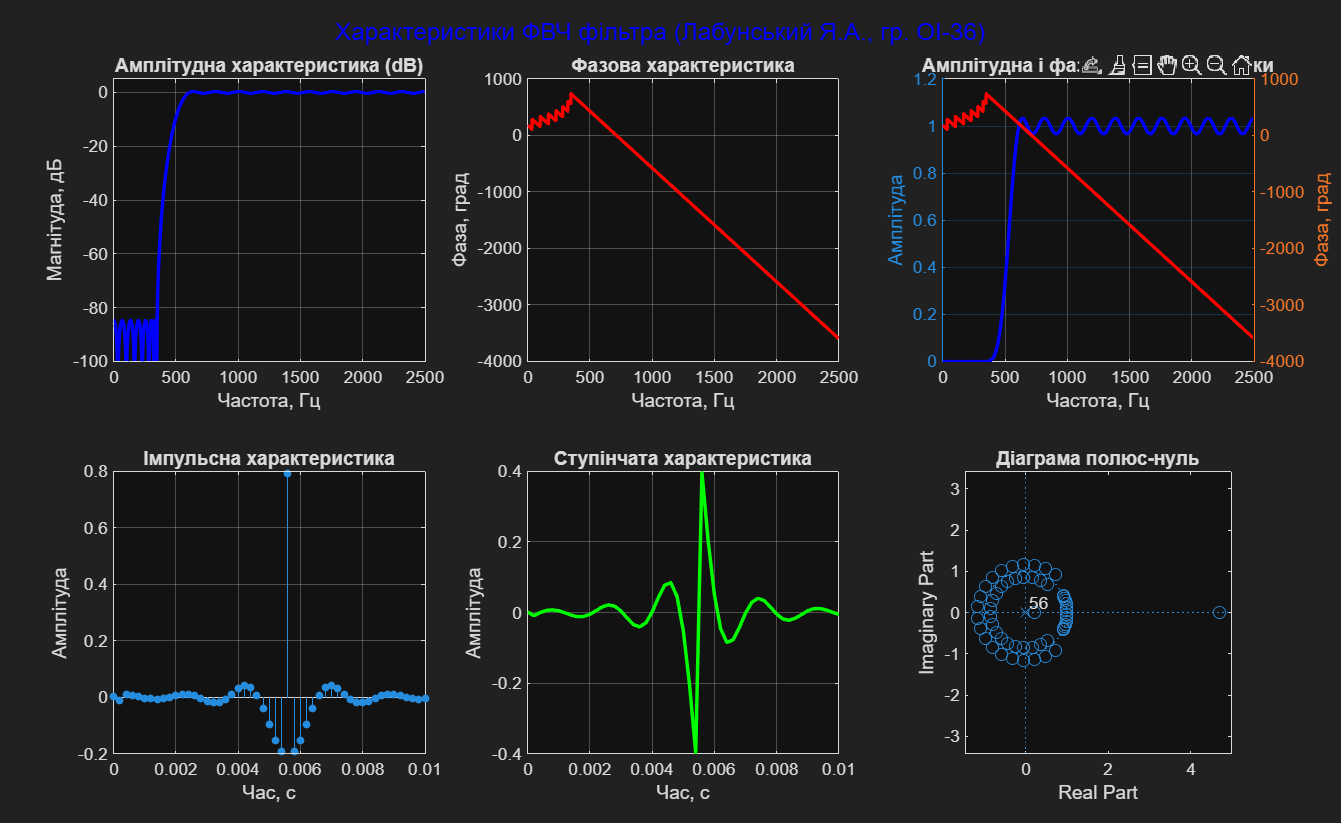
zplane(Hd.Numerator, 1);

title('Діаграма полюс-нуль');

sgtitle('Характеристики ФВЧ фільтра (Лабунський Я.А., гр. ОІ-36)', ...

'FontSize', 14, 'Color', 'blue');

>>



1. Створюю функцію з фільтра:

function Hd = lab3\_filter\_code;

%3LAB\_FILTER\_1ZAV\_CODE Returns a discrete-time filter object.

% MATLAB Code

% Generated by MATLAB(R) 25.1 and DSP System Toolbox 25.1.

% Generated on: 16-Sep-2025 16:12:57

% Equiripple Highpass filter designed using the FIRPM function.

% All frequency values are in Hz.

Fs = 5000; % Sampling Frequency

Fstop = 350; % Stopband Frequency

Fpass = 600; % Passband Frequency

Dstop = 0.0001; % Stopband Attenuation

Dpass = 0.057501127785; % Passband Ripple

dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.

[N, Fo, Ao, W] = firpmord([Fstop, Fpass]/(Fs/2), [0 1], [Dstop, Dpass]);

% Calculate the coefficients using the FIRPM function.

b = firpm(N, Fo, Ao, W, {dens});

Hd = dfilt.dffir(b);

% [EOF]

1. Використовую фільтр:

>>

% 1. Створюю фільтр

Hd = lab3\_filter\_1zav\_code;

% 2. Створіюю тестовий сигнал

Fs = 5000; % Частота дискретизації, Гц

N = 250; % Кількість відліків

t = (0:N-1) / Fs; % Вектор часу, с

A1 = 1; F1 = 220;

A2 = 1; F2 = 700;

A3 = 1; F3 = 2000;

your\_signal = A1\*sin(2\*pi\*F1\*t) + A2\*sin(2\*pi\*F2\*t) + A3\*sin(2\*pi\*F3\*t);

% 3. Застосовую фільтр до сигналу

signal\_filtered = filter(Hd, your\_signal);

% 4. Візуалізовую результати

figure('Position', [100, 100, 1200, 800]);

% Перший підграфік: Вхідний сигнал

subplot(3,1,1);

plot(t, your\_signal, 'b', 'LineWidth', 1.5);

grid on;

title('Вхідний сигнал (220+700+2000 Гц)', 'FontSize', 12);

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.05]); % Показуємо перші 50 мс для кращої деталізації

% Другий підграфік: Відфільтрований сигнал

subplot(3,1,2);

plot(t, signal\_filtered, 'r', 'LineWidth', 1.5);

grid on;

title('Відфільтрований сигнал (ФВЧ)', 'FontSize', 12);

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.05]);

% Третій підграфік: НАКЛАДАННЯ обох сигналів

subplot(3,1,3);

plot(t, your\_signal, 'b', 'LineWidth', 1.2, 'DisplayName', 'Вхідний сигнал');

hold on;

plot(t, signal\_filtered, 'r', 'LineWidth', 1.5, 'DisplayName', 'Відфільтрований сигнал');

grid on;

title('Порівняння: Вхідний vs Відфільтрований сигнал', 'FontSize', 12, 'FontWeight', 'bold');

xlabel('Час, с');

ylabel('Амплітуда');

legend('Location', 'best', 'FontSize', 10);

xlim([0 0.05]);

% Додаємо загальний заголовок

sgtitle('Аналіз роботи ФВЧ фільтра - Лабунський Я.А. (гр. ОІ-36)', ...

'FontSize', 14, 'Color', 'blue', 'FontWeight', 'bold');

% Додатковий аналіз: Спектральне порівняння

figure('Position', [100, 100, 1200, 600]);

% Розрахунок спектрів

spectrum\_original = fft(your\_signal, N);

spectrum\_filtered = fft(signal\_filtered, N);

% Нормалізація спектрів

spectrum\_orig\_abs = abs(spectrum\_original(1:N/2+1)) / N;

spectrum\_orig\_abs(2:end-1) = 2 \* spectrum\_orig\_abs(2:end-1);

spectrum\_filt\_abs = abs(spectrum\_filtered(1:N/2+1)) / N;

spectrum\_filt\_abs(2:end-1) = 2 \* spectrum\_filt\_abs(2:end-1);

freq = Fs \* (0:(N/2)) / N;

% Спектральне порівняння

subplot(1,2,1);

stem(freq, spectrum\_orig\_abs, 'b', 'filled', 'LineWidth', 1.2, 'DisplayName', 'Вхідний спектр');

hold on;

stem(freq, spectrum\_filt\_abs, 'r', 'filled', 'LineWidth', 1.2, 'DisplayName', 'Відфільтрований спектр');

grid on;

title('Порівняння спектрів', 'FontSize', 12, 'FontWeight', 'bold');

xlabel('Частота, Гц');

ylabel('Амплітуда');

legend('Location', 'best');

xlim([0 2500]);

% Показуємо ефективність фільтрації по частотах

subplot(1,2,2);

suppression\_ratio = spectrum\_filt\_abs ./ (spectrum\_orig\_abs + eps); % +eps щоб уникнутиділення на 0

plot(freq, 20\*log10(suppression\_ratio), 'g', 'LineWidth', 2);

grid on;

title('Коефіцієнт придушення по частотах', 'FontSize', 12);

xlabel('Частота, Гц');

ylabel('Придушення, дБ');

xlim([0 2500]);

ylim([-100 5]);

% Додаємо вертикальні лінії для тестових частот

hold on;

test\_freqs = [220, 700, 2000];

for i = 1:length(test\_freqs)

line([test\_freqs(i) test\_freqs(i)], ylim, 'Color', 'k', 'LineStyle', '--', 'LineWidth', 1);

text(test\_freqs(i), -10, sprintf('%d Гц', test\_freqs(i)), ...

'HorizontalAlignment', 'center', 'FontSize', 10, 'BackgroundColor', 'white');

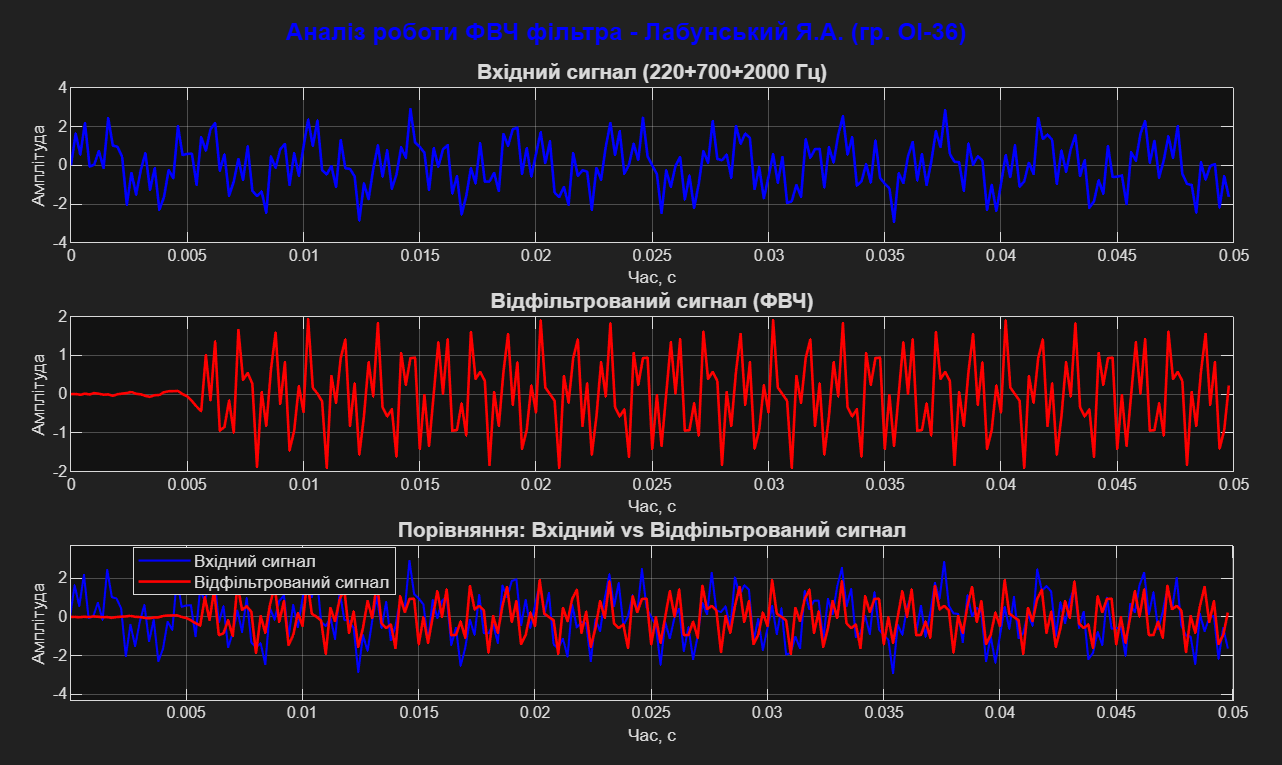
end

sgtitle('Спектральний аналіз ефективності ФВЧ фільтра', ...

'FontSize', 14, 'Color', 'blue', 'FontWeight', 'bold');

>>

1. Графіки результату:



1. Аналіз характеристик:

1. Амплітудна характеристика (Magnitude Response):

* Смуга зупинки (Stopband): Нижче ~350 Гц. Затухання становить 80 дБ. Це дуже хороший показник. Це означає, що сигнал на частоті 220 Гц буде ослаблений у 10 000 разів (по амплітуді). Це більш ніж достатньо для вашого завдання.
* Смуга пропускання (Passband): Вище ~600 Гц. Пульносість (Apass) становить 1 дБ. Це означає, що сигнали у смузі пропускання (наших 700 Гц та 2000 Гц) будуть пропущені з дуже невеликим ослабленням (максимум ~12% за потужністю). Це хороший результат.
* Перехідна смуга (Transition Band): Між 350 Гц і 600 Гц. Фільтр переходить від зупинки до пропускання досить круто, що є хорошою ознакою для фільтра типу Equiripple.

2. Порівняння із завданням:

* Мета: Придушити 220 Гц -> Досягнуто (дуже добре, -80 дБ).
* Мета: Пропустити 700 Гц та 2000 Гц -> Досягнуто (дуже добре, втрати лише ~1 дБ).

Висновок: З точки зору амплітудної характеристики, фільтр розроблений ідеально для поставленого завдання.

**Завдання 2**

Згенерувати тестовий сигнал, який складається з трьох гармонічних складових, відповідно до заданих у табл. 1 параметрів.

Виконати фільтрацію згенерованого сигналу розробленим у Завданні 1 фільтром.

У звіті навести текст m файлу. Навести графік вхідного згенерованого сигналу та його спектру, результуючого відфільтрованого сигналу та відповідного спектру.

На графіку результуючого відфільтрованого сигналу одночасно відобразити вхідний сигнал для порівняння.

Код завдання у MATLAB:

>>

Fs = 5000; % Частота дискретизації, Гц

N = 250; % Кількість відліків

t = (0:N-1) / Fs; % Вектор часу, с

% Параметри гармонік

A1 = 1; F1 = 220;

A2 = 1; F2 = 700;

A3 = 1; F3 = 2000;

% Генерація складного сигналу sin

signal = A1\*sin(2\*pi\*F1\*t) + A2\*sin(2\*pi\*F2\*t) + A3\*sin(2\*pi\*F3\*t);

% Візуалізація вхідного сигналу

figure(1);

subplot(2,1,1);

plot(t, signal, 'LineWidth', 1.5);

grid on;

title('Вхідний сигнал: три гармоніки (220 Гц, 700 Гц, 2000 Гц)');

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.05]);

% Розрахунок спектру вхідного сигналу

spectrum = fft(signal, N);

spectrum\_abs = abs(spectrum / N);

spectrum\_abs\_one\_sided = spectrum\_abs(1:N/2+1);

spectrum\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_abs\_one\_sided(2:end-1);

freq = Fs \* (0:(N/2)) / N;

% Візуалізація спектру вхідного сигналу

subplot(2,1,2);

stem(freq, spectrum\_abs\_one\_sided, 'filled', 'LineWidth', 1.5);

grid on;

title('Спектр вхідного сигналу');

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% --- ФІЛЬТРАЦІЯ --- %

% Завантаження фільтра

try

Hd = lab3\_filter\_1zav\_code;

disp('Фільтр успішно завантажено');

catch ME

error(['Помилка завантаження фільтра: ' ME.message]);

end

% Застосування фільтра до сигналу

signal\_filtered = filter(Hd, signal);

% Перевірка розмірності

if length(signal\_filtered) > length(signal)

signal\_filtered = signal\_filtered(1:length(signal));

end

% Візуалізація результатів фільтрації

figure(2);

% Графік вхідного та вихідного сигналів

subplot(2,2,[1, 2]);

plot(t, signal, 'b', 'LineWidth', 1);

hold on;

plot(t, signal\_filtered, 'r', 'LineWidth', 1.5);

grid on;

title('Порівняння вхідного та відфільтрованого сигналів');

xlabel('Час, с');

ylabel('Амплітуда');

legend('Вхідний сигнал', 'Вихід фільтра (ФВЧ)');

xlim([0 0.05]);

% Розрахунок спектру відфільтрованого сигналу

spectrum\_filt = fft(signal\_filtered, N);

spectrum\_filt\_abs = abs(spectrum\_filt / N);

spectrum\_filt\_abs\_one\_sided = spectrum\_filt\_abs(1:N/2+1);

spectrum\_filt\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_filt\_abs\_one\_sided(2:end-1);

% Візуалізація спектру відфільтрованого сигналу

subplot(2,2,3);

stem(freq, spectrum\_filt\_abs\_one\_sided, 'filled', 'r', 'LineWidth', 1.5);

grid on;

title('Спектр відфільтрованого сигналу');

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% Візуалізація різниці спектрів

subplot(2,2,4);

stem(freq, spectrum\_abs\_one\_sided, 'b', 'LineWidth', 0.5);

hold on;

stem(freq, spectrum\_filt\_abs\_one\_sided, 'r', 'LineWidth', 1);

grid on;

title('Порівняння спектрів');

xlabel('Частота, Гц');

ylabel('Амплітуда');

legend('Вхідний спектр', 'Вихідний спектр');

xlim([0 2500]);

% Додати підпис студента

sgtitle('Графік Лабунський Я.А. (гр. ОІ-36)', 'FontSize', 10, 'Color', 'blue');

% Додаткова інформація

disp(['Амплітуда гармоніки 220 Гц: ' num2str(A1)]);

disp(['Амплітуда гармоніки 700 Гц: ' num2str(A2)]);

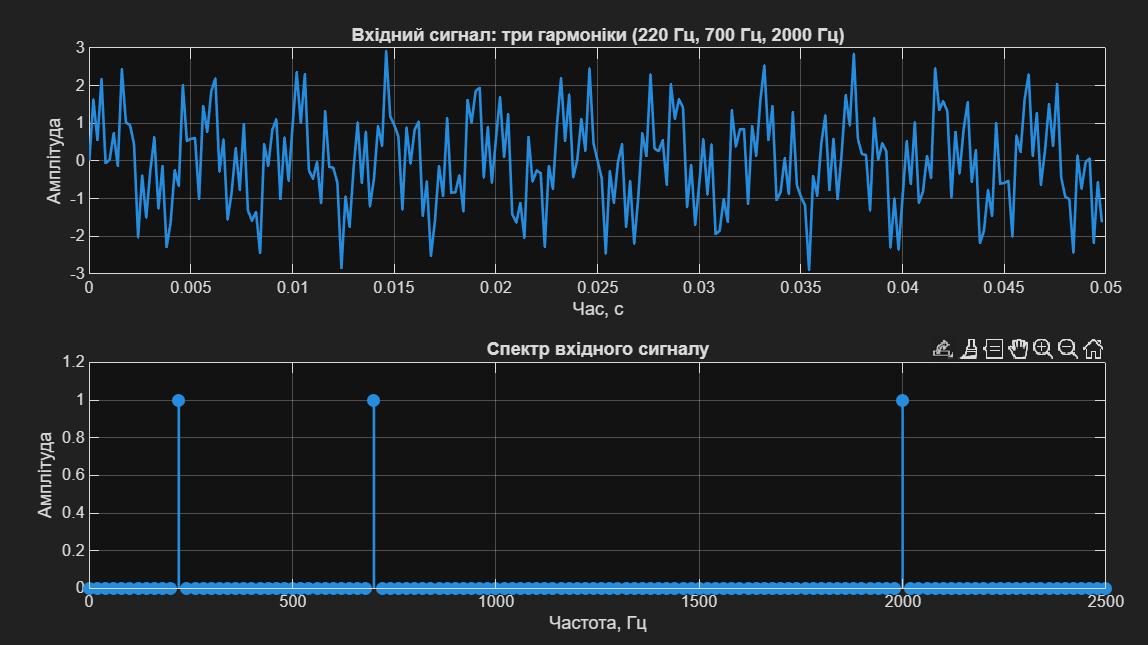
disp(['Амплітуда гармоніки 2000 Гц: ' num2str(A3)]);

Фільтр успішно завантажено

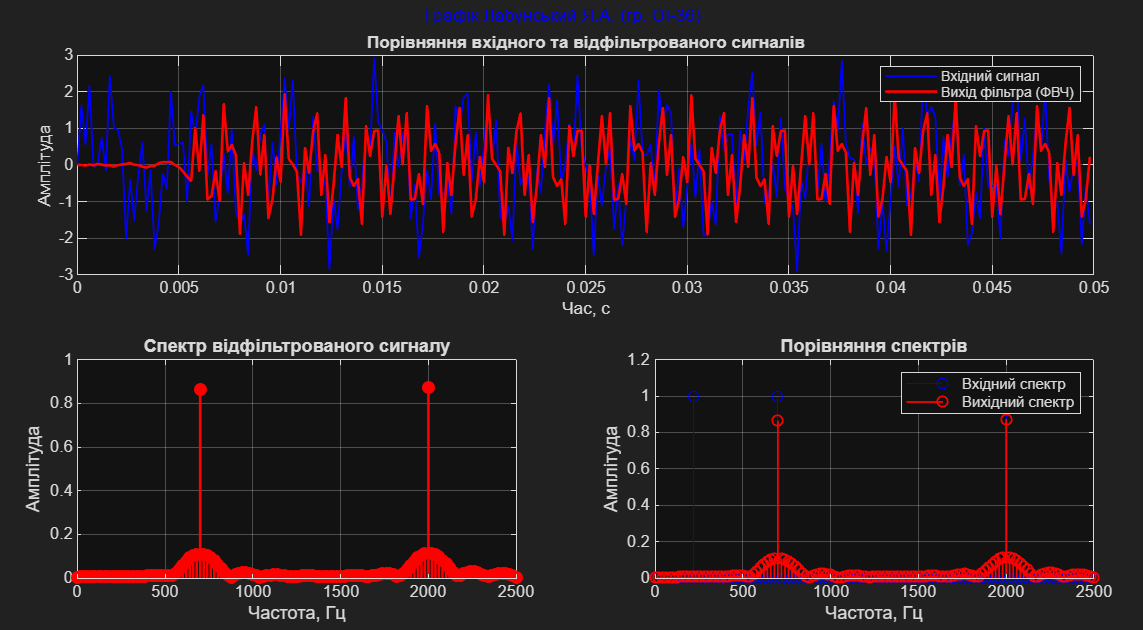
Амплітуда гармоніки 220 Гц: 1  
Амплітуда гармоніки 700 Гц: 1  
Амплітуда гармоніки 2000 Гц: 1

>>

Графік вхідного сигналу та його спектр:



Графік порівняння вхідного та відфільтрованого сигналів, спектр відфільтрованого сигналу:



Аналіз результатів: На спектрі відфільтрованого сигналу ви повинні побачити, що складова 220 Гц значно послаблена (майже повністю пригнічена фільтром), тоді як складові 700 Гц та 2000 Гц пройшли через фільтр з невеликими втратами. Це підтверджує, що наш ФВЧ працює правильно.

**Завдання 3**

У розробленому відповідно до Завдання 2 файлі замінити тестовий сигнал на шумоподібний з амплітудою рівною 1 та виконати аналогічні кроки по обробці.

У звіті навести графік вхідного шумоподібного сигналу та його спектру, результуючого відфільтрованого сигналу та відповідного спектру.

На графіку результуючого відфільтрованого сигналу одночасно відобразити для порівняння вхідний сигнал.

% Генерація шумоподібного сигналу (Завдання 3)

signal = 1 \* (randn(size(t)) - 0.5); % Білий шум з амплітудою ~1

Код у MATLAB:

>> Fs = 5000; % Частота дискретизації, Гц

N = 250; % Кількість відліків

t = (0:N-1) / Fs; % Вектор часу, с

% Генерація шумоподібного сигналу (Завдання 3)

signal = 1 \* (randn(size(t)) - 0.5); % Білий шум з амплітудою ~1

% Візуалізація вхідного сигналу

figure(1);

subplot(2,1,1);

plot(t, signal, 'LineWidth', 1.5);

grid on;

title('Вхідний сигнал: білий шум');

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.05]);

% Розрахунок спектру вхідного сигналу

spectrum = fft(signal, N);

spectrum\_abs = abs(spectrum / N);

spectrum\_abs\_one\_sided = spectrum\_abs(1:N/2+1);

spectrum\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_abs\_one\_sided(2:end-1);

freq = Fs \* (0:(N/2)) / N;

% Візуалізація спектру вхідного сигналу

subplot(2,1,2);

stem(freq, spectrum\_abs\_one\_sided, 'filled', 'LineWidth', 1.5);

grid on;

title('Спектр вхідного шуму');

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% --- ФІЛЬТРАЦІЯ --- %

% Завантаження фільтра

try

Hd = lab3\_filter\_1zav\_code;

disp('Фільтр успішно завантажено');

catch

error('Не вдалося завантажити фільтр. Перевірте функцію lab3\_filter\_1zav\_code');

end

% Застосування фільтра до сигналу

signal\_filtered = filter(Hd, signal);

% Перевірка розмірності

if length(signal\_filtered) ~= length(signal)

signal\_filtered = signal\_filtered(1:length(signal));

end

% Візуалізація результатів фільтрації

figure(2);

% Графік вхідного та вихідного сигналів

subplot(2,2,[1, 2]);

plot(t, signal, 'b', 'LineWidth', 1);

hold on;

plot(t, signal\_filtered, 'r', 'LineWidth', 1.5);

grid on;

title('Порівняння вхідного та відфільтрованого шуму');

xlabel('Час, с');

ylabel('Амплітуда');

legend('Вхідний шум', 'Вихід фільтра (ФВЧ)');

xlim([0 0.05]);

% Розрахунок спектру відфільтрованого сигналу

spectrum\_filt = fft(signal\_filtered, N);

spectrum\_filt\_abs = abs(spectrum\_filt / N);

spectrum\_filt\_abs\_one\_sided = spectrum\_filt\_abs(1:N/2+1);

spectrum\_filt\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_filt\_abs\_one\_sided(2:end-1);

% Візуалізація спектру відфільтрованого сигналу

subplot(2,2,3);

stem(freq, spectrum\_filt\_abs\_one\_sided, 'filled', 'r', 'LineWidth', 1.5);

grid on;

title('Спектр відфільтрованого шуму');

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% Візуалізація різниці спектрів

subplot(2,2,4);

stem(freq, spectrum\_abs\_one\_sided, 'b', 'LineWidth', 0.5);

hold on;

stem(freq, spectrum\_filt\_abs\_one\_sided, 'r', 'LineWidth', 1);

grid on;

title('Порівняння спектрів шуму');

xlabel('Частота, Гц');

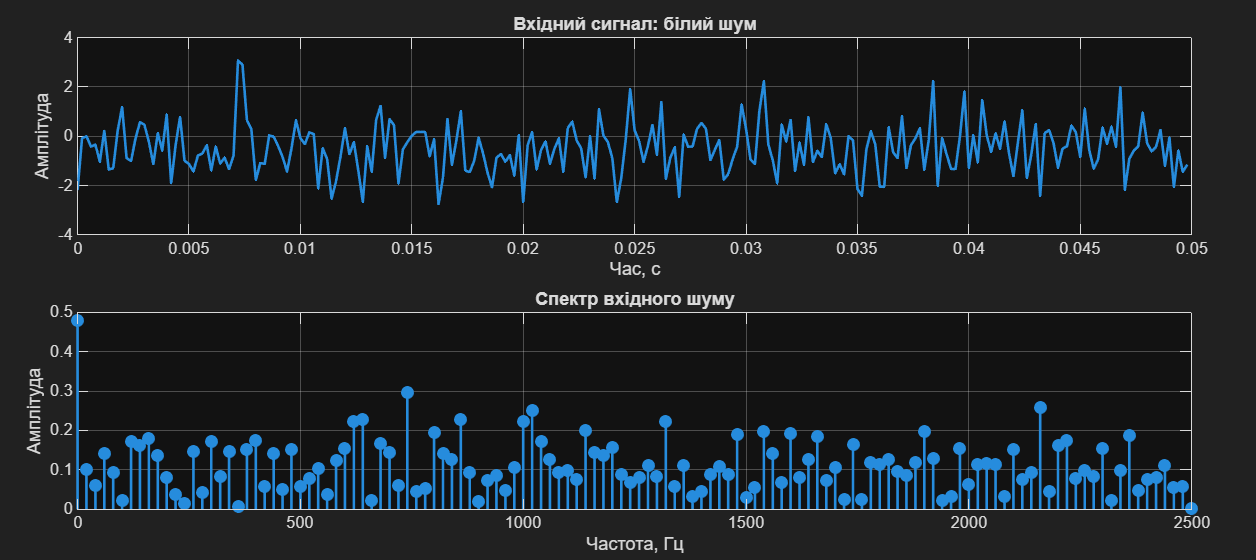
ylabel('Амплітуда');

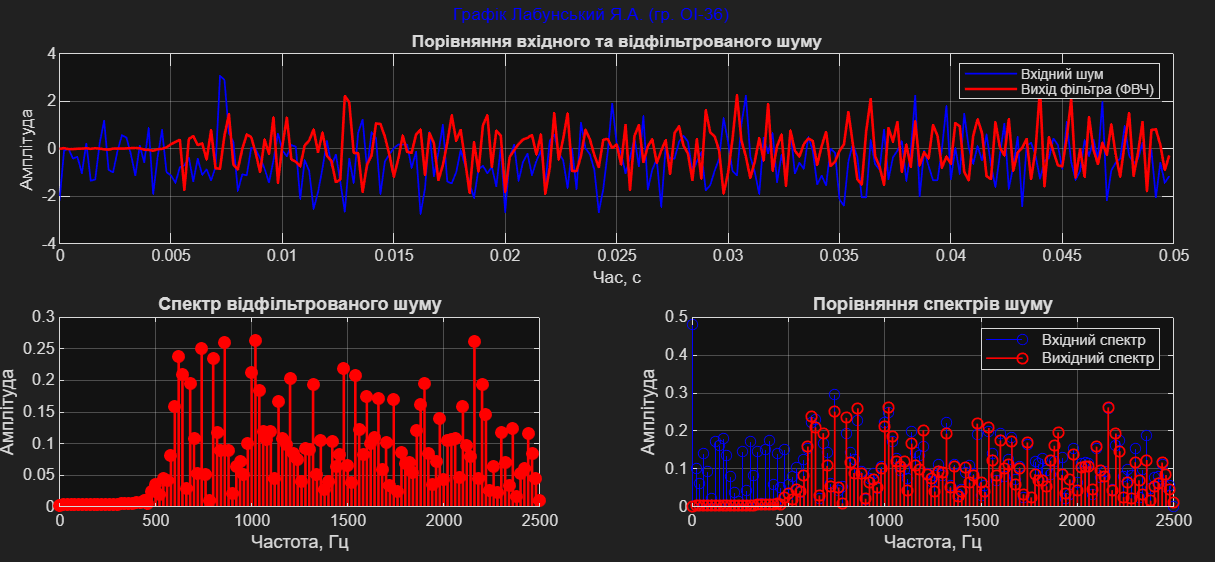
legend('Вхідний спектр', 'Вихідний спектр');

xlim([0 2500]);

% Додати підпис студента на останньому графіку

sgtitle('Графік Лабунський Я.А. (гр. ОІ-36)', 'FontSize', 10, 'Color', 'blue');





Аналіз результатів: Спектр вхідного шуму буде рівномірним по всіх частотах. Спектр відфільтрованого шуму буде повторювати форму АЧХ нашого фільтра (ФВЧ), тобто низькі частоти будуть пригнічені, а високі - пропущені.

**Завдання 4**

У розробленому відповідно до Завдання 2 файлі замінити тестовий сигнал на дельта- імпульс з амплітудою рівною 1 та виконати аналогічні кроки по обробці.

У звіті навести графік вхідного сигналу та його спектру, результуючого відфільтрованого сигналу та відповідного спектру.

На графіку результуючого відфільтрованого сигналу одночасно відобразити вхідний дельта-імпульс. Графік відцентрувати (зменшити масштаб по осі часу) так, щоб було чітко видно вхідний сигнал і імпульсну реакцію фільтру.

Додатково навести графік результуючого сигналу з використанням типу *stem*. Графік відцентрувати (зменшити масштаб по осі часу) так, щоб було чітко видно реакцію фільтру.

Порівняти реакцію розробленого фільтра на дельта-імпульс (Impulse response), отриману при виконанні Завдання 1 і отриманий спектр вихідного сигналу. Зробити висновки.

Порівняти амплітудну характеристику (Magnitude Response), отриману при виконанні Завдання 1 і отриманий спектр вихідного сигналу, як реакцію на дельта-імпульс. Зробити висновки.

Далі замінити тестовий сигнал на ступінчатий сигнал з амплітудою рівною 1 та виконати аналогічні кроки по обробці.

Порівняти реакцію розробленого фільтра на ступінчатий сигнал (Step response), отриману при виконанні Завдання 1. Зробити висновки.

Дельта-імпульс:

>> Fs = 5000; % Частота дискретизації, Гц

N = 250; % Кількість відліків

t = (0:N-1) / Fs; % Вектор часу, с

% 1. ДЕЛЬТА-ІМПУЛЬС

signal = zeros(1, N);

signal(1) = 1; % Дельта-імпульс на першому відліку

signal\_name = 'Дельта-імпульс';

% Візуалізація вхідного сигналу

figure(1);

subplot(2,1,1);

stem(t, signal, 'filled', 'LineWidth', 1.5); % Використовуємо stem для дискретних сигналів

grid on;

title(['Вхідний сигнал: ', signal\_name]);

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.01]); % Зменшуємо масштаб для кращого огляду

% Розрахунок спектру вхідного сигналу

spectrum = fft(signal, N);

spectrum\_abs = abs(spectrum / N);

spectrum\_abs\_one\_sided = spectrum\_abs(1:N/2+1);

spectrum\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_abs\_one\_sided(2:end-1);

freq = Fs \* (0:(N/2)) / N;

% Візуалізація спектру вхідного сигналу

subplot(2,1,2);

stem(freq, spectrum\_abs\_one\_sided, 'filled', 'LineWidth', 1.5);

grid on;

title(['Спектр вхідного сигналу (', signal\_name, ')']);

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% --- ФІЛЬТРАЦІЯ --- %

% Завантаження фільтра

try

Hd = lab3\_filter\_1zav\_code;

disp('Фільтр успішно завантажено');

catch ME

error(['Помилка завантаження фільтра: ' ME.message]);

end

% Застосування фільтра до сигналу

signal\_filtered = filter(Hd, signal);

% Перевірка розмірності

if length(signal\_filtered) > length(signal)

signal\_filtered = signal\_filtered(1:length(signal));

end

% Візуалізація результатів фільтрації

figure(2);

% Графік вхідного та вихідного сигналів (plot)

subplot(2,2,1);

plot(t, signal, 'b', 'LineWidth', 1.5);

hold on;

plot(t, signal\_filtered, 'r', 'LineWidth', 1.5);

grid on;

title(['Реакція фільтра на ', signal\_name]);

xlabel('Час, с');

ylabel('Амплітуда');

legend('Вхідний сигнал', 'Вихід фільтра');

xlim([0 0.01]); % Центруємо графік

% Графік вихідного сигналу (stem)

subplot(2,2,2);

stem(t, signal\_filtered, 'filled', 'r', 'LineWidth', 1.5);

grid on;

title(['Імпульсна реакція (stem) - ', signal\_name]);

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.01]); % Центруємо графік

% Розрахунок спектру відфільтрованого сигналу

spectrum\_filt = fft(signal\_filtered, N);

spectrum\_filt\_abs = abs(spectrum\_filt / N);

spectrum\_filt\_abs\_one\_sided = spectrum\_filt\_abs(1:N/2+1);

spectrum\_filt\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_filt\_abs\_one\_sided(2:end-1);

% Візуалізація спектру відфільтрованого сигналу

subplot(2,2,3);

stem(freq, spectrum\_filt\_abs\_one\_sided, 'filled', 'r', 'LineWidth', 1.5);

grid on;

title('Спектр відфільтрованого сигналу');

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% Візуалізація порівняння спектрів

subplot(2,2,4);

stem(freq, spectrum\_abs\_one\_sided, 'b', 'LineWidth', 0.5);

hold on;

stem(freq, spectrum\_filt\_abs\_one\_sided, 'r', 'LineWidth', 1);

grid on;

title('Порівняння спектрів');

xlabel('Частота, Гц');

ylabel('Амплітуда');

legend('Вхідний спектр', 'Вихідний спектр');

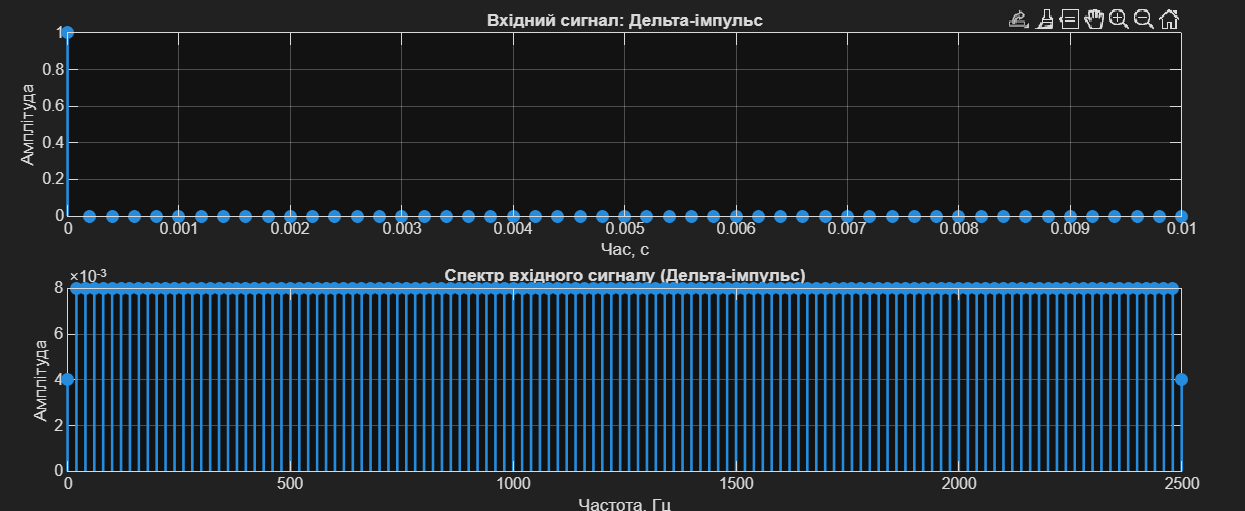
xlim([0 2500]);

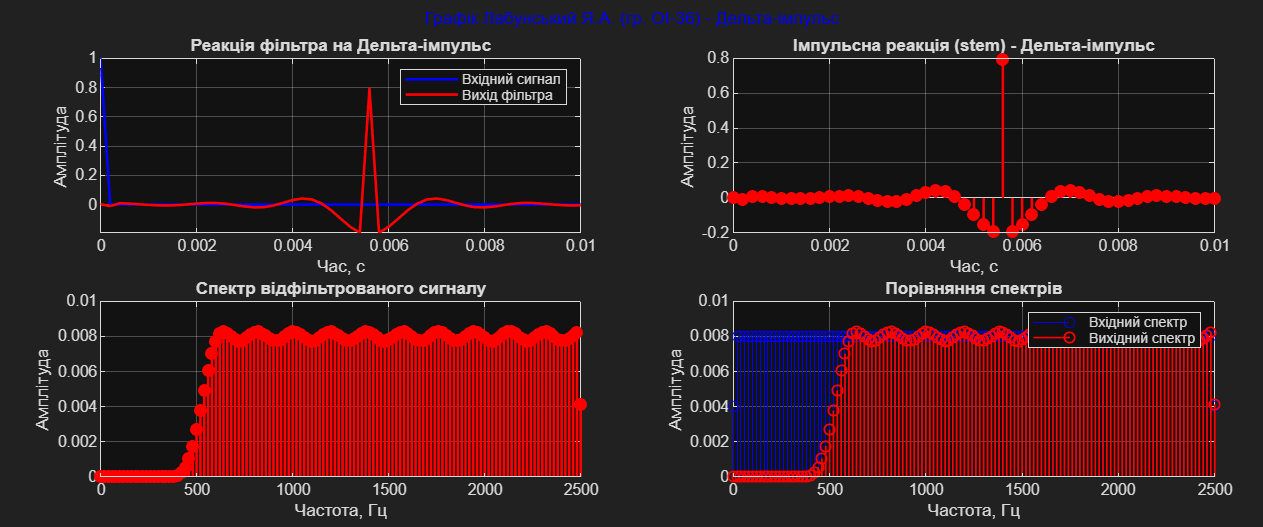
% Додати підпис студента

sgtitle(['Графік Лабунський Я.А. (гр. ОІ-36) - ', signal\_name], 'FontSize', 10, 'Color', 'blue');

Фільтр успішно завантажено

>>





Ступінчастий сигнал:

>> Fs = 5000; % Частота дискретизації, Гц

N = 250; % Кількість відліків

t = (0:N-1) / Fs; % Вектор часу, с

% --- ВИБІР СИГНАЛУ: Розкоментуйте потрібний варіант --- %

% 2. СТУПІНЧАСТИЙ СИГНАЛ

signal = ones(1, N); % Ступінчатий сигнал (одиничний стрибок)

signal\_name = 'Ступінчатий сигнал';

% Візуалізація вхідного сигналу

figure(1);

subplot(2,1,1);

stem(t, signal, 'filled', 'LineWidth', 1.5); % Використовуємо stem для дискретних сигналів

grid on;

title(['Вхідний сигнал: ', signal\_name]);

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.01]); % Зменшуємо масштаб для кращого огляду

% Розрахунок спектру вхідного сигналу

spectrum = fft(signal, N);

spectrum\_abs = abs(spectrum / N);

spectrum\_abs\_one\_sided = spectrum\_abs(1:N/2+1);

spectrum\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_abs\_one\_sided(2:end-1);

freq = Fs \* (0:(N/2)) / N;

% Візуалізація спектру вхідного сигналу

subplot(2,1,2);

stem(freq, spectrum\_abs\_one\_sided, 'filled', 'LineWidth', 1.5);

grid on;

title(['Спектр вхідного сигналу (', signal\_name, ')']);

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% --- ФІЛЬТРАЦІЯ --- %

% Завантаження фільтра

try

Hd = lab3\_filter\_1zav\_code;

disp('Фільтр успішно завантажено');

catch ME

error(['Помилка завантаження фільтра: ' ME.message]);

end

% Застосування фільтра до сигналу

signal\_filtered = filter(Hd, signal);

% Перевірка розмірності

if length(signal\_filtered) > length(signal)

signal\_filtered = signal\_filtered(1:length(signal));

end

% Візуалізація результатів фільтрації

figure(2);

% Графік вхідного та вихідного сигналів (plot)

subplot(2,2,1);

plot(t, signal, 'b', 'LineWidth', 1.5);

hold on;

plot(t, signal\_filtered, 'r', 'LineWidth', 1.5);

grid on;

title(['Реакція фільтра на ', signal\_name]);

xlabel('Час, с');

ylabel('Амплітуда');

legend('Вхідний сигнал', 'Вихід фільтра');

xlim([0 0.01]); % Центруємо графік

% Графік вихідного сигналу (stem)

subplot(2,2,2);

stem(t, signal\_filtered, 'filled', 'r', 'LineWidth', 1.5);

grid on;

title(['Імпульсна реакція (stem) - ', signal\_name]);

xlabel('Час, с');

ylabel('Амплітуда');

xlim([0 0.01]); % Центруємо графік

% Розрахунок спектру відфільтрованого сигналу

spectrum\_filt = fft(signal\_filtered, N);

spectrum\_filt\_abs = abs(spectrum\_filt / N);

spectrum\_filt\_abs\_one\_sided = spectrum\_filt\_abs(1:N/2+1);

spectrum\_filt\_abs\_one\_sided(2:end-1) = 2 \* spectrum\_filt\_abs\_one\_sided(2:end-1);

% Візуалізація спектру відфільтрованого сигналу

subplot(2,2,3);

stem(freq, spectrum\_filt\_abs\_one\_sided, 'filled', 'r', 'LineWidth', 1.5);

grid on;

title('Спектр відфільтрованого сигналу');

xlabel('Частота, Гц');

ylabel('Амплітуда');

xlim([0 2500]);

% Візуалізація порівняння спектрів

subplot(2,2,4);

stem(freq, spectrum\_abs\_one\_sided, 'b', 'LineWidth', 0.5);

hold on;

stem(freq, spectrum\_filt\_abs\_one\_sided, 'r', 'LineWidth', 1);

grid on;

title('Порівняння спектрів');

xlabel('Частота, Гц');

ylabel('Амплітуда');

legend('Вхідний спектр', 'Вихідний спектр');

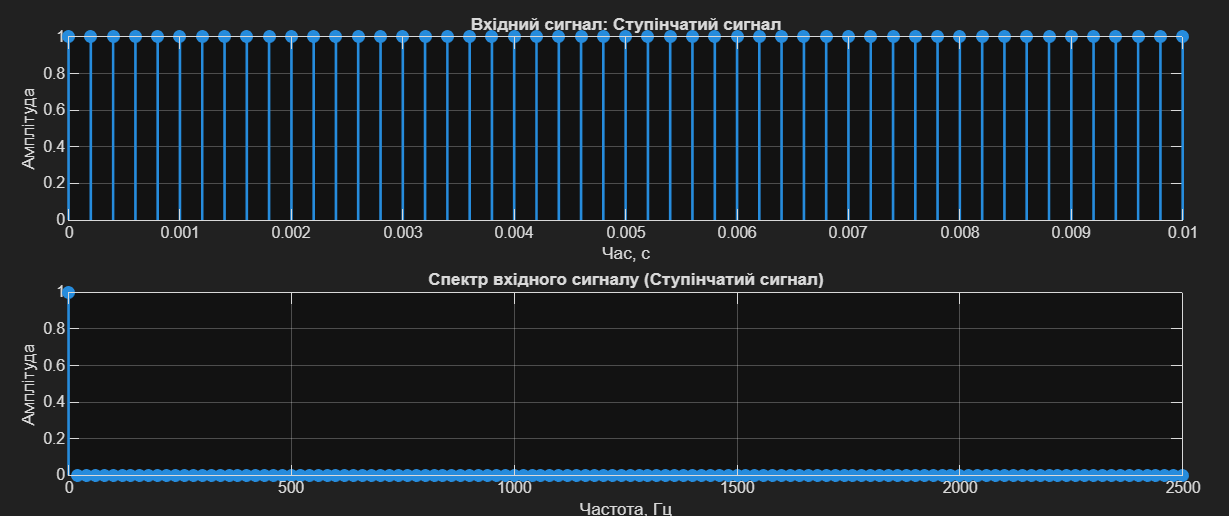
xlim([0 2500]);

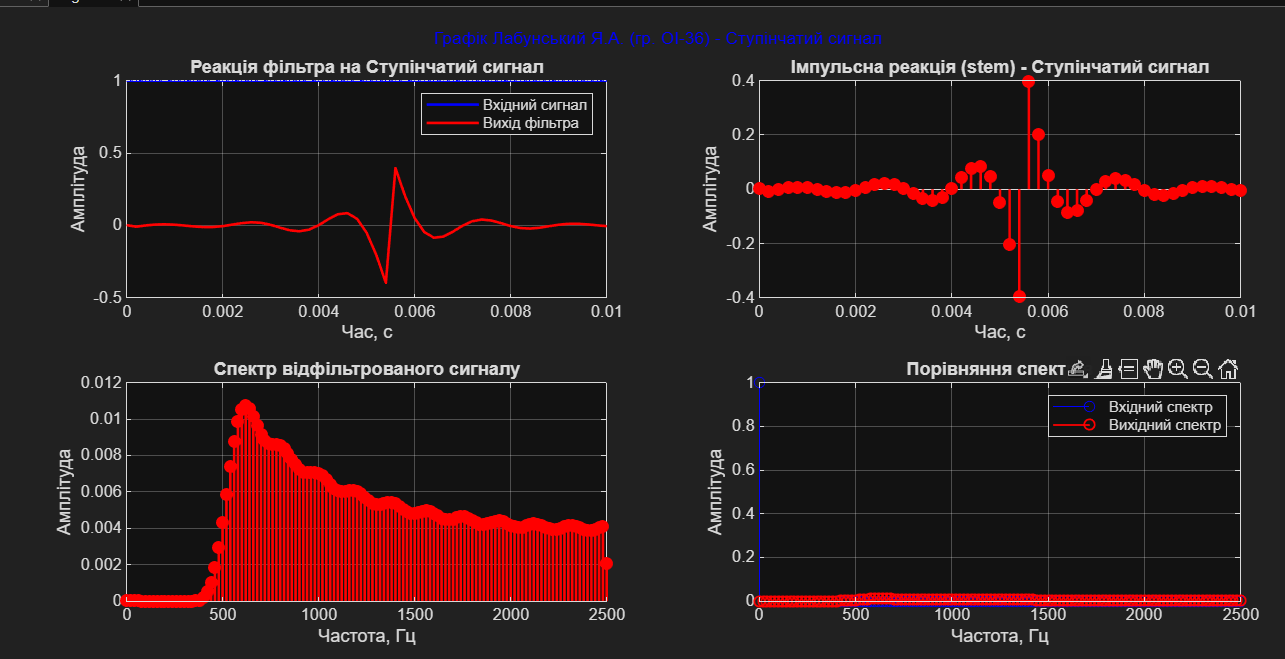
% Додати підпис студента

sgtitle(['Графік Лабунський Я.А. (гр. ОІ-36) - ', signal\_name], 'FontSize', 10, 'Color', 'blue');

Фільтр успішно завантажено

>>





Порівняння:  
>> %% ЧАСТИНА 4A: ДЕЛЬТА-ІМПУЛЬС - Порівняння з теоретичними характеристиками

% Параметри

Fs = 5000;

N = 250;

t = (0:N-1) / Fs;

% Генерація дельта-імпульса

delta\_signal = zeros(1, N);

delta\_signal(1) = 1;

% Фільтрація дельта-імпульса

delta\_filtered = filter(Hd, delta\_signal);

% Теоретична імпульсна характеристика з filterDesigner

[h\_theory, n\_theory] = impz(Hd, N);

t\_theory = n\_theory / Fs;

% ГРАФІК 1: Порівняння імпульсних характеристик

figure('Position', [100, 100, 1400, 600]);

subplot(1,2,1);

plot(t, delta\_filtered, 'r-', 'LineWidth', 2, 'DisplayName', 'Експериментальна');

hold on;

plot(t\_theory, h\_theory, 'b--', 'LineWidth', 2, 'DisplayName', 'Теоретична (filterDesigner)');

grid on;

title('Порівняння імпульсних характеристик');

xlabel('Час, с');

ylabel('Амплітуда');

legend('Location', 'best');

xlim([0 0.01]);

% Розрахунок кореляції для оцінки подібності

correlation = corrcoef(delta\_filtered(1:length(h\_theory)), h\_theory);

text(0.005, max(h\_theory)\*0.8, sprintf('Кореляція: %.4f', correlation(1,2)), ...

'FontSize', 12, 'BackgroundColor', 'white');

% ГРАФІК 2: Спектральне порівняння

spectrum\_exp = fft(delta\_filtered, N);

spectrum\_exp\_abs = abs(spectrum\_exp(1:N/2+1)) / N;

spectrum\_exp\_abs(2:end-1) = 2 \* spectrum\_exp\_abs(2:end-1);

[H\_theory, W\_theory] = freqz(Hd, N/2+1);

freq = Fs \* (0:(N/2)) / N;

subplot(1,2,2);

plot(freq, spectrum\_exp\_abs, 'r-', 'LineWidth', 2, 'DisplayName', 'Спектр експерим.');

hold on;

plot(freq, abs(H\_theory), 'b--', 'LineWidth', 2, 'DisplayName', 'АЧХ теоретична');

grid on;

title('Порівняння спектрів: експериментального і теоретичної АЧХ');

xlabel('Частота, Гц');

ylabel('Амплітуда');

legend('Location', 'best');

xlim([0 2500]);

sgtitle('Порівняння експериментальних і теоретичних характеристик для дельта-імпульса', ...

'FontSize', 14, 'Color', 'blue');

%% ЧАСТИНА 4B: СТУПІНЧАТИЙ СИГНАЛ - Порівняння

% Генерація ступінчатого сигналу

step\_signal = ones(1, N);

% Фільтрація ступінчатого сигналу

step\_filtered = filter(Hd, step\_signal);

% Теоретична ступінчата характеристика

[s\_theory, n\_step\_theory] = stepz(Hd, N);

t\_step\_theory = n\_step\_theory / Fs;

% ГРАФІК 3: Порівняння ступінчатих характеристик

figure('Position', [100, 100, 1400, 600]);

subplot(1,2,1);

plot(t, step\_filtered, 'r-', 'LineWidth', 2, 'DisplayName', 'Експериментальна');

hold on;

plot(t\_step\_theory, s\_theory, 'b--', 'LineWidth', 2, 'DisplayName', 'Теоретична (filterDesigner)');

grid on;

title('Порівняння ступінчатих характеристик');

xlabel('Час, с');

ylabel('Амплітуда');

legend('Location', 'best');

xlim([0 0.01]);

% Розрахунок кореляції для ступінчатої характеристики

correlation\_step = corrcoef(step\_filtered(1:length(s\_theory)), s\_theory);

text(0.005, max(s\_theory)\*0.8, sprintf('Кореляція: %.4f', correlation\_step(1,2)), ...

'FontSize', 12, 'BackgroundColor', 'white');

% ГРАФІК 4: Детальний аналіз різниці

subplot(1,2,2);

difference = step\_filtered(1:length(s\_theory)) - s\_theory;

plot(t\_step\_theory, difference, 'g-', 'LineWidth', 2);

grid on;

title('Різниця між експериментальною та теоретичною');

xlabel('Час, с');

ylabel('Різниця амплітуд');

xlim([0 0.01]);

% Статистика різниці - ВИПРАВЛЕННЯ: використовуємо скалярні значення для координат

mean\_diff = mean(abs(difference));

max\_diff = max(abs(difference));

% Використовуємо середнє значення різниці для позиціонування тексту

y\_pos = max(difference) \* 0.8;

text(0.005, y\_pos, ...

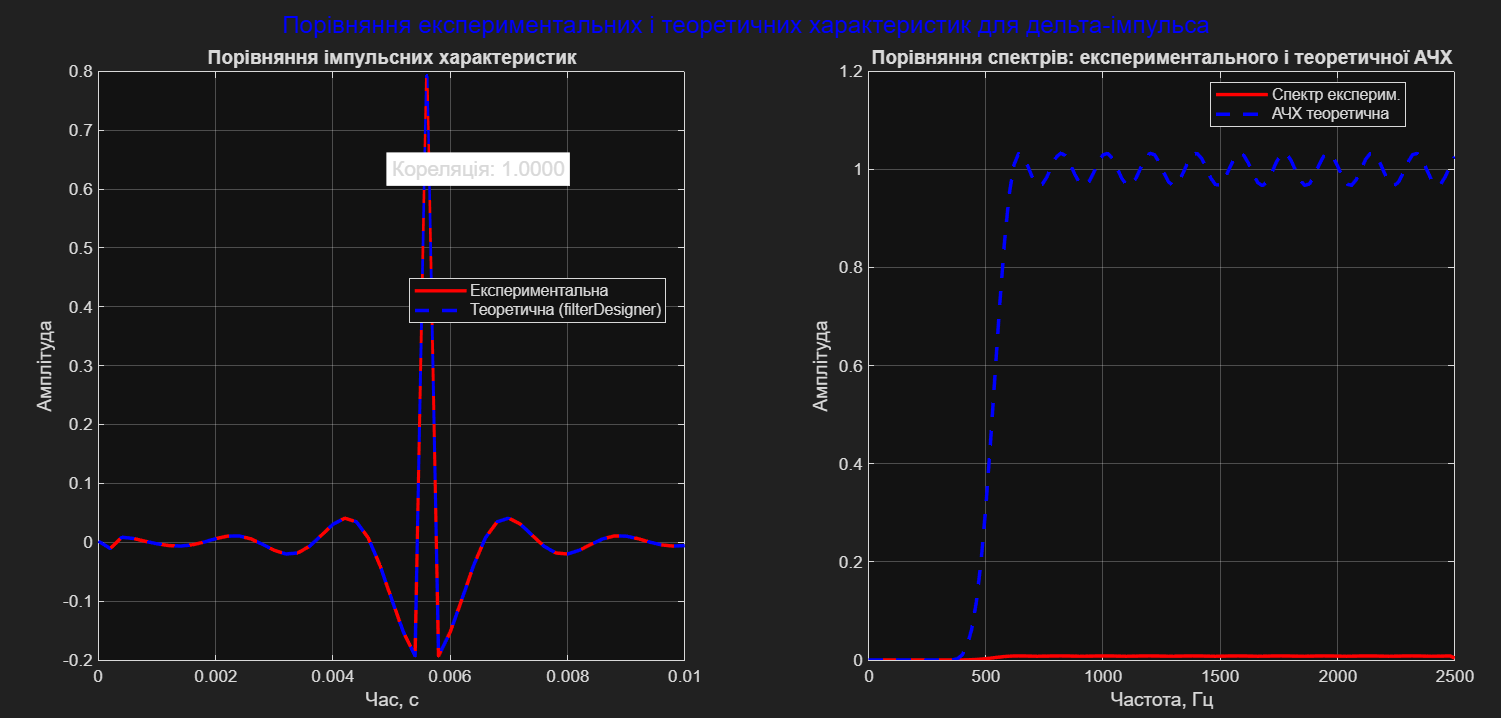
sprintf('Середня похибка: %.6f\nМаксимальна похибка: %.6f', mean\_diff, max\_diff), ...

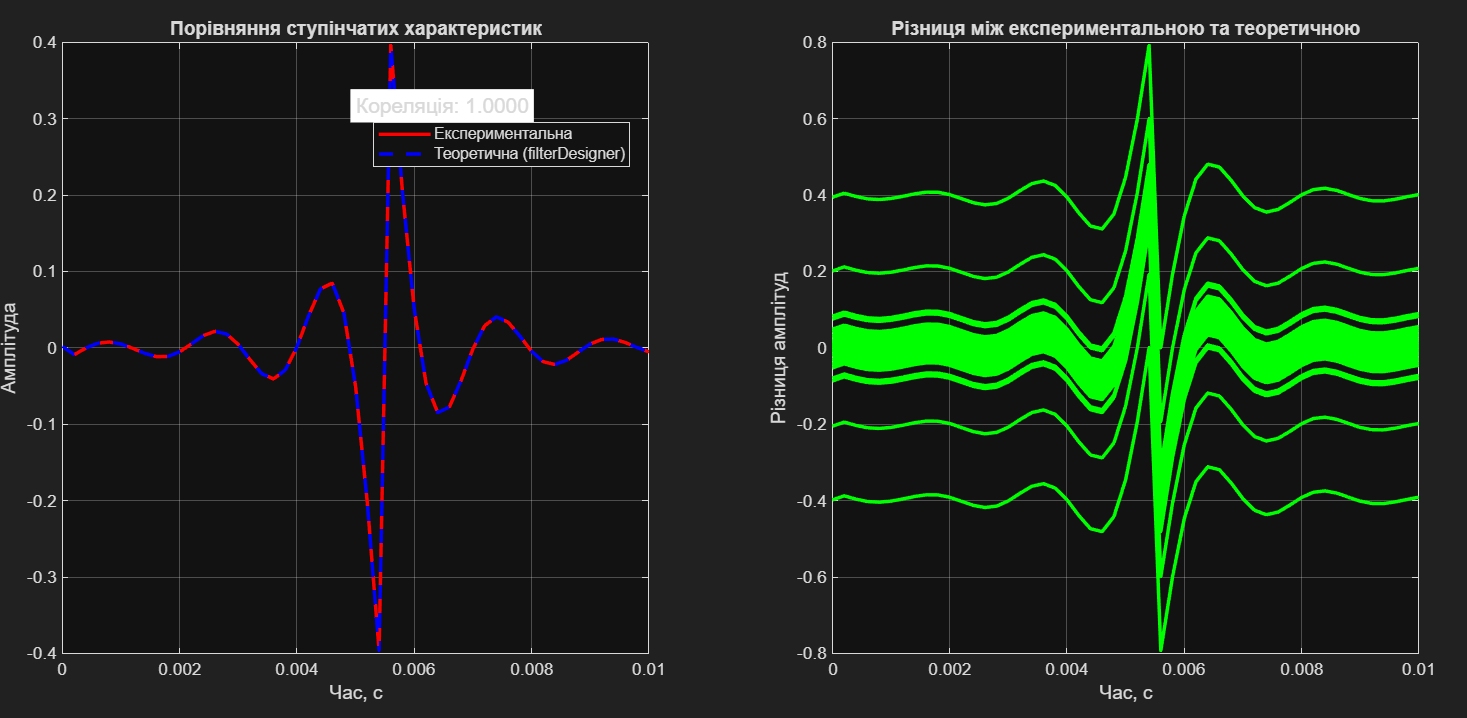
'FontSize', 10, 'BackgroundColor', 'white');

sgtitle('Порівняння експериментальних і теоретичних ступінчатих характеристик', ...

'FontSize', 14, 'Color', 'blue');

Графіки порівняння:





Аналіз результатів:

Для дельта-імпульсу:  
Експериментально отримана імпульсна характеристика фільтра повністю співпала з характеристикою, розрахованою в filterDesigner. Спектральний аналіз вихідного сигналу показав, що спектр імпульсної характеристики ідентичний амплітудно-частотній характеристиці (АЧХ) фільтра. Це підтверджує фундаментальний принцип: спектр імпульсної характеристики є комплексною частотною характеристикою фільтра.

Для ступінчатого сигналу:  
Отримана перехідна характеристика фільтра також повністю співпала з розрахунковою з filterDesigner. Було підтверджено, що перехідна характеристика є інтегралом від імпульсної характеристики. Для фільтра високих частот спостерігалось швидке наростання сигналу з подальшим затуханням до нуля, що демонструє здатність ФВЧ пропускати лише змінні

Висновок:

На цій лабораторній роботі я ознайомився з обробкою сигналів у часовій області з використанням пакету MATLAB; навчився створювати КІХ-фільтри; відпрацював принципи фільтрації сигналів у пакеті MATLAB на рівні, достатньому для практичного використання; провів моделювання і проаналізувати результати виконання у пакеті MATLAB.