

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**К. П. Вонсевич, М. О. Безуглий**

# **АНАЛІЗ ВИМІРЮВАЛЬНИХ СИГНАЛІВ КОМП'ЮТЕРНИЙ ПРАКТИКУМ**

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня магістра за освітньою програмою  
«Комп'ютерно-інтегровані системи та технології в приладобудуванні»  
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»*

Київ  
КПІ ім. Ігоря Сікорського  
2022

Рецензенти      *Защепкіна Н.М.*, д.т.н., професор, професор кафедри інформаційно-вимірювальних технологій, КПІ ім. Ігоря Сікорського  
*Захарченко О.С.*, старший викладач кафедри радіоінженерії, КПІ ім. Ігоря Сікорського

Відповідальний редактор      *Тимчик Г.С.*, д.т.н., професор

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 01.12.2022 р.)  
за поданням Вченої ради приладобудівного факультету (протокол № 12/22 від 28.11.2022 р.)*

Електронне мережне навчальне видання

*Вонсевич Костянтин Петрович*, канд. техн. наук  
*Безуглий Михайло Олександрович*, д-р. техн. наук, професор

# АНАЛІЗ ВИМІРЮВАЛЬНИХ СИГНАЛІВ КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Аналіз вимірювальних сигналів: Комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / К. П. Вонсевич, М. О. Безуглий ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 5,06 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 105 с.

У навчальному посібнику викладено теоретичні відомості та інструкції, необхідні для виконання практичних робіт з дисципліни «Аналіз вимірювальних сигналів». Наведено інформацію щодо основ аналізу в межах часової та частотної області, описано особливості розрахунку часових характеристик, методів нормалізації та розрахунку спектру сигналу. Розглянуто принципи функціонування та особливості застосування типових рекурсивних фільтрів, типові методи машинної класифікації, охарактеризовано основні види архітектури штучних нейронних мереж, що можуть бути використані для класифікації сигналів. Запропоновано приклади блоків коду для реалізації типових задач аналізу сигналів у середовищі Matlab.

Посібник призначений для здобуття ступеня магістра за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» і буде також корисним для закріплення академічних знань, набутих студентом впродовж теоретичного навчання, сприятиме розвитку прикладних навичок та умінь при обробці вимірювальних сигналів, вдосконаленню рішень у самостійних студентських проектах.

© К. П. Вонсевич, М. О. Безуглий, 2022  
© КПІ ім. Ігоря Сікорського, 2022

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	4
ВСТУП .....	5
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №1. Елементарний аналіз. Візуалізація та дискретизація сигналів.....	6
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №2. Часовий аналіз. Нормалізація і визначення характеристик сигналу.....	19
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №3. Фільтри сигналів. Застосування фільтрів у середовищі Matlab.....	31
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4. Спектральний аналіз. Застосування швидких перетворень Фур'є в Matlab.....	51
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №5. Машинне навчання. Основи проектування і тренування нейронної мережі прямого поширення.....	68
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №6. Класифікація сигналів за допомогою каскадної нейронної мережі.....	86
ДОДАТКИ. ДОДАТОК А. Вимоги до оформлення практичних робіт ....	102
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	104

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CFNN – каскадна нейронна мережа прямого поширення

DFT – дискретні перетворення Фур'є

HN – мережа Хопфілда

FFT – швидке перетворення Фур'є

FFNN – нейронна мережа прямого поширення

TDF – часові характеристики сигналу

АЧХ – амплітудно-частотна характеристика

АФ – активаційна функція

БД – база даних

ФНЧ – фільтр низьких частот

ФВЧ – фільтр високих частот

ШНМ – штучна нейронна мережа

## ВСТУП

Комп'ютерні практикуми, наведені у навчальному посібнику, містять інструкції та завдання до виконання шести практичних робіт у двох програмних середовищах: Microsoft Excel і Matlab. Теоретичний матеріал, представлений у кожному практикумі, є доповненням до інформації, що подається на лекційних заняттях з дисципліни «Аналіз вимірювальних сигналів», а його структура визначає змістовність практичної роботи. У практикумах подано теоретичні відомості, приклади підготовки вхідного набору інформації для подальшого аналізу, параметри робочих програм, приклади фрагментів коду для розрахунку характеристик сигналів, їх візуалізації та класифікації, окремі математичні викладки. Кожен практикум містить опис завдання для подальшого виконання студентом, а також набір персональних налаштувань і значень амплітуди вимірюваних сигналів для різних варіантів. Для виконання практикумів студенти повинні знати: базові принципи роботи у середовищах Microsoft Excel і Matlab, сутність окремих математичних і статистичних методів та обчислень.

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1

## ЕЛЕМЕНТАРНИЙ АНАЛІЗ. ВІЗУАЛІЗАЦІЯ ТА ДИСКРЕТИЗАЦІЯ СИГНАЛІВ

**Мета:** Ознайомитись з основами візуалізації сигналів у середовищі Microsoft Excel та особливостями підготовки вхідної інформації для її подальшого аналізу.

**Змістовність роботи:** основні відомості про сигнал та принципи дискретизації. Теорема Котельникова та частота Найквіста. Часові вікна. Використання функцій середовища Microsoft Excel для імпортування та візуалізації аналітичної інформації.

### 1.1. Теоретичні відомості

#### 1.1.1. Сигнал. Основні принципи дискретизації сигналів

При роботі з вимірювальною та цифровою технікою вхідні сигнали різної природи розглядаються як аналогові (неперервні) функції часу. Аналоговий сигнал є безперервною або кусково-неперервною функцією, аргумент якої також є безперервним (рис. 1.1). Однак, у практиці аналізу сигналів перед виокремленням їх корисної складової початкова аналогова функція повинна пройти декілька проміжних етапів перетворень, а розрахунок окремих параметрів сигналу у більшості випадків здійснюється для його дискретної форми представлення.

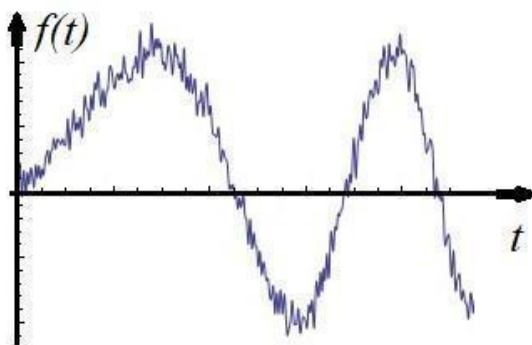


Рис. 1.1. Приклад аналогової форми представлення сигналу

Дискретний сигнал є кінцевим типом сигналу, який визначений лише у певних проміжках часу (певних значеннях аргументу) з деяким інтервалом. Більше того, при цифровій обробці сигнал представляється у вигляді послідовності чисел і є дискретним рядом значень (**discrete series**). Числа, що визначають значення сигналу в певний момент часу називаються відліками (**samples**). Як правило, для аналізу відліки беруться з певним інтервалом через деякі проміжки часу. Цей інтервал має назву періоду або кроку дискретизації (**sample time**) та позначається як  $T$ . Величина, що є зворотною до періоду дискретизації, називається частотою дискретизації (**sampling frequency**) і визначається за формулою (1.1):

$$f = \frac{1}{T} \quad . \quad (1.1)$$

Варто зазначити, що послідовність чисел, яка представляє сигнал при цифровій обробці (дискретний ряд) не повністю відповідає його аналоговій формі та залежить від частоти дискретизації. Для прикладу розглянемо дискретний сигнал (рис. 1.2).

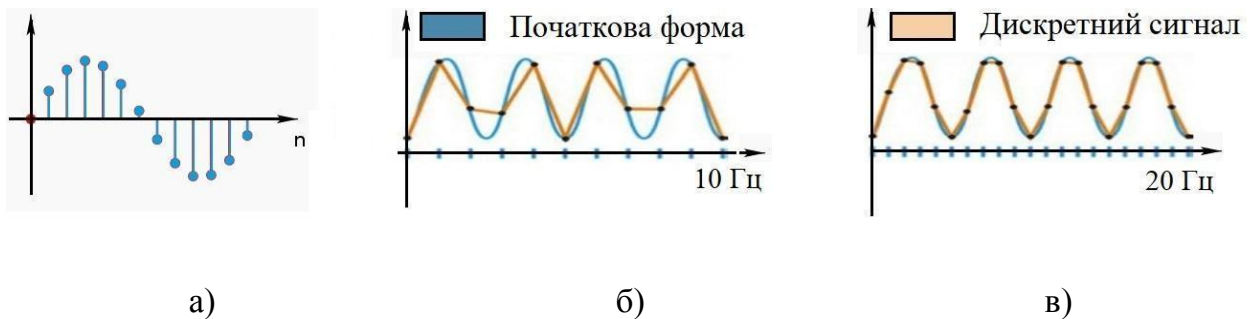


Рис. 1.2. Дискретний сигнал: приклад дискретної форми (а); візуалізація впливу частоти дискретизації 10 Гц (б); візуалізація впливу частоти дискретизації 20 Гц (в)

З рисунку видно, що при частоті дискретизації 20 Гц форма початкового сигналу відтворюється більш достовірно ніж при частоті 10 Гц, однак, перетворення аналогового сигналу в послідовність дискретних відліків все-одно призводить до втрати частини інформації, адже поведінка сигналу в проміжках між відліками є невідомою. Втрата інформації не відбувається лише при забезпеченні виконання умов **теорему Котельникова**, у відповідності до якої

аналогові сигнали можуть бути точно відновлені за значеннями своїх дискретних відліків.

Згідно з теоремою Котельникова гармонічний сигнал буде адекватно представлений дискретними відліками за умови, якщо його частота не перевищуватиме половини частоти дискретизації. Така частота має назву **частоти Найквіста** і розраховується за формулою (1.2):

$$f_N = \frac{1}{2T}. \quad (1.2)$$

Залежно від того, чи дотримується співвідношення між частотою Найквіста і частотою вхідного гармонічного сигналу при дискретизації, можливі три випадки вихідних результатів:

- Якщо частота вхідного сигналу буде меншою частоти Найквіста, то дискретні відліки можуть достовірно відтворити аналогову форму сигналу (рис. 1.3).

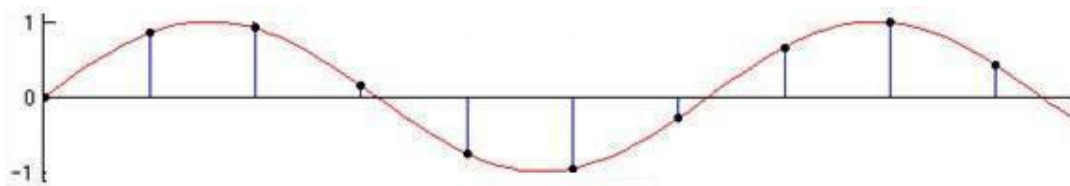


Рис. 1.3. До пояснення теореми Котельникова. Випадок перший

- Якщо частота вхідного сигналу дорівнюватиме частоті Найквіста, то дискретні відліки зможуть відтворити аналоговий гармонічний сигнал на тій же частоті, однак амплітуда і фаза відновленого сигналу (пунктирна лінія на рис. 1.4) можуть бути зміненими (в окремих випадках усі дискретні відліки синусоїди можуть навіть дорівнювати нулю).

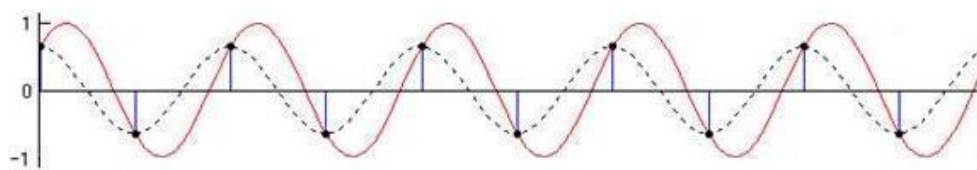


Рис. 1.4. До пояснення теореми Котельникова. Випадок другий



- Якщо частота вхідного сигналу буде більшою ніж частота Найквіста, то відновлений за дискретними відліками аналоговий сигнал також буде гармонічним, але його частота вже буде іншою (рис. 1.5). Такий ефект отримав назву **аліасинг** (поява хибних частот при відтворенні аналогового сигналу).

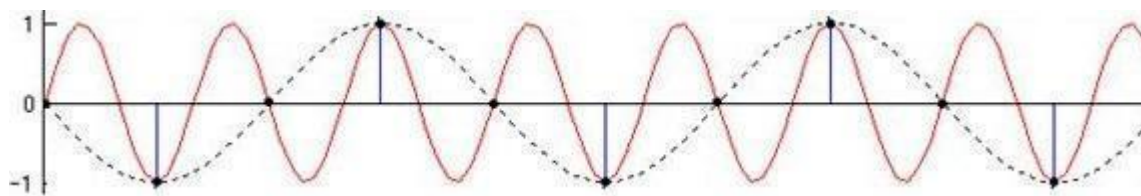


Рис. 1.5. До пояснення теореми Котельникова. Випадок третій

У переважній більшості випадків перед процесом визначення значущих параметрів сигналу здійснюється його попереднє підсилення та обробка за допомогою різноманітних фільтрів (низьких та високих частот, смугових фільтрів тощо), які в подальшому надсилаються на аналогово-цифровий перетворювач (АЦП), де здійснюється **дискретизація** і **квантування** сигналу за рівнем. Процеси квантування і дискретизації є незалежними один від одного, але зазвичай виконуються всередині однієї мікросхеми. Процес квантування передбачає перетворення деякої кількості дискретних відліків у цифрову форму, що має обмежений (кінцевий) набір значень. При квантуванні область зміни сигналу поділяється на визначені рівні з певним кроком (кроком квантування), а кожному рівню присвоюється деяке значення. Вхідний сигнал, який потребує квантування, у кожен момент часу порівнюється з рівнем квантування і на виході створюється нова форма вихідного сигналу зі значеннями, які були найближчими до цього рівня. У результаті на виході процесора деякої вимірювальної системи отримується набір чисел з відліками квантованого (оцифрованого) сигналу.

### 1.1.2. Візуалізація інформації у середовищі Microsoft Excel. Часові вікна

Для візуалізації вхідної інформації про сигнал та побудови графіків вхідних функцій може використовуватись різноманітне програмне забезпечення (ПЗ), наприклад: Microsoft Excel, Mathcad, Matlab тощо. Розглянемо приклад візуалізації сигналу та імпорту вхідних даних за допомогою Microsoft Excel. Імпортувати до робочого поля дані з текстового файлу (який містить, наприклад, значення з результатами вимірювання амплітуди сигналу електроміограми) можна за наступним алгоритмом:

- У робочому полі середовища Microsoft Excel створюємо новий бланк з назвою «Початкова форма сигналів».
- У довільний стовпець створеного бланку (наприклад, стовпець A на рис. 1.6) записуємо набір значень від 0 до 550 в порядку зростання та підписуємо створений набір значень як «Вісь X» у першому рядку стовпця.

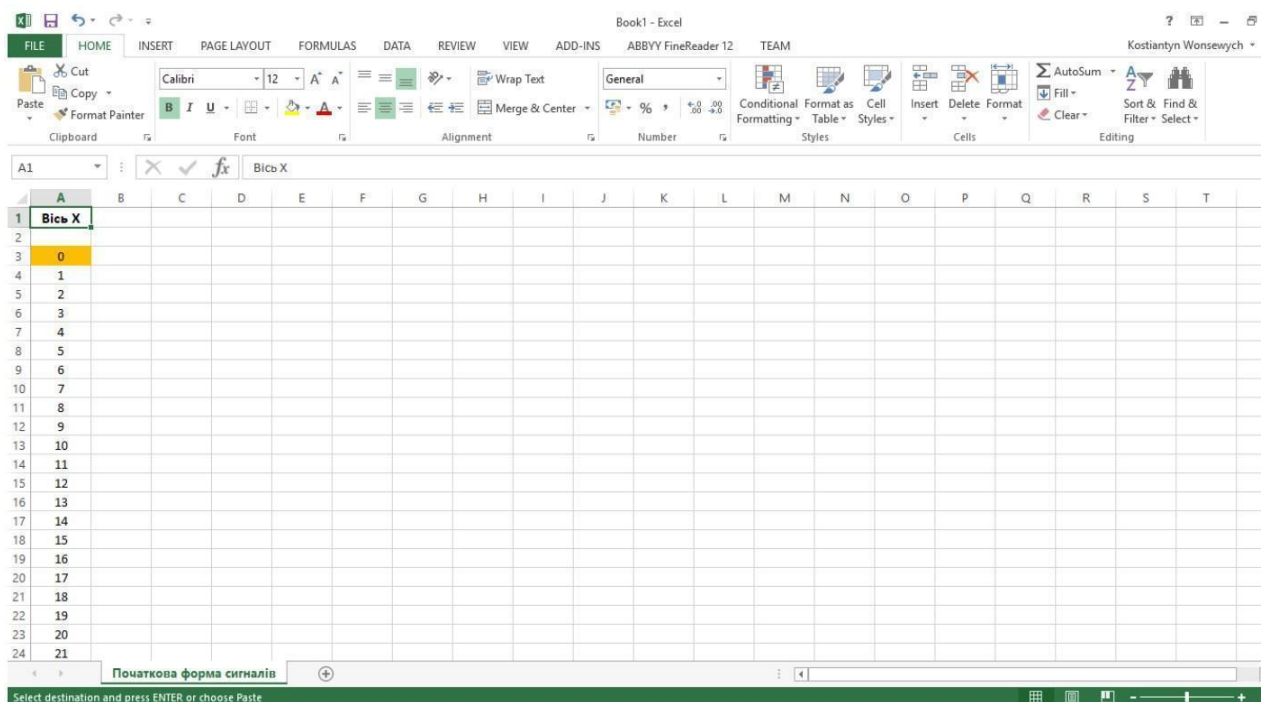


Рис. 1.6. Приклад додавання інформації до бланку у середовищі Microsoft Excel

- Імпортуємо інформацію з текстового файлу у довільний стовпець створеного бланку (наприклад, стовпець C) та задаємо у першому рядку його назву (наприклад «Сигнал\_1»). Для цього лівою кнопкою миші (ЛКМ) натискаємо на вкладку «DATA» → «From Text» → у вікні «Import text file» обираємо необхідний файл із вхідною інформацією → «Import» (Рис. 1.7).

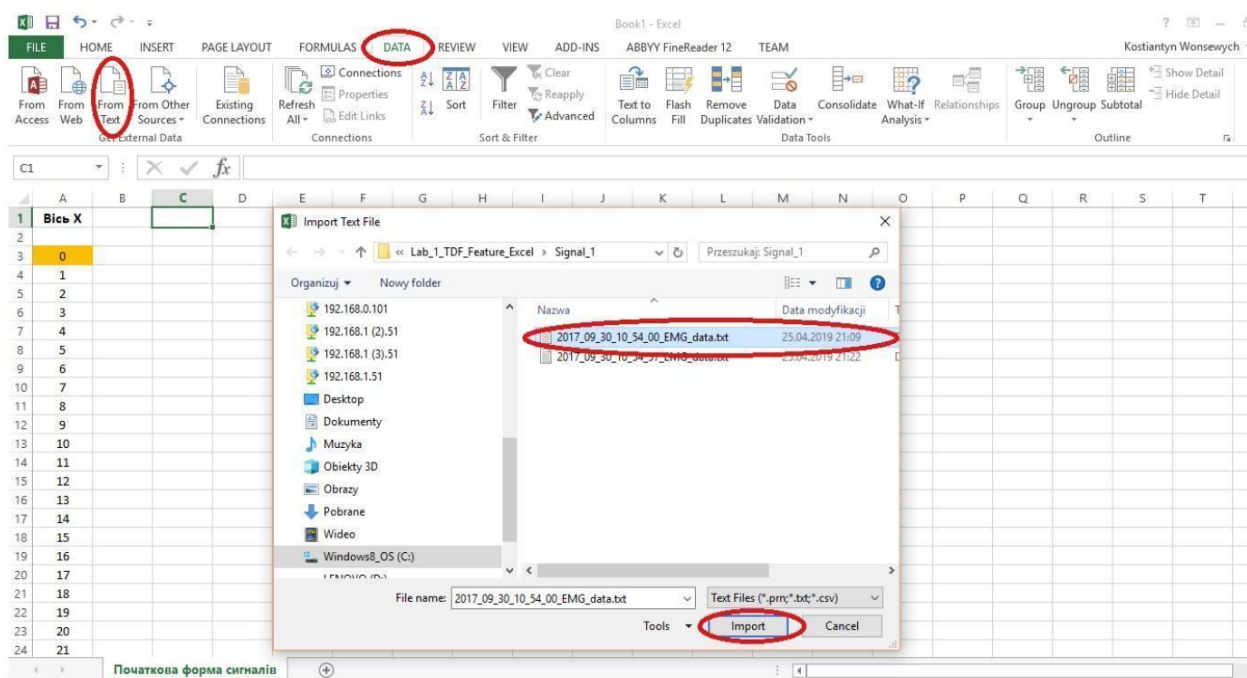


Рис. 1.7. Процес імпортування інформації з текстового файлу у середовищі Microsoft Excel

- У вікні «Text Import Wizard» (рис. 1.8) приймаємо запропоновані налаштування без внесення змін (натискаючи кнопки «Next», «Finish» та «Ok» відповідно).

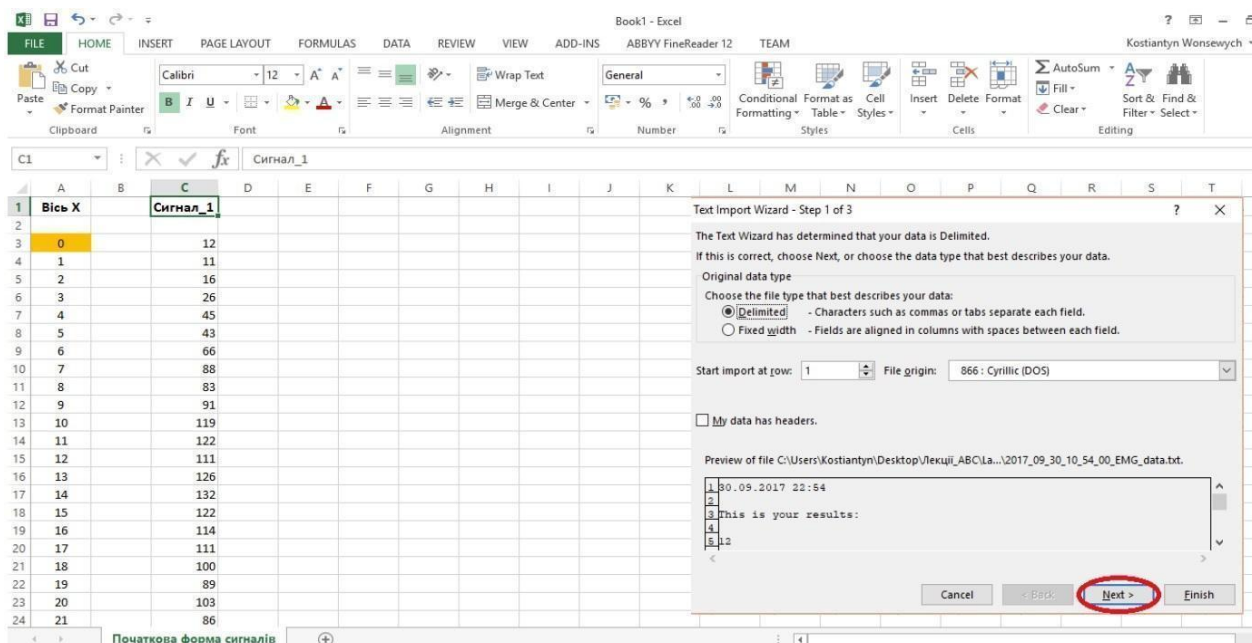


Рис. 1.8. Завершення налаштувань для імпортування інформації з текстового файлу у середовищі Microsoft Excel

- Для побудови графіку сигналу за інформацією, що була імпортована з текстового файлу, необхідно: обрати вкладку «Insert» → «Line Chart» → «Select Data» → «Add» → «Series values» та виділити необхідний масив значень амплітуди сигналу, який буде розміщено по осі «Y». Для того, щоб задати значення для осі «X», у вікні «Select Data Source» → «Horizontal (Category) Axis Labels» → «Edit» → виділити необхідні значення → «OK» (рис. 1.9).

Варто зазначити, що на практиці аналіз сигналу та визначення його характеристик часто здійснюються не для всієї довжини сигналу, а лише у межах так званих «часових вікон».

Часове вікно має чітко визначену тривалість та розподіляє один загальний сигнал на однакові інтервали фіксованої довжини, для яких здійснювати підрахунок параметрів зручніше.

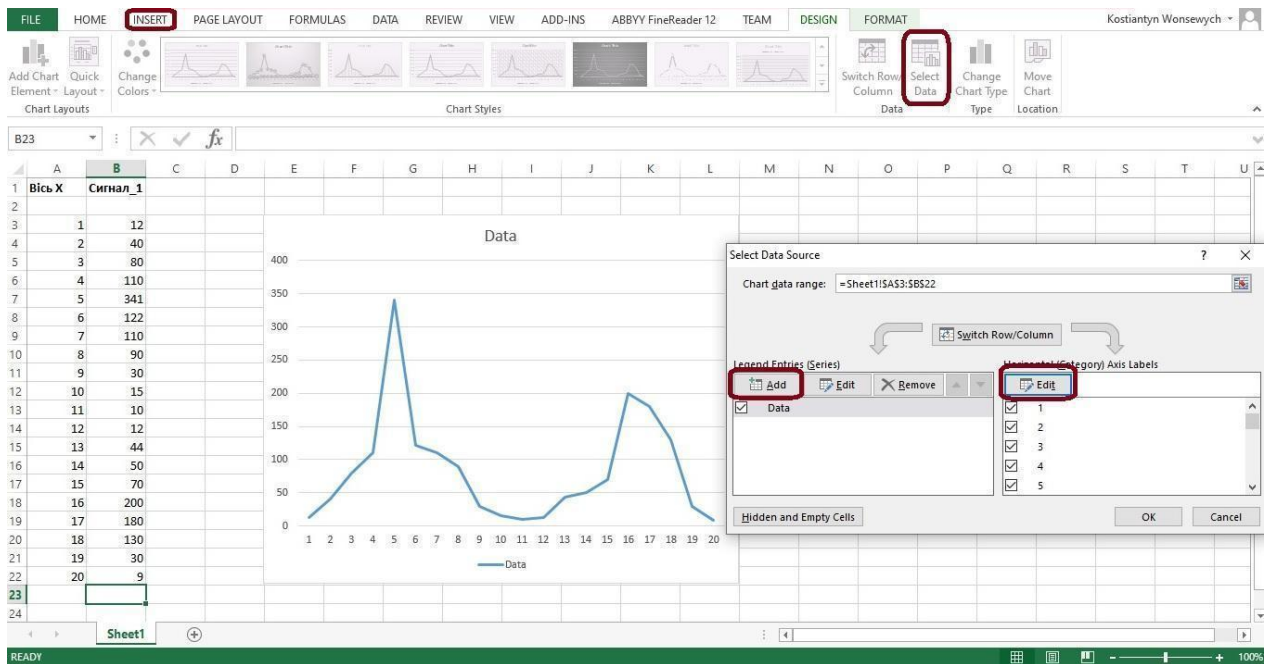


Рис. 1.9. Послідовність дій для побудови графіка сигналу в середовищі Microsoft Excel

Довжина часового вікна може варіюватися в залежності від виду сигналу та мети конкретного дослідження (наприклад, при аналізі сигналу електроміограми довжина часового вікна задається в межах від 30 до 500 мс).

## 1.2. Завдання для виконання

Створити робочу папку з назвою: **AVS\_PR1\_Шифр групи\_Прізвище студента\_№ варіанту** (PR1 розшифровується як: «Практична робота №1»). Назва папки задається латиницею (англомовними символами)!

**Завдання 1.1.** У робочій папці створити Excel-файл з назвою **AVS\_PR1\_№ варіанту**, що містить інформацію про початкову форму вхідних сигналів (значення амплітуди та їх відповідні графіки).

### Порядок виконання Завдання 1.1.

1. У робочому вікні середовища Microsoft Excel створити бланк з назвою «Початкова форма сигналів».
2. У довільному стовпці створеного бланку задати набір значень від 0 до 550 (в порядку зростання) та підписати створений набір значень як «Вісь X».

3. Імпортувати значення амплітуди сигналу з першого текстового файлу (який міститься у відповідній робочій папці за номером варіанту) та підписати його як «Сигнал\_1».

4. Повторити дії, описані у п.3, імпортувавши інформацію для решти текстових файлів, розміщених у робочій папці, та підписати відповідні стовпці як «Сигнал\_2», «Сигнал\_3» тощо.

5. Створити окремі графіки сигналів, у яких по осі «X» для кожного з графіків вказати значення зі стовпця «Вісь X» (від 0 до 550), а по осі «Y» – відповідні імпортовані значення амплітуди зі стовпців «Сигнал\_1», «Сигнал\_2» тощо (рис. 1.10). Кожному графіку повинна бути дана назва та підписана вісь «Y» (Amplitude, cV).

**Завдання 1.2.** У вже створеному Excel-файлі із назвою **AVS\_PR1\_№ варіанту** додати бланки для окремого аналізу кожного з вхідних сигналів у межах часових вікон та побудувати графіки їх ітерацій.

### Порядок виконання Завдання 1.2.

1. Створити додатковий бланк (нову вкладку) з назвою «TDF\_Sig\_1», для якого в довільних стовпцях задати набір значень для осі «X» та імпортувати амплітуду першого сигналу (див. п.2 та п.3 у завданні 1.1).

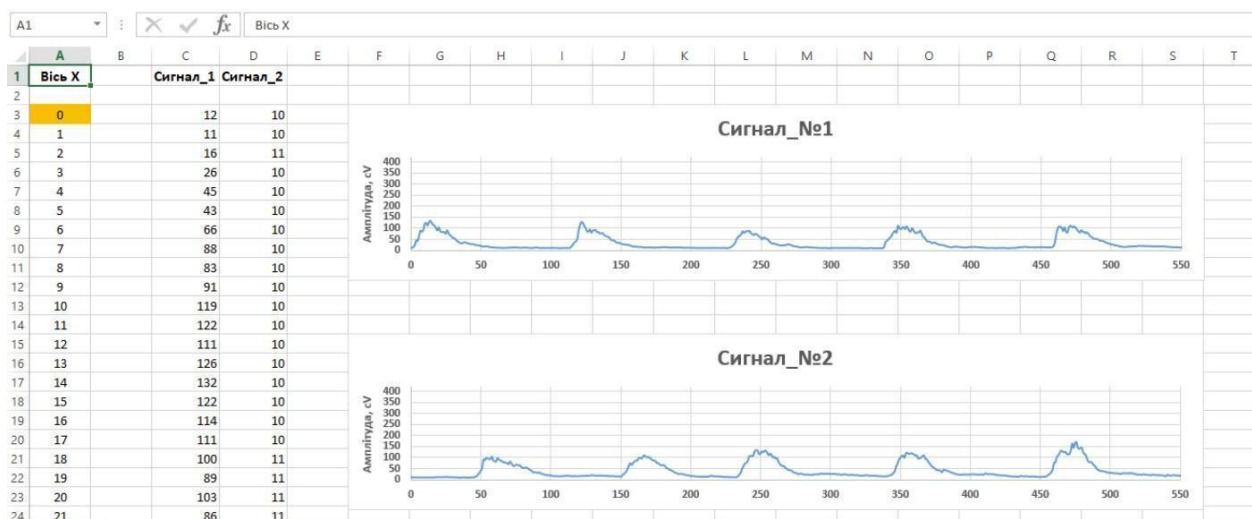


Рис. 1.10. Приклад побудови окремих графіків імпортованих значень сигналів



2. Розподілити імпортовані значення амплітуди сигналу в межах конкретних часових вікон у відповідності до індивідуального завдання (табл. 1.1) та побудувати графіки до кожної з ітерацій першого сигналу. Для цього: створити п'ять окремих графіків, що по осі «X» матимуть інтервал у  $N$  точок (представлений з кроком  $p$  на графіку), а по осі «Y» відповідні значення амплітуди для кожної окремої ітерації. Наприклад, якщо по завданню часове вікно має розмір у 100 мс, то для побудови графіку першої ітерації сигналу (першого часового вікна) по осі «X» має бути задано значення від 0 до 99, а по осі «Y» обрано перших 100 значень амплітуди (п.1 у завданні 1.2). Для другої ітерації (другого часового вікна): по осі «X» значення від 100 до 199, а по осі «Y» - обрано других 100 значень амплітуди тощо.

3. Для кожного зі створених графіків у межах часових вікон (рис. 1.11) вісь «Y» підписати як «Амплітуда, сВ» та задати назву графіку кожної ітерації.

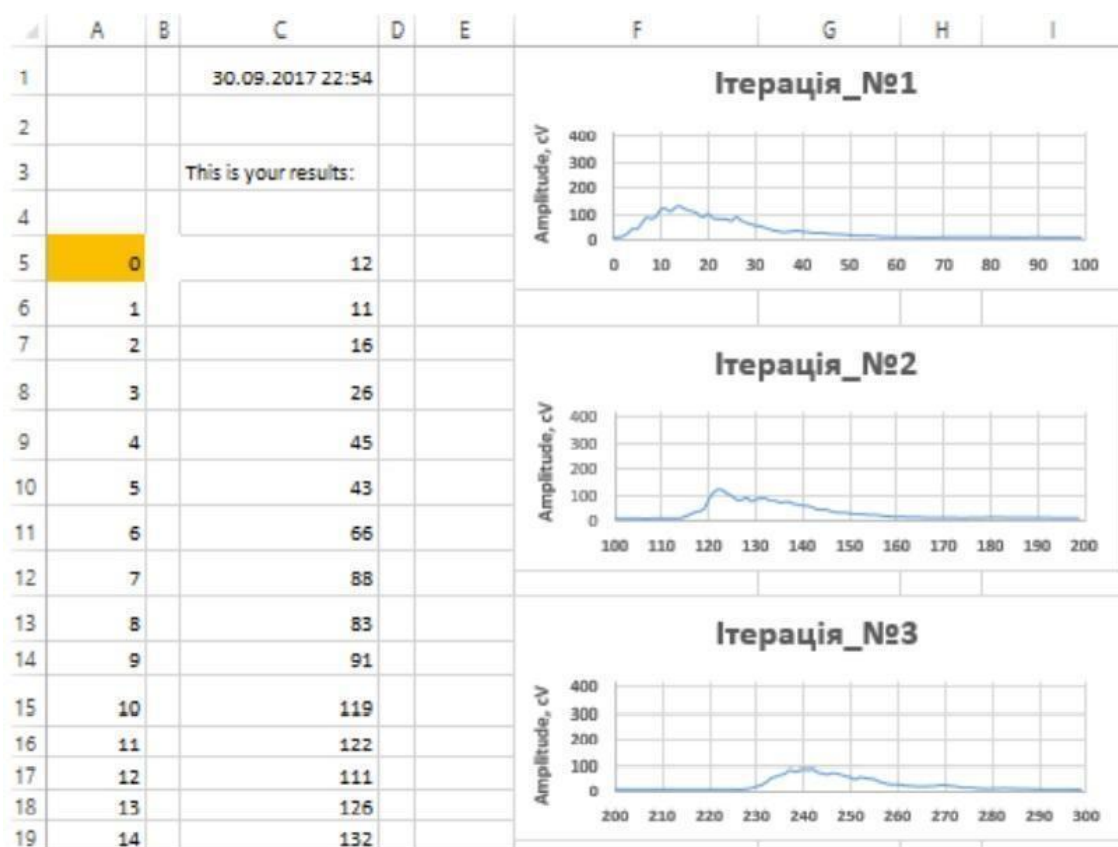


Рис. 1.11. Приклад побудови графіків окремих ітерацій сигналу в межах часових вікон

4. Створити бланки, розподілити значення амплітуди і побудувати графіки для решти сигналів, виконавши аналогічні дії у відповідності з алгоритмом, поданим у п.1 - п.3 у завданні 1.2.
5. Зберегти створені робочі файли та оформити звіт до практичної роботи.

### **1.3. Вимоги до оформлення звіту**

1. Звіт повинен бути представлений в електронному вигляді. Вихідний файл звіту повинен бути збережений у форматі «\*.docx» та «\*.pdf».
2. Назва електронного файлу зі звітом подається у форматі: «AVS\_PR1\_Шифр групи\_Прізвище студента\_№варіанту».
3. Структура звіту повинна містити наступні елементи:
  - титульний аркуш;
  - мету роботи;
  - номер варіанту та індивідуальне завдання (у відповідності до табл. 1.1);
  - покроковий опис виконаних дій з рисунками, таблицями тощо;
  - висновки по роботі.
4. Якість зображення та розмір шрифту на схемах, рисунках і таблицях повинні чітко відображати представлену в них інформацію. Усі рисунки, таблиці, схеми та розділи у звіті повинні бути підписані (вимоги див. у додатку А).
5. Файл звіту додається у загальну робочу папку (див. п. 1.2), яка надсилається викладачеві на перевірку (у «\*.zip» архіві). Для проходження перевірки робоча папка обов'язково повинна містити: робочий Excel-файл та звіт до практичної роботи у двох форматах.

**Якщо оформлення звіту, робочих файлів і папки не відповідають вимогам, практична робота до захисту не допускається!**



#### **1.4. Контрольні запитання**

1. Що таке сигнал?
2. У чому полягає різниця між дискретним і аналоговим сигналом?
3. Що таке крок та частота дискретизації?
4. На що впливає вибір частоти дискретизації сигналу?
5. У чому полягає суть теореми Котельникова?
6. Що таке частота Найквіста та для чого вона потрібна?
7. Що таке аліасинг?
8. Що таке квантування та крок квантування?

Таблиця 1.1. Початкові умови до завдання 1.2

№ варіанту	$N$ (мс)	$p$
1	110	5
2	100	2
3	105	1
4	110	4
5	100	10
6	105	5
7	110	1
8	100	5
9	105	10
10	110	2

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ №2

## ЧАСОВИЙ АНАЛІЗ. НОРМАЛІЗАЦІЯ І ВИЗНАЧЕННЯ ХАРАКТЕРИСТИК СИГНАЛУ

**Мета:** Ознайомитись з методами нормалізації значень та принципами визначення часових характеристик сигналу з використанням інструментів середовища Microsoft Excel.

**Змістовність роботи:** особливості розрахунку та огляд окремих вбудованих функцій Microsoft Excel для визначення основних часових характеристик. Основні методи та особливості нормалізації значень амплітуди сигналу.

### 2.1. Теоретичні відомості

#### 2.1.1. Особливості визначення характеристик сигналу. Часові характеристики

Для повноцінного аналізу сигналу самого лише набору числових значень та візуального оцінювання його графіку недостатньо. З метою порівняння декількох сигналів між собою та їх детального опису застосовують практику визначення набору значущих характеристик, які в подальшому аналізуються дослідником чи відповідними автоматизованими системами. Розрахунок таких характеристик може проводитись в межах часового, частотного або частотно-часового аналізу.

Категорія часових характеристик сигналу (TDF) не потребує складних математичних перетворень і розраховується для функцій у певній часовій області, де значення сигналу відоме для всіх дійсних чисел. Серед основних часових характеристик, які часто застосовуються і у статистиці, можна виділити: дисперсію функції, амплітуду Вільсона, середнє абсолютне значення, інтегроване абсолютне значення, середнє абсолютне значення нахилу кривої, максимальне значення амплітуди, кількість перетинів нульової лінії тощо. Розглянемо математичний опис окремих часових характеристик.

## Дисперсія:

$$VAR_i = \frac{1}{N} \sum_{k=1}^N (x_i(k) - \underline{x_i})^2. \quad (2.1)$$

Дисперсія показує наскільки поточне значення функції (амплітуди сигналу) відрізняється від її середнього значення. Наприклад, для електроміографічного сигналу, графік якого показаний на рис. 2.1, дисперсія розраховуватиметься як:

$$VAR_i = \frac{1}{N} \sum_{i=1}^N (U_i(t) - \underline{U_i})^2,$$

де

$$\underline{U_i} = \frac{1}{N} \sum_{i=1}^N |U_i(t)|.$$

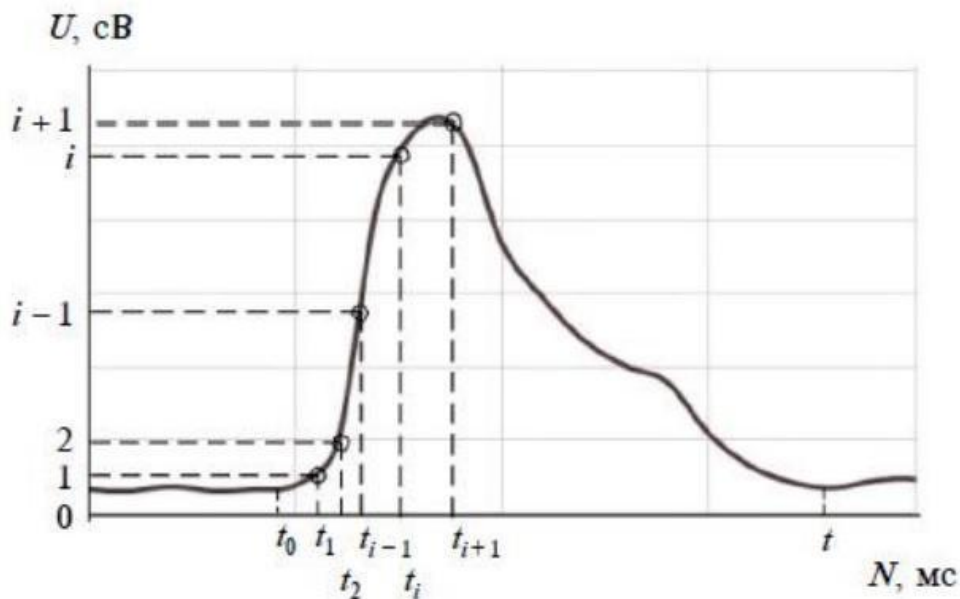


Рис. 2.1. До пояснення розрахунку дисперсії сигналу

Дисперсію сигналу на певному інтервалі значень досить зручно розраховувати шляхом застосування вбудованих функцій середовища Microsoft Excel. Зокрема, для розрахунку дисперсії в Excel передбачено функцію *VAR* та її окремі різновиди (*VAR.S*, *VARP*, *VAR.P*, *VARPA* тощо). Для вибору функції у

вкладці «Insert» обираємо пункт «Function» → «Statistic» та знаходимо необхідну назву функції. Також можна скористатись формою запису у комірці робочого поля виразу вигляду:

- $= VAR (number1, [number2], ...),$

де «number1, [number2], ...» є набором значень функції, для якого проводиться розрахунок.

**Середнє абсолютне відхилення ( $MAD$ ):**

$$MAD_i = \frac{1}{N} \sum_{i=1}^N |x_i - X|, \quad (2.2)$$

де

$$X = \frac{1}{N} \sum_{i=1}^N x_i.$$

Є мірою мінливості функції, що характеризує розсіювання значень деякої величини навколо центру розподілу. Чим більшим є значення середнього абсолютного відхилення, тим більшим є і розкид значень навколо умовного центру. Для вибору функції в Excel у вкладці «Insert» обираємо пункт «Function» → «Statistic» та знаходимо назву функції (AVEDEV). Також можна скористатись формою запису у комірці робочого поля виразу вигляду:

- $= AVEDEV (number1, [number2], ...),$

де «number1, [number2], ...» є набором значень функції, для якого проводиться розрахунок.

**Амплітуда Віллісона ( $WAMP$ ):**

$$WAMP_i = \sum_{k=1}^N f(|x_i(k) - x_i(k + 1)|), f(x) = \begin{cases} 1, \text{ якщо } x > \text{порогу} \\ 0, \text{ якщо } x < \text{порогу} \end{cases}. \quad (2.3)$$

Амплітуда Віллісона визначає кількість разів коли дві послідовні вибірки значень перевищують деякий заданий поріг.

**Довжина хвилі ( $WL$ ):**

$$WL_i = \sum_{k=1}^{N-1} (|x_i(k) - x_i(k + 1)|). \quad (2.4)$$

Параметр довжини хвилі є мірою сукупної зміни сигналу, що визначається для двох сусідніх зразків деякої функції.

**Середнє абсолютне значення (MAV):**

$$MAV_i = \frac{1}{N} \sum_{k=1}^N |x_i(k)|. \quad (2.5)$$

Для розрахунку середнього абсолютного значення в Excel, достатньо у комірці робочого поля записати вираз вигляду:

- $= AVERAGE (ABS(number1, [number2], ...))$ ,

де «number1, [number2], ...» є набором значень функції для якого проводиться розрахунок.

**Інтегроване абсолютне значення (IAV):**

$$IAV_i = MAV_i \cdot N. \quad (2.6)$$

**Максимальне значення функції (Peak):**

$$Peak_i = \max_{[a,b]} x_i. \quad (2.7)$$

Також у якості елементарної часової характеристики сигналу може використовуватись і розрахунок його максимального значення на певному часовому інтервалі, що визначається формулою (2.7), а в середовищі Excel у вкладці «Insert» обираємо пункт «Function» → «Statistic» та знаходимо назву функції (MAX). Або записуємо у комірці робочого поля вираз вигляду:

- $= MAX (number1, [number2], ...)$ ,

де «number1, [number2], ...» є набором значень функції, для якого визначається максимальне значення.

### **2.1.2. Нормалізація амплітуди сигналів**

Як згадувалось у п.п.1.1.2 аналіз та визначення характеристик часто здійснюються не для всієї тривалості сигналу, а у межах часових вікон певної

довжини. Крім того, для додаткової уніфікації перед етапом безпосередніх TDF розрахунків може проводитись **нормалізація** виміряних значень амплітуди сигналу. Такий процес дає змогу стандартизувати різнорівневі значення амплітуди сигналів однієї природи в межах деякої спільної шкали. Серед основних методів нормалізації можна виділити:

- масштабування на основі мінімальних і максимальних значень (Min-Max нормалізація).
- нормалізацію по стандартному відхиленню.
- Robust - нормалізацію.
- амплітудну нормалізацію по середньому або максимальному значенню.

**Min-Max нормалізація** є одним із найбільш поширених методів масштабування значень сигналу, у якому вихідний набір даних розраховується за формулою (2.8).

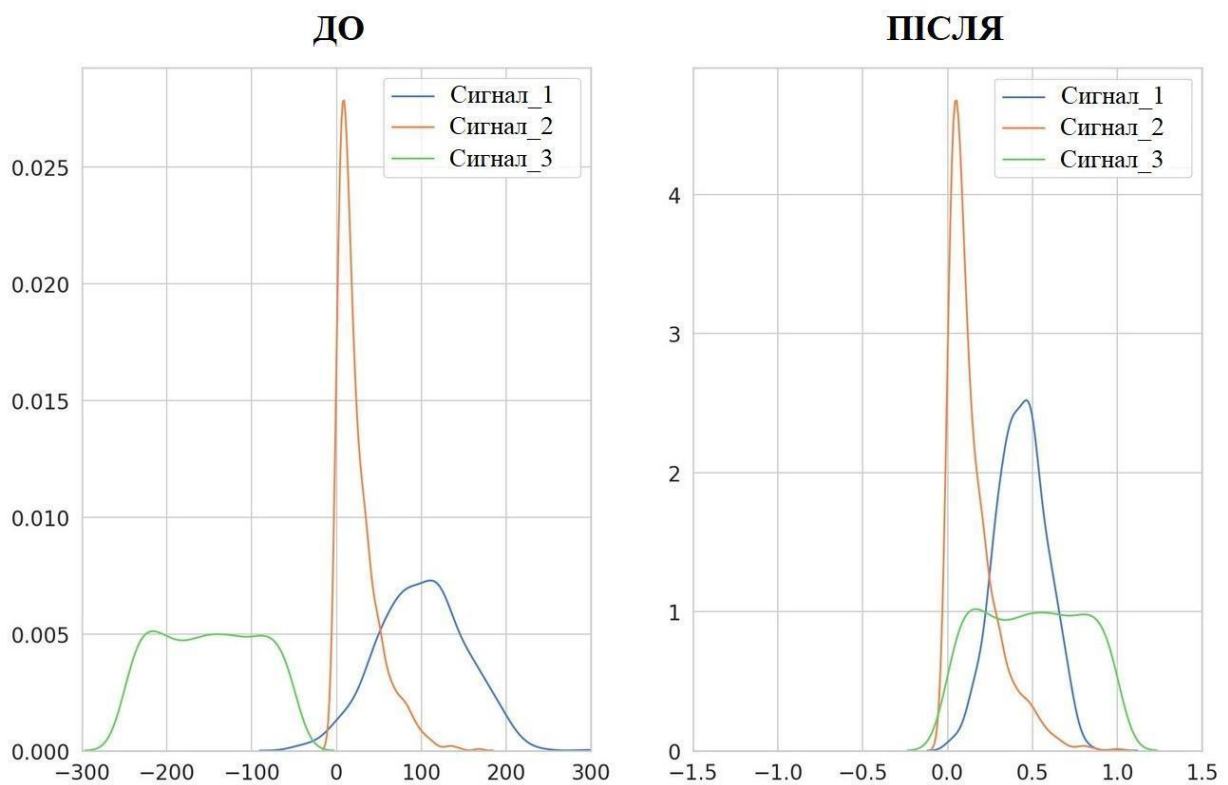


Рис. 2.2. Приклад набору сигналів до та після Min-Max нормалізації

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \cdot \quad (2.8)$$

Як видно з рис. 2.2 при Min-Max нормалізації зберігається форма вихідного розподілу але такий метод є чутливим до різких викидів.

**Нормалізація по стандартному відхиленню** (Z-score нормалізація) змінює масштаб вхідної функції, у процесі видаляючи середнє значення та ділячи вихідний результат на стандартне відхилення. Такий метод базується на створенні вихідного набору значень сигналу, що розподіляє значення з центром у нульовому відліку та стандартним відхиленням, що дорівнює одиниці (рис. 2.3). Формула для Z-score нормалізації визначається як (2.9)

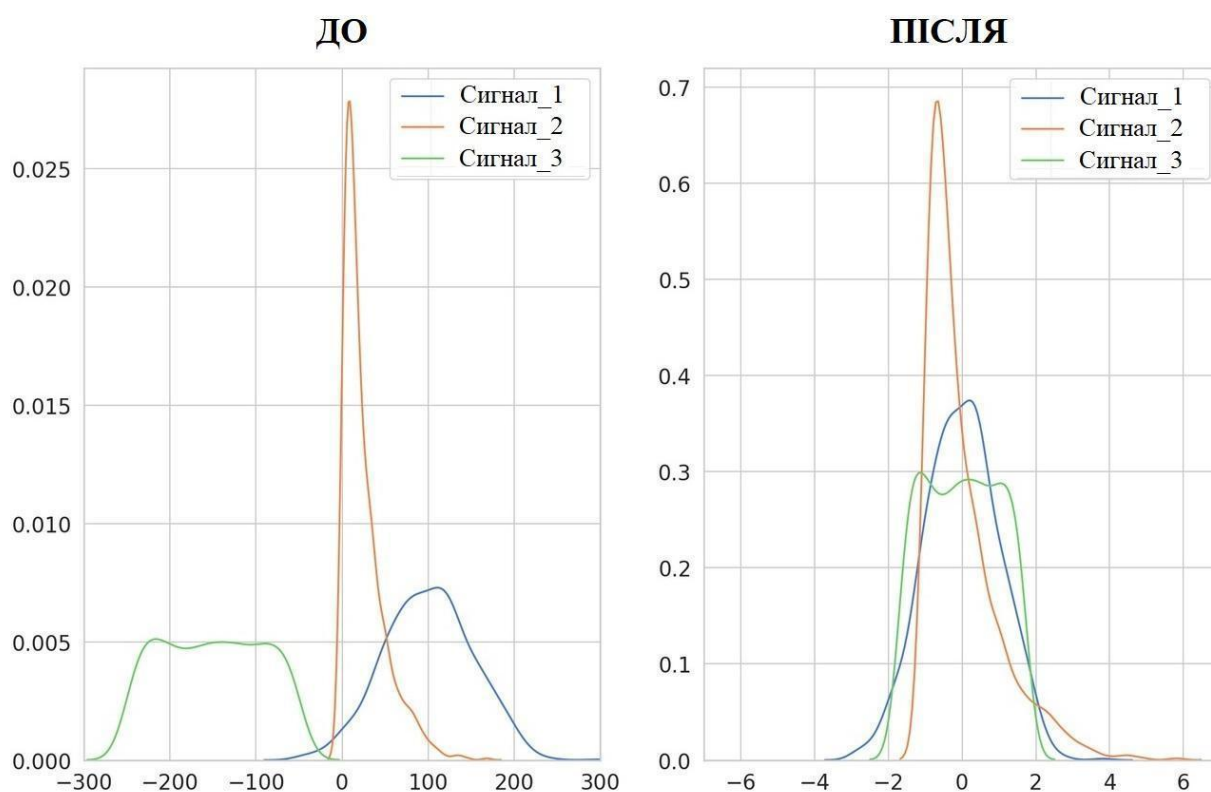


Рис. 2.3. Приклад набору сигналів до та після Z-score нормалізації

$$y = \frac{x - \bar{x}}{\sigma(x)}, \quad (2.9)$$

де  $\bar{x}$  – середнє значення функції, що визначається як  $\bar{x} = \frac{1}{N} \left( \sum_{i=1}^N x_i \right)$ ;

а  $\sigma(x)$  – стандартне відхилення, що розраховується за формулою (2.10):



$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \underline{x})^2}. \quad (2.10)$$

**Robust - нормалізація** є схожою на Min-Max метод нормалізації інформації, однак у такому підході кожна функція масштабується за допомогою кватилів та описується за допомогою формули (2.11):

$$X = \frac{x - Q_1(x)}{Q_3(x) - Q_1(x)}, \quad (2.11)$$

де  $Q$  – кватилі функції.

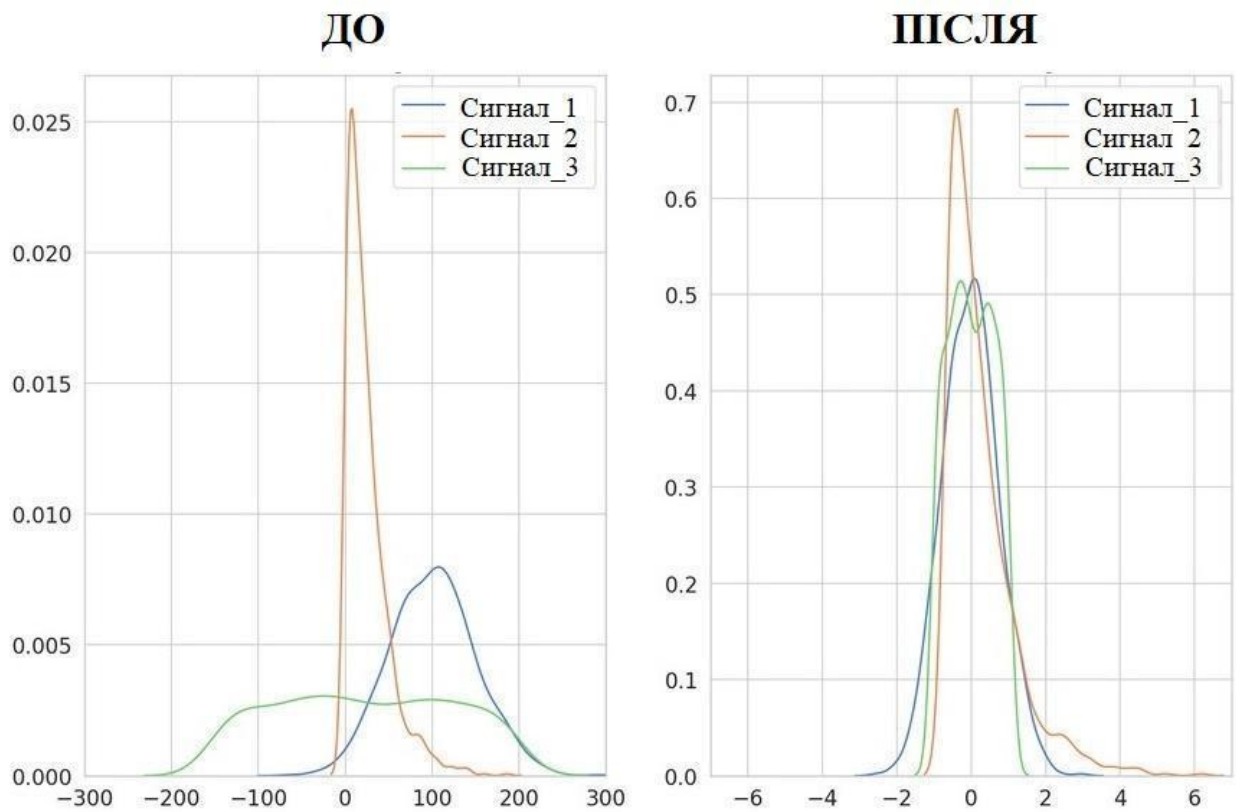


Рис. 2.4. Приклад набору сигналів до та після Robust - нормалізації

Як видно з рис. 2.4 Robust - нормалізація є більш стійкою до різких викидів сигналу і розподіляє більшу частину його щільності в межах нульового відліку, майже не змінюючи початкову форму кривої.

**Амплітудна нормалізація** по середньому або максимальному значенню є найбільш простим способом масштабування вхідної інформації про сигнал, що передбачає ділення поточного значення функції на її максимальне (або, в окремих випадках, мінімальне) значення (2.12):

$$y = \frac{x_i}{\max(x)} \cdot 100. \quad (2.12)$$

## 2.2. Завдання для виконання

Створити робочу папку з назвою: **AVS\_PR2\_Шифр групи\_Прізвище студента\_№ варіанту** (PR2 розшифровується як: «Практична робота №2»). Назва папки задається латиницею (англомовними символами)!

**Завдання 2.1.** У робочій папці створити Excel-файл з назвою **AVS\_PR2\_№ варіанту**, що містить інформацію про початкову форму вхідних сигналів і базується на практичній роботі №1.

### Порядок виконання Завдання 2.1.

1. У робочому Excel-файлі, який було створено у першій практичній роботі, відкрити бланк із назвою «TDF\_Sig\_1» та розрахувати набір часових характеристик сигналу у відповідності до індивідуального завдання (табл. 2.1). Розрахунок характеристик повинен здійснюватись окремо для кожної з ітерацій сигналу в межах її часового вікна (п.2 у завданні 1.2).

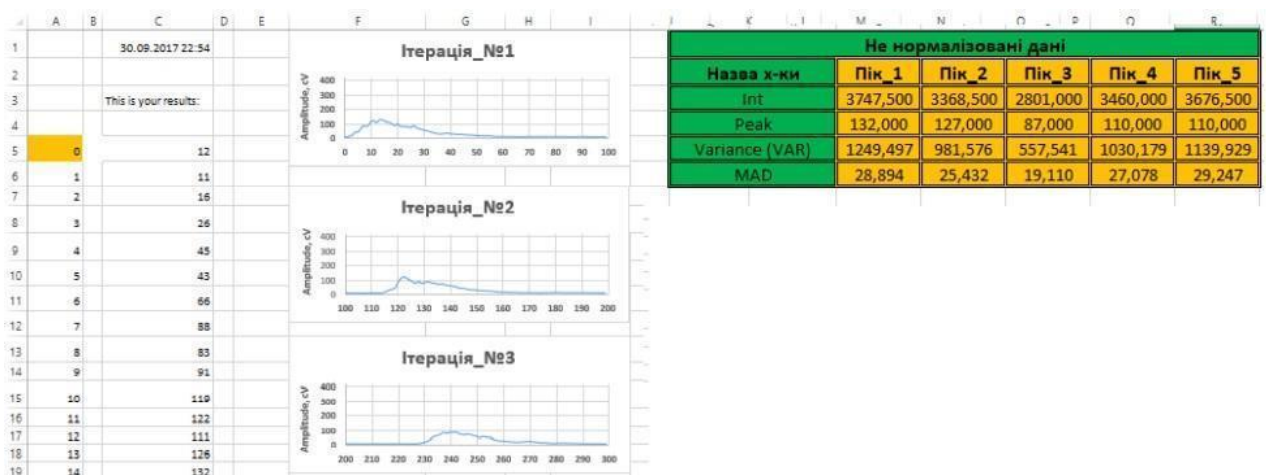


Рис. 2.5. Приклад оформлення таблиці з розрахованими характеристиками сигналу

2. Набір розрахованих характеристик для всіх ітерацій першого

сигналу подати у вигляді таблиці, зразок якої показано на рис. 2.5.

3. Відкрити бланк «TDF\_Sig\_2» та повторити дії, описані в п.1 та п.2 завдання 2.1. Розрахунок характеристик сигналів та оформлення результатів для кожного бланку (кожного сигналу в Excel-файлі) здійснюється аналогічно.

**Завдання 2.2.** У тому ж Excel-файлі (AVS\_PR2\_№ варіанту) провести нормалізацію значень амплітуди вхідного сигналу та розрахувати заданий набір характеристик для нормалізованих значень.

### Порядок виконання Завдання 2.2.

1. Відкрити бланк із назвою «TDF\_Sig\_1» та провести розрахунок нових значень амплітуди сигналу (згідно з методом нормалізації, вказаним у індивідуальному завданні, що подане в табл. 2.1). Нормалізовані значення амплітуди записати в окремому стовпці бланку.

2. Здійснити розрахунок набору часових характеристик для нормалізованого сигналу, аналогічно до завдання 2.1 (окремо для кожної з ітерацій в межах її часового вікна).

3. Набір розрахованих характеристик для всіх ітерацій нормалізованого сигналу також подати у вигляді таблиці, зразок якої показано на рис. 2.6

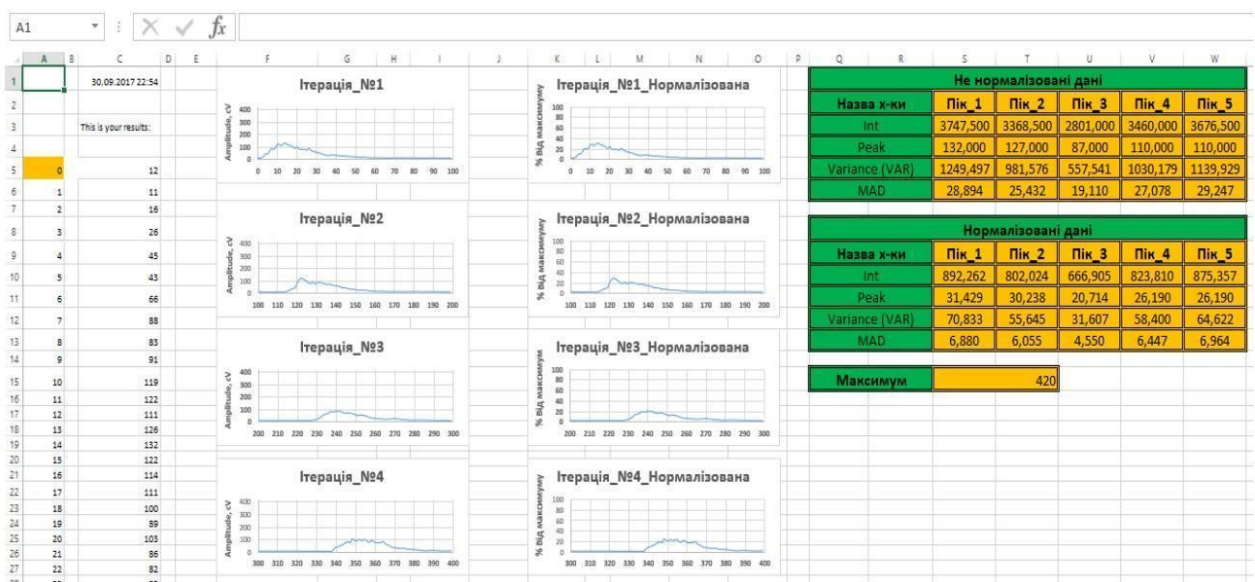


Рис. 2.6. Приклад оформлення таблиці з розрахованими характеристиками нормалізованого сигналу

4. Повторити дії описані у п.1 - п.3 для кожного з сигналів та бланків, що розміщені у робочому Excel-файлі.

5. Зберегти результати та оформити звіт до практичної роботи.

### **2.3. Вимоги до оформлення звіту**

1. Звіт повинен бути представлений в електронному вигляді. Вихідний файл звіту повинен бути збережений у форматі «**\*.docx**» та «**\*.pdf**».

2. Назва електронного файлу зі звітом подається у форматі: «AVS\_PR2\_Шифр групи\_Прізвище студента\_№варіанту».

3. Структура звіту повинна містити наступні елементи:

- титульний аркуш;
- мету роботи;
- номер варіанту та індивідуальне завдання (у відповідності до табл. 2.1);
- покроковий опис виконаних дій із рисунками, формулами, тощо;
- **таблиці з розрахованими наборами характеристик для нормалізованих і не нормалізованих сигналів;**
- висновки по роботі.

4. Якість зображення та розмір шрифту на схемах, рисунках і таблицях повинні чітко відображати представлену в них інформацію. Усі рисунки, таблиці, схеми та розділи у звіті повинні бути підписані (вимоги див. у додатку А).

5. Файл звіту додається у загальну робочу папку (див. п. 2.2), яка надсилається викладачеві на перевірку (у «\*.zip» архіві). Для проходження перевірки робоча папка обов'язково повинна містити: робочий Excel-файл та звіт до практичної роботи у двох форматах.

**Якщо оформлення звіту, робочих файлів і папки не відповідають вимогам, практична робота до захисту не допускається!**

## **2.4. Контрольні запитання**

1. Що таке характеристики сигналу? Для чого вони необхідні?
2. У межах яких видів аналізу сигналу зазвичай здійснюється розрахунок характеристик? Чим вони відрізняються?
3. Які часові характеристики сигналу ви знаєте?
4. Що таке нормалізація значень сигналу та для чого вона потрібна?
5. Які основні методи нормалізації ви знаєте? У чому їх особливість?
6. Опишіть суть методу Min-Max нормалізації.
7. Опишіть суть методу Z-score нормалізації.
8. Опишіть суть методу Robusy - нормалізації
9. У чому суть амплітудної нормалізації по максимальному значенню?

Таблиця 2.1. Початкові умови до завдання 2.2 та завдання 2.2

№ варіанту	$N$ (мс)	$p$	Набір TDF для розрахунку	Метод нормалізації
1	110	5	MAV, VAR, MAD	Min-Max
2	100	2	IAV, WL, MAD	Robust
3	105	1	MAD, Peak, VAR	Амплітудна по максимальному значенню $X_{max} = 420$
4	110	4	MAV, Peak MAD	Z-score
5	100	10	VAR, MAV, WL	Амплітудна по максимальному значенню $X_{max} = 500$
6	105	5	MAV, MAD, WL	Robust
7	110	1	VAR, MAV, IAV	Min-Max
8	100	5	MAD, IAV, Peak	Z-score
9	105	10	WL, Peak, MAV	Амплітудна по максимальному значенню $X_{max} = 450$
10	110	2	WL, VAR, Peak	Min-Max

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ №3

## ФІЛЬТРИ СИГНАЛІВ. ЗАСТОСУВАННЯ ФІЛЬТРІВ У СЕРЕДОВИЩІ MATLAB

**Мета:** Ознайомитись з поняттям фільтра сигналу, основними визначеннями та характеристиками типових фільтрів, а також особливостями їх застосування у середовищі Matlab.

**Змістовність роботи:** фільтр сигналу, основні визначення та поняття. ФНЧ, ФВЧ та смугові фільтри. Фільтри Чебишева, Баттерворта, еліптичний фільтр. Приклад запису параметрів фільтра в середовищі Matlab. Покроковий опис процесу створення і практичного застосування елементарних фільтрів до набору вхідних сигналів.

### 3.1. Теоретичні відомості

#### 3.1.1. Фільтри сигналів. Основні визначення та поняття

*Фільтр сигналу* – це електронний пристрій або набір математичних перетворень, що застосовується для виділення деякої частини спектру сигналу та пригнічення його небажаних складових. Ідеальний фільтр повинен виділяти чітко визначену смугу частот і повністю пригнічувати решту коливань, аж до повного їх загасання (рис. 3.1.а). Проте практична реалізація такого фільтра не можлива. Тому при проектуванні реальних фільтрів їх параметри підбирають таким чином аби АЧХ фільтра була максимально наближеною до ідеальної. Оцінювання якості фільтра зазвичай здійснюється за величиною нахилу кривої його АЧХ нижче *частоти зрізу* та вимірюється у «дБ / октаву» (рис. 3.1.б).

Для визначення частоти зрізу фільтру необхідно на графіку його АЧХ знайти рівень -3 дБ від максимального значення і точку перетину цього рівня з кривою АЧХ. Частота, що відповідає знайденій точці, і буде частотою зрізу фільтру. Наприклад, на рис. 3.2 показано, що точка на рівні -3 дБ, яка відповідає частоті зрізу фільтру, знаходиться на частоті 159 Гц. Що означає запис «дБ / октаву»? Враховуючи, що октава – це співвідношення частот два до одного, то нахил кривої АЧХ (наприклад 9 дБ / октаву) свідчить про те, що при збільшенні частоти у два рази крива АЧХ зростатиме або спадатиме на 9 дБ.

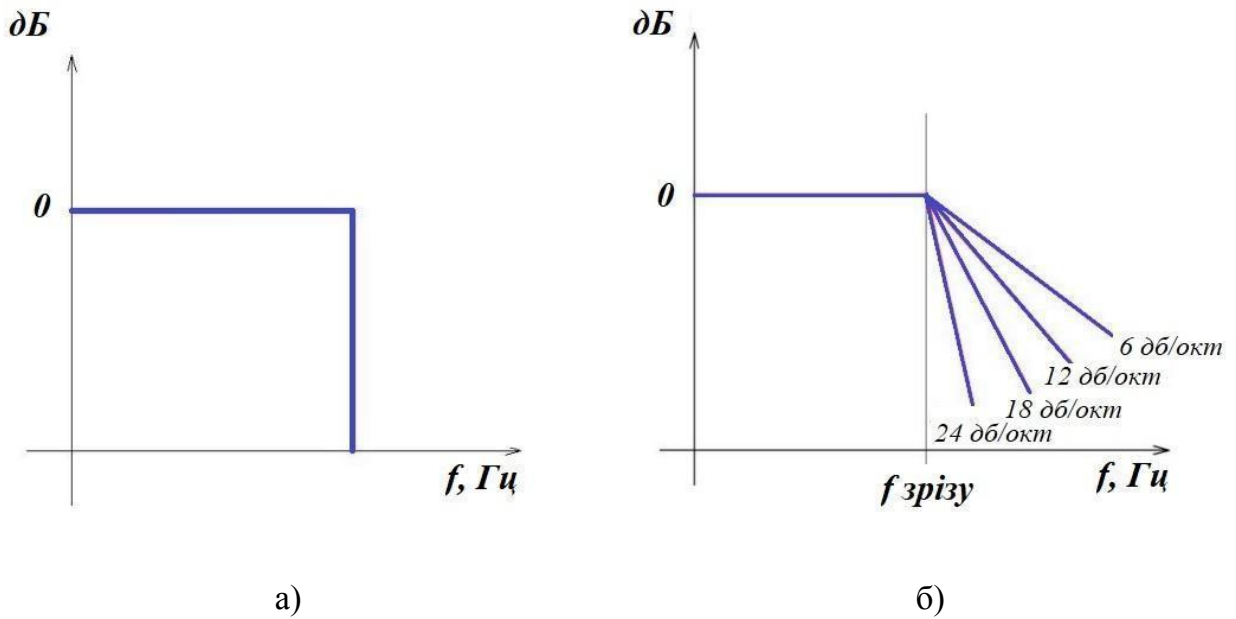


Рис. 3.1. До пояснення властивостей фільтру:  
АЧХ ідеального фільтру (а); нахили АЧХ реального фільтру (б)

Чим більшою є крутизна нахилу АЧХ, тим вищим буде його **порядок** і кращою буде «обираюча» здатність фільтру. За частотою пропускання фільтри поділяють на: фільтри низьких частот (ФНЧ), фільтри високих частот (ФВЧ), смугові (пропускаючі та загороджуючі) і фазові.



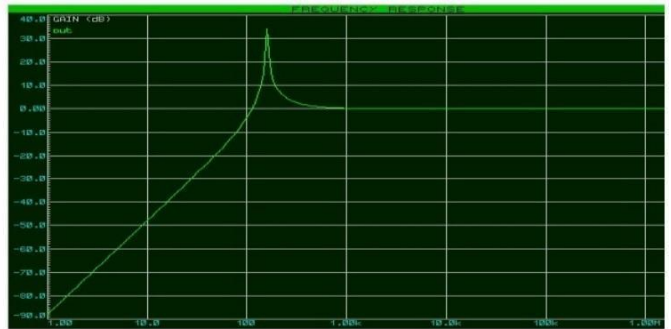
Рис. 3.2. До пояснення частоти зрізу фільтру



**ФНЧ-фільтр** пропускає низькочастотну складову вхідного сигналу, а при підвищенні його робочої частоти – пригнічує високочастотну складову (рис. 3.3.а). Водночас, **ФВЧ-фільтр**, навпаки, послаблює низькі частоти і пропускає високі (рис. 3.3.б).



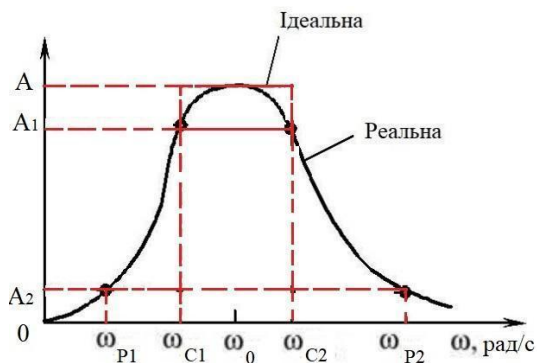
а)



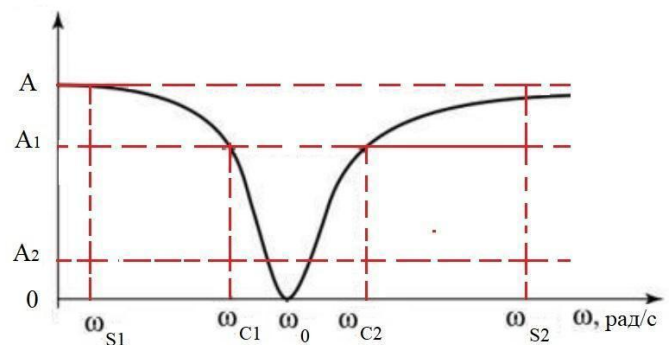
б)

Рис. 3.3. Амплітудно-частотні характеристики фільтрів:  
АЧХ фільтра низьких частот (а); АЧХ фільтра високих частот (б)

**Смуговий фільтр** пропускає вхідний сигнал лише у певній, визначеній смузі частот (наприклад, від 1 Гц до 10 Гц, від 30 Гц до 55 Гц тощо) та поділяється на смугові **пропускаючі** та смугові **загороджуючі** фільтри.



а)



б)

Рис. 3.4. АЧХ смугових фільтрів:  
АЧХ пропускаючого фільтра (а); АЧХ загороджуючого фільтра (б)

Конструктивно такі фільтри є послідовністю об'єднаних в одній системі ФВЧ і ФНЧ, а їх АЧХ характеризується наступними параметрами (рис. 3.4):

- нижня частота зрізу  $\omega_{c1}$ ;
- верхня частота зрізу  $\omega_{c2}$ ;
- верхня межа смуги пропускання  $\omega_{p1}$  (або смуги пригнічення  $\omega_{s1}$ );
- нижня межа смуги пропускання  $\omega_{p2}$  (або смуги пригнічення  $\omega_{s2}$ ).

Особливістю **фазового фільтра** є те, що він пропускає всі частоти вхідного сигналу з однаковим рівнем підсилення, але при цьому змінює його початкову фазу (рис. 3.5).

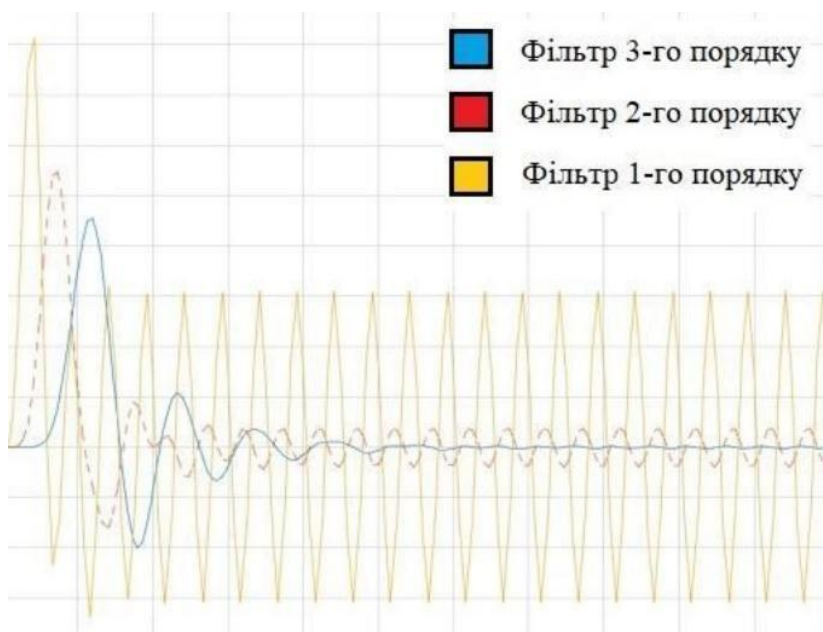


Рис. 3.5. Приклад застосування фазових фільтрів різних порядків до вхідного сигналу

Зазвичай фазові фільтри характеризуються лише одним параметром: граничною частотою, на якій фазовий зсув досягає показника 90°.

### 3.1.2. Фільтри Чебишева, Баттерворта та еліптичний фільтр. Проектування елементарного фільтра в Matlab

Серед основних рекурсивних фільтрів, що застосовуються в практиці аналізу сигналів, виділяють: фільтр Чебишева, фільтр Баттерворта та еліптичний фільтр.

**Фільтр Чебишева**, у порівнянні з іншими видами фільтрів, має досить крутий спад і суттєві пульсації амплітудно-частотної характеристики на частотах смуг пропускання і пригнічення (рис. 3.6). **Фільтр Чебишева I роду** має рівно-хвильові пульсації АЧХ в смузі пропускання, амплітуда яких визначається показником пульсації. Фільтр **Чебишева II роду** має менш крутий спад амплітудної характеристики та відсутність пульсації в смузі пропускання. Однак у смузі пригнічення такого фільтра пульсації присутні, що призводить до збільшення числа його складових компонентів в порівнянні з фільтром Чебишева I роду.

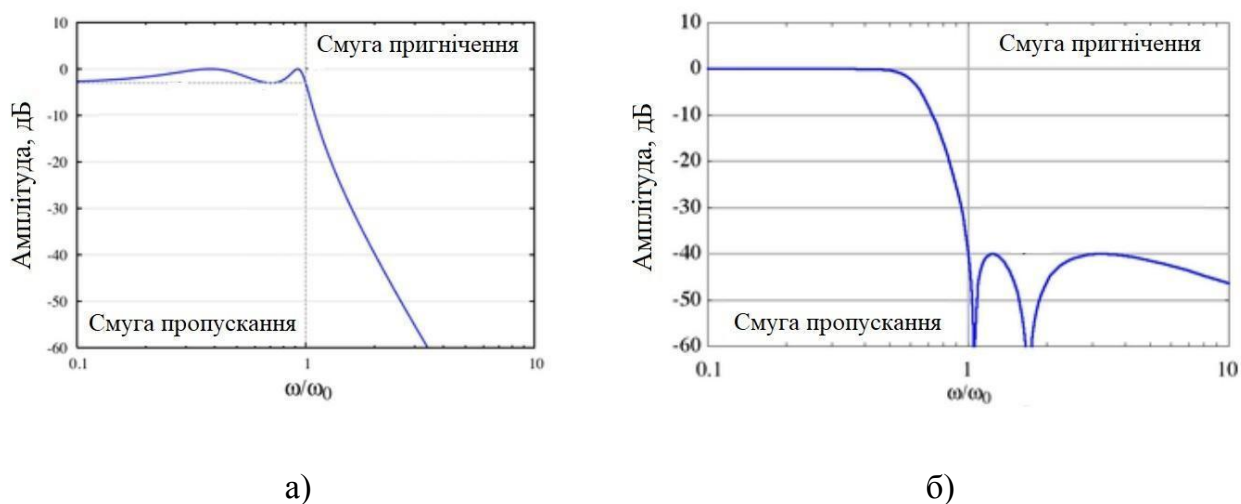


Рис. 3.6. Фільтр Чебишева:  
АЧХ фільтра Чебишева I роду (а); АЧХ фільтра Чебишева II роду (б)

Варто зазначити, що число рівновеликих пульсацій у смузі пропускання такого фільтра зростає разом з його порядком, а використання фільтра Чебишева є зручним у випадках, коли необхідно за допомогою фільтра низького порядку забезпечити сильне пригнічення окремих частот, а гладкість АЧХ (на частотах смуг пропускання і пригнічення) не є критичною.

**Фільтр Баттерворта** проектується таким чином, щоб його АЧХ була максимально гладкою на частотах смуги пропускання і зменшувалась майже до нуля на частотах смуги пригнічення. Вважається, що це єдиний з цифрових фільтрів, який зберігає стабільну форму АЧХ навіть для більш високих порядків. Частотна характеристика фільтра Баттерворта є максимально плоскою (без пульсацій), а його смуга пропускання проектується таким чином, щоб

смуга від 0 Гц до частоти зрізу -3 дБ не мала пульсацій. У фільтрі Баттерворта першого порядку АЧХ згасає з крутизною -20 Дб/дек, для другого порядку АЧХ загасає на -40 Дб/дек, для фільтра третього порядку на -60 дБ, тощо. Частотну характеристику фільтра Баттерворта і його стандартні апроксимації для різних порядків фільтра показано на рис. 3.7.

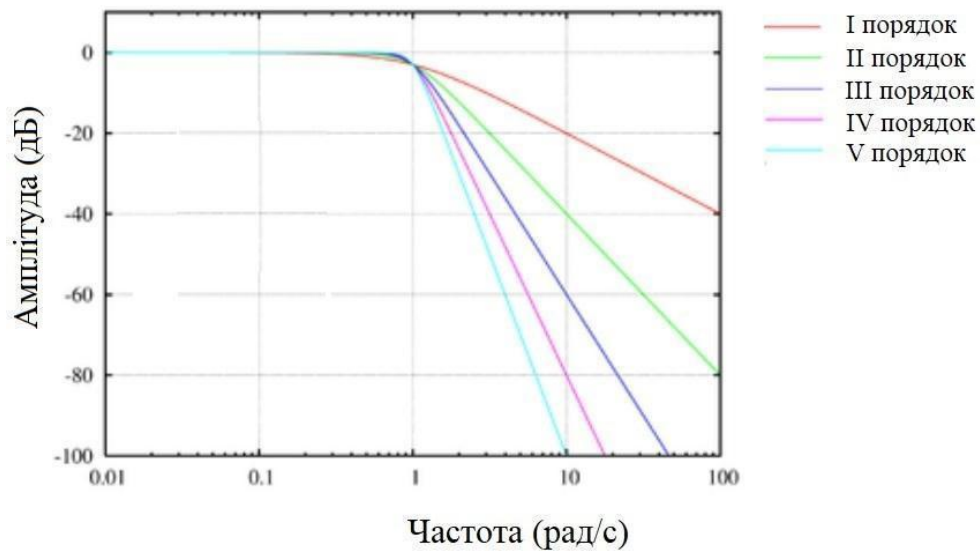


Рис. 3.7. Приклад ідеальної частотної характеристики різних порядків фільтра Баттерворта

Однак, варто зазначити, що фільтр Баттерворта досягає такої “площинності” смуги пропускання внаслідок широкої смуги переходу, а фазові характеристики фільтра є досить низькими.

**Еліптичний фільтр** має пульсації амплітудно-частотної характеристики як у смузі пропускання, так і у смузі пригнічення, а величина пульсацій в кожній зі смуг є незалежною одна від одної (рис. 3.8). Характерною особливістю такого фільтра є дуже крутий спад його амплітудної характеристики (що часто застосовується розробниками у якості інструмента для більш ефективного поділу частот). Еліптичні фільтри об'єднують в собі властивості фільтрів Чебишева (I та II роду) і дозволяють отримати високу крутизну АЧХ фільтра.

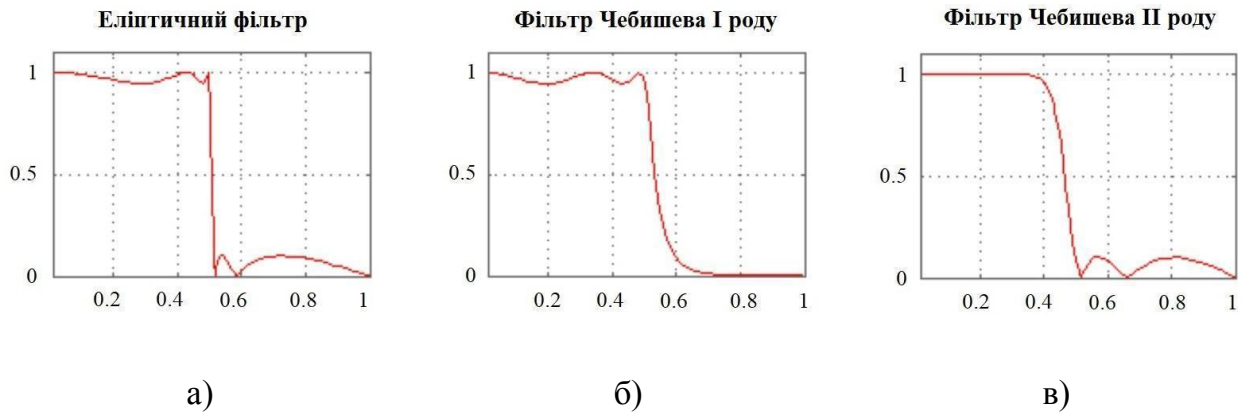


Рис. 3.7. Амплітудно-частотні характеристики: еліптичного фільтра (а); фільтра Чебишева I роду (б); фільтра Чебишева II роду (в)

Якщо пульсації в смузі пригнічення такого фільтра дорівнюють нулю, то еліптичний фільтр перетворюється у фільтр Чебишева I роду. Якщо пульсації дорівнюють нулю в смузі пропускання, то фільтр стає фільтром Чебишева II роду. Якщо ж пульсації відсутні по всій амплітудній характеристиці, то еліптичний фільтр за своїм функціоналом перетворюється на фільтр Баттерворта.

Можливості програмного середовища Matlab дозволяють здійснювати проектування різноманітних фільтрів сигналу як за допомогою вбудованих модулів програми, так і «вручну», шляхом написання відповідного програмного коду. Розглянемо приклад написання коду та визначення робочих параметрів для чотирьох видів ФНЧ фільтрів.

- ФНЧ фільтр Баттерворта:

```
[b a] = butter(1, 0.55, 'low');
```

Тут **butter** - назва фільтра, **1** - його порядок, **0.55** - частота зрізу, **low** - тип фільтра (ФНЧ)

- Еліптичний ФНЧ фільтр:

```
[b a] = ellip(1, 5, 30, 0.55, 'low');
```

Тут **ellip** - назва фільтра, **1** - його порядок, **5** - початкова частота, **30** - кінцева частота, **0.55** - частота зрізу, **low** - тип фільтра (ФНЧ)

- ФНЧ фільтр Чебишева I роду:

```
[b a] = cheby1(1, 20, 0.55, 'low');
```

Тут **cheby1** - назва фільтра, **1** - його порядок, **20** - кінцева частота, **0.55** - частота зрізу, **low** - тип фільтра (ФНЧ)

- ФНЧ фільтр Чебишева II роду:

```
[b a] = cheby2(1, 25, 0.55, 'low');
```

Тут **cheby2** - назва фільтру, **1** - його порядок, **25** - кінцева частота, **0.55** - частота зрізу, **low** - тип фільтру (ФНЧ)

Варто зазначити, що показаний приклад коду не є єдиною можливою формою запису коду для проектування фільтрів сигналу в Matlab. Однак, такої форми запису в загальному коді достатньо для створення елементарного фільтра та подальшого базового аналізу його впливу на зміну початкової форми сигналу.

### 3.2. Завдання для виконання

Виконання третьої практичної роботи базується на результатах, що визначені студентом у першій та другій практичних роботах. На відміну від попередніх завдань третя і всі наступні практичні роботи виконуються у середовищі Matlab. Для цього, необхідно створити робочу папку з назвою: **AVS\_PR3\_Шифр групи\_Прізвище студента\_№варіанту** (PR3 розшифровується як: «Практична робота №3»). **Назва папки задається латиницею (англомовними символами)!**

**Завдання 3.1.** Створити робочий код, що задаватиме алгоритм побудови фільтра *першого порядку* (у відповідності до індивідуального завдання, розміщеного в табл. 3.2) та графіків *однієї ітерації сигналу*, до та після застосування фільтра. Визначити набір нових значень амплітуди для відфільтрованого сигналу та коефіцієнти фільтра, який було застосовано при обробці вхідного сигналу.

#### Порядок виконання Завдання 3.1.

1. У робочому вікні середовища Matlab вибрати робочу папку, у якій в подальшому будуть зберігатись створені файли (рис. 3.8.)
2. Створити нову базу даних (БД), натиснувши правою клав'шею миші (ПКМ) у вікні «Workspace» та обравши пункт «New» (рис. 3.9). Задати назву файлу для БД як «Raw\_Signal\_1».



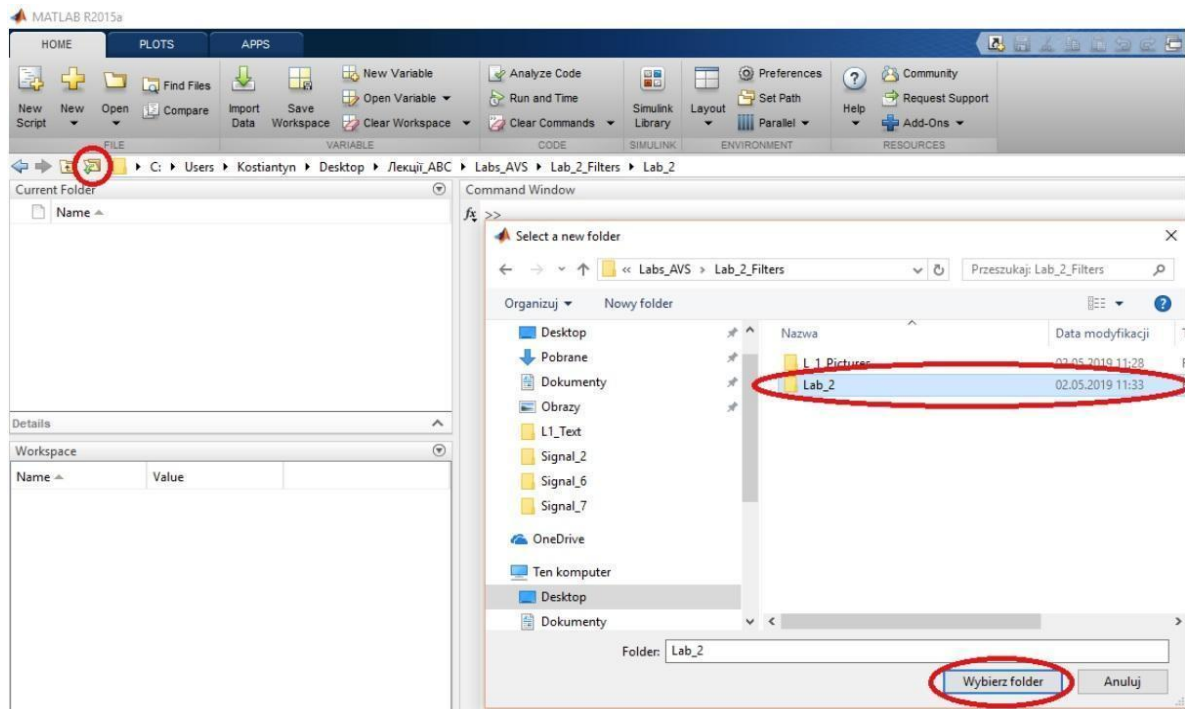


Рис. 3.8. Вибір робочої папки у середовищі Matlab

Відкривши створений файл, розмістити в ньому нормалізовані значення амплітуди для першого сигналу (див. завдання 2.2). Значення амплітуди для кожної ітерації сигналу повинні бути попередньо *транспоновані* (зі стовпця в рядок) та розміщені по порядку (кожна ітерація в окремому рядку).

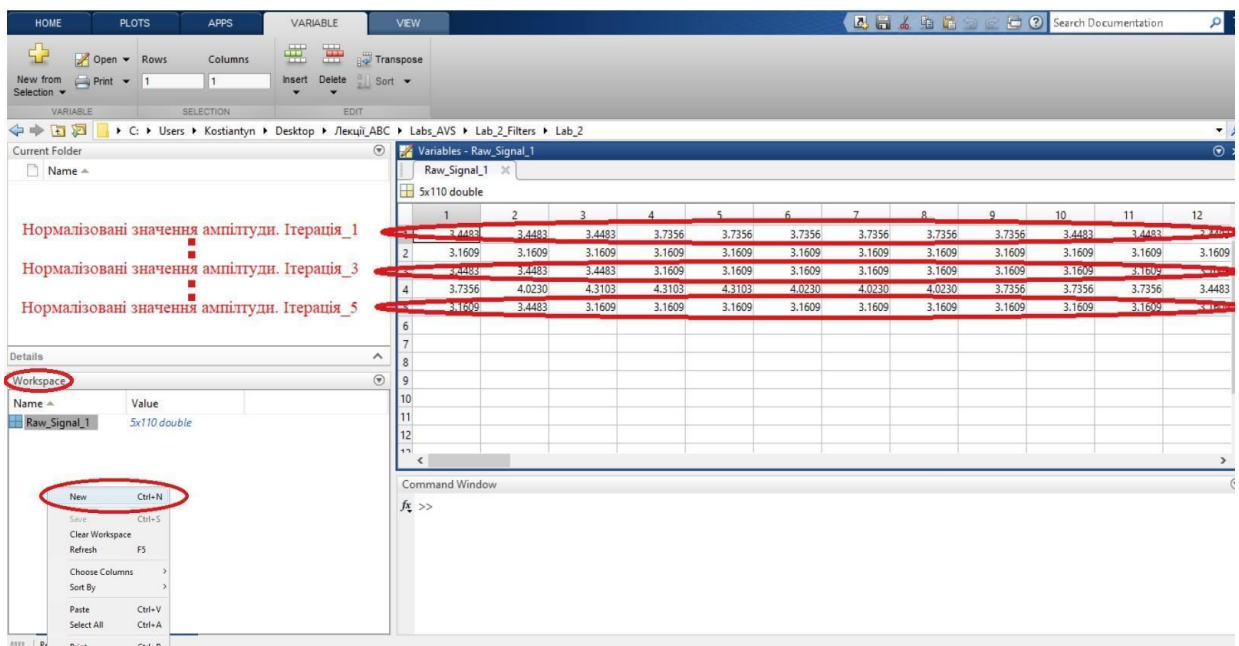


Рис. 3.9. Послідовність створення бази даних і заповнення її вмісту в Matlab

3. Повторити дії описані в попередньому пункті, створивши файл «Raw\_Signal\_2» та заповнивши його нормалізованими значеннями амплітуди для другого сигналу. Алгоритм дій для значень кожного наступного сигналу буде аналогічним.

4. Додати до створеної БД новий файл з назвою «t» і розмістити в ньому набір значень від 0 до  $N$  (значення, які в подальшому визначатимуть вісь «X» на графіках функції) у порядку зростання (рис. 3.10).

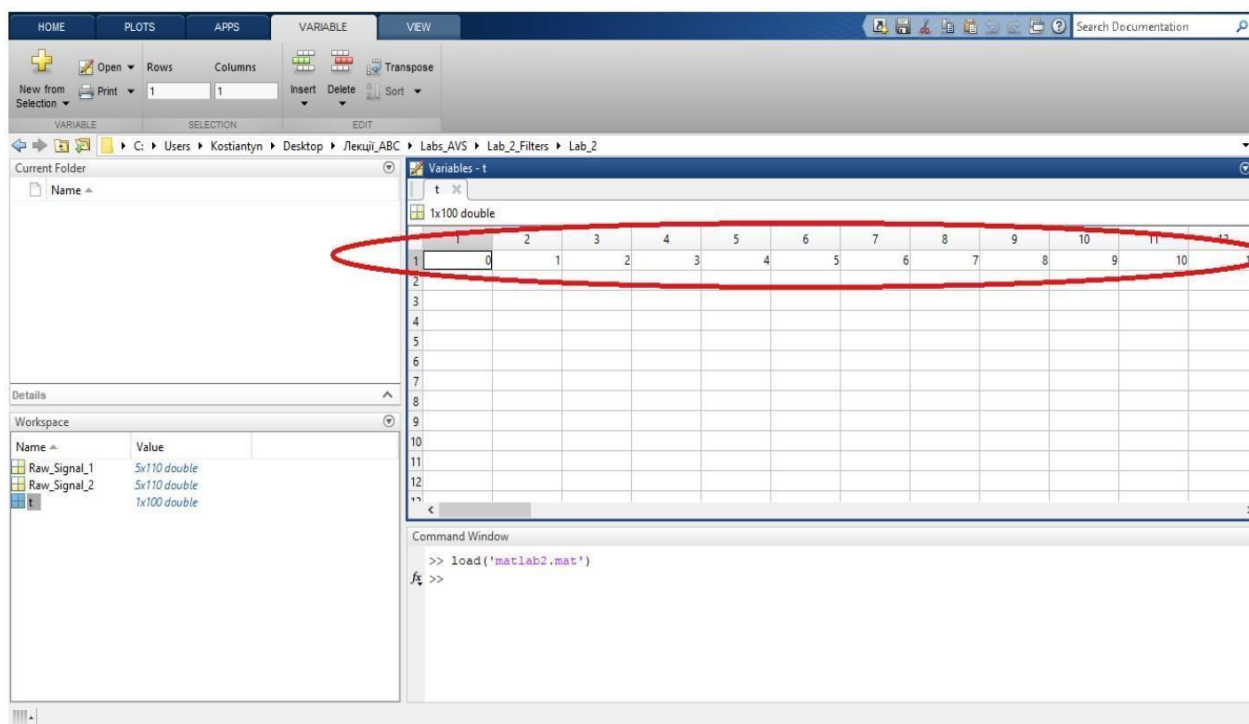


Рис. 3.10. Послідовність створення бази даних і заповнення її вмісту в Matlab

5. Закрити всі створені вкладки та натиснувши на полі «Workspace» ПКМ → «Save», зберегти створену БД з назвою «Data» у попередньо створену робочу папку.

6. Створити новий файл типу «Script» і прописати в ньому робочий код, що визначатиме алгоритм побудови фільтрів і графіків сигналу на основі інформації, що розміщена у БД. Для цього на верхній панелі «Home» обрати пункт «New» → «Script». У новому вікні «Untitled» записати код, показаний на рис. 3.11. та рис. 3.12.



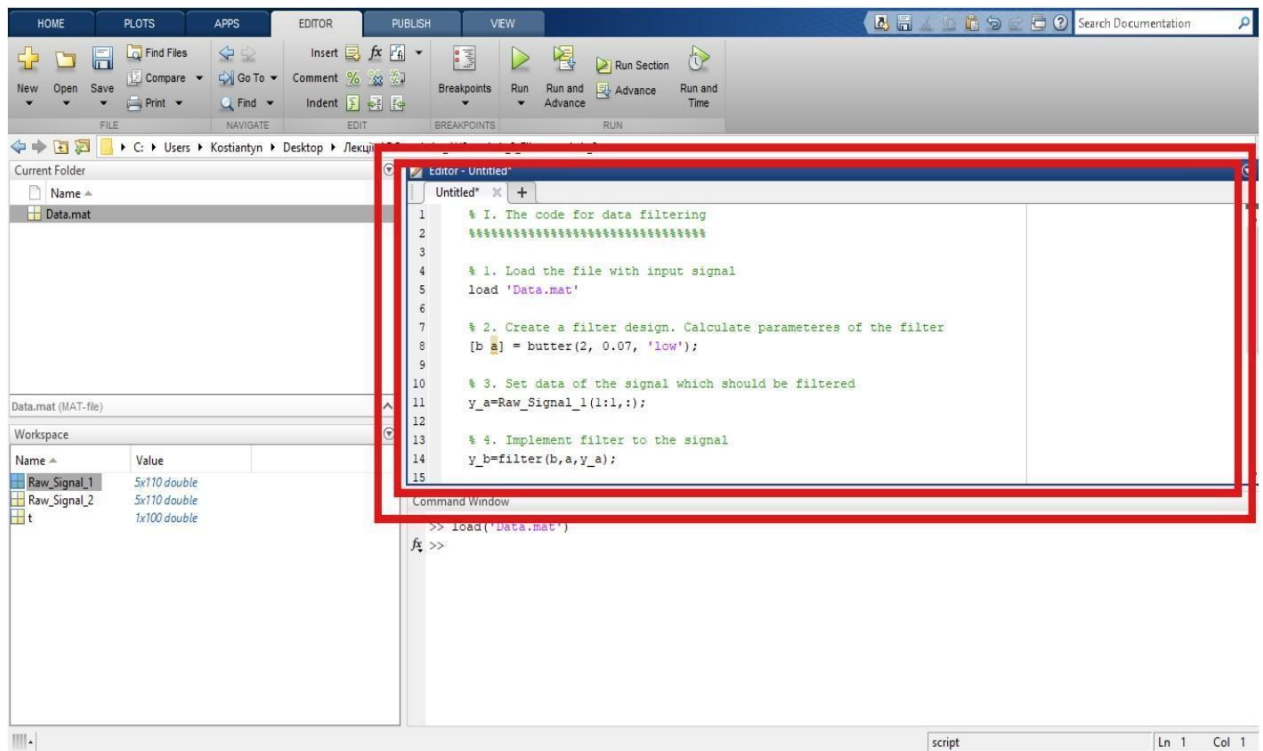


Рис. 3.11. Приклад робочого коду для застосування фільтра Баттерворта до першого сигналу

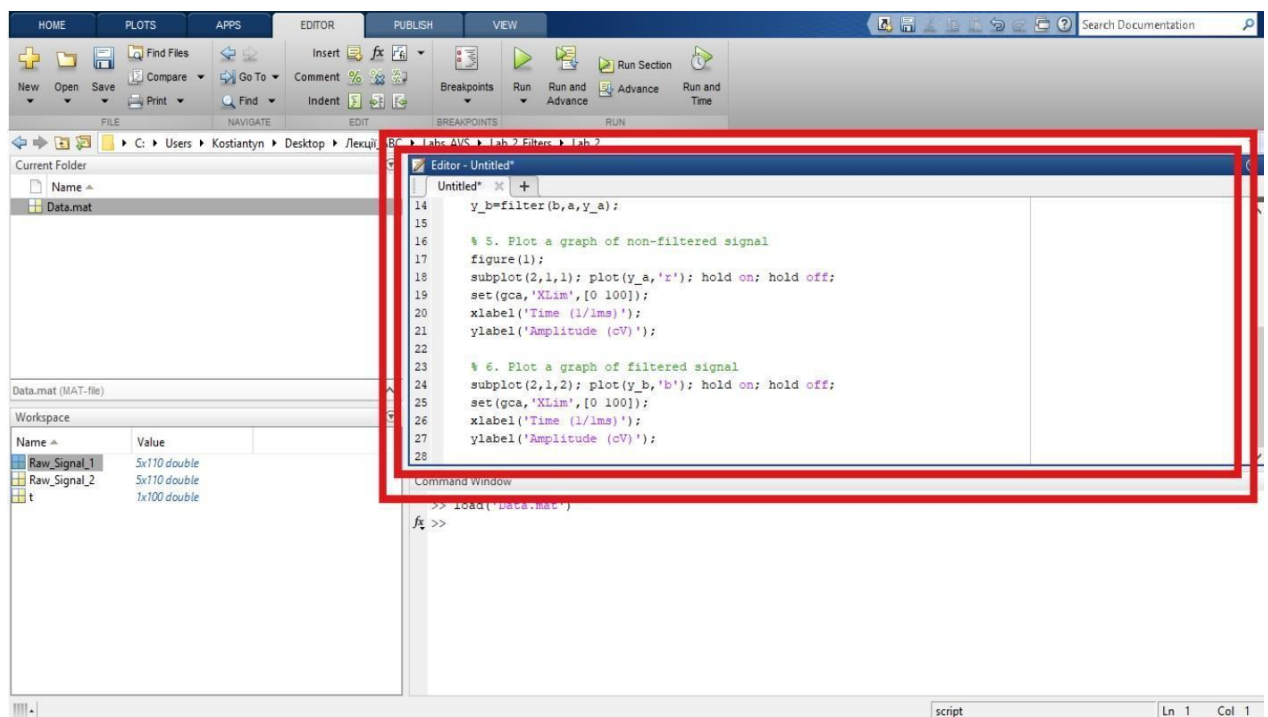


Рис. 3.12. Приклад робочого коду для застосування фільтра Баттерворта до першого сигналу (продовження коду)

Варто зазначити, що код прописаний після символу «%» (на рисунку 3.11 виділений зеленим кольором) є коментарем, та не впливає на хід виконання програми. Всі інші коди (виділені чорним кольором) безпосередньо впливають на програму та фінальний результат. Приклад коду, показаний на рис. 3.11 та рис. 3.12, створено для обробки першого сигналу БД, заданого у файлі «Raw\_Signal\_1». *При аналізі другого та всіх наступних сигналів (файл «Raw\_Signal\_2», «Raw\_Signal\_3» тощо) відповідні місця коду (у яких використовується назва «Raw\_Signal\_1») необхідно змінити на нову назву.*

7. Зберегти файл зі створеним кодом у робочій папці з назвою «Filter\_one\_iteration» у автоматично обраному форматі «\*.m».

8. Після збереження здійснити запуск створеного коду на моделювання, натиснувши кнопку «Run», розміщену у верхній частині робочого вікна (рис. 3.13). Якщо код програми написано без помилок, система Matlab створить графіки початкової та відфільтрованої форми *першої ітерації* сигналу, які необхідно зберегти у робочій папці під назвою «I\_Filter\_Signal\_1\_IT1» у форматі «\*.bmp».

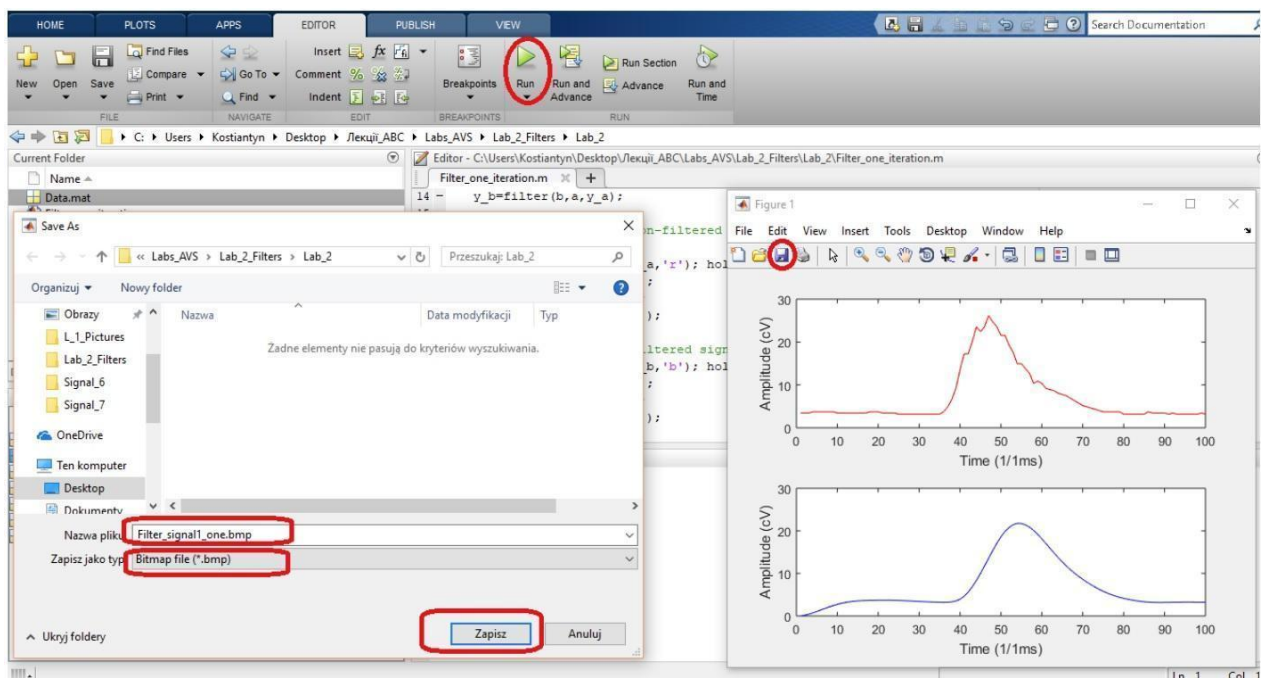


Рис. 3.13. Приклад результату моделювання робочого коду та збереження отриманих результатів

*Якщо у прописаному коді допущено помилки, то інформація про них буде наведена у вікні «Command Window», середовища Matlab. У такому випадку необхідно виправити допущені помилки та повторити запуск робочого коду повторно.*

9. Записати коефіцієнти «a» і «b» фільтра, що автоматично розраховані системою Matlab та подані в додаткових файлах, які розміщуються у полі «Workspace» після виконання коду (рис. 3.14).

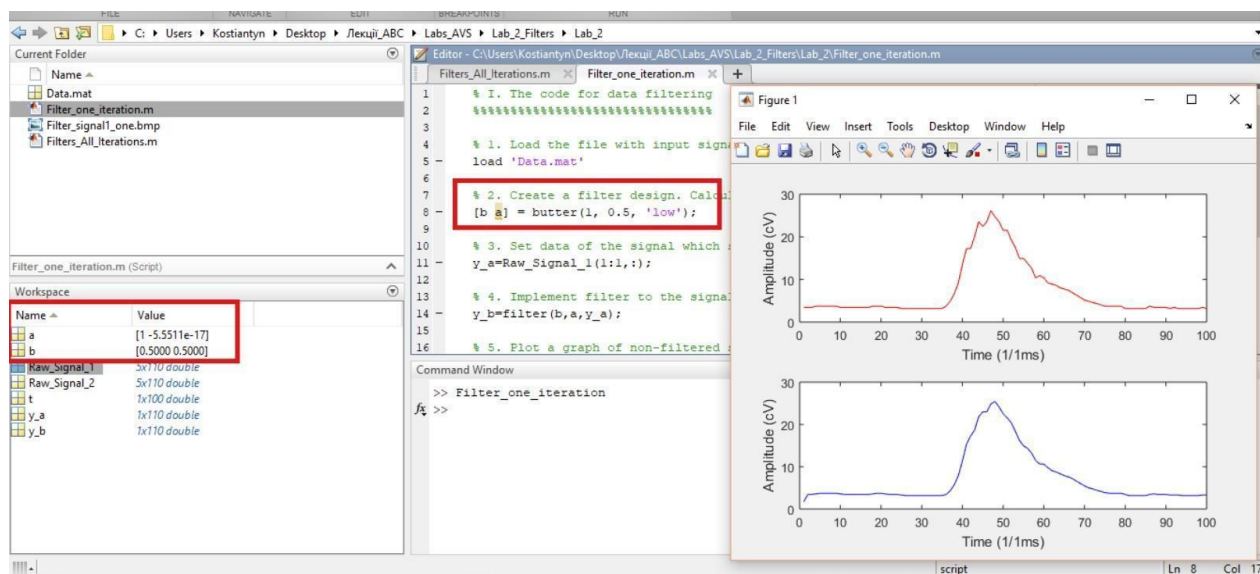


Рис. 3.14. Розміщення значень коефіцієнтів фільтра в середовищі Matlab

10. Визначені коефіцієнти фільтра необхідно записати у вигляді таблиці в протоколі до практичної роботи. Приклад запису значень коефіцієнтів фільтра показано в табл 3.1.

Таблиця 3.1. Приклад запису коефіцієнтів фільтру

Назва коефіцієнту	Числове значення		
a	1	-1,6909963768	0,73272603037
b	0,01043241337	0,02086482674	0,01043241337

11. Застосувати фільтр для решти сигналів, що розміщені у створеній БД, змінивши відповідні місця коду у відповідності до назв файлів, які містять

значення амплітуди (див. п.6). Зберегти графіки початкової та відфільтрованої форми *першої ітерації* для кожного з сигналів (у робочій папці в форматі «\*.bmp», під назвами «I\_Filter\_Signal\_2\_IT1», «I\_Filter\_Signal\_3\_IT1» тощо).

12. Після запису коефіцієнтів та збереження отриманих графіків необхідно закрити створений код та у вікні «Command Window» по чергово прописати команди для очищення робочого середовища (рис. 3.15).

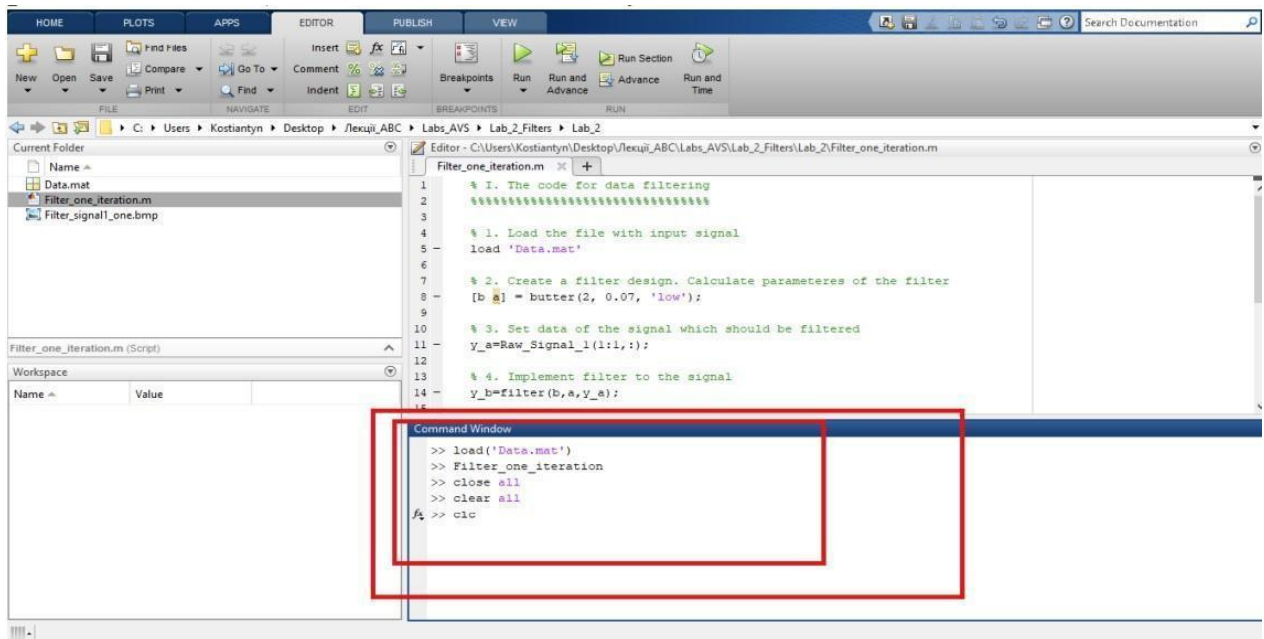


Рис. 3.15. Приклад запису команд для очищення робочого середовища

Кожна із команд записаних у вікні «Command Window» активується натисканням клавіші «Enter».

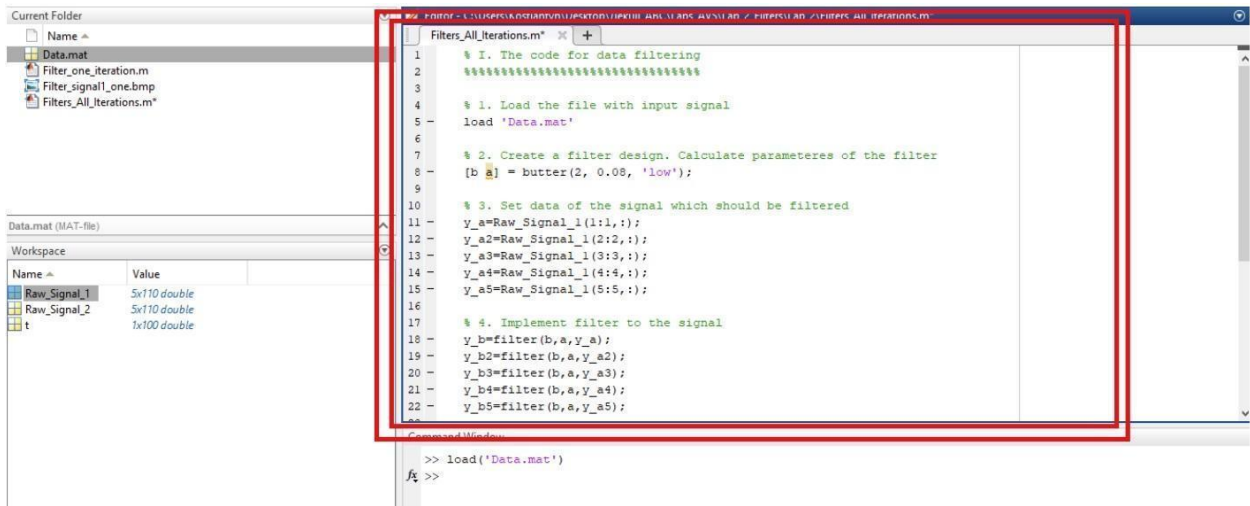
**Завдання 3.2.** Створити робочий код, що задаватиме алгоритм побудови фільтра *першого порядку* (у відповідності до індивідуальних завдань, розміщених у табл. 3.2) та графіків для *всіх ітерацій* сигналу одночасно.

### Порядок виконання Завдання 3.2.

1. Створити новий файл типу «Script» і прописати в ньому робочий код, що визначатиме алгоритм побудови фільтрів і графіків сигналу на основі інформації, що розміщена у БД одразу для всіх ітерацій сигналу. Для цього на верхній панелі «Home» обрати пункт «New» → «Script». У новому вікні «Untitled» записати код, показаний на рис. 3.16 та рис. 3.17.

2. Створений код зберегти у робочій папці під назвою

«Filter\_All\_Iterations» у автоматично обраному форматі «\*.m». Здійснити запуск робочого коду на моделювання та зберегти отримані графіки *всіх ітерацій* сигналу, попередньо відкривши їх «на повний екран» (рис. 3.18), з назвою «I\_Filter\_Signal\_1\_IT\_ALL» у форматі «\*.bmp» у робочій папці.





БД, змінивши відповідні місця коду у відповідності до назв файлів, які містять значення амплітуди (див. п.6, завдання 3.1). Зберегти графіки початкової та відфільтрованої форми *всіх ітерацій* для кожного з сигналів (у робочій папці в форматі «\*.bmp» з назвами «I\_Filter\_Signal\_2\_IT\_ALL», «I\_Filter\_Signal\_3\_IT\_ALL» тощо).

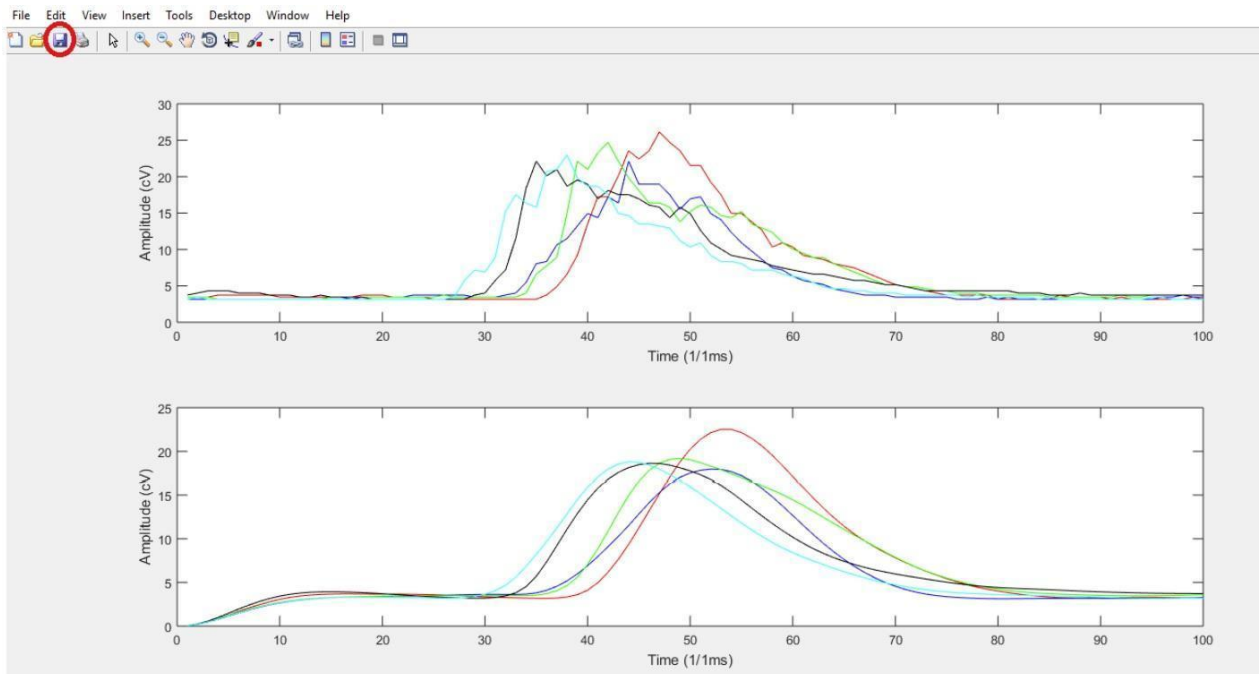


Рис. 3.18. Приклад графіку до та після фільтрації для всіх ітерацій сигналу одночасно

**Завдання 3.3.** Оновити робочий код, що задаватиме алгоритм побудови фільтра *другого* порядку (у відповідності до індивідуальних завдань, розміщених у табл. 3.2) та графіків для *першої ітерації* сигналу.

#### **Порядок виконання Завдання 3.3.**

1. Відкрити збережений файл коду «Filter\_one\_iteration» та змінити параметри фільтра у відповідності до індивідуального завдання. Повторити дії, описані у п.8 - п.11 завдання 3.1, для нових параметрів фільтра. Отримані графіки сигналу зберегти з назвами «II\_Filter\_Signal\_1\_IT1», «II\_Filter\_Signal\_2\_IT1» тощо. Визначені коефіцієнти фільтра подати у протоколі до роботи у вигляді таблиці, оформленої аналогічно до табл. 3.1.

**Завдання 3.4.** Оновити робочий код, що задаватиме алгоритм побудови фільтра *другого* порядку (у відповідності до індивідуальних завдань, розміщених у табл. 3.2) та графіків для *всіх ітерацій* сигналу одночасно.

**Порядок виконання Завдання 3.4.**

1. Відкрити збережений файл коду «Filter\_All\_Iterations» та змінити параметри фільтра у відповідності індивідуального завдання. Повторити дії, описані у п.2 - п.3 завдання 3.2, для нових параметрів фільтра. Отримані графіки сигналу зберегти у форматі «\*.bmp» з назвами «II\_Filter\_Signal\_1\_IT\_ALL», «II\_Filter\_Signal\_2\_IT\_ALL» тощо.

2. Оформити звіт до практичної роботи.

**3.3. Вимоги до оформлення звіту**

1. Звіт повинен бути представлений в електронному вигляді. Вихідний файл звіту повинен бути збережений у форматі «\*.docx» та «\*.pdf».
2. Назва електронного файлу зі звітом подається у форматі: «AVS\_PR3\_Шифр групи\_Прізвище студента\_№варіанту».
3. Структура звіту повинна містити наступні елементи:
  - титульний аркуш;
  - мету роботи;
  - номер варіанту та індивідуальне завдання (у відповідності до табл. 3.2);
  - покроковий опис виконаних дій;
  - **таблицю та графіки, отримані при застосуванні фільтра першого порядку:**
    - таблицю зі значеннями коефіцієнтів фільтра;
    - графіки першої ітерації для кожного з сигналів;
    - графіки всіх ітерацій одночасно для кожного з сигналів;
  - **таблицю та графіки, отримані при застосуванні фільтра другого порядку:**
    - таблицю зі значеннями коефіцієнтів фільтра;
    - графіки першої ітерації для кожного з сигналів;
    - графіки всіх ітерацій одночасно для кожного з сигналів;

- висновки по роботі.
4. Якість зображення та розмір шрифту на схемах, рисунках і таблицях повинні чітко відображати представлену в них інформацію. Усі рисунки, таблиці, схеми та розділи у звіті повинні бути підписані (вимоги див. у додатку А).
  5. Файл звіту додається у загальну робочу папку (див. п. 2.2), яка надсилається викладачеві на перевірку (у «\*.zip» архіві). Для проходження перевірки робоча папка обов'язково повинна містити такі файли: створену БД, код для однієї ітерації сигналу, код для всіх ітерацій сигналу, збережені рисунки та звіт до практичної роботи у двох форматах.

**Якщо оформлення звіту, робочих файлів і папки не відповідають вимогам, практична робота до захисту не допускається!**

### **3.4. Контрольні запитання**

1. Що таке фільтр сигналу?
2. У чому різниця між ідеальним та реальним фільтром?
3. Що таке частота зрізу фільтра? Як її визначити?
4. Поясніть, що означає запис дБ / октаву?
5. Чим відрізняються ФНЧ та ФВЧ фільтри?
6. Які ви знаєте види смугових фільтрів? У чому між ними різниця?
7. Що таке фазовий фільтр?
8. У чому полягає відмінність між фільтром Чебишева I та II роду?
9. Що таке фільтр Баттерворта?
10. Чим характерний еліптичний фільтр? Які його особливості?
11. Напишіть код для задання ФНЧ фільтру Баттерворта першого порядку в середовищі Matlab. Поясніть всі складові запису
12. Напишіть код для задання ФНЧ фільтру Чебишева I роду, першого порядку, в середовищі Matlab. Поясніть всі складові запису.



Таблиця 3.2. Початкові умови до завдання 3.1-3.4

№ варіанту	N (мс)	Тип фільтра	Параметри фільтра			
			Порядок фільтра	Початкова частота смуги пропускання	Кінцева частота смуги пропускання	Нормалізована частота зрізу
1	110	ФНЧ Баттерворта	I	-	-	0.6
			II	-	-	0.06
2	100	Еліптичний ФНЧ	I	0.5	30	0.05
			II	0.5	30	0.1
3	105	ФНЧ Чебишева I роду	I	-	3	0.1
			II	-	3	0.1
4	110	ФНЧ Чебишева II роду	I	-	0.5	0.05
			II	-	0.5	0.1
5	100	ФНЧ Баттерворта	I	-	-	0.55
			II	-	-	0.09
6	105	ФНЧ Чебишева II роду	I	-	1	0.07
			II	-	1	0.2
7	110	Еліптичний ФНЧ	I	1	25	0.07
			II	1	25	0.2
8	100	ФНЧ Чебишева I роду	I	-	2	0.15
			II	-	2	0.15

Продовження Таблиці 3.2. Початкові умови до завдання 3.1-3.4

<b>9</b>	105	ФНЧ Баттерворта	I	-	-	0.7
			II	-	-	0.04
<b>10</b>	110	Еліптичний ФНЧ	I	0.5	40	0.03
			II	0.5	40	0.3

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

## СПЕКТРАЛЬНИЙ АНАЛІЗ. ЗАСТОСУВАННЯ ШВИДКИХ ПЕРЕТВОРЕНЬ ФУР'Є В MATLAB

**Мета:** Ознайомитись з поняттям спектрального аналізу сигналів, особливостями розрахунку і застосування Фур'є перетворень у середовищі Matlab.

**Змістовність роботи:** гармонічний сигнал та його основні параметри, спектральні діаграми. Перетворення Фур'є, особливості розрахунку дискретних та швидких перетворень Фур'є. Покроковий алгоритм з підготовки вхідної інформації, використання вбудованих функцій і візуалізації результатів обчислень отриманих методом швидких перетворень Фур'є в середовищі Matlab.

### 4.1. Теоретичні відомості

#### 4.1.1. Спектральний аналіз. Перетворення Фур'є

У попередніх практичних роботах було представлено послідовність та особливості розрахунку детермінованих сигналів у часовій області шляхом визначення набору їх часових характеристик. Ще одним потужним інструментом, що застосовується для пошуку залежностей і властивостей сигналу, є спектральний (частотний) аналіз. Зокрема, частотний аналіз сигналу дозволяє визначати наявність або відсутність у ньому періодичних складових, порівняти вхідний сигнал з еталоном або здійснити його перетворення для подальшої інтерпретації. Серед ортогональних функцій, які можуть бути використані як базис для представлення сигналів, найчастіше застосовуються гармонічні (синусоїдальні або косинусоїдальні) функції. Спектральне розкладання деякого вхідного сигналу передбачає його представлення у вигляді суми гармонічних коливань на різних частотах.

*Гармонічний сигнал* задається виразом (4.1):

$$s(t) = A_m \cos(\omega t + \varphi) = A_m \cos\psi(t), \quad (4.1)$$

де  $A_m$  – амплітуда сигналу,  $\omega$  – його кутова швидкість (кутова частота),  $\psi$  – початкова фаза.

Теоретично гармонічні сигнали можуть існувати у необмеженому часовому інтервалі та задовольняють умовам періодичності. А саме – повторенню миттєвих значень такого сигналу через деякий проміжок часу. Такий проміжок часу має назву **період**. Для гармонічного сигналу за один період відбувається зміна повної фази сигналу на  $2\pi$  радіанів, а період сигналу можна визначити за формулою:

$$T = \frac{2\pi}{\omega} = \frac{1}{f}, \quad (4.2)$$

де  $f$  – частота коливань у герцах [Гц], що може бути визначена із кутової швидкості за співвідношенням  $\omega = 2\pi f$ .

Суть спектрального опису складного сигналу полягає в представленні його у вигляді нескінченної суми елементарних гармонічних сигналів, які мають різну **амплітуду**, **частоту** і **початкову фазу**. Сукупність елементарних гармонічних сигналів, які разом утворюють складний сигнал, називають **спектром** у базисі гармонічних коливань.

У рамках спектрального аналізу до складного сигналу, який періодично повторюється з певною кутовою частотою, застосовують один із основних методів обробки – перетворення Фур'є. Оскільки вхідний сигнал може бути поданий як набір часових відліків або як неперервна функція часу, то існує чотири основних варіанти Фур'є перетворень:

- перетворення Фур'є з неперервним часом;
- перетворення Фур'є з дискретним часом;
- ряд Фур'є з неперервним часом;
- ряд Фур'є з дискретним часом.

Якщо складний сигнал, що періодично повторюється на частоті  $\omega_0 = \frac{2\pi}{T_0}$ , описати рядом Фур'є у тригонометричній формі як базис гармонічних функцій з кратними частотами, то формула (4.1) набуває вигляду:

$$s(t \pm nT_0) = A_0 + \sum_{k=1}^{\infty} A_k \cos(k\omega_0 t + \varphi_k), \quad (4.3)$$

де  $A_0$  – постійна складова (відображає середнє значення сигналу на певному періоді),  $A_k$  – амплітуда  $k$ -ї гармонічної складової,  $\varphi_k$  – початкова фаза  $k$ -ї гармонічної складової.

Такий вираз описує спектр складного періодичного сигналу, який складається з постійної складової  $A_0$  та необмеженої кількості гармонічних складових (частоти яких описуються дискретним рядом значень  $k\omega_0$  кратних основній частоті  $\omega_0$ , а значення  $k = 1, 2, 3, \dots, n$ ). Такі складові мають назву **гармонік** періодичного сигналу. Гармоніку, яка має номер  $k = 1$ , називають **основною гармонікою**.

Для явного представлення спектра використовують його графічне зображення у вигляді двох **спектральних діаграм**: амплітудної та фазової. У **амплітудній** спектральній діаграмі (рис. 4.1.а) вісь абсцис відповідає за частоту сигналу, а вісь ординат – за амплітуду гармонік  $A_k$ .

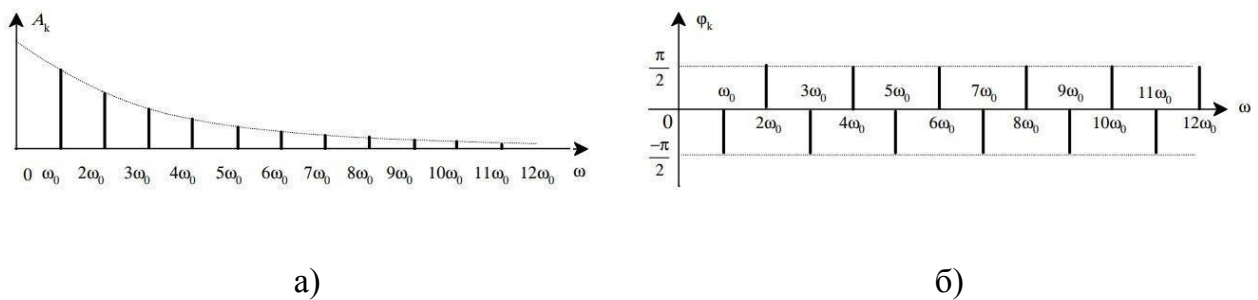


Рис. 4.1. Приклад спектральної діаграми сигналу:  
амплітудна спектральна діаграма (а); фазова спектральна діаграма (б)

У **фазовій** спектральній діаграмі (рис. 4.1.б) вісь абсцис також відповідає за частоту сигналу, а вісь ординат – за початкову фазу  $\varphi_k$ . Варто зазначити, що спектри періодичних сигналів є дискретними, а їх діаграми мають лінійчасту форму (рис. 4.1).

#### 4.1.2. Дискретні та швидкі перетворення Фур'є

Одним з базових алгоритмів обробки вимірювальних сигналів у частотній області є метод дискретного перетворення Фур'є (DFT). Використання DFT та окремих алгоритмів його обчислення (таких як швидке перетворення Фур'є (FFT)) здійснюється для сигналів, що задаються у вигляді дискретної послідовності.

Для дискретної послідовності сигналу  $S(n)$  пряме та обернене DFT обчислюється за формулами (4.4) та (4.5), відповідно:

$$S(k) = \sum_{n=0}^{N-1} S(n) \exp\left[-j \frac{2\pi nk}{N}\right], \quad k = \underline{0, N-1}, \quad (4.4)$$

$$S(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) \exp\left[j \frac{2\pi nk}{N}\right], \quad n = \underline{0, N-1}. \quad (4.5)$$

Тут  $k$  – номер гармоніки на частоті  $f_k$ ,  $N$  – обсяг вибірки.

Пряме DFT, визначене як комплексний спектр сигналу, можна описати формулою (4.6):

$$s(k) = A(k) - jB(k) = C(k)e^{-j\varphi(k)}. \quad (4.6)$$

У такому випадку АЧХ сигналу визначатиметься за формулою (4.7), а його ФЧХ за формулою (4.8):

$$C(k) = \sqrt{A^2(k) + B^2(k)}. \quad (4.7)$$

$$\varphi(k) = \arctg \frac{B(k)}{A(k)} + 2\pi m. \quad (4.8)$$

Обернене DFT також можна подати у вигляді (4.9):

$$S(n) = \frac{1}{N} \sum_{k=0}^{N-1} C(k) \cos\left[\varphi(k) + \frac{2\pi}{N} kn\right], \quad n = \underline{0, N-1}. \quad (4.9)$$

Ще однією відомою формою запису DFT є рівняння вигляду (4.10):

$$F(\Delta\xi) = \sum_{k=0}^{N-1} x(k\Delta x) \delta(x - k\Delta * x) e^{-j2\pi k\Delta\xi\Delta x} = k * \sum_{k=0}^{N-1} x_k * e^{-j\frac{2\pi}{N}kn}, \quad (4.10)$$

$$\text{де } \Delta x = \frac{2x_{\max}}{N}; \Delta\xi = \frac{1}{2x_{\max}}; \text{ а } \Delta x * \Delta\xi = \frac{1}{N}.$$

Аналогічно для оберненого перетворення:

$$x_m = k \sum F_n * e^{j\frac{2\pi}{N}nn}, \quad (4.11)$$

$$\text{де } k = \frac{1}{\sqrt{N}}.$$

Таким чином DFT можна представити у вигляді  $F = \frac{1}{\sqrt{N}} E_N X$ , де  $E_N$  подається у матричній формі (4.12):

$$E_N = \begin{bmatrix} W^0 & W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & W^2 & \dots & W^{N-1} \end{bmatrix}. \quad (4.12)$$

Разом з цим  $W^k = e^{-j\frac{2\pi k}{N}}$ , а рядки такої матриці визначають набір ортогональних функцій (базис, згідно з яким виконується розкладання сигналу). У процесі виконання DFT кожен елемент вектора результату визначає внесок певної ортогональної функції у формування вихідного сигналу. Оскільки обчислення DFT потребує значної кількості операцій множення (орієнтовно  $N^2$ ) і розрахунку синусів, то для прискорення обчислень більш часто застосовують метод FFT.

Алгоритм FFT по суті є прискореним алгоритмом DFT, що забезпечує більш швидкий розрахунок усіх необхідних параметрів приблизно за  $N \cdot \log_2 N$  операцій множення. У цьому алгоритмі використання властивостей комплексної експоненти (для зручності обчислень позначається  $W_N^{kn}$ ) дозволяє зменшити значну кількість повторюваних операцій, які виникають при DFT розрахунках.

$$W_N^{kn} = e^{-j(\frac{2\pi k}{N})kn} \quad (4.13)$$

$$X(jk) = \sum_{n=0}^{N-1} x(n)e^{-j(\frac{2\pi k}{N})kn} \quad (4.14)$$

При FFT розрахунках період обирається рівним числу точок FFT (довжині реалізації сигналу, яка буде оброблятися), а експоненті  $W_N^{kn}$  відповідає період  $\frac{N}{p}$ , де  $p$  – цілі числа, на які буде ділитись показник  $N$  (розмір або довжина FFT). Особливістю FFT алгоритму є те, що він може застосовуватись лише у випадках коли довжина сигналу  $N$  є степеню двійки.

Використання швидких перетворень Фур'є дозволяє розрахувати спектр частот для сигналів, які змінюються у часі, та візуалізувати його за допомогою спектральних діаграм (рис. 4.2).

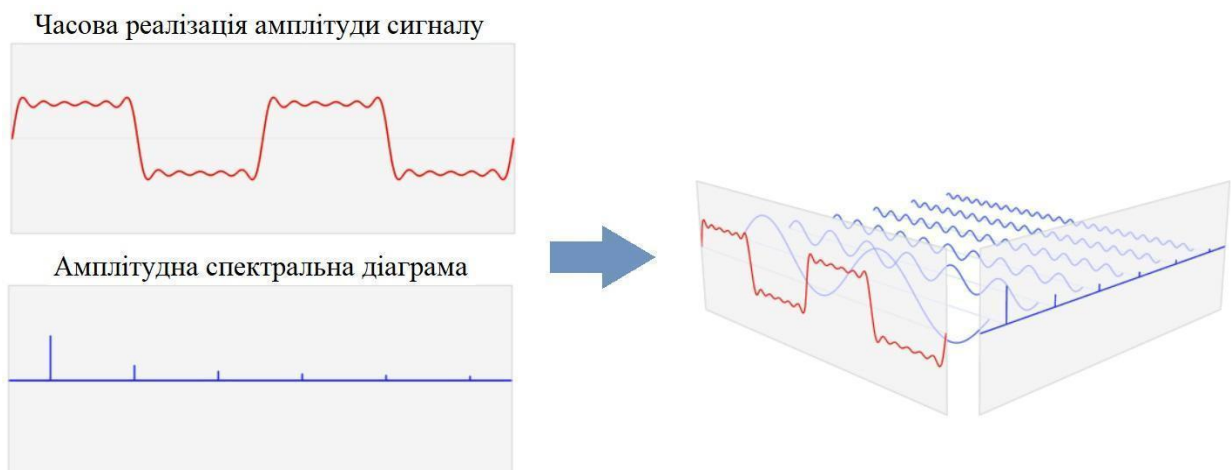


Рис. 4.2. До пояснення спектрального та часового представлення сигналу



У процесі аналізу частіше використовуються саме амплітуди окремих гармонік сигналу, а не їх фази. У результаті це призводить до побудови спектру у вигляді графіка залежності амплітуди від частоти. Шкала амплітуди при цьому зазвичай градується в децибелах.

## 4.2. Завдання для виконання

Створити робочу папку з назвою: **AVS\_PR4\_Шифр групи\_Прізвище студента\_№варіанту** (PR4 розшифровується як «Практична робота №4»). **Назва папки задається латиницею (англомовними символами)!** Скопіювати до створеної папки файли «Data», «Filter\_one\_iteration» та «Filter\_All\_Iterations» з попередньої практичної роботи. Переназвати їх як «Filter\_one\_iteration\_FFT» і «Filter\_All\_Iterations\_FFT». Файл «Data» залишити без змін.

**Завдання 4.1.** Застосувати метод швидких перетворень Фур'є (Fast Fourier Transform) для розрахунку магнітуди для *першої ітерації* сигналу на визначеному інтервалі частот.

### Порядок виконання Завдання 4.1.

1. Відкрити файл «Filter\_one\_iteration\_FFT» та доповнити його додатковими рядками коду, показаними на рис. 4.3 та рис. 4.4 Приклад коду, показаний на рисунку, створено для обробки першого сигналу БД, заданого у файлі «Raw\_Signal\_1».

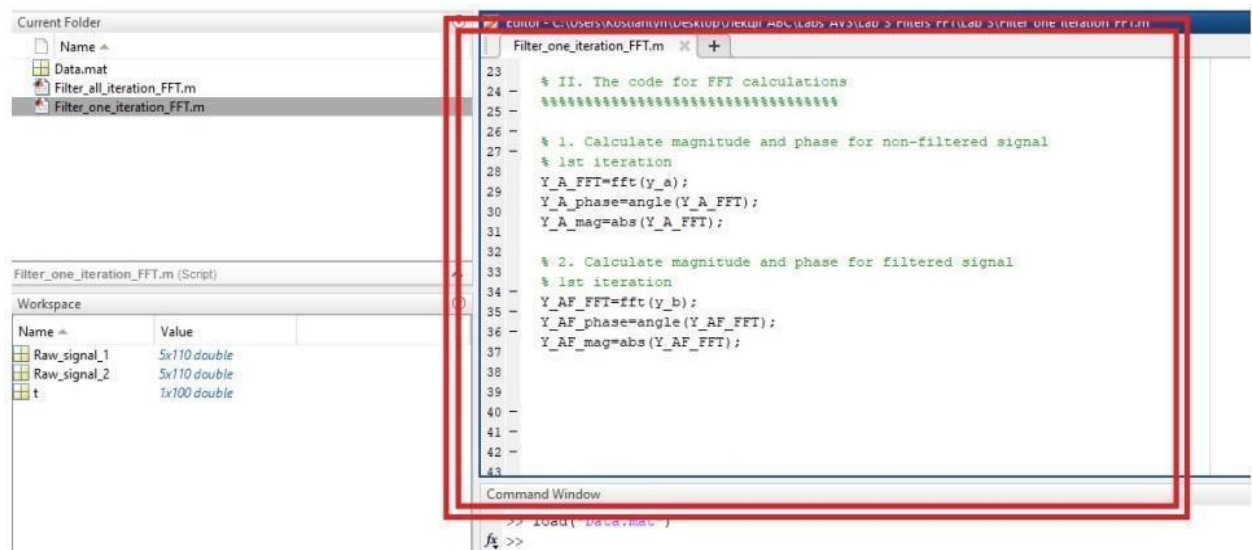
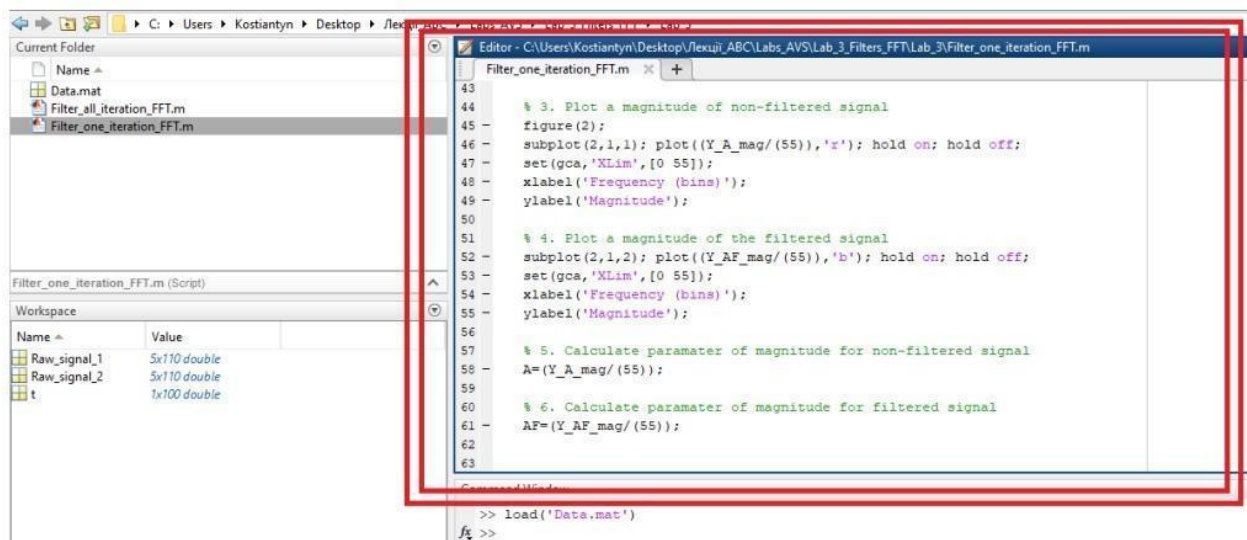


Рис. 4.3. Приклад коду для застосування FFT-перетворень

При аналізі другого та всіх наступних сигналів (файл «Raw\_Signal\_2», «Raw\_Signal\_3» тощо) відповідні місця коду (у яких використовується назва «Raw\_Signal\_1») необхідно змінити у відповідності до нової назви.



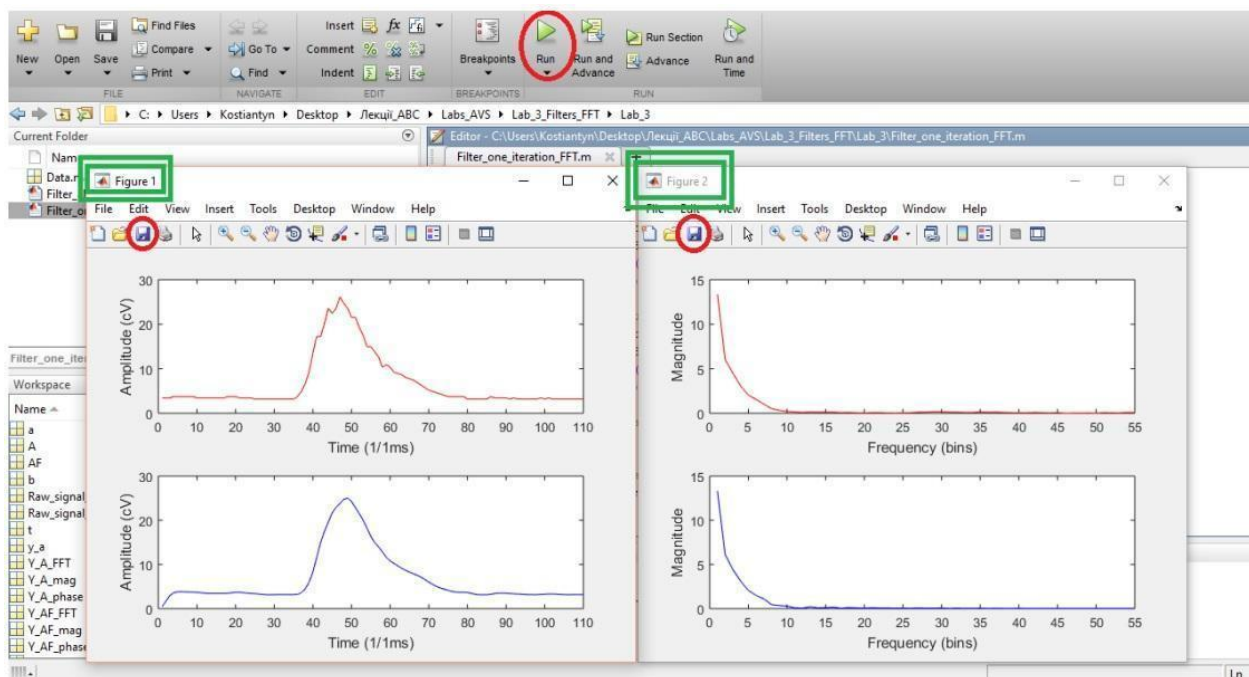


Рис. 4.5. Приклад графіків однієї ітерації сигналу та їх частотного спектру

4. Записати у вигляді таблиці значення магнітуди початкової та відфільтрованої форми сигналів на частоті [1 Гц, 2 Гц, 5 Гц, 7 Гц, 8 Гц] і розмістити їх у протоколі до практичної роботи. Приклад таблиці зі значеннями магнітуди показано в табл 4.1.

Таблиця 4.1. Приклад запису значень магнітуди сигналу на інтервалі частот

Частота	Розраховані значення магнітуди	
	Початкова форма першого сигналу	Після застосування фільтру
1 Гц	13,3751	13,2954
2 Гц	6,0232	6,1004
5 Гц	2,0323	2,0480
7 Гц	1,0180	1,0936
8 Гц	0,5232	0,4460

Автоматично розраховані системою Matlab значення магнітуди розміщуються в файлах «А» (початкова форма) та «AF» (після застосування фільтру), які створюються у полі «Workspace» після успішного виконання створеного коду (рис. 4.6).

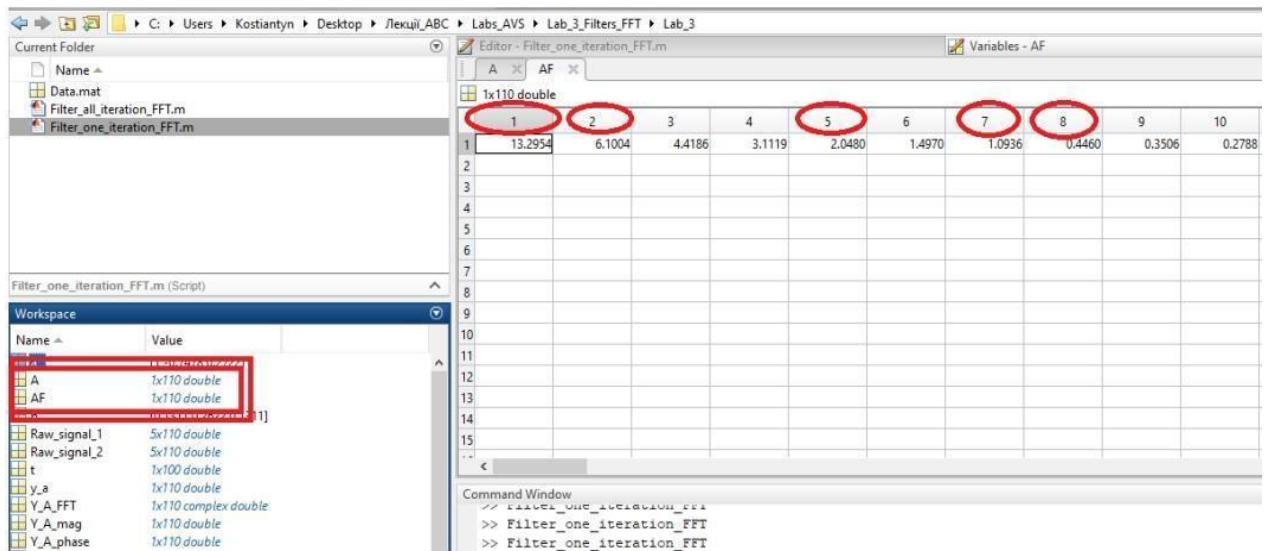


Рис. 4.6. Приклад розташування та вмісту робочих файлів з визначеною магнітудою сигналу

5. Зберегти і закрити створений код; у вікні «Command Window» почергово прописати команди для очищення робочого середовища (див. п. 11 завдання 3.1).

**Завдання 4.2.** Застосувати метод швидких перетворень Фур'є (Fast Fourier Transform) для розрахунку магнітуди одночасно для *всіх ітерацій* сигналу на визначеному інтервалі частот.

#### Порядок виконання Завдання 4.2.

1. Відкрити файл «Filter\_All\_Iterations\_FFT» та доповнити його додатковими рядками коду, показаними на рис. 4.7 - рис. 4.10. Приклад коду, показаний на рисунку, створено для обробки першого сигналу БД, заданого у файлі «Raw\_Signal\_1». *При аналізі другого та всіх наступних сигналів (файл «Raw\_Signal\_2», «Raw\_Signal\_3» тощо) відповідні місця коду (у яких використовується назва «Raw\_Signal\_1») необхідно змінити у відповідності до нової назви.*

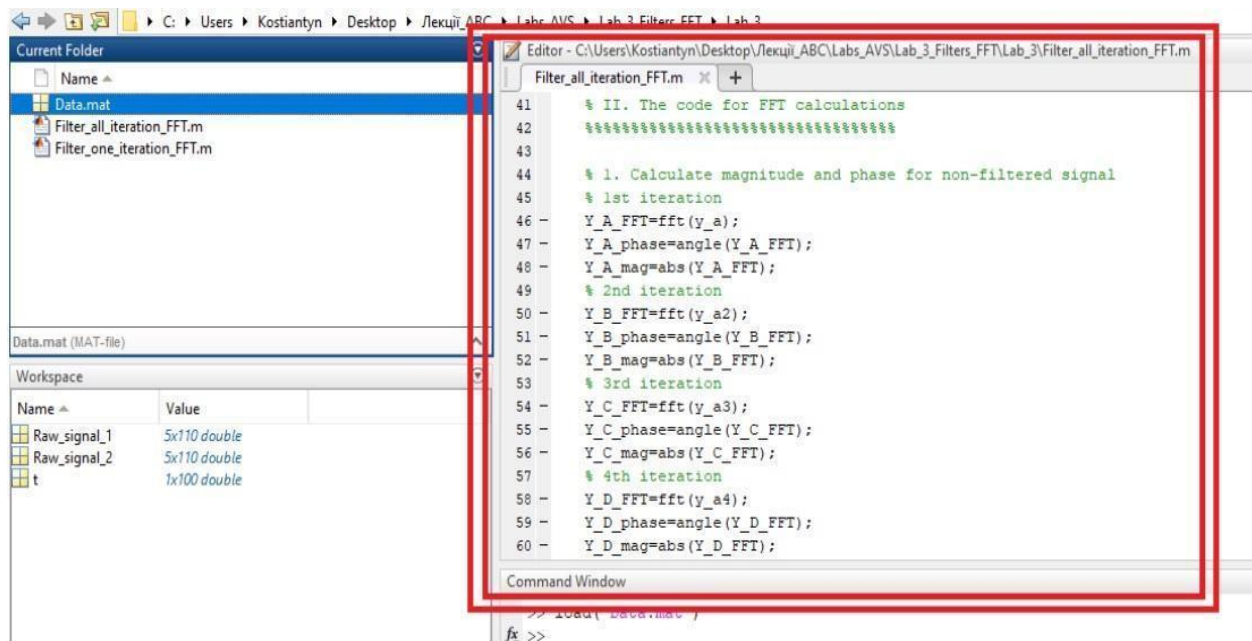


Рис. 4.7. Приклад коду для виконання завдання 4.2

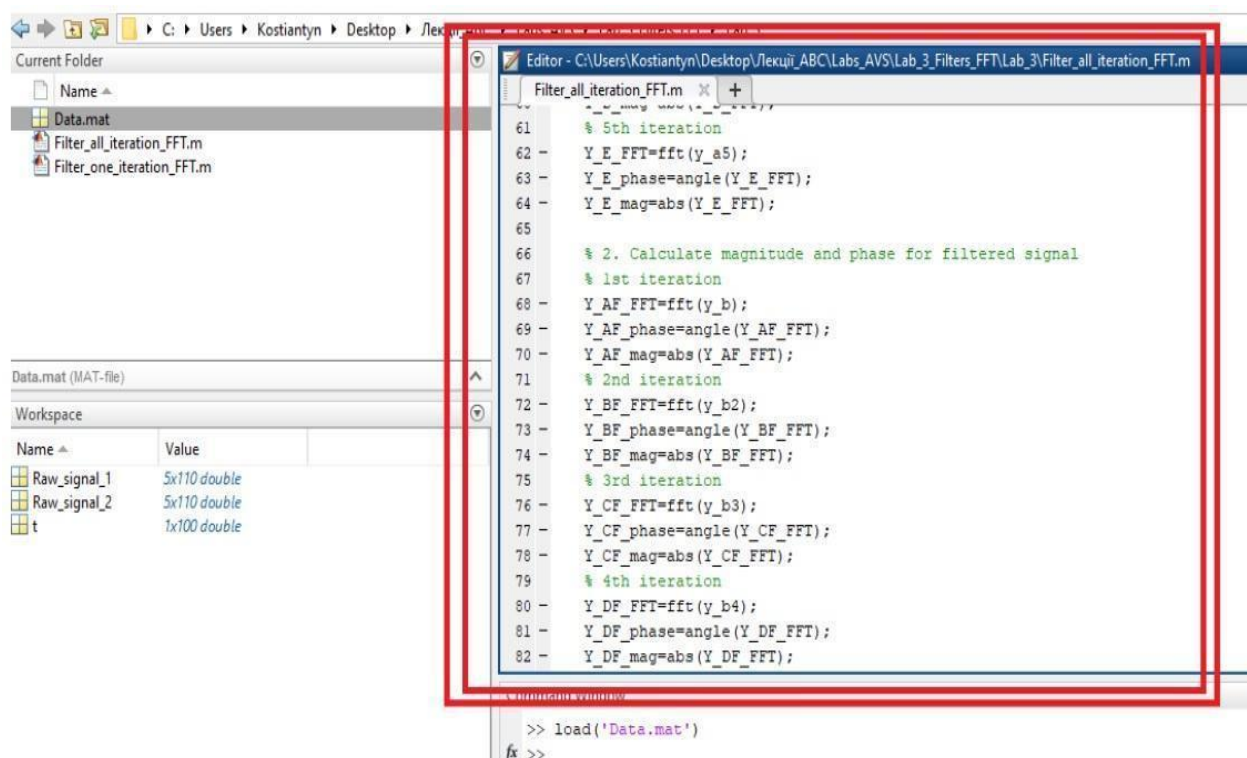


Рис. 4.8. Приклад коду для виконання завдання 4.2 (друга частина)



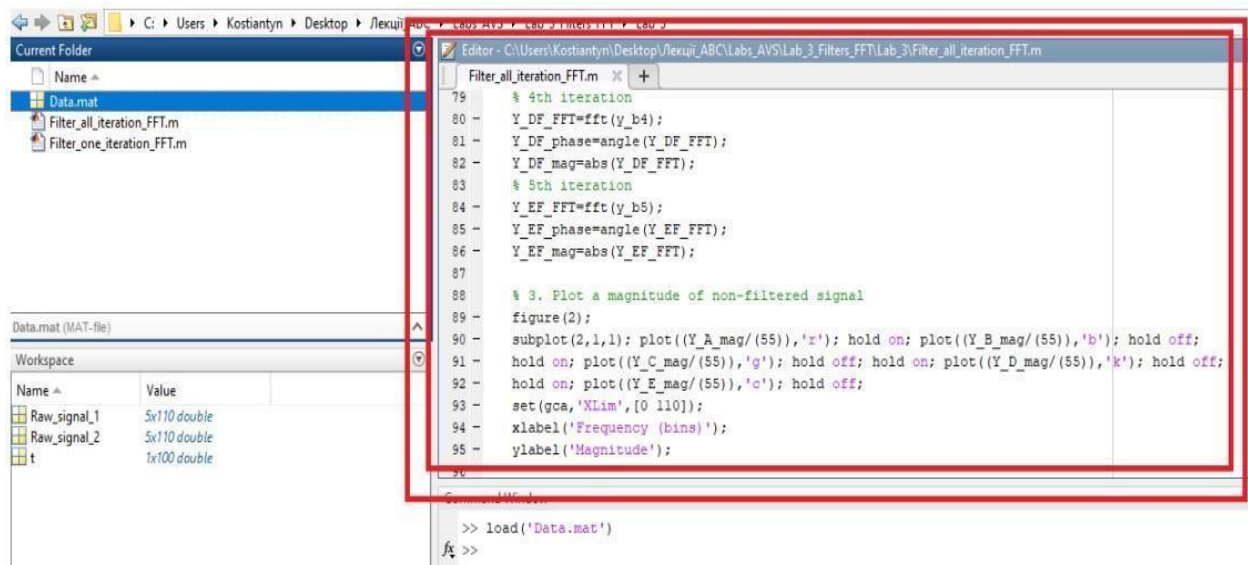


Рис. 4.9. Приклад коду для виконання завдання 4.2 (третя частина)

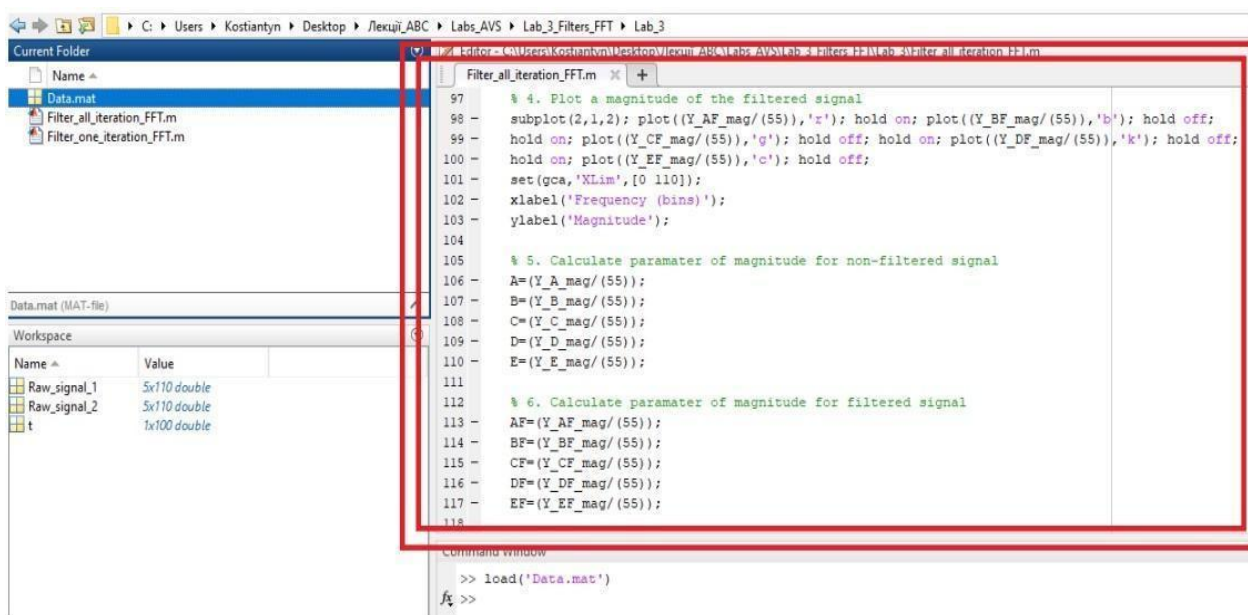


Рис. 4.10. Приклад коду для виконання завдання 4.2 (четверта частина)

2. Зберегти програму і здійснити запуск створеного коду на моделювання, натиснувши кнопку «Run». Якщо код не містить помилок, то в результаті моделювання програмою буде згенеровано два графіки (рис. 4.11), на яких зображено амплітуду *всіх ітерацій* сигналу в початковій та відфільтрованій формах («Figure 1»), а також їх магнітуду на визначеному інтервалі частот («Figure 2»).

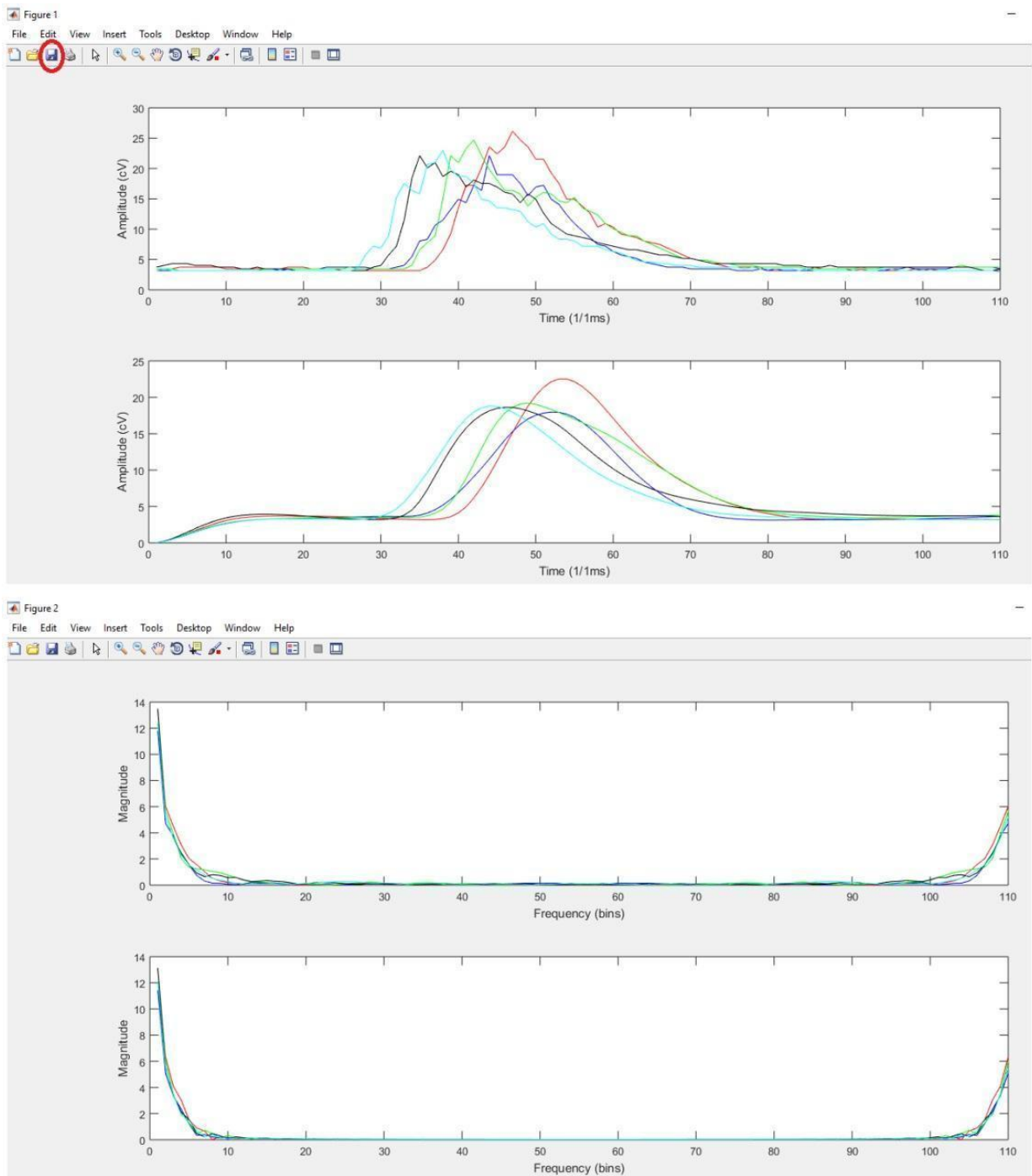


Рис. 4.11. Приклад графіків всіх ітерацій сигналу одночасно та їх частотного спектру

3. Зберегти отримані графіки у робочій папці з розширенням «\*.bmp», з назвами: «A\_II\_Filter\_Signal\_1\_IT\_ALL» для графіка, розміщеного у вікні «Figure 1», та «F\_II\_Filter\_Signal\_1\_IT\_ALL» для графіка, розміщеного у вікні «Figure 2», відповідно.

4. Записати у вигляді зведеної таблиці значення магнітуди початкової та відфільтрованої форми всіх ітерацій сигналу на частоті [1 Гц, 2 Гц, 5 Гц, 7 Гц, 8 Гц] і розмістити їх у протоколі до практичної роботи. Приклад таблиці зі значеннями магнітуди показано в табл 4.2.

Таблиця 4.2. Приклад запису значень магнітуди всіх ітерацій сигналу

Частота	Розраховані значення магнітуди					
	Початкова форма 1-го сигналу	...	Початкова форма М-го сигналу	Сигнал №1 після застосування фільтра	...	Сигнал №М після застосування фільтра
1 Гц	13,3751	...	12,9237	13,2954	...	12,5354
2 Гц	6,0232	...	5,1208	6,1004	...	5,2304
5 Гц	2,0323	...	2,2543	2,0480	...	2,0390
7 Гц	1,0180	...	1,1331	1,0936	...	1,0221
8 Гц	0,5232	...	0,4650	0,4460	...	0,2350

Автоматично розраховані системою Matlab значення магнітуди розміщуються в файлах «А», «В», «С», «D», «Е» (початкова форма) та «AF», «BF», «CF», «DF», «EF» (після застосування фільтра), які створюються у полі «Workspace» після успішного виконання створеного коду.

5. Повторити п.2 - п.4. для кожного з сигналів БД, змінюючи назви рисунків і таблиць («A\_I\_Filter\_Signal\_2\_IT\_ALL», «F\_I\_Filter\_Signal\_2\_IT\_ALL» тощо).

6. Оформити звіт до практичної роботи.

### 4.3. Вимоги до оформлення звіту

1. Звіт повинен бути представлений в електронному вигляді. Вихідний файл звіту повинен бути збережений у форматі «\*.docx» та «\*.pdf».
2. Назва електронного файлу зі звітом подається у форматі: «AVS\_PR4\_Шифр групи\_Прізвище студента\_№варіанту».
3. Структура звіту повинна містити наступні елементи:
  - титульний аркуш;



- мету роботи;
  - номер варіанту та індивідуальне завдання (у відповідності до табл. 4.3);
  - покроковий опис виконаних дій;
  - **таблицю та графіки, отримані при застосуванні фільтру для однієї ітерації першого сигналу:**
    - таблицю зі значеннями магнітуди першої ітерації сигналу до та після застосування фільтру;
    - графік амплітуди першої ітерації до та після застосування фільтру;
    - графік магнітуди першої ітерації до та після застосування фільтру;
  - **таблиці та графіки, отримані при застосуванні фільтру для всіх ітерацій одночасно (для кожного із сигналів):**
    - таблиці зі значеннями магнітуди всіх ітерацій сигналу до та після застосування фільтру (для кожного з сигналів, розміщених у БД).
    - графіки амплітуди всіх ітерацій до та після застосування фільтру (для кожного з сигналів, розміщених у БД);
    - графіки магнітуди всіх ітерацій до та після застосування фільтру (для кожного з сигналів, розміщених у БД);
  - висновки по роботі;
4. Якість зображення та розмір шрифту на схемах, рисунках і таблицях повинні чітко відображати представлену в них інформацію. Усі рисунки, таблиці, схеми та розділи у звіті повинні бути підписані (вимоги див. у додатку А).
  5. Файл звіту додається у загальну робочу папку (див. п. 2.2), яка надсилається викладачеві на перевірку (у «\*.zip» архіві). Для проходження перевірки робоча папка обов'язково повинна містити такі файли: створену БД, код для однієї ітерації сигналу, код для всіх ітерацій сигналу, збережені рисунки та звіт до практичної роботи у двох форматах.

**Якщо оформлення звіту, робочих файлів і папки не відповідають вимогам, практична робота до захисту не допускається!**

#### 4.4. Контрольні запитання

1. Що таке гармонічний сигнал? У чому його особливість?
2. Що таке амплітуда, частота та початкова фаза сигналу?
3. Що таке гармоніки та основна гармоніка?
4. Що таке спектр сигналу та спектральна діаграма?
5. Які види спектральних діаграм ви знаєте? У чому їх особливість?
6. Що таке перетворення Фур'є?
7. У чому полягає особливість дискретних перетворень Фур'є?
8. Чим швидкі перетворення Фур'є відрізняються від дискретних?
9. У чому полягає різниця спектрального та часового аналізу сигналів?
10. Як в середовищі Matlab викликається метод швидких перетворень Фур'є? Які параметри повинні бути вказані в коді програми для реалізації FFT перетворень?

Таблиця 4.3. Початкові умови до завдання 4.1 та 4.2

№ варіанту	N (мс)	Тип фільтра	Параметри фільтра			
			Порядок фільтра	Початкова частота полоси пропускання	Кінцева частота полоси пропускання	Нормалізована частота зрізу
1	110	ФНЧ Баттерворта	II	-	-	0.06
2	100	Еліптичний ФНЧ	I	0.5	30	0.05
3	105	ФНЧ Чебишева I роду	I	-	3	0.1
4	110	ФНЧ Чебишева II роду	II	-	0.5	0.1
5	100	ФНЧ Баттерворта	I	-	-	0.55
6	105	ФНЧ Чебишева II роду	II	-	1	0.2
7	110	Еліптичний ФНЧ	I	1	25	0.07
8	100	ФНЧ Чебишева I роду	II	-	2	0.15
9	105	ФНЧ Баттерворта	II	-	-	0.04
10	110	Еліптичний ФНЧ	I	0.5	40	0.03

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ №5

## МАШИННЕ НАВЧАННЯ. ОСНОВИ ПРОЕКТУВАННЯ І ТРЕНУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ПРЯМОГО ПОШИРЕННЯ

**Мета:** Ознайомитись з поняттям машинного навчання та особливістю побудови і тренування штучної нейронної мережі у середовищі Matlab.

**Змістовність роботи:** поняття машинного навчання, його види та основні методи. Штучна нейронна мережа, основні визначення та складові частини штучної нейронної мережі. Покроковий опис процесу створення вхідного набору інформації для тренування мережі, проектування її структури та алгоритму навчання у середовищі Matlab.

### 5.1. Теоретичні відомості

#### 5.1.1. Машинне навчання. Види машинного навчання

Машинне навчання (МН) є підгалуззю інформатики, що застосовує різноманітні методи прогнозування, статистики і оптимізації вхідної інформації для «навчання» автоматизованих систем поступово покращувати свою продуктивність у вирішенні конкретних завдань, без явного їх програмування. МН досліджує принцип побудови алгоритмів, які мають здатність виконувати різноманітні передбачення, базуючись на побудові моделей, що використовують наявний, обмежений набір показників з емпіричних входів.

Методи МН зазвичай поділяють на дві великі категорії:

- **Навчання з учителем** (supervised learning).

Тут автоматизованій системі чітко вказують приклади емпіричних входів та бажаний результат (виходи математичної моделі), які попередньо задаються розробником («вчителем»). Метою такого навчання є напрацювання чіткої залежності або загального правила, яке відображатиме реакцію виходів моделі на стан її не завжди ідеальних входів.

- **Навчання без учителя** (unsupervised learning).

У такому випадку автоматизована система (АС) не має початкових міток чи підказок, а її завданням є самостійний пошук взаємозв'язків та прихованих відповідностей внутрішньої структури, що можуть пов'язати між собою входи та виходи моделі. При цьому навчання без учителя може бути метою саме по собі (виявлення прихованих закономірностей у вхідній інформації) або засобом досягнення більш загальної мети (навчання системи виявленню потрібних ознак чи характеристик). Навчання з учителем також можна поділити на декілька загальних категорій:

- **Напіваавтоматичне навчання** (англ. semi-supervised learning)

Де АС надається неповний тренувальний набір інформації, у якому відсутня значна частина цільових входів (наприклад, лише окрема частина вимірюного сигналу)

- **Активне навчання** (англ. active learning):

Тут АС отримує тренувальні мітки лише для обмеженого набору вхідної інформації та змушена оптимізувати власний вибір об'єктів для відповідних міток.

- **Навчання з підкріпленням** (англ. reinforcement learning)

Набір інформації для тренування математичної моделі надається лише як зворотний зв'язок на дії програми в динамічному середовищі (у вигляді винагороди або покарання за її рішення).

Серед широкого різноманіття методів та алгоритмів, що застосовуються у МН, можна виділити кілька основних: дерево рішень,  $k$ -ті найближчі сусіди, опорні вектори, розмиту логіку, Баєсові мережі, приховані моделі Маркова, штучні нейронні мережі тощо. Ефективність кожного з них залежить від завдання конкретного дослідження і якості вхідного набору інформації для навчання алгоритму. Одним з найбільш універсальних методів МН, що часто застосовується і в автоматизації, і в практиці аналізу сигналів, зокрема є метод штучних нейронних мереж.

### **5.1.2. Штучна нейронна мережа. Основні визначення та поняття**

Метод *штучних нейронних мереж* (ШНМ) є спрощеною математичною інтерпретацією моделі біологічних нейронних мереж, котрі застосовуються для

вирішення складних задач, що передбачають здійснення аналітичних розрахунків подібних тим, які виконує людський мозок. Фактично, модель ШНМ, подібно біологічному аналогові, є послідовністю нейронів, що з'єднуються між собою нейронними зв'язками (синапсами).

Розглянемо типову структуру ШНМ прямого поширення (рис. 5.1). Вона поєднує у собі декілька ключових речей: *шари*, *нейрони* та *синапси*. *Нейрон* – це елементарна обчислювальна одиниця ШНМ, функціональним призначенням якої є отримання вхідної інформації, здійснення над нею простих обчислень і подальше транспортування до наступної функціональної одиниці.

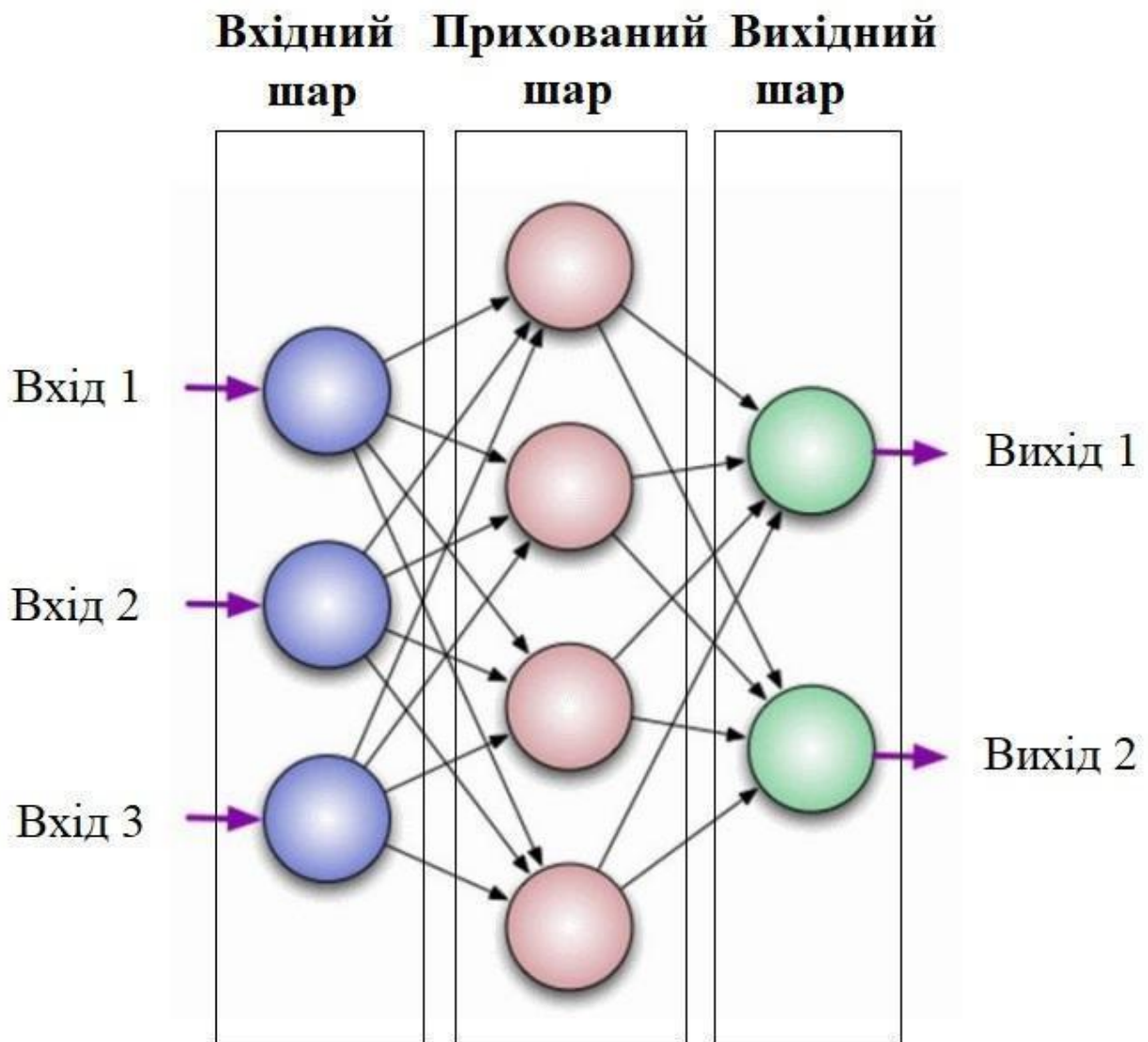


Рис. 5.1. Узагальнена структура штучної нейронної мережі прямого поширення

Як видно з рис. 5.1 нейрони поділяють на три основних категорії: вхідні, приховані та вихідні (показані на рисунку синім, червоним і зеленим кольором, відповідно). Якщо ШНМ складається зі значної кількості нейронів, їх об'єднують у спеціальні «шари». У результаті структура ШНМ прямого поширення містить вхідний шар нейронів (отримує вхідну інформацію),  $n$ -кількість прихованих шарів (відповідають за здійснення основних розрахунків) та вихідний шар (містить результати, які було досягнуто нейронною мережею). Нейрони здійснюють розрахунки в діапазоні чисел від  $[0; 1]$  або  $[-1; 1]$ . У випадку коли значення, що подаються на вхід нейрону, не потрапляють до такого діапазону чисел, застосовується один із методів нормалізації інформації (наприклад, показаних у п. 2.1.2). **Синапс** виконує роль зв'язку між двома нейронами (рис. 5.2) та має один функціональний параметр – **вагу**. Завдяки вазі синапсу вхідна інформація, що передається між нейронами, змінюється, а значення ваги коректується в процесі навчання мережі аж до моменту досягнення нею бажаного вихідного результату. Як видно з рис. 5.2 якщо у ШНМ використовується, наприклад, три нейрони то до них буде під'єднано і три синапси з різними значеннями ваги для кожного з них ( $w_1$ - $w_3$ ).

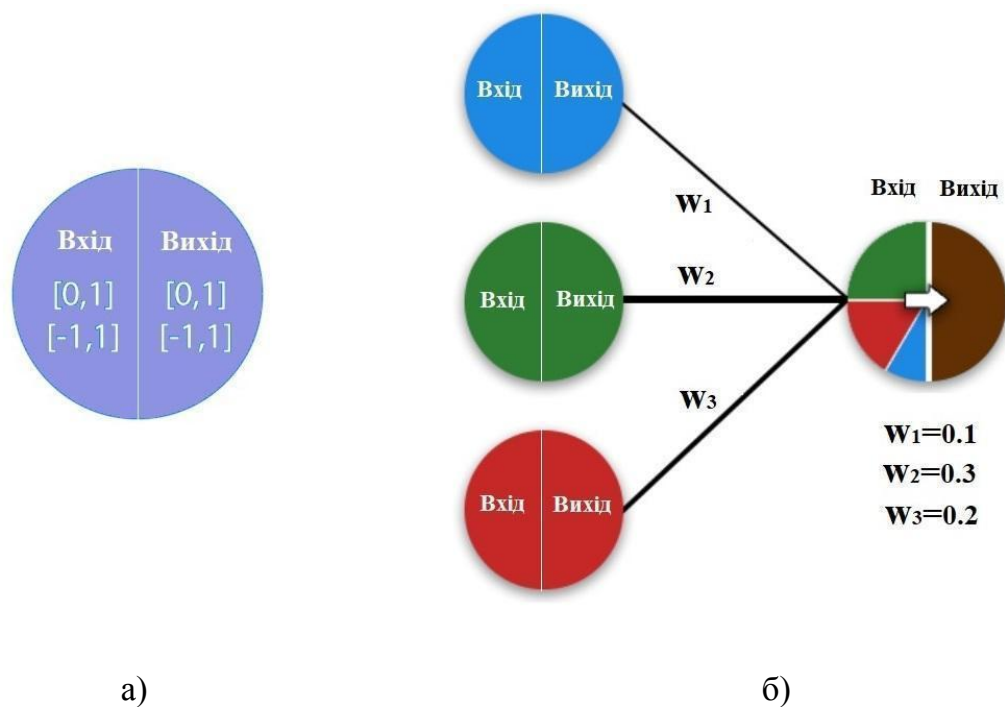


Рис. 5.2. До пояснення поняття нейронів та синапсів:  
 структура нейрону (а); нейрони, об'єднані синапсами з певною вагою (б)

При цьому найбільшу частку вхідної інформації в нейроні наступного шару займатиме та, для якої вага вхідного синапсу була найвищою. Для організації процесу постійної нормалізації вхідної інформації у нейронах на кожному з шарів ШНМ використовуються спеціальні «**активаційні функції**». Якщо значення на вході нейрону задати як аргумент для функції активації, то в результаті можна отримати вихідне значення у робочому діапазоні нейрону ( $[0; 1]$  або  $[-1; 1]$ ). Існує декілька активаційних функцій (АФ), серед яких найчастіше використовують *лінійну*, *сигмоїдну* та функцію *тангенційного сигмоїда*.

**Лінійна функція** АФ (*purelin*) описується формулою (5.1), передає початкове значення без спотворень і використовується, як правило, у випадках, коли необхідно провести тестування ШНМ.

$$f(x) = x. \quad (5.1)$$

**Сигмоїдна** АФ (*logsig*) описується формулою (5.2), а вихідний діапазон значень такої функції лежить в діапазоні  $[0; 1]$ . Сигмоїдна функція є найбільш доцільною у використанні у тих випадках, коли робота ШНМ з від'ємними значеннями не передбачається.

$$f(x) = \frac{1}{1+e^{-x}}. \quad (5.2)$$

АФ **тангенційного сигмоїду** (*tansig*) описується формулою (5.3), а вихідний діапазон значень такої функції лежить в діапазоні  $[-1; 1]$ . Використання такої функції є доцільним у випадках, коли набір вхідної інформації у ШНМ може включати як від'ємні, так і додатні значення.

$$f(x) = \frac{e^{2x}-1}{e^{2x}+1}. \quad (5.3)$$

**Тренувальна серія** – це набір вхідних значень, на яких повинно здійснюватись навчання ШНМ. Чим більш різноманітною і великою буде



тренувальна серія значень, тим більш якісно ШНМ буде тренуватись на поданих їй прикладах.

У випадку коли ШНМ під час навчання повністю обробила набір значень у тренувальній серії спрацьовує умовний лічильник, що має назву *ітерація*. Значення такого лічильника збільшуватиметься кожного разу, коли нейронна мережа застосує значення з тренувального сету для певного набору вагових коефіцієнтів (ваги синапсів).

**Епоха** – це значення, яке збільшуватиметься кожного разу, коли ШНМ застосує весь набір тренувальних сетів для набору вагових коефіцієнтів. Тобто в процесі тренування нейронної мережі спочатку збільшується значення відпрацьованих ітерацій, а вже після нього – значення епох. За правилами навчання нейронних мереж не можна спочатку навчати ШНМ тільки на одній тренувальній серії, а потім на іншій. Нейронна мережа в процесі навчання повинна пройти всі ітерації під час епохи, а вже потім перейти до наступної епохи та тренувального сету з новими ваговими коефіцієнтами синапсів. Чим вищою є кількість епох (задається як один з параметрів для ШНМ перед початком її тренування), тим більш натренованою буде мережа. Однак і час на навчання мережі буде збільшуватись.

**Похибка** – це один з параметрів, що задаються для ШНМ перед початком її навчання, та є відсотковою величиною, яка відображає розбіжність між очікуваним і фактично досягнутим результатами. Поточне значення похибки розраховується системою під час кожної епохи і в переважній більшості випадків повинно зменшуватись у процесі тренування.

## 5.2. Завдання для виконання

Створити робочу папку з назвою: **AVS\_PR5\_Шифр групи\_Прізвище студента\_№варіанту** (PR5 розшифровується як: «Практична робота №5»). Назва папки задається латиницею (англомовними символами)! Скопіювати до створеної папки Excel файл, який використовувався студентом для другої практичної роботи, та назвати його «PR5\_Test». У скопійованому Excel файлі:

- Змінити назву створених раніше бланків з «TDF\_Sig\_1», «TDF\_Sig\_2»,...,«TDF\_Sig\_n» на «TDF\_Sig\_1\_Test»,

«TDF\_Sig\_2\_Test»»,...«TDF\_Sig\_n\_Test» і видалити імпортовані значення амплітуди сигналу в цих бланках.

- Імпортувати нові значення амплітуди: з текстового файлу для першого тестового сигналу у відповідний стовпець бланку «TDF\_Sig\_1\_Test», для другого тестового сигналу – в «TDF\_Sig\_2\_Test» тощо. Часові характеристики для кожної з ітерацій тестового сигналу перерахуються автоматично.

**Завдання 5.1.** Підготувати вхідний набір інформації для подальшої роботи зі штучною нейронною мережею. Визначити вхідні та вихідні класи, тренувальний та тестувальний сет для нейронної мережі.

### Порядок виконання Завдання 5.1.

1. У вікні «Workspace» створити новий файл БД з назвою «input», у якому буде розміщуватись вхідний набір характеристик сигналу для тренування нейронної мережі.
2. Скопіювати у перші п'ять стовпчиків файлу «input» набір часових характеристик першого нормалізованого сигналу (вкладка «TDF\_Sig\_1» у Excel-файлі «AVS\_PR2\_№ варіанту»), які було розраховано у практичній роботі №2 (рис. 5.3).

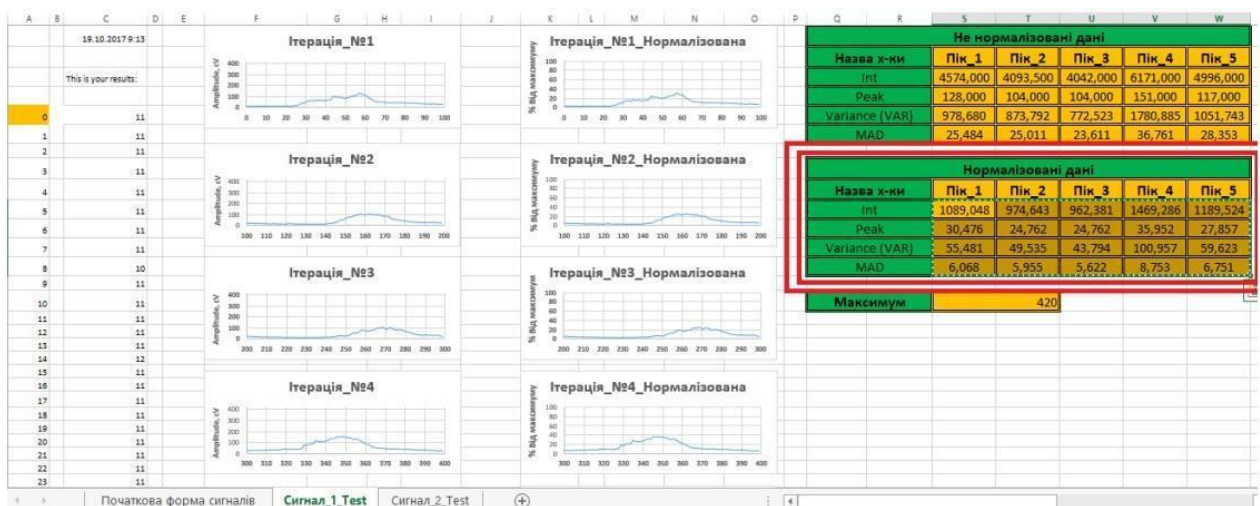


Рис. 5.3. Вікно Excel-файлу з розрахованими характеристиками сигналу

У наступні п'ять стовпчиків скопіювати значення характеристик другого

нормалізованого сигналу тощо (рис. 5.4). Подібні дії провести для всіх сигналів, що досліджувались у Excel-файлі.

Current Folder: C:\Users\Kostiantyn\Desktop\Лекції\_ABC\Лаб\_АВС\Lab\_4\_ANN\Lab\_4\_Matlab

Variables - input

	1	2	3	4	5	6	7	8	9	10	11
1	621.6670	611.9050	585.3570	622.9760	709.5240	790.4760	690	782.2620	761.5480	853.9290	
2	24.7620	20.7140	20	20.2380	19.0480	35.4760	24.5240	33.3330	31.4290	31.4290	
3	38.0610	29.7120	26.0680	23.3760	29.5870	76.6150	41.4470	62.5150	60.4360	66.7670	
4	4.7580	4.3690	3.8720	3.7750	4.5880	6.7280	5.0110	6.0410	6.1090	6.6760	

Details

Workspace

Name	Value
input	4x10 double

Command Window

fz >>

Рис. 5.4. Приклад заповнення файлу «input»

3. У вікні «Workspace» створити новий файл БД з назвою «output», у якому будуть розміщуватись вихідні дані (набір відповідей), що необхідні для тренування нейронної мережі. Додати до файлу «output» набір бінарних значень (0 та 1), що визначатимуть відповідність часових характеристик з файлу «input» класам, які повинна визначити нейронна мережа (рис. 5.5).

Current Folder: C:\Users\Kostiantyn\Desktop\Лекції\_ABC\Лаб\_АВС\Lab\_4\_ANN\Lab\_4\_Matlab

Variables - output

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	0	0	0	0	0	
2	0	0	0	0	0	1	1	1	1	1	

Details

Workspace

Name	Value
input	4x10 double
output	2x10 double

Command Window

fz >>

Рис. 5.5. Приклад заповнення файлу «output»

Наприклад, якщо ШНМ повинна визначити приналежність деякої ітерації сигналу до одного з двох вихідних класів (Сигнал «А» чи Сигнал «В»), то:

- рядки файлу «output» визначатимуть відповідний клас (перший рядок – Сигнал «А», другий рядок – Сигнал «В»), а стовпчики № ітерації у тренувальному наборі даних, яку потрібно співвіднести з відповідним класом;
- 0 – це негативна відповідь (на цій ітерації сигнал відносився до категорії «А», проте належав до категорії «В»);
- 1 – це позитивна відповідь (на цій ітерації сигнал належав до категорії «А» і не відносився до категорії «В»).

4. У вікні «Workspace» створити файл з назвою «real\_class», у якому буде розміщуватись реальний перелік класів, яким відповідає кожна ітерація сигналу. **Наприклад:** 1 – перший сигнал, 2 – другий сигнал, 3 – третій сигнал тощо (рис. 5.6).

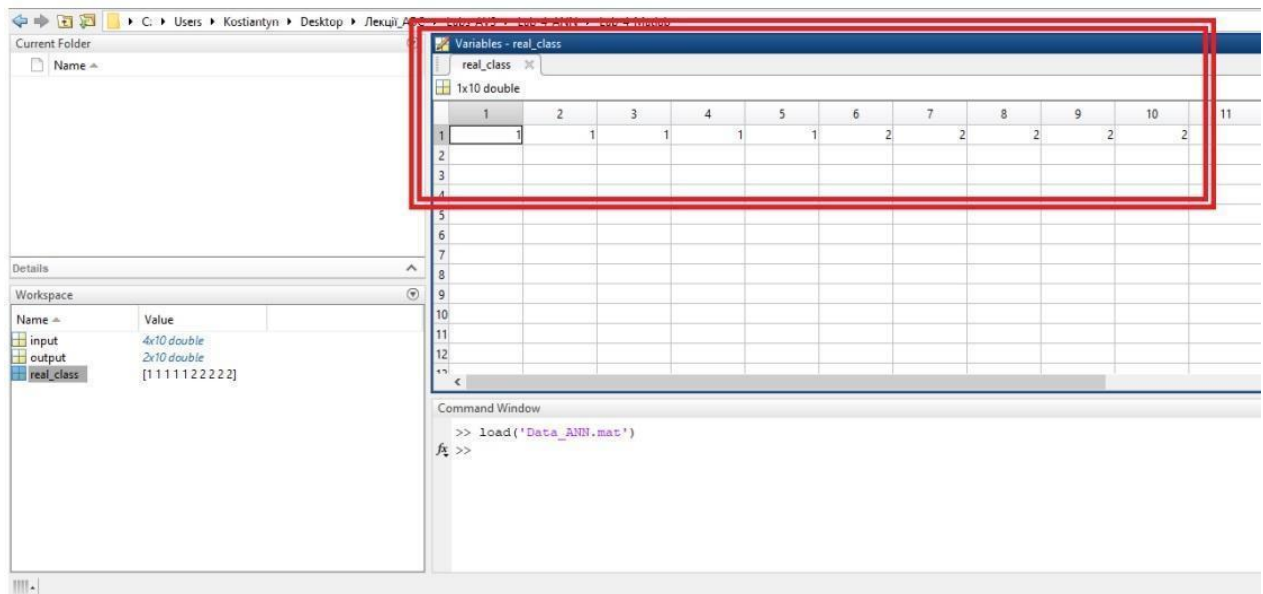


Рис. 5.6. Приклад заповнення файлу «real\_class»

5. У вікні «Workspace» створити новий файл БД з назвою «test\_in», у якому будуть розміщуватись вхідні дані для тестування нейронної мережі. Для цього необхідно відкрити Excel-файл «PR5\_Test» і скопіювати у файл «test\_in» набір часових характеристик для нормалізованих сигналів «TDF\_Sig\_1\_Test», «TDF\_Sig\_2\_Test» тощо (див. п.2 завдання 5.1).

6. У вікні «Workspace» створити новий файл БД з назвою «test\_out», у якому будуть розміщуватись вихідні дані (набір відповідей), необхідні для тестування нейронної мережі. Додати до файлу «test\_out» набір бінарних значень (0 та 1), що визначатимуть відповідність часових характеристик з файлу «test\_in» класам, які повинна визначити нейронна мережа при тестуванні (див. п.3 завдання 5.1).

7. Натиснути ПКМ на полі «Workspace» → «Save» і зберегти всі створені файли БД з назвою «Data\_ANN» та розширенням «\*.mat» у створеній робочій папці.

8. Помістити у робочу папку файли «confusion\_test», «Feedback\_training» та «plotConfMat», які надаються викладачем перед початком практичної роботи.

**Завдання 5.2.** Адаптувати робочий код та провести налаштування параметрів штучної нейронної мережі. Здійснити тренування ШНМ у відповідності до заданих параметрів.

#### **Порядок виконання Завдання 5.2.**

1. Відкрити файл «Feedback\_training» і задати в ньому параметри тренування нейронної мережі у відповідності до індивідуального завдання. На рис. 5.7 показано:

- *net* – тип нейронної мережі;
- *net.layers(1).transferFcn* – активаційна функція першого прихованого шару;
- *net.layers(2).transferFcn* – активаційна функція другого прихованого шару;
- *net.divideParam.trainRatio* – кількість (відсоток) даних із загального тренувального набору, що будуть використані для тренування ШНМ;
- *net.divideParam.valRatio* – кількість (відсоток) даних із загального тренувального набору, що будуть використані для валідування;
- *net.divideParam.testRatio* – кількість (відсоток) даних із загального тренувального набору, що будуть використані для тестування ШНМ;
- *net.trainParam.goal* – точність (кількість знаків після коми), до

досягнення якої нейронна мережа повинна тренуватись;

- *net.trainParam.epochs* – максимальна кількість тренувальних епох;
- *net.trainParam.max\_fail* – максимальна кількість похибок, яку може допустити ШНМ при тренуванні.

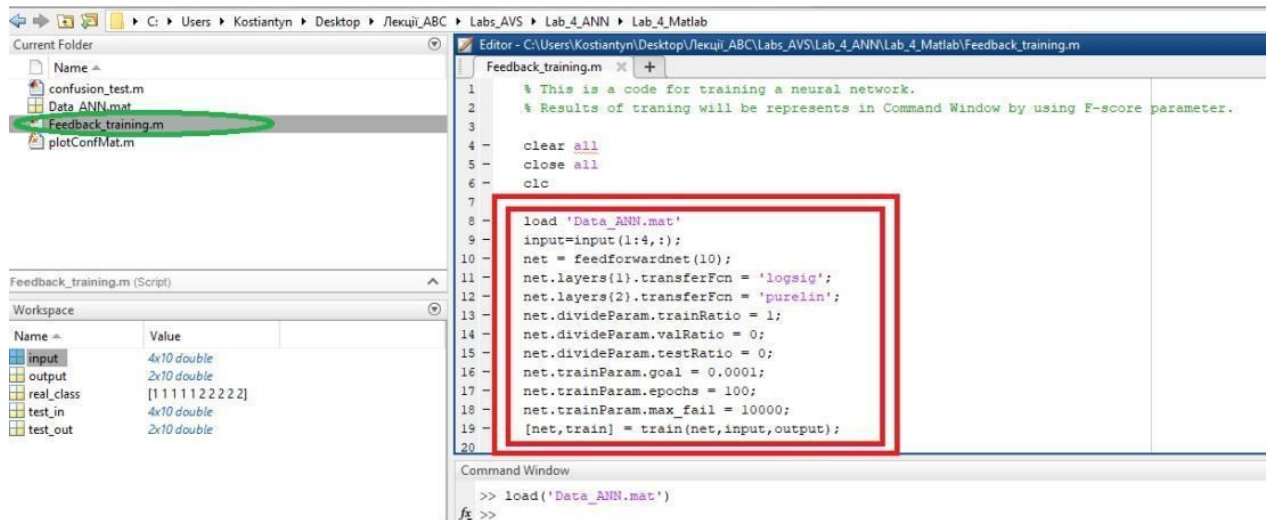


Рис. 5.7. Приклад робочого коду для налаштування параметрів ШНМ

2. Задавши всі необхідні параметри необхідно натиснути кнопку «Run», розміщену у верхній частині робочого вікна середовища Matlab.

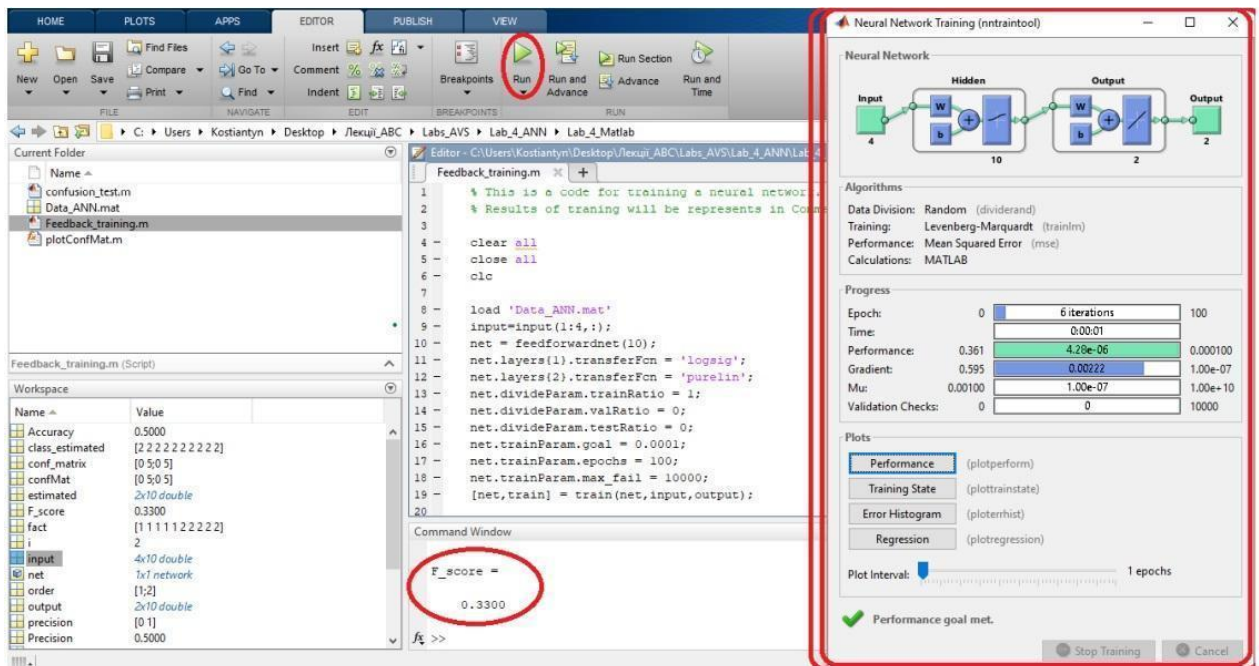


Рис. 5.8. Вигляд робочого вікна середовища Matlab в процесі тренування ШНМ



Якщо робочий код з параметрами мережі задано без помилок, програмою буде запущено вікно тренування мережі (рис. 5.8). Після завершення процесу вікно з параметрами тренування необхідно зберегти у форматі зображення (зробити screenshot) з назвою «PR5\_ANN\_structure\_1». Фінальна точність класифікації згідно показника F-score, яку досягла нейронна мережа в процесі тренування буде показана у вікні «Command Window». Наприклад, на рис. 5.8 це значення F-score = 0.33, яке відповідає точності класифікації у межах 33%.

3. Орієнтуючись на результати (показник F-score) здійснити ще п'ять тренувальних спроб (запусків коду на моделювання) з незмінними параметрами ШНМ для того ж тренувального набору даних.

4. Зберегти в окремий файл ШНМ мережу з найкращим результатом тренування. Для збереження нейронної мережі після закінчення процесу її тренування у вікні Command Window необхідно ввести такий код:

`>> genFunction(net, 'C:\Users\Kostiantyn\Desktop\PR_AVS\PR5_ANN\Net_1.m');`

Де:

- C:\Users\Kostiantyn\Desktop\PR\_AVS\PR5\_ANN - шлях до створеної папки, у яку буде збережено мережу.
- «Net\_1» - назва штучної нейронної мережі.

Якщо код введено без помилок, то у вікні Command Window з'явиться повідомлення про успішне збереження ШНМ (рис. 5.9), а файл з мережею з'явиться у робочій папці.

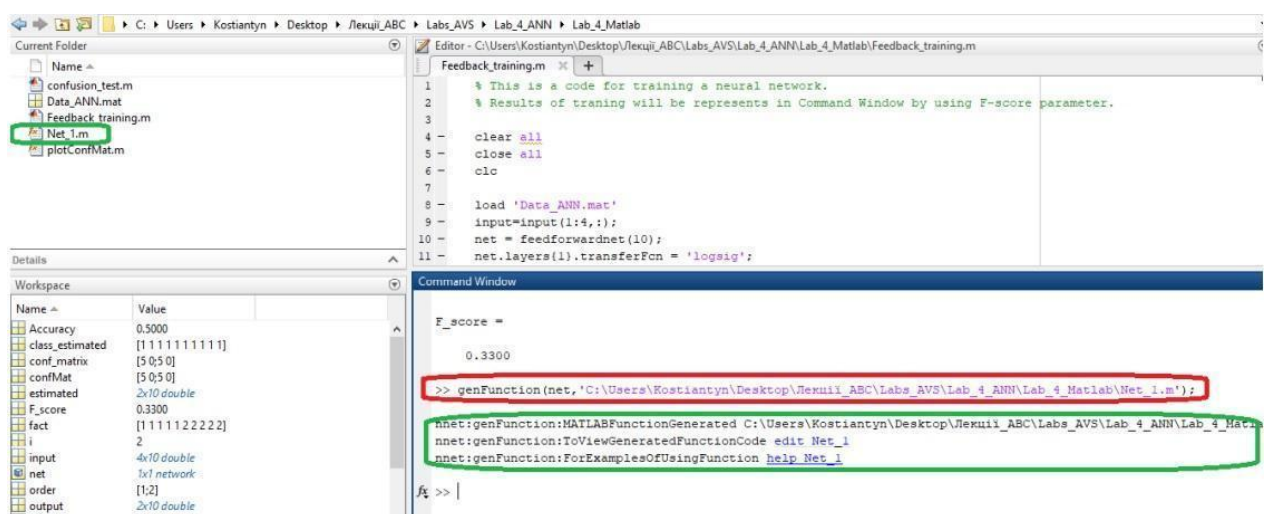


Рис. 5.9. Приклад запису коду для збереження структури ШНМ

**Завдання 5.3.** Адаптувати робочий код та провести тестування збереженої моделі нейронної мережі на новому наборі даних.

### Порядок виконання Завдання 5.3.

1. Відкрити файл «confusion\_test», задати в ньому назву збереженої у п.4 завдання 5.2 нейронної мережі (рис. 5.10) і запустити процес тестування мережі (натиснувши кнопку «Run»).

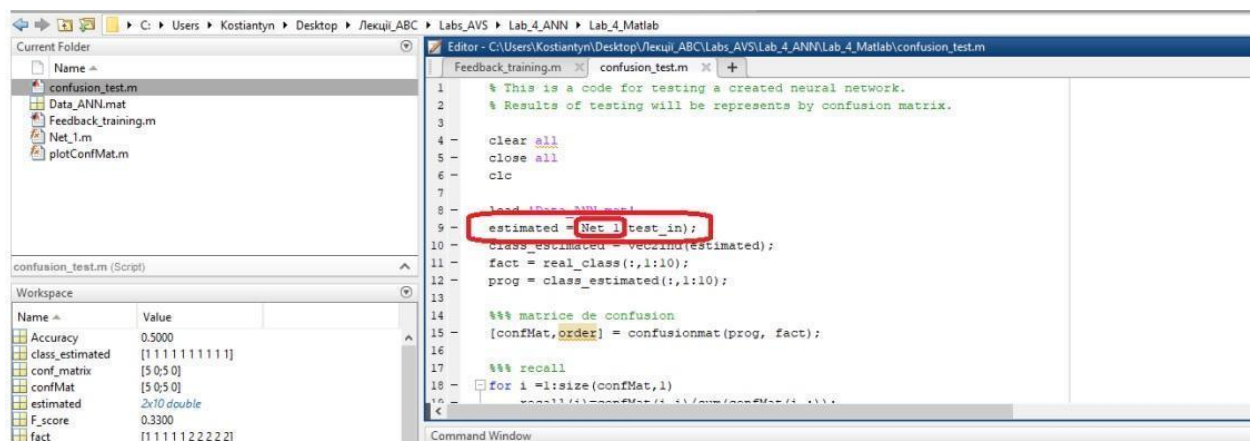


Рис. 5.10. Приклад зміни робочого коду для подальшого тестування збереженої ШНМ

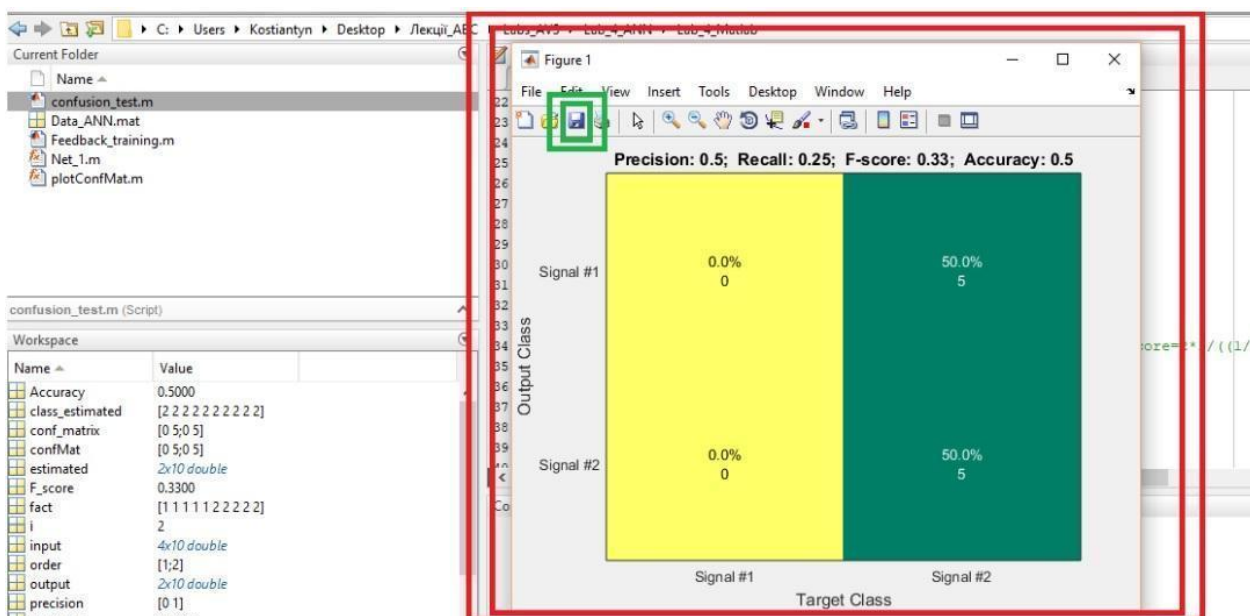


Рис. 5.11. Приклад матриці похибок створеної у середовищі Matlab



Якщо робочий код адаптовано без помилок, програмою буде візуалізовано матрицю похибок (рис. 5.11) і розраховано чотири параметри для оцінювання точності класифікації, що були досягнуті у результаті тестування ШНМ: precision, recall, F-score та accuracy.

2. Зберегти отриману матрицю похибок у форматі «\*.bmp» з назвою «PR5\_Confusion\_test\_1», а розраховані параметри для оцінювання точності класифікації подати у вигляді таблиці в протоколі до практичної роботи (табл. 5.1).

*Таблиця 5.1. Приклад запису значень параметрів оцінювання точності*

Назва параметру	Розраховане значення
Перша архітектура ШНМ	
Precision	0.5
Recall	0.25
F-score	0.33
Accuracy	0.5
Друга архітектура ШНМ	
Precision	0.6
Recall	0.6
F-score	0.6
Accuracy	0.6

3. Повторити всі пункти завдання 5.2 та завдання 5.3 для другої архітектури нейронної мережі, змінивши відповідні місця робочого коду в файлах «Feedback\_training» та «confusion\_test», у відповідності до індивідуального завдання. Зображення для другої структури ШНМ зберігати під назвами «PR\_ANN\_structure\_2» і «PR\_Confusion\_test\_2», а натреновану мережу – з назвою «Net\_2», відповідно. Розраховані параметри для оцінювання точності класифікації додати до табл. 5.1.

4. Оформити звіт до практичної роботи.

### 5.3. Вимоги до оформлення звіту

1. Звіт повинен бути представлений в електронному вигляді. Вихідний файл звіту повинен бути збережений у форматі «\*.docx» та «\*.pdf».
2. Назва електронного файлу зі звітом подається у форматі: «AVS\_PR5\_Шифр групи\_Прізвище студента\_№варіанту».
3. Структура звіту повинна містити наступні елементи:
  - титульний аркуш;
  - мету роботи;
  - номер варіанту та індивідуальне завдання (у відповідності до табл. 5.2);
  - покроковий опис виконаних дій;
  - зображення з архітектурами штучних нейронних мереж;
  - зображення з матрицями похибок для кожної з архітектур;
  - таблицю зі значеннями розрахованих параметрів для оцінювання точності класифікації;
  - висновки по роботі.
4. Якість зображення та розмір шрифту на схемах, рисунках і таблицях повинні чітко відображати представлену в них інформацію. Усі рисунки, таблиці, схеми та розділи у звіті повинні бути підписані (вимоги див. у додатку А).
5. Файл звіту додається у загальну робочу папку (див. п. 5.2), яка надсилається викладачеві на перевірку (у «\*.zip» архіві). Для проходження перевірки робоча папка обов'язково повинна містити такі файли: Excel-файл з набором даних для тренування мережі, Excel-файл з набором даних для тестування мережі, Matlab-файли (створену БД, робочі коди, файли зі збереженими архітектурами ШНМ), зображення з результатами тестування (матриці похибок).

**Якщо оформлення звіту, робочих файлів і папки не відповідають вимогам, практична робота до захисту не допускається!**

## 5.4. Контрольні запитання

1. Що таке машинне навчання?
2. Які категорії машинного навчання ви знаєте? У чому між ними різниця?
3. Які бувають види «навчання з вчителем»?
4. Які основні методи машинного навчання ви знаєте?
5. Що штучна нейронна мережа? З яких компонентів вона складається?
6. Що таке нейрон, синапс і вага?
7. Що таке тренувальна серія та активаційна функція?
8. Які активаційні функції для ШНМ ви знаєте?
9. Що таке епоха, похибка та ітерація?
10. На що впливає вага синапсу?
11. Як в середовищі Matlab задати активаційну функцію? Наведіть приклад коду
12. Як в середовищі Matlab задати кількість нейронів у шарі ШНМ? Наведіть приклад коду

Таблиця 5.2. Початкові умови до завдання 5.2

№ варіанту	Набір TDF	Параметри для архітектури штучної нейронної мережі					
		Номер мережі	Тип і кількість нейронів	Активаційна функція першого шару	Активаційна функція другого шару	trainRatio / valRatio / testRatio	goal / epochs / max_fail
1	MAV, VAR, MAD	I	feedforwardnet(10)	tansig	tansig	[1/0/0]	[0.001/100/10000]
		II	feedforwardnet(20)	tansig	tansig		[0.00001/100/10000]
2	IAV, WL, MAD	I	feedforwardnet(5)	logsig	logsig	[1/0/0]	[0.01/100/10000]
		II	feedforwardnet(35)	logsig	logsig		[0.00001/100/10000]
3	MAD, Peak, VAR	I	feedforwardnet(50)	purelin	purelin	[1/0/0]	[0.01/100/10000]
		II	feedforwardnet(100)	purelin	purelin		[0.001/100/10000]
4	MAV, Peak, MAD	I	feedforwardnet(15)	logsig	logsig	[1/0/0]	[0.001/100/10000]
		II	feedforwardnet(40)	purelin	purelin		[0.00001/100/10000]
5	VAR, MAV, WL	I	feedforwardnet(20)	tansig	tansig	[1/0/0]	[0.00001/100/10000]
		II	feedforwardnet(70)	purelin	purelin		[0.00001/100/10000]
6	MAV, MAD, WL	I	feedforwardnet(25)	tansig	tansig	[1/0/0]	[0.001/100/10000]
		II	feedforwardnet(15)	logsig	logsig		[0.01/100/10000]
7	VAR, MAV, IAV	I	feedforwardnet(30)	logsig	logsig	[1/0/0]	[0.01/100/10000]
		II	feedforwardnet(80)	tansig	tansig		[0.000001/100/10000]

Продовження Таблиці 5.2. Початкові умови до завдання 5.2

<b>8</b>	MAD, IAV, Peak	I	feedforwardnet(33)	tansig	logsig	[1/0/0]	[0.001/100/10000]
		II	feedforwardnet(66)	purelin	purelin		[0.001/100/10000]
<b>9</b>	WL, Peak, MAV	I	feedforwardnet(22)	logsig	logsig	[1/0/0]	[0.001/100/10000]
		II	feedforwardnet(44)	purelin	logsig		[0.001/100/10000]
<b>10</b>	WL, VAR, Peak	I	feedforwardnet(11)	logsig	tansig	[1/0/0]	[0.01/100/10000]
		II	feedforwardnet(55)	purelin	purelin		[0.01/100/10000]

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ №6

## КЛАСИФІКАЦІЯ СИГНАЛІВ ЗА ДОПОМОГОЮ КАСКАДНОЇ НЕЙРОННОЇ МЕРЕЖІ

**Мета:** Ознайомитись з основними видами та архітектурою штучних нейронних мереж і особливістю побудови та тренування каскадної нейронної мережі прямого поширення у середовищі Matlab.

**Змістовність роботи:** основні види штучних нейронних мереж та їх знакові особливості. Методи оцінювання точності класифікації, ROC-крива, матриця похибок. Покроковий опис процесу проектування і тренування каскадної нейронної мережі у середовищі Matlab.

### 6.1. Теоретичні відомості

#### 6.1.1. Основні види та архітектури штучних нейронних мереж

Штучні нейронні мережі є одним із найбільш часто застосовуваних методів машинного навчання, що використовується для вирішення багатьох задач як у практиці аналізу сигналів, так і в обробці зображень, прогнозуванні, системах комп'ютерного зору тощо. Тому існує багато видів нейронних мереж, що відрізняються за архітектурою, призначенням та методами їх навчання. Коротко розглянемо деякі із них.

*Нейронні мережі прямого поширення* (FFNN-мережі) є найбільш прямолінійним класифікатором сигналів, у якому інформація передається поступово з входу до виходу мережі (див. п. 5.2). У FFNN нейрони одного шару не пов'язані між собою, однак мають зв'язок з нейронами у сусідніх шарах (рис. 6.1.a). Такі мережі зазвичай навчаються методом зворотного поширення помилки, що передбачає надання ШНМ необмеженої кількості вхідних і вихідних даних.

*Мережі-автокодувальники* (AE) – це різновид ШНМ, який за своєю сутністю подібний до FFNN, а за формою нагадує піщаний годинник. Основний принцип такої архітектури полягає в автоматичному кодуванні інформації мережею з метою її стиснення. У AE приховані симетричні шари, що менші за

розмірами вхідного та вихідного шарів (рис. 6.1.б). Навчання таких мереж зазвичай проводиться методом зворотного поширення помилки, а похибка такої ШНМ повинна розраховуватись як різниця між її входом та виходом.

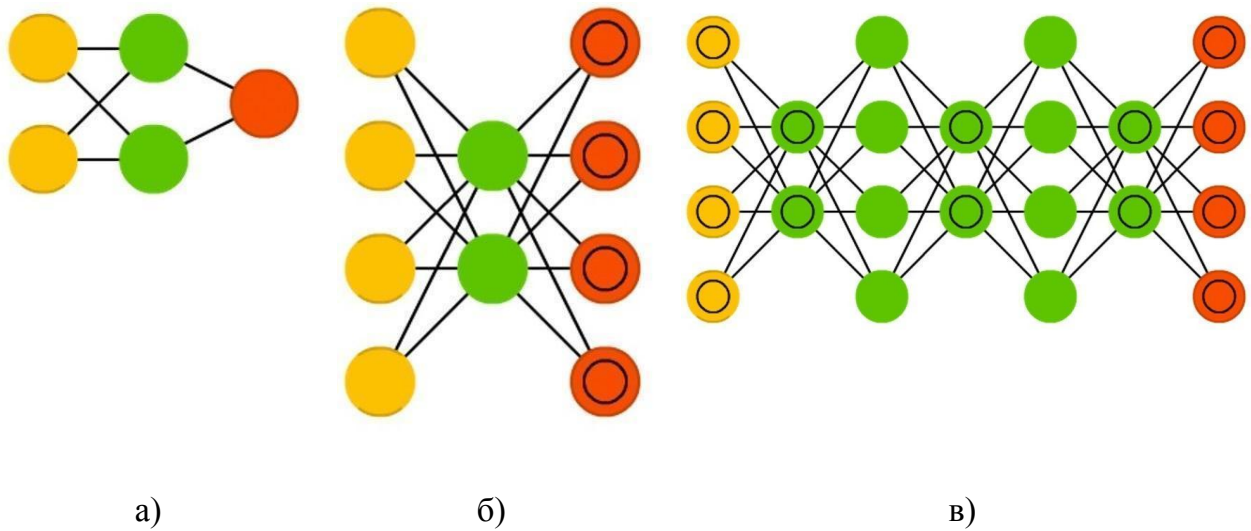


Рис. 6.1. Архітектура штучних нейронних мереж:

ШНМ прямого поширення (а); ШНМ-автокодувальник (б); Deep Belief-мережа (в)

**Deep Belief-мережі (DBN)** – це ШНМ, що навчаються по-блочно, причому кожному складовому блоку такої мережі для навчання потрібно лише правильно закодувати інформацію з попереднього блоку (рис. 6.1.в). Така техніка навчання отримала назву «жадібного навчання» і полягає у виборі локальних оптимальних рішень, які однак не гарантують оптимальний фінальний результат. За допомогою методу зворотного поширення помилки *DBN* можна навчити відображати інформацію у вигляді ймовірнісної моделі, а у випадку використання такої ШНМ для навчання без вчителя – стабілізовану модель можна використовувати і для генерування нового набору інформації.

**Нейронні мережі Хопфілда (HN)** – це нейронні мережі, що містять симетричну матрицю нейронних зв'язків. Кожен вузол такої мережі є її входом; у процесі навчання ці вузли відіграють роль прихованих нейронів, а на фінальному етапі – перетворюються на виходи системи (рис. 6.2.а). HN навчається за наступним алгоритмом: значення нейронів задаються у відповідності до бажаного шаблону, після чого розраховується вага синапсів, яка в подальшому не змінюється. Після того як HN навчилася на одному або декількох вхідних шаблонах, в подальшому вона завжди буде зводитися до

одного з таких шаблонів (хоча шаблон не завжди відповідатиме бажаному результату). Такі мережі стабілізуються в залежності від рівня загальної «енергії» та «температури» ШНМ, а у кожного нейрона є свій поріг активації при проходженні через який нейрон набуває одного з двох можливих значень  $[-1; 1]$ , іноді  $[0; 1]$ ).

**Мережа типу «машина Больцмана» (BM)** – це вид стохастичної нейронної мережі, яка за принципом свого функціонування дуже схожа на мережу Хопфілда. Однак в BM деякі нейрони позначені як вхідні, решта – як приховані (рис. 6.2.б). У процесі навчання такої мережі вхідні нейрони перетворюються на вихідні, а навчання відбувається за методом зворотного поширення помилки або алгоритмом порівняльної розбіжності.

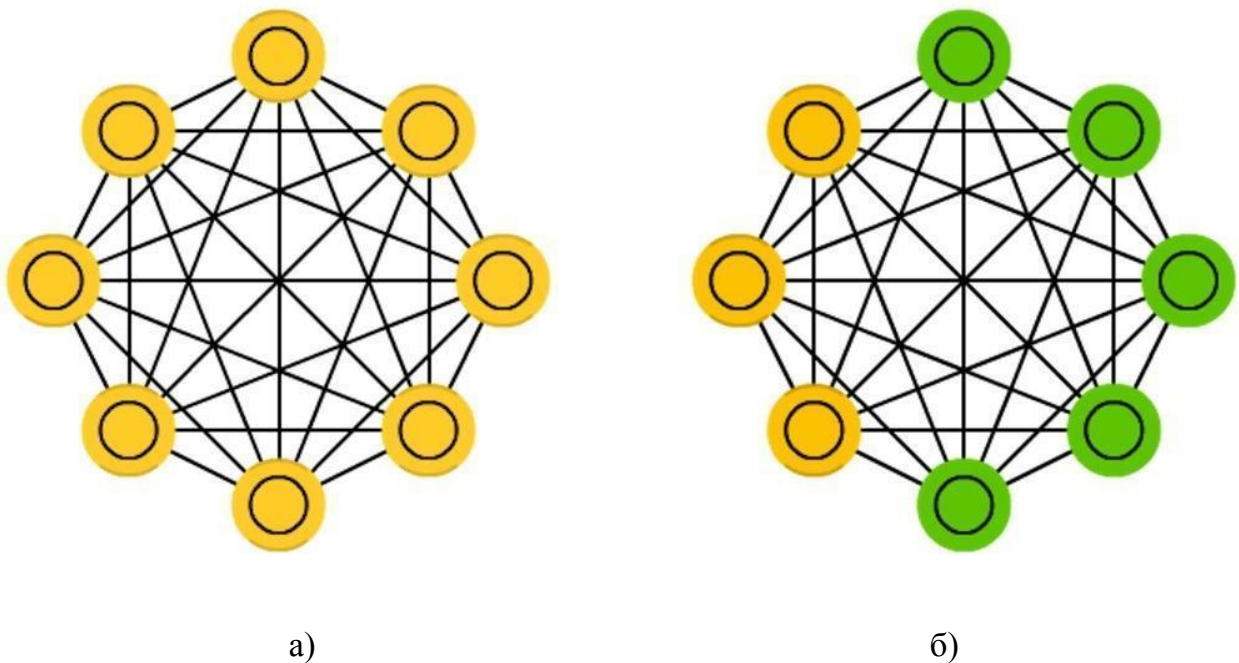


Рис. 6.2. Архітектура штучних нейронних мереж:  
ШНМ Хопфілда (а); Машина Больцмана (б)

**Згорткові нейронні мережі (CNN)** часто застосовуються для обробки вхідних зображень або аудіо сигналів. Такі мережі, як правило, використовують як «сканер», що здійснює поступову обробку вхідного набору даних «по шматкам». Наприклад, при аналізі деякого зображення розміром  $200 \times 200$  пікселів така ШНМ не аналізуватиме одразу повний масив інформації, а спочатку розрахує шмат розміром розміру  $20 \times 20$  пікселів, після



чого зміститься на один піксель і почне розрахунок нового вхідного набору. У CNN набір вхідної інформації передається через згорткові шари, вузли яких не завжди з'єднані між собою (рис. 6.3.а). Кількість нейронів у шарах CNN має властивість зменшуватись поступово з кожним наступним шаром (від вхідного до вихідного шару). Водночас, при побудові архітектури таких ШНМ зменшення кількості нейронів у їх шарах здійснюється в більшості випадків за спадаючою зі степеню двійки: 32, 16, 8, 4 тощо.

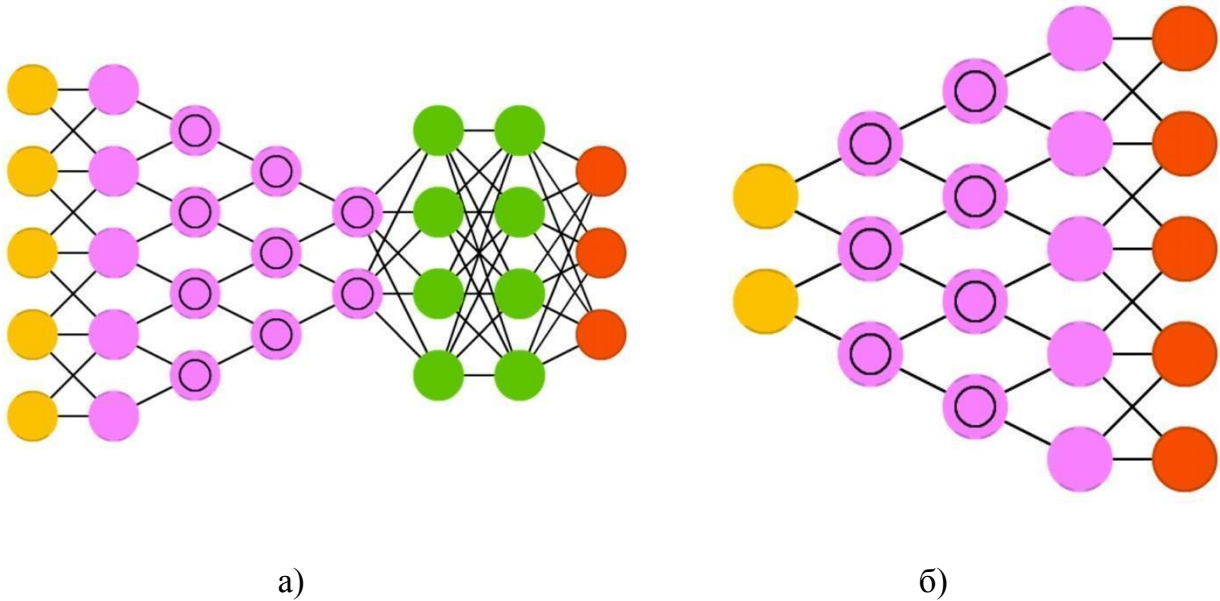


Рис. 6.3. Архітектура штучних нейронних мереж:  
згорткова нейронна мережа (а); деконволюційна ШНМ (б)

**Деконволюційні нейронні мережі (DCN)** – це ШНМ, які є зворотними до CNN. Як видно з рис. 6.3.б, кількість нейронів у такій мережі поступово збільшується зі входу до виходу, що й відрізняє її від інших архітектур. Принцип роботи такої мережі полягає у генеруванні нею декількох вихідних варіантів на один вхідний запит. Наприклад, при заданні на вхід DCN одного слова-запиту (скажімо, **«цифровий сигнал»**) така мережа на виході може генерувати набір зображень, на яких буде візуалізовано різноманітні сигнали цифрової форми. Варто зауважити, що в більшості випадків на вихід DCN передається бінарний вектор, що відповідає певному набору варіантів, наприклад:  $\langle 0; 1 \rangle$  – цифровий сигнал,  $\langle 1; 0 \rangle$  – аналоговий сигнал, а  $\langle 1; 1 \rangle$  – детермінований сигнал тощо.

**Каскадні нейронні мережі прямого поширення (CFNN)** за принципом функціонування та тренування є подібними до FFNN але мають вагові зв'язки (синапси), які з'єднують нейрони вхідного шару не лише з наступним, а й з кожним з шарів у ШНМ (рис. 6.4).

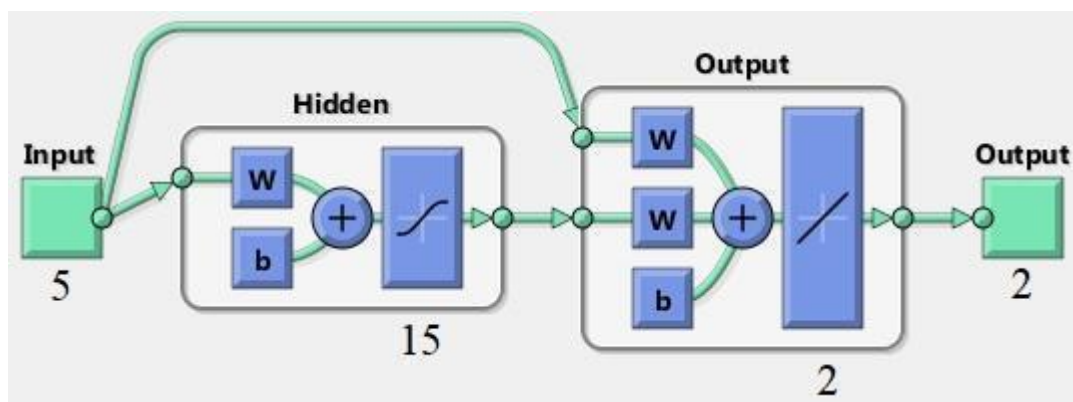


Рис. 6.4. Приклад архітектури каскадних нейронних мереж прямого поширення

У результаті швидкість тренування CFNN є вищою за штучні нейронні мережі прямого поширення і в окремих випадках дозволяє досягати вищої точності класифікації. Рівняння, що описують модель CFNN можна записати у вигляді формули (6.1):

$$y = \sum_{i=1}^n f^i \omega_i^i x_i + f^0 \left( \sum_{j=1}^k \omega_j^0 f_j^h \left( \sum_{i=1}^n \omega_{ji}^h x_i \right) \right), \quad (6.1)$$

де  $f^i$  – функція активації між вхідним та вихідним шаром, а  $\omega_i^i$  – вага синапса між вхідним та вихідним шаром.

### 6.1.2. Оцінювання результатів класифікації. ROC-крива і матриця похибок

При проектуванні та тренуванні архітектури штучної нейронної мережі важливо оцінити проміжну і результуючу ефективність класифікації, яку можна отримати при її застосуванні. Оцінювання точності класифікації найчастіше здійснюється шляхом використання параметрів ROC-кривої або матриці

похибок (МП). **Метод ROC-кривої** відображає співвідношення між часткою вірно класифікованих та помилково-позитивних результатів, отриманих під час тестування мережі. Кількісна інтерпретація ROC представляється у вигляді розрахунку площі під ROC-кривою, яка обмежується віссю помилково-позитивних відповідей (рис. 6.5), а саме показника AUC (area under ROC curve).

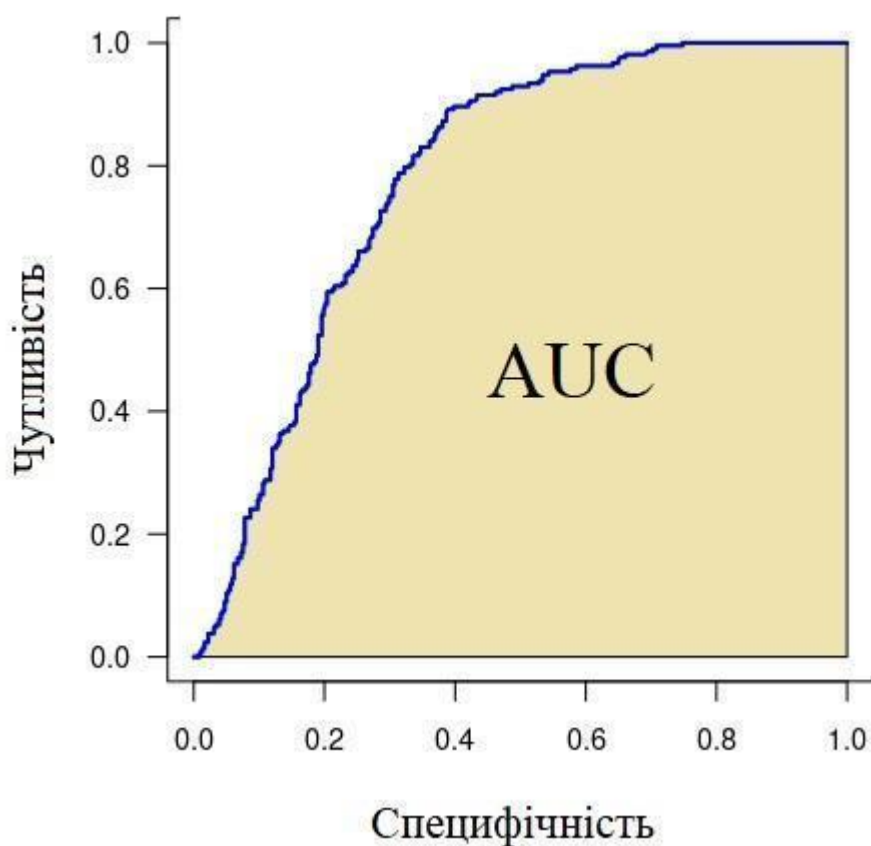


Рис. 6.5. Приклад графіку ROC-кривої

Чим більшим є показник AUC, тим вища точність результатів класифікації, наданих ШНМ. Як видно з рис. 6.5 для розрахунку показника AUC необхідно враховувати також параметри чутливості та специфічності мережі. Чутливість ROC-кривої є співвідношенням кількості вірно-позитивних результатів до загальної кількості можливих класів, якими оперує мережа. А специфічність це співвідношення помилково-позитивних категорій до загальної кількості можливих класів.

**Метод матриці похибок**, на відміну від розрахунку ROC-кривої, дозволяє здійснити одночасну візуалізацію та аналіз результатів не лише для

всього набору класів загалом, а й кожному з класів окремо в межах одного графіку (матриці). Матриця похибок (МП) представляє собою матрицю розміром N на M класів (рис. 6.6), де колонки є набором реальних класів, які повинна визначити нейронна мережа, а рядки – фактичними класами, які були визначені ШНМ як реальні.

Реальні класи	$N_1$	$P_{11}$ $Q_{11}$	$P_{12}$ $Q_{12}$	$P_{13}$ $Q_{13}$	$P_{14}$ $Q_{14}$
	$N_2$	$P_{21}$ $Q_{21}$	$P_{22}$ $Q_{22}$	$P_{23}$ $Q_{23}$	$P_{24}$ $Q_{24}$
	$N_3$	$P_{31}$ $Q_{31}$	$P_{32}$ $Q_{32}$	$P_{33}$ $Q_{33}$	$P_{34}$ $Q_{34}$
	$\dots$				
	$N_n$	$P_{n1}$ $Q_{n1}$	$P_{n2}$ $Q_{n2}$	$P_{n3}$ $Q_{n3}$	$P_{nm}$ $Q_{nm}$
		$M_1$	$M_2$	$M_3 \dots M_m$	
		Фактичні класи			

Рис. 6.6. Приклад матриці похибок

Під час розрахунку та заповнення МП фактично визначені результати розташовуються у рядку і співвідносяться зі стовпцем реальних класів, для яких паралельно підраховується кількість вірно позитивних (TP), помилково позитивних (FP) та помилково негативних відповідей (FN).

$$Precision = \frac{TP}{TP+FP}, \quad (6.2)$$

$$Recall = \frac{TP}{TP+FN}, \quad (6.3)$$

$$F - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (6.4)$$

Ефективність МП найчастіше оцінюється шляхом розрахунку трьох параметрів: Precision, Recall і F-score (6.2) – (6.4).

## 6.2. Завдання для виконання

Створити робочу папку з назвою: **AVS\_PR6\_Шифр групи\_Прізвище студента\_№варіанту** (де PR6 розшифровується як: «Практична робота №6»). Назва папки задається латиницею (англомовними символами)! Використовуючи шаблони, створені у четвертій практичній роботі за аналогічним алгоритмом, застосувати відповідні фільтри сигналів і розрахувати набір значень магнітуди для всіх ітерацій тестових сигналів (див. п. 5.2). Створити зведену таблицю з розрахованими значеннями магнітуди всіх ітерацій сигналів до та після застосування фільтру на частоті [1Гц, 2Гц, 5Гц, 7Гц, 8Гц] і розмістити їх у протоколі до практичної роботи (див. табл 4.2).

**Завдання 6.1.** Підготувати вхідний набір інформації для подальшої роботи зі штучною нейронною мережею. Створити БД для подальшої класифікації сигналів, базуючись на частотних характеристиках визначених для *початкової форми* сигналів.

### Порядок виконання Завдання 6.1.

1. У середовищі Matlab обрати створену робочу папку до шостої практичної роботи, а у вікні «Workspace» створити новий файл БД з назвою «input\_FFT», у якому буде розміщуватись вхідний набір розрахованих значень магнітуди для початкової форми сигналів (частотних характеристик).

2. Скопіювати у перші п'ять стовпчиків файлу «input\_FFT» набір часових характеристик для всіх ітерацій першого сигналу, у наступні п'ять стовпчиків значення характеристик для другого сигналу тощо (рис. 6.7).

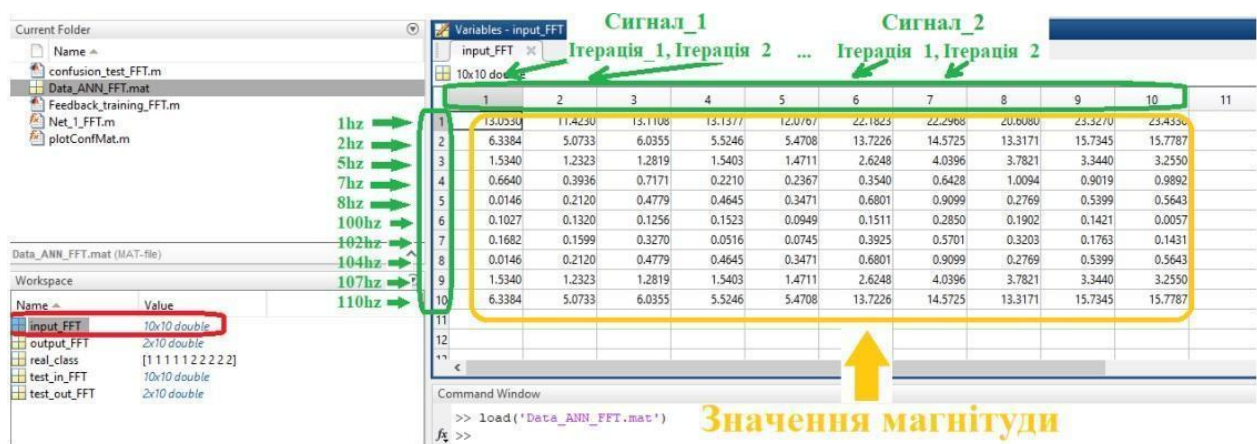


Рис. 6.7. Приклад заповнення файлу «input\_FFT»

Подібні дії провести для всіх сигналів, що досліджувались студентом у попередніх роботах. Варто зазначити, що у файлі «input\_FFT» стовпці відповідають номеру відповідної ітерації, а рядки – значенню частоти по порядку (перший рядок – 1 Гц, другий рядок – 2 Гц, третій рядок – 5 Гц тощо).

3. У вікні «Workspace» створити новий файл БД з назвою «output\_FFT», у якому будуть розміщуватись вихідні дані (набір відповідей), що необхідні для тренування нейронної мережі. Додати до файлу «output» набір бінарних значень (0 та 1), що визначатимуть відповідність часових характеристик з файлу «input\_FFT» класам, які повинна визначити нейронна мережа (див. п.3 завдання 5.1).

4. У вікні «Workspace» створити файл з назвою «real\_class», у якому буде розміщуватись реальний перелік класів, яким відповідає кожна ітерація сигналу. **Наприклад:** 1 – перший сигнал, 2 – другий сигнал, 3 – третій сигнал тощо (див. п.4 завдання 5.1).

5. У вікні «Workspace» створити новий файл БД з назвою «test\_in\_FFT», у якому будуть розміщуватись вхідні дані для тестування нейронної мережі. Для цього додати у файл (див. п.2 завдання 6.1) розраховані значення магнітуди початкової форми тестових сигналів.

6. У вікні «Workspace» створити новий файл БД з назвою «test\_out\_FFT», у якому будуть розміщуватись вихідні дані (набір відповідей), необхідні для тестування нейронної мережі. Додати до файлу «test\_out\_FFT»

набір бінарних значень (0 та 1), що визначатимуть відповідність частотних характеристик з файлу «test\_in\_FFT» класам, які повинна визначити нейронна мережа при тестуванні (див. п.3 завдання 5.1).

7. Натиснути ПКМ на полі «Workspace» → «Save» і зберегти всі створені файли БД з назвою «Data\_ANN\_FFT» та розширенням «\*.mat» у створеній робочій папці.

8. Помістити у робочу папку файли «confusion\_test\_FFT», «Feedback\_training\_FFT» та «plotConfMat», які надаються викладачем перед початком практичної роботи.

**Завдання 6.2.** Адаптувати робочий код та провести налаштування параметрів штучної нейронної мережі для подальшої класифікації на основі частотних характеристик сигналу. Здійснити тренування ШНМ згідно заданих параметрів.

**Порядок виконання Завдання 6.2.**

1. Відкрити файл «Feedback\_training\_FFT» і задати в ньому параметри тренування нейронної мережі у відповідності до індивідуального завдання (аналогічно п.1 завдання 5.2).

2. Задавши всі необхідні параметри натиснути кнопку «Run», розміщену у верхній частині робочого вікна середовища Matlab. Якщо робочий код з параметрами мережі задано без помилок, то програмою буде запущено вікно тренування мережі (рис. 6.8). Після завершення процесу вікно з параметрами тренування необхідно зберегти у форматі зображення (зробити screenshot) з назвою «PR6\_ANN\_FFT\_structure\_1».

3. Орієнтуючись на результати (показник F-score) здійснити ще п'ять тренувальних спроб (запусків коду на моделювання) з незмінними параметрами ШНМ для того ж тренувального набору даних.

4. Зберегти в окремий файл ШНМ мережу з найкращим результатом тренування.



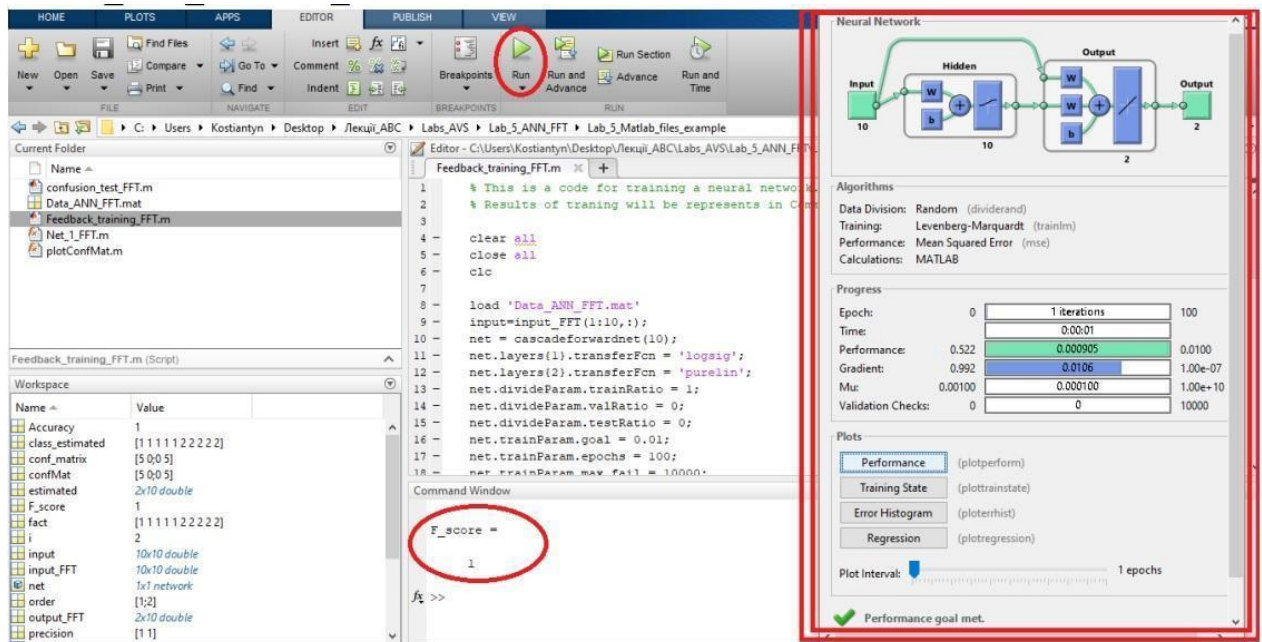


Рис. 6.8. Приклад вікна тренування каскадної нейронної мережі у середовищі Matlab

Для збереження нейронної мережі після закінчення процесу її тренування у вікні Command Window необхідно ввести такий код:

«genFunction(net,'C:\Users\Kostiantyn\Desktop\PR\_AVS\PR6\_ANN\_FFT\Net\_1\_FFT.m');»

Де:

- C:\Users\Kostiantyn\Desktop\PR\_AVS\PR6\_ANN\_FFT - шлях до створеної папки, у яку буде збережено мережу.
- «Net\_1\_FFT» - назва штучної нейронної мережі.

Якщо код введено без помилок, то у вікні Command Window з'явиться повідомлення про успішне збереження ШНМ, а файл з мережею з'явиться у робочій папці.

**Завдання 6.3.** Адаптувати робочий код та провести тестування збереженої моделі штучної нейронної мережі на наборі даних, що не використовувались при її тренуванні.

### Порядок виконання Завдання 6.3.

1. Відкрити файл «confusion\_test\_FFT», задати в ньому назву збереженої у завданні 6.2 штучної нейронної мережі (аналогічно з п.1 завдання



5.3) і запустити процес її тестування (натиснувши кнопку «Run»). Якщо робочий код адаптовано без помилок, програмою буде візуалізовано матрицю похибок і розраховано чотири параметри для оцінювання точності класифікації.

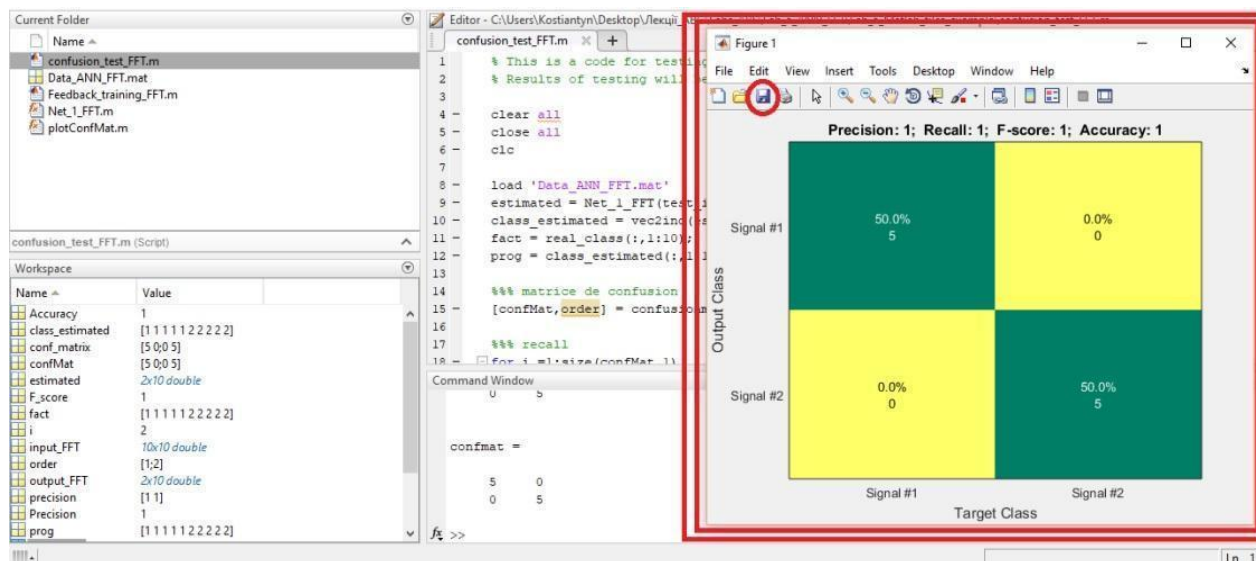


Рис. 6.9. Приклад матриці похибок розрахованої для каскадної нейронної мережі

2. Зберегти отриману матрицю похибок у форматі «\*.bmp» з назвою «PR6\_Confusion\_FFT\_test\_1», а розраховані параметри для оцінювання точності класифікації подати у вигляді таблиці 6.1 в протоколі до практичної роботи (оформленої аналогічно до табл. 5.1).

3. Повторити всі пункти завдання 6.2 та завдання 6.3 для другої архітектури ШНМ, змінивши відповідні місця робочого коду в файлах «Feedback\_training\_FFT» та «Feedback\_training\_FFT» у відповідності до індивідуального завдання. Зображення для другої структури ШНМ зберігати під назвами «PR6\_ANN\_FFT\_structure\_2» і «PR6\_Confusion\_FFT\_test\_2», а натреновану мережу з назвою «Net\_2\_FFT», відповідно. Розраховані параметри для оцінювання точності класифікації додати до табл. 6.1.

**Завдання 6.4.** Здійснити класифікацію за допомогою ШНМ, базуючись на частотних характеристиках визначених для сигналів *після застосування фільтру*.

#### Порядок виконання Завдання 6.4.

1. Повторити всі пункти завдання 6.1 - 6.3 для значень частотних

характеристик, які розраховані для **відфільтрованої форми** сигналів. Назви збережених зображень і натренованих архітектур ШНМ задавати зі словом «\_filtered». Наприклад, «PR6\_ANN\_FFT\_structure\_1\_filtered», «PR6\_Confusion\_FFT\_test\_1\_filtered», «Net\_1\_FFT\_filtered» тощо. Розраховані параметри для оцінювання точності класифікації відфільтрованих сигналів додати до табл. 6.2.

2. Оформити звіт до практичної роботи.

### **Вимоги до оформлення звіту**

1. Звіт повинен бути представлений в електронному вигляді. Вихідний файл звіту повинен бути збережений у форматі «**\*.docx**» та «**\*.pdf**».
2. Назва електронного файлу зі звітом подається у форматі: «AVS\_PR6\_Шифр групи\_Прізвище студента\_№варіанту».
3. Структура звіту повинна містити наступні елементи:
  - титульний аркуш;
  - мету роботи;
  - номер варіанту та індивідуальне завдання (у відповідності до табл. 6.1);
  - покроковий опис виконаних дій;
  - **зображення з архітектурами штучних нейронних мереж;**
  - **зображення з матрицями похибок для кожної з архітектур;**
  - **таблиці зі значеннями розрахованих параметрів для оцінювання точності класифікації (для початкової і відфільтрованої форми сигналів);**
  - висновки по роботі.
4. Якість зображення та розмір шрифту на схемах, рисунках і таблицях повинні чітко відображати представлену в них інформацію. Усі рисунки, таблиці, схеми та розділи у звіті повинні бути підписані (вимоги див. у додатку А).
5. Файл звіту додається у загальну робочу папку (див. п. 6.2), яка надсилається викладачеві на перевірку (у «\*.zip» архіві). Для проходження перевірки робоча папка обов'язково повинна містити такі файли: Matlab-файли

(створену БД, робочі коди, файли зі збереженими архітектурами ШНМ), зображення з результатами тестування (матриці похибок), протокол до практичної роботи у двох форматах.

**Якщо оформлення звіту, робочих файлів і папки не відповідають вимогам, практична робота до захисту не допускається!**

#### **6.4. Контрольні запитання**

1. Які основні види нейронних мереж ви знаєте?
2. Чим відрізняються FFNN та CFNN мережі?
3. У чому полягає різниця між згортковими та деконволюційною мережею?
4. Що таке ШНМ Хопфілда? У чому її особливість?
5. Що таке Deep Belief - мережі?
6. Як можна оцінити точність класифікації сигналів, отриману за допомогою ШНМ?
7. Що таке ROC-крива? Як виражається її кількісна інтерпретація?
8. Що таке чутливість та специфічність ROC-кривої?
9. Що таке матриця похибок? За якими кількісними параметрами вона оцінюється?
10. Що таке TP, FP, FN відповіді?

Таблиця 6.1. Початкові умови до шостої практичної роботи

№ варіанту	Набір TDF	Параметри для архітектури штучної нейронної мережі					
		Номер мережі	Тип і кількість нейронів	Активаційна функція першого шару	Активаційна функція другого шару	trainRatio / valRatio / testRatio	goal / epochs / max_fail
1	MAV, VAR, MAD	I	cascadeforwardnet(30)	tansig	tansig	[1/0/0]	[0.001/100/10000]
		II	cascadeforwardnet(50)	tansig	tansig		[0.00001/100/10000]
2	IAV, WL, MAD	I	cascadeforwardnet(25)	logsig	logsig	[1/0/0]	[0.01/100/10000]
		II	cascadeforwardnet(45)	logsig	logsig		[0.00001/100/10000]
3	MAD, Peak, VAR	I	cascadeforwardnet(5)	purelin	purelin	[1/0/0]	[0.01/100/10000]
		II	cascadeforwardnet(10)	purelin	purelin		[0.001/100/10000]
4	MAV, Peak MAD	I	cascadeforwardnet(25)	logsig	logsig	[1/0/0]	[0.001/100/10000]
		II	cascadeforwardnet(45)	purelin	purelin		[0.00001/100/10000]
5	VAR, MAV, WL	I	cascadeforwardnet(30)	tansig	tansig	[1/0/0]	[0.00001/100/10000]
		II	cascadeforwardnet(50)	purelin	purelin		[0.00001/100/10000]
6	MAV, MAD, WL	I	cascadeforwardnet(15)	tansig	tansig	[1/0/0]	[0.001/100/10000]
		II	cascadeforwardnet(45)	logsig	logsig		[0.01/100/10000]
7	VAR, MAV, IAV	I	cascadeforwardnet(20)	logsig	logsig	[1/0/0]	[0.01/100/10000]
		II	cascadeforwardnet(60)	tansig	tansig		[0.000001/100/10000]

Продовження Таблиці 6.1. Початкові умови до шостої практичної роботи

<b>8</b>	MAD, IAV, Peak	I	cascadeforwardnet(11)	tansig	logsig	[1/0/0]	[0.001/100/10000]
		II	cascadeforwardnet(55)	purelin	purelin		[0.001/100/10000]
<b>9</b>	WL, Peak, MAV	I	cascadeforwardnet(15)	logsig	logsig	[1/0/0]	[0.001/100/10000]
		II	cascadeforwardnet(20)	purelin	logsig		[0.001/100/10000]
<b>10</b>	WL, VAR, Peak	I	cascadeforwardnet(35)	logsig	tansig	[1/0/0]	[0.01/100/10000]
		II	cascadeforwardnet(65)	purelin	purelin		[0.01/100/10000]

## ДОДАТКИ

### ДОДАТОК А. ВИМОГИ ДО ОФОРМЛЕННЯ ПРАКТИЧНИХ РОБІТ

У додатку наведено вимоги та представлено приклади оформлення основних складових елементів протоколу до практичної роботи. Порядок нумерації рисунків, таблиць, формул подано у додатку як приклад.

#### Оформлення основного тексту в протоколі

**Шрифт:** Times New Roman 14 пт, **інтервал:** 1.5, **абзац:** перший рядок 1.25 см. **Вирівнювання основного тексту:** по ширині, **вирівнювання заголовків:** посередині, **вирівнювання підзаголовків:** по ширині (з абзацу). Назви заголовків, підзаголовків, розділів та підрозділів виконуються жирним шрифтом.

#### Оформлення рисунків та підписів до них

**Розміщення рисунку:** посередині, **розміщення підпису рисунку:** знизу, посередині. Рисунок підписується у форматі: «Рис. № практичної роботи.№ рисунку. Назва рисунку» звичайним шрифтом. Наприклад:



Рис. 1.1. Початкова форма сигналу

## Оформлення таблиць та підписів до них

**Розміщення таблиці:** зверху, вирівнювання по правому краю. Текст у таблиці виконується шрифтом Times New Roman 14 пт, або Times New Roman 12 пт, **інтервал:** одиничний.

Підписи таблиць виконуються у форматі: «Таблиця № практичної роботи. № таблиці. Назва таблиці» звичайним шрифтом. Наприклад:

Таблиця 1.1. Розраховані коефіцієнти

Назва коефіцієнту	Числове значення		
a	1	-1,6909	0,73272
b	0,0104	0,0208	0,01043

## Оформлення нумерації формул

**Розміщення формул:** посередині листа. **Нумерація:** наскрізна, розміщується по правому краю листа, у форматі: «(№ практичної роботи.№ формули)». Наприклад:

$$\Delta x = \frac{2x_{max}}{N} \quad (1.1)$$

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 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.
2. Комп'ютерне моделювання процесів та систем. Чисельні методи : підручник / С.П. Вислоух, О.В. Волошко, Г.С. Тимчик, М.В. Філіппова. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. – 228 с. ISBN 978-966-990-028-9.
3. Андруник В.А. Чисельні методи в комп'ютерних науках:навчальний посібник / В.А. Андруник, В.А.Висоцька, В.В.Пасічник, Л.Б. Чирун Л.Б., Л.В. Чирун – Львів: Видавництво «Новий світ -2000», 2017. – 470с. ISBN 978-617-7519-06-4.
4. Перетворення Фур'є, Лапласа: узагальнення та застосування: навч.-метод. посіб. [для студентів та аспірантів мех.-мат. і природн. спец. ВНЗ] / Г. П. Лопушанська, А. О. Лопушанський, О. М. М'яус ; М-во освіти і науки України, Львів. нац. ун-т ім. І. Франка. — Львів: ЛНУ, 2014. — 152 с. — Бібліогр.: с. 147—152 (90 назв).
5. Кветний Р. Н. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 2 : навчальний посібник / [Р. Н. Кветний, І. В. Богач, О. Р. Бойко та інші]; за заг. ред. Р. Н. Кветного. – Вінниця : ВНТУ, 2013. – 235 с. ISBN 978-966-641-521-2 (частина 2).
6. Майданюк В. П. Обробка сигналів: навчальний посібник / В. П. Майданюк, А. М. Петух. – Вінниця : ВНТУ, 2012. – 144.
7. Основи теорії електронних кіл: Підручник (друге видання: доопрацьоване і доповнене) / Ю.Я. Бобало, Б.А. Мандзій, П.Г. Стахів, Л.Д. Писаренко, Ю.І. Якименко; За ред. проф. Ю.І. Якименка. – Київ: Видавництво Національного технічного університету України “Київський політехнічний інститут”, 2011. – 332 с.



8. Коваль, О. А. Методи синтезу та аналізу вимірних сигналів : конспект лекцій [Електронний ресурс] / О. А. Коваль ; Харків. нац. автомоб.-дор. ун-т. - Харків, 2018. - 367 с.

9. Теорія сигналів. Методичні вказівки до практичних занять та самостійної роботи. Частина 2. Спектральний аналіз періодичних сигналів. [Електронний ресурс]: навч. посіб. для студентів вищих навчальних закладів спеціальності 125 «Кібербезпека», освітня програма «Системи технічного захисту інформації» / КПІ ім. Ігоря Сікорського; уклад.: С. М. Куш, Д. О. Прогонов, Смирнов В.П.. – Електронні текстові дані (1 файл: 0,3 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2019. – 36 с.

10. NIELSEN, Michael A. Neural networks and deep learning. San Francisco, CA, USA: Determination press, 2015, 225 p.