МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни
„Цифрова обробка сигналів та зображень”
для студентів спеціальності
172 Електронні комунікації та радіотехніка
174 Автоматизація, комп’ютерно інтегровані технології та робототехніка
усіх форм навчання

2024

–

Методичні вказівки до лабораторних робіт з дисципліни „Цифрова обробка сигналів та зображень” для студентів спеціальності 172 Електронні комунікації та радіотехніка, 174 Автоматизація, комп’ютерно інтегровані технології та робототехніка усіх форм навчання, Укл. М.В. Єфименко – Запоріжжя : НУЗП, 2024. – 44с.

Укладач: М. Єфименко, доцент, д.т.н.

Рецензент: Н.О. Миронова, доцент, к.т.н.

Відповідальний за випуск: Т.І. Куляба, старший викладач.

Затверджено
на засіданні кафедри ІТЕЗ
протокол № 1 від 10.09.24 р.

Рекомендовано до видання на
засіданні НМК ФІБЕК
протокол № 2 від 19.09.24 р.

Зміст

Лабораторна робота 1.

Вивчення основ роботи з пакетом для математичних та інженерних розрахунків MATLAB

Мета роботи – ознайомитися з системою **MATLAB** та вивчити основні принципи роботи з цією системою.

1.1. Теоретичні відомості

1.1.1. Робочий простір системи MATLAB і її командне вікно

Після запуску програми MATLAB на екрані з'являється командне вікно системи MATLAB, що містить меню, лінійку з кнопками і клієнтську частину із знаком запрошення >>. Після знака >> можна вводити з клавіатури числа, імена змінних і знаки операцій, що складають деякий вираз. Імена змінних повинні починатися з букви і складатися з послідовності букв, цифр і знаків підкреслення. Найпростіші знаки операцій: **+, -, *, /, =.** Після натиснення клавіші **Enter** обчислюється вираз і результат виводиться на екран, як це показано на рис. 1.1.

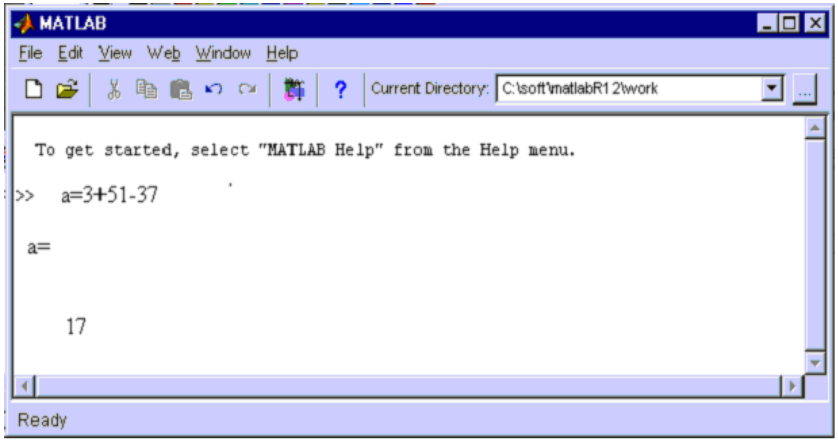


Рисунок 1.1

Обчислення повторюються багато разів: вводяться з клавіатури нові числові дані, змінні і нові символьні вирази, що показуються в командному вікні. Якщо не вистачає вільного міста для цих даних, то використовується *вертикальна прокрутка,* що зсовує рядки вгору, а в знизу вікна з'являється вільний рядок для введення нових даних і знак >>. Для поглядання всіх рядків програми використовується *смуга протяжки*. (рис. 1.2).

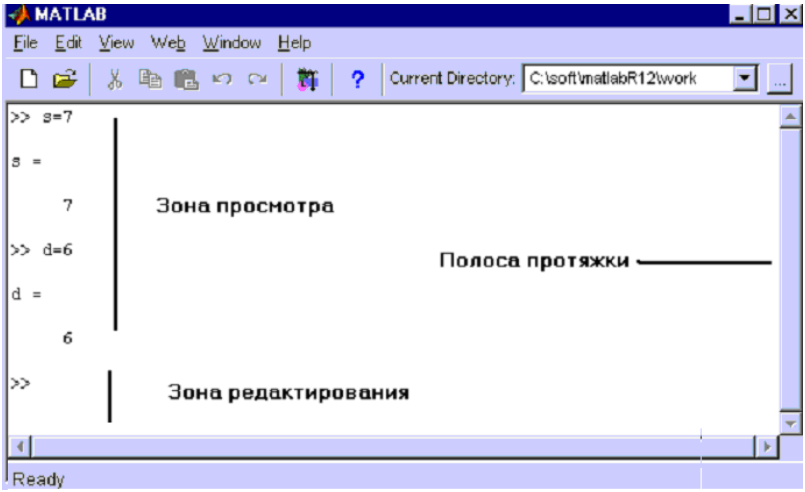


Рисунок 1.2

Можна також здійснювати протяжку змісту командного вікна системи **MATLAB** за допомогою клавіш курсору, **`PageUp`**, **`PageDown`**, **`Ctrl+Home`**, **`Ctrl+End`**. Натискання клавіші „↑” дозволяє повернути в рядок раніше введені команди і іншу вхідну інформацію, яка запам'ятовується в стеку команд. Клавіша „↓” здійснює прокрутку команд в протилежному напрямі.

Вся видима інформація у вікні системи **MATLAB** розташовується в двох різних зонах: *перегляду* і *редагування*. В зоні *перегляду* нічого не можна виправити; хоча в неї і можна помістити курсор, проте реакцією на введення з клавіатури буде автоматичне переміщення курсору в рядок введення, розташований в зоні редагування. В зоні *перегляду* можна виділяти за допомогою миші будь-яку інформацію і копіювати її в буфер обміну (**`Clipboard`**), щоб потім вставити її або в текст, або в рядок введення. Зона *редагування* займає один рядок командного вікна системи **MATLAB**, в якому знаходиться знак запрошення >> та називається *рядком введення*, який може займати декілька фізичних рядків командного вікна. Для цього не можна просто натискувати клавішу **`Enter`**, оскільки при цьому введення інформації буде закінчено і **MATLAB** приступить до обчислень та подальшого показу результату. Для продовження введення з показом видимої інформації на наступних фізичних рядках треба набрати три або більше крапок, а потім натиснути **`Enter`**, що і показано на рис. 1.3.

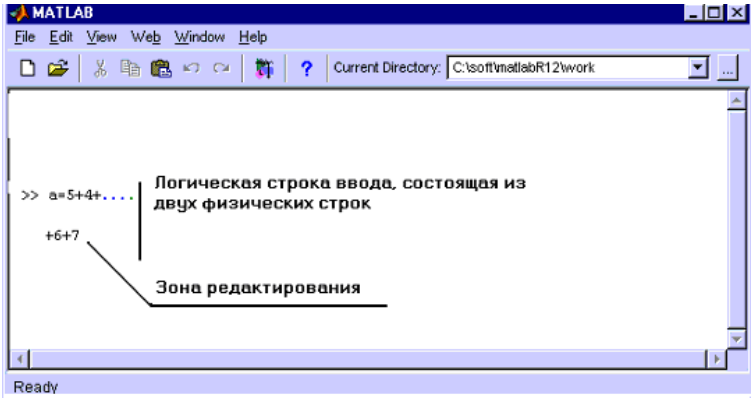


Рисунок 1.3

Проте і в цьому випадку зона *редагування* розповсюджується тільки на останній рядок і вона вже не містить знака запрошення >>, а в попередніх фізичних рядках рядка *введення* змінити нічого не можна. Крім того, сумарна довжина рядка введення обмежена 256 символами.

Всі значення змінних, обчислені протягом поточного сеансу роботи, зберігаються в спеціально області пам'яті, яка називається *робочим простором системи* **MATLAB** (по-англійські – *Matlab Workspace*).

Командою **`clc`** можна стерти видимий зміст командного вікна системи **MATLAB**, проте це не торкнеться змісту робочого простору. Якщо після цього набрати ім'я раніше обчисленої змінної **`a`**, то після натиснення **`Enter`** знову побачимо її значення:

```
>>a
a =
20
```

При збільшенні розміру робочого простору ефективність роботи знижується. Тому, коли зникає необхідність в зберіганні змінних в поточному сеансі роботи, можна стерти з пам'яті комп'ютера змінні з іменами **`имя1`** і **`имя2`** командою

```
clear имя1 имя2
```

Щоб видалити відразу всі змінні, потрібно використовувати команду

```
clear.
```

Якщо ви хочете перевірити, які змінні залишилися в робочому просторі, то для цього потрібно виконати команду **`who`**,

яка виведе список всіх змінних, що входять на даний момент в робочий простір системи **MATLAB** (рис. 1.4).

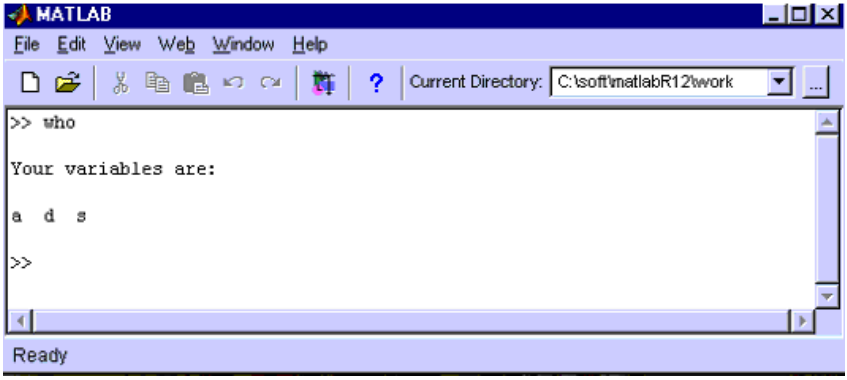


Рисунок 1.4

Для перегляду значення будь-якої змінної з поточного робочого простору системи **MATLAB** достатньо набрати її ім'я і натискувати **Enter**. Після закриття сеансу роботи з системою **MATLAB** всі раніше обчислені змінні втрачаються. Щоб зберегти у файлі зміст робочого простору, потрібно виконати команду меню **File|Save Workspace As** та задати каталог на диску і ім'я файлу в стандартному діалоговому вікні ОС Windows. Розширення імені файлу повинне бути **mat**, тому такі файли прийнято називати **MAT-файлами**. Також можна безпосередньо в командному вікні системи MATLAB набрати команду: **save путь_к_файлу\имя_MAT-файла** та натиснути **Enter**. Результат буде той же самий. В подальших сеансах роботи для завантаження раніш збереженого на диску **MAT-файла** потрібно виконати команду меню **File|Load Workspace** та вказати потрібний **Мат-файл** в діалоговому вікні **Load.mat_file**. Якщо виконати цю команду кілька разів з різними файлами, то можна з'єднати зміст декількох попередніх сеансів роботи. Але, якщо імена змінних з різних сеансів співпадають, в поточному робочому просторі буде представлена змінна з останнього відкритого **Мат-файлу**. Замість розглянутої команди меню можна в командному вікні системи **MATLAB** набрати команду **load имя_MAT-файла**. Можна також із записаного на диску **Мат-файлу** ввести в робочий простір значення окремих змінних. Для цього потрібно попередню команду доповнити іменами змінних:

load имя_MAT-файла имя1, имя2

В результаті з **Мат-файлу** будуть введені змінні з іменами **имя1**, **имя2** і т.д. Якщо **Мат-файл** вказано без повного шляху до нього, то він повинен знаходитися в поточному каталозі системи **MATLAB**, який завжди можна визначити за допомогою команди **cd**, а змінити його можна командою **cd шлях_до_нового_каталогу.**

В кожному сеансі роботи з системою **MATLAB** доцільно поточним каталогом задавати той каталог, з файлами якого належить працювати частіше. Відзначимо деякі особливості команд системи **MATLAB**. Під командами користувача ми розуміємо вказівку системі виконати деяку дію, наприклад, показати поточний каталог.

Основні команди системи **MATLAB** можна задавати різними способами: за допомогою меню головного вікна або кнопок на смузі інструментів, введенням з клавіатури ключових слів з подальшим натисненням **Enter**. Деякі команди можна реалізувати тільки введенням з клавіатури відповідних їм ключових слів (наприклад, команда **cd**). Частина команд вимагає додаткової інформації від користувача. Наприклад, команда **clear** використовує імена змінних, що підлягають видаленню з поточного робочого простору. Цю додаткову інформацію указують в командному рядку через прогалину після ключового слова. Проте є і ще одна можливість – укласти додаткові параметри в круглі дужки. Наприклад, раніше розглянуту команду видалення змінних з робочого простору можна записати і в іншому вигляді, в формі функціонального виклику:

```
clear ('имя1', 'имя2')
```

Про ці дві можливості говорять як про *дуальність* форми виклику команд системи **MATLAB**. Тут же відзначимо, що при функціональній формі виклику команд явно задані імена треба укладати в *апострофи*.

По будь-якій команді системи **MATLAB** можна отримати швидку довідку, виконавши команду:

```
help им'я_команди.
```

Працюючи з командним вікном системи **MATLAB**, тобто вводячи команди, задаючи числові значення змінних і конструюючи різні математичні вирази, що підлягають обчисленню, а також викликаючи вбудовані в систему **MATLAB** математичні функції, легко можна виконати серйозні обчислення і відображати результати. Такий режим роботи називається *інтерактивним*.

Якщо вбудованих, надзвичайно обширних можливостей системи **MATLAB** виявиться недостатньо для вирішення конкретної задачі, користувач може самостійно запрограмувати необхідні для вирішення проблеми функції. Це можна виконати як на внутрішній **M-мові** системи **MATLAB**, так і на мовах Fortran, C і C++.

1.1.2. Створення протоколу сеансу роботи в пакеті MATLAB

Сеанс роботи з **MATLAB** називається *сесією* (session). Сесія є поточним документом, який відображає роботу користувача з системою **MATLA**B. В ній є рядки введення, виведення та повідомлень про помилки. Змінні і функцій, які використовувались в сесії, можна записати на диск в файлі з розширенням **.mat**, використовуючи команду **save**. Команда **load** дозволяє

завантажити з диска дані в робочу область. Фрагменти сесії можна оформити у вигляді щоденника за допомогою команди **diary** (щоденник):

diary filename – веде запис на диск усіх команд в рядках введення і результатів у вигляді текстового файлу ім'ям **filename**;

diary off – припиняє запис у файл;

diary on – знов починає запис у файл.

Таким чином, чергуючи команди **diary off** і **diary on**, можна зберігати потрібні фрагменти сесії в їх формальному вигляді. Команду **diary** можна задати і у вигляді функції **diary('file')**, де рядок **'file'** задає ім'я файлу. Наступний приклад пояснює техніку застосування команди **diary**:

```
diary myfile.m
1+2
ans =
3
diary off
2+3
ans =5
diary on
sin(1)
ans =0.8415
diary off
```

В даному прикладі перша операція – **1+2=3** – буде записана у файл **myfile.m**, друга – **2+3=5** — не буде записана, третя операція – **sin(1)=0.8415** – знову буде записана. Таким чином, буде створений файл сценарію (**Script-файл**) наступного вигляду:

```
1+2
ans = 3
diary off sin(1)
ans =0.8415
diary off
```

Script-файл приведено в тому вигляді, як записано, тобто з пропусками поміж рядками. Одна з поширених помилок – спроба запустити подібний файл в командному рядку вказівкою його імені:

```
myfile
??? ans =
Missing variable or function.
Error in ==> C:\MATLAB\bin\niyfile.m
On line 3 --> ans =
```

Це приводить до помилок, оскільки даний файл – це просто текстовий запис команд і результатів їх виконання, який не перевіряється на

коректність і містить ряд рядків, помилкових з позицій синтаксису мови програмування MATLAB. Наприклад, вираз ans =. Зате команда type дозволяє проглянути текст такого файлу зі всіма записаними діями:

```
type myfile
1+2
ans=
3
diary off.
ans=
0.8415
diary off
```

Щоб уникнути подібних помилок, рекомендується записувати файл з розширенням, відмінним від .m, наприклад .txt. Це дозволить вбудовувати подібні текстові файли щоденника сесії в документи, що містять її опис.

1.1.3. Основи програмування в MATLAB

MATLAB підтримує ще один режим роботи – *пакетний*, в якому можна складати програми, що є послідовністю команд користувача і зберігаються на диску у вигляді окремого файлу. Файли, які містять команди мови MATLAB (М-мови), називаються М-файлами. Створити М-файл можна в будь-якому текстовому редакторі, але зручніше для цієї мети використовувати спеціальний *редактор* М-файлів, що входить до складу MATLAB. Підготовлений і записаний на диск m-файл стає частиною системи, і його можна викликати як з командного рядка, так і з іншого m-файла. Є два типи m-файлів: *файли-сценарії* і *файли-функції*. Важливо, що в процесі свого створіння вони проходять синтаксичний контроль за допомогою вбудованого в систему MATLAB редактора – відладчика m-файлів.

1.1.4. Файл-сценарій

Файл-сценарій, або Script-файл, є просто записом серії команд без вхідних і вихідних параметрів, це найпростіша програма на мові програмування MATLAB. Він має наступну структуру: основний коментар, додатковий коментар, тіло файлу з будь-якими виразами. Важливі наступні особливості *файлів-сценаріїв*: вони не мають вхідних і вихідних аргументів; працюють з даними з робочої області; в процесі иконання не компілюються; є послідовністю операцій, яка зафіксована у вигляді файлу і повністю аналогічна тієї, що використовується в сесії. Основний коментар – це перший рядок текстових коментарів, а додатковий – подальші рядки. Основний коментар виводиться при виконанні команд lookfor і help ім'я_каталогу. Повний коментар виводиться при виконанні команди help ім'я_файлу. Розглянемо наступний *файл-сценарій:* моделювання трьохчастотного сигналу

на фоні сильного шуму, що створено генератором випадкових чисел): % моделювання трьохчастотного сигналу на фоні сильного % шуму, створюваного генератором випадкових чисел

```
f1=150;
f2=200;
f3=250;
t=0:0.0005:1;
x=sin(2*pi*f2*t)+0.4*sin(2*pi*f1*t)+0.4*sin(2*pi*f3*t);
y=x+2*randn(size(t));
plot(y(1:100),'b')
```

Перші два рядки – це коментар, інші – тіло файлу. Знак % в коментарях повинен починатися з першої позиції рядка. В даному випадку команда help name не приймає коментар (іноді це може знадобитися) і поверне повідомлення вигляду No help comments found in-name.m. Змінні, що використовуються у файлах-сценаріях, є глобальними, тобто вони діють однаково в командах сесії і усередині програмного блоку. Тому задані в сесії значення змінних використовуються і в тілі файлу. Імена файлів-сценаріїв не можна використовувати як параметри функцій, оскільки файли-сценарії не повертають значень. Сигнал, що промоделювали, (рис.1.5) має середню частоту 200 рад/с і два бічні сигнали з частотами 150 і 250 рад/с, що відповідає амплітудно-модульованому сигналу з частотою модуляції 50 рад/с і глибиною модуляції 0.8 (амплітуда бічних частот складає 0.4 від амплітуди центрального сигналу). З вигляду сигналу не видно, що корисний сигнал – амплітудно-модульоване коливання, настільки його забито шумами.

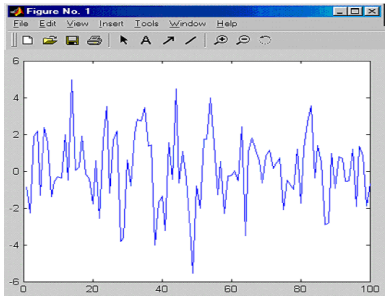


Рисунок 1.5 – Форма сигналу з шумом

Побудуємо графік спектральної густини отриманого сигналу за допомогою прямого перетворення Фур'є, перетворюючи часове зображення сигналу в частотне. Цей графік в області частот до **300 Гц** будується за допомогою наступних команд:

```
Y=fft(y,1024);
```

```
Pyy=Y.*conj(Y)/1024;
f=2000*(0:150)/1024;
plot(f,Pyy(1:151),'b');
```

Графік спектральної густини сигналу, що побудовано в даному прикладі, показано на рис. 1.6. Спектрограма сигналу має явний пік на середній частоті амплітудно-модульованого сигналу і два бічні піки. Всі ці три частотні складові сигналу явно виділяються на загальному шумовому фоні. Таким чином, даний приклад наочно ілюструє техніку виявлення слабих сигналів на фоні шумів, що лежить в основі роботи радіоприймальних пристроїв.

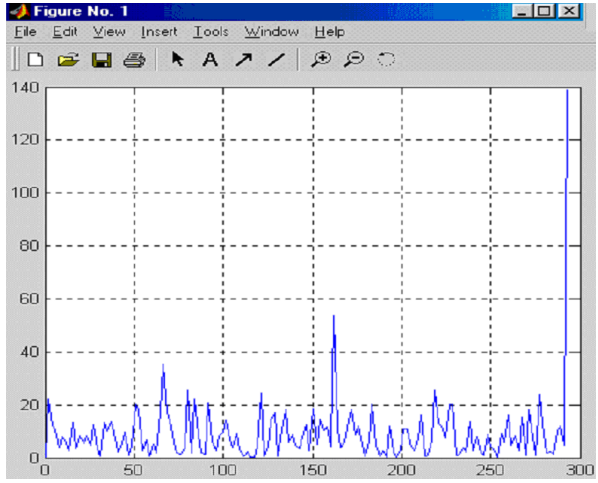


Рисунок 1.6 – Графік спектральної густини сигналу

Щоб створити **М-файл**, необхідно клацнути кнопкою **New M-File** головної панелі інструментів **MATLAB**. З'явиться нове вікно – текстовий редактор з готовим для редагування порожнім документом (рис. 1.7). Після набору тексту *Файл-сценарій* і його відлагодження необхідно створити **М-файл,** для чого скористатися пунктом меню **File\Save As ім'я М-файлу.**

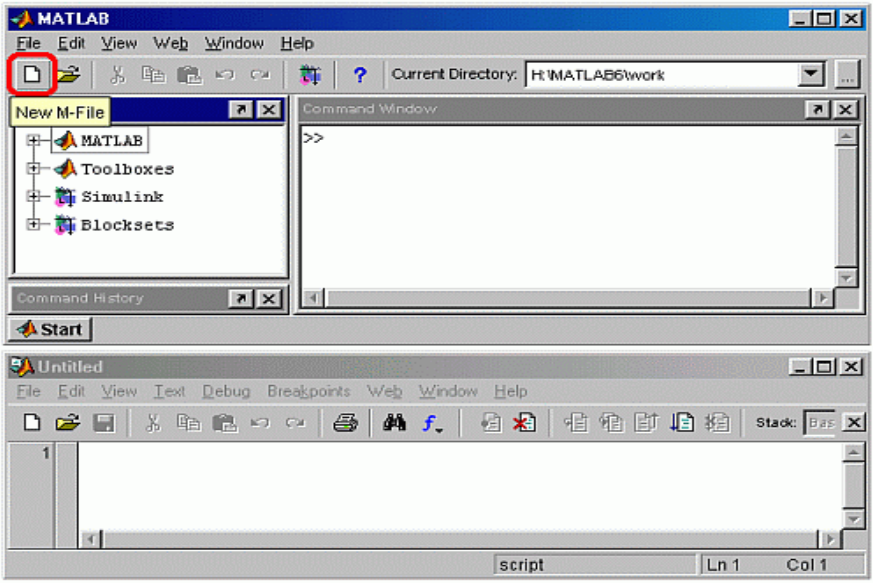


Рисунок 1.7 – Створення **M-файла**

1.1.5. Створення М-функцій

М-файл-функція є типовим об'єктом мови програмування системи **MATLAB**. Одночасно це повноцінний модуль з погляду структурного програмування, оскільки містить вхідні і вихідні параметри і використовує апарат локальних змінних. Структура такого модуля з одним вихідним параметром виглядає таким чином:

```
function var=f_name(Список_параметрів)
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами
var=вираз
```

М-файл-функція має наступні властивості: функція починається з оголошення **function**, після якого указується ім'я змінної **var** – вихідного параметра, ім'я самої функції і список її вхідних параметрів; функція повертає своє значення і може використовуватися у вигляді **name(Список_параметрів)** в математичних виразах; всі змінні, що є в тілі файлу-функції, є локальними, тобто діють тільки в межах тіла функції; файл-функція є самостійним програмним модулем, який спілкується з іншими модулями через свої вхідні і вихідні параметри; правила виведення коментарів ті ж, що у файлів-сценаріїв; файл-функція служить засобом

розширення системи **MATLAB**; при виявленні файлу-функції він компілюється і виконується, а створені машинні коди зберігаються в робочій області системи **MATLAB**. Остання конструкція **`var=вираз`** вводиться, якщо треба, щоб функція повертала результат обчислень. Приведена форма файлу-функції характерна для функції з одним вихідним параметром. Якщо вихідних параметрів більше, то вони указуються в квадратних дужках після слова **`function`**. При цьому структура модуля має наступний вигляд:

```
function [varl,var2....]=f_name(Список_параметрів)
%Основний коментар
%Допоміжний коментар
Тіло файлу з будь-якими виразами
var1=вираз
var2=вираз
```

Таку функцію не можна просто використовувати безпосередньо в математичних виразах, оскільки вона повертає не єдиний результат, а безліч результатів – по числу вихідних параметрів. Якщо функція, яка використовується, має єдиний вихідний параметр, але має ряд вихідних параметрів, то для повернення значення буде використовуватися перший з них. Це часто веде до помилок в математичних обчисленнях. Тому, як наголошувалося, дана функція використовується як окремий елемент програм вигляду:

```
[varl,var2... ]=f_name(Список_параметрів)
```

Після її застосування змінні виходу varl, var2... стають визначеними і їх можна використовувати в подальших математичних виразах і інших сегментах програми. Якщо функція записана у вигляді name(Список_параметрів), то повертається значення тільки першого вихідного параметра – змінної varl. Далі приведено приклад М-функції для обчислення трьохчастотного сигналу на фоні сильного шуму, створюваного генератором випадкових чисел.

```
function s=sig(f1,f2,f3);
t=0:0.0005:1;
x=sin(2*pi*f2*t)+0.4*sin(2*pi*f1*t)+0.4*sin(2*pi*f3*t);
y=x+2*randn(size(t));
plot(y(1:100),'b');
s=Y;
```

М-функція створюється аналогічно, як і М-файл. Після набору тексту М-функції та її відлагодження необхідно створити М-файл**,** для чого скористатися пунктом меню **`File\Save As ім'я М-файлу`.**

1.1.6. Порядок виконання роботи

Використовуючи методичний матеріал, ознайомтесь з командним вікном системи **MATLAB**. Обчисліть наступні математичні вирази:

```
2+3
x=pi/4
2.301*sin(x)
4+exp(3)/5
y=8
sqrt(y)/2
sin(pi/2)
```

Збережіть зміст робочого вікна у файлі. Очистіть робочий простір.

Зітріть видимий зміст робочого простору. Завантажте зміст робочого простору з файлу. Створіть **Файл-сценарій** розрахунку трьохчастотного сигналу на фоні сильного шуму, створюваного генератором випадкових чисел у відповідності з таблицею 1.1. Клацнувши кнопку **Run** на панелі інструментів редактора **М-файлів**, запустіть програму. З'явиться графік сигналу з шумом. Запишіть **Файл-сценарій** на диск і запустіть його з командного рядка **MATLAB**. Створіть **Файл-сценарій** для побудови графіка спектральної густини отриманого сигналу за допомогою прямого перетворення Фур'є, що перетворює часове зображення сигналу в частотне. Виконайте отриманий **Файл-сценарій** з командного рядка і побудуйте спектр сигналу, що синтезується. Створіть **М-функцію** для розрахунку трьохчастотного сигналу на фоні сильного шуму, створеного генератором випадкових чисел. Відладьте її і запустіть з командного рядка, використовуючи вхідні дані з табл.1.1.

Таблиця 1.1

№ вар.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
F_1	75	300	225	90	450	120	150	375	135	600
F_2	100	400	300	120	600	160	200	500	180	800
F_3	125	500	375	150	750	200	250	625	225	1000

1.1.7. Зміст звіту

Титульний лист, виконаний відповідно до СТП.

Мета роботи.

Протокол сеансу роботи з **MATLAB.**

Графік сигналу, що синтезується.

Спектр сигналу.

Відповіді на контрольні питання.

1.1.8. Контрольні питання

З яких частин складається головне вікно системи **MATLAB**?

Яка частина головного вікна називається командним вікном системи **MATLAB**?
Як записати у файл зміст робочого простору?
Що таке рядок введення в системі **MATLAB**?
Як стерти видимий вміст робочого простору?
Як створити протокол сеансу роботи в пакеті **MATLAB**?
Що таке **Файл-сценарій**? Його структура.

Лабораторна робота 2.

Сигнал і його перетворення при цифровий обробці

Мета роботи: вивчення ефектів які виникають при перетворенні аналогового сигналу в цифрову форму. Засоби які застосовуються при виконанні лабораторної роботи: лабораторна робота виконується на персональному комп'ютері в середовищі "MATLAB" і "SIMULINK".

2.1. Порядок виконання лабораторної роботи

Лабораторна робота складається з чотирьох частин. У першій частині вивчаються ефекти, що виникають при дискретизації аналогового сигналу. У другій частині розглядаються випадок дискретизації імпульсних сигналів. У третій і четвертій частинах вивчаються шуми квантування, що виникають при аналого-цифровому і цифро-аналоговому перетворенні для різних методів квантування з урізанням і округленням.

2.2. Експеримент 1. Вивчення ефектів які виникають при дискретизації аналогового сигналу

Для проведення експерименту, необхідно зібрати схему (рис.2.1) з типових елементів, використовуючи при цьому браузер бібліотеки Simulink. Схема використовується для вивчення ефектів, які виникають при дискретизації аналогового сигналу

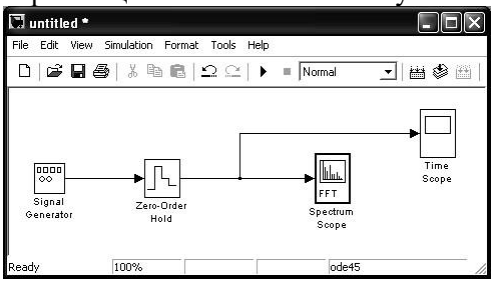


Рисунок 2.1 - Схема для вивчення ефектів накладення при дискретизації аналогового сигналу

Провести настройку генератора сигналів, вибравши вид сигналу SIN, амплітуду вихідного сигналу 1 вольт і частоту 1 кГц. Налаштувати екстраполятор нульового порядку (Zero-Order Hold) на частоту дискретизації 48 кГц, (рис. 2).

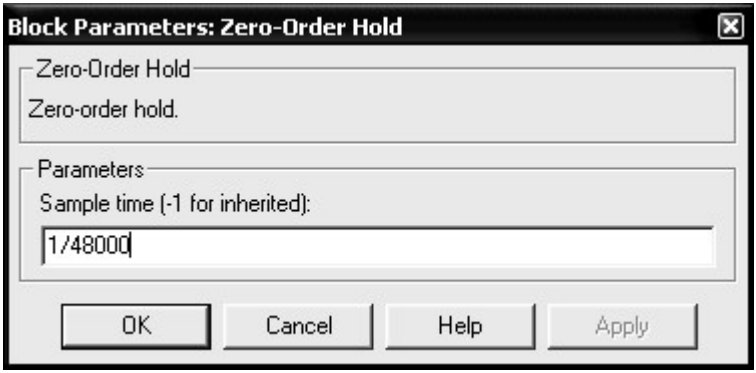
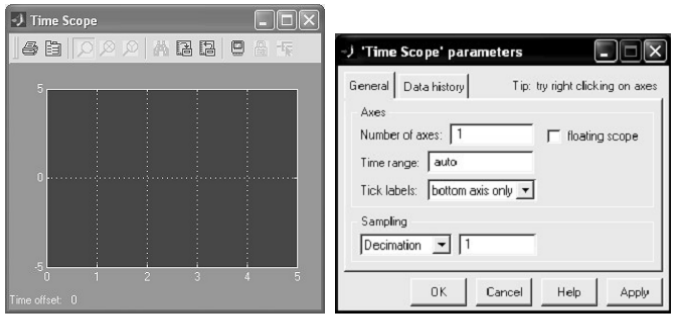


Рисунок 2.2 – Вікно настройки параметрів екстраполятор нульового порядку

Далі необхідно налаштувати осцилограф (рис. 2.3), причому у вкладці "історія даних" ("Data history") необхідно прибрати галочку обмеження обсягу відображаються точок.



а) в)

Рисуноу 2.3 - Вікна осцилографа: а) - основне вікно; б) - вікно настройки

Аналізатор спектра необхідно налаштувати, як показано на рис. 2.4, вибравши розмір вхідного буфера (Buffer size) 4096, перекриття буферів (Buffer overlap) 64 і довжину БПФ (FFT Length) 4096.

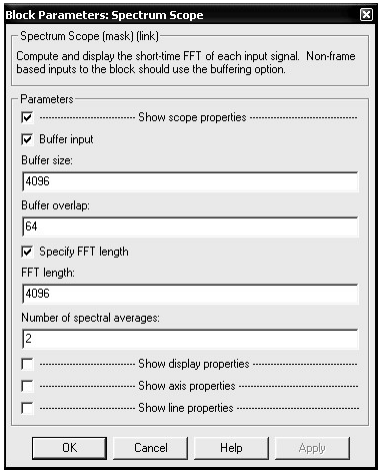


Рисунок 2.4 - Вікно налаштування спектроаналізатора

Запустивши схему на виконання необхідно подивитися сигнал, одержуваний у вікнах Time Scope і Spectrum Scope. Після чого потрібно провести експеримент для частот в діапазоні від 1 КГц до 96 КГц, з кроком 20 кГц. Відобразити отримані результати з поясненнями в звіті.

2.3. Експеримент 2. Вивчення впливу обмеження спектра аналогового сигналу при дискретизації.

Зібрати схему показану на рис.2.5. Провести настройку генератора імпульсних сигналів (Pulse Generator) вибравши амплітуду імпульсів 1 вольт частоту проходження 1КГц і тривалість імпульсу 20%.

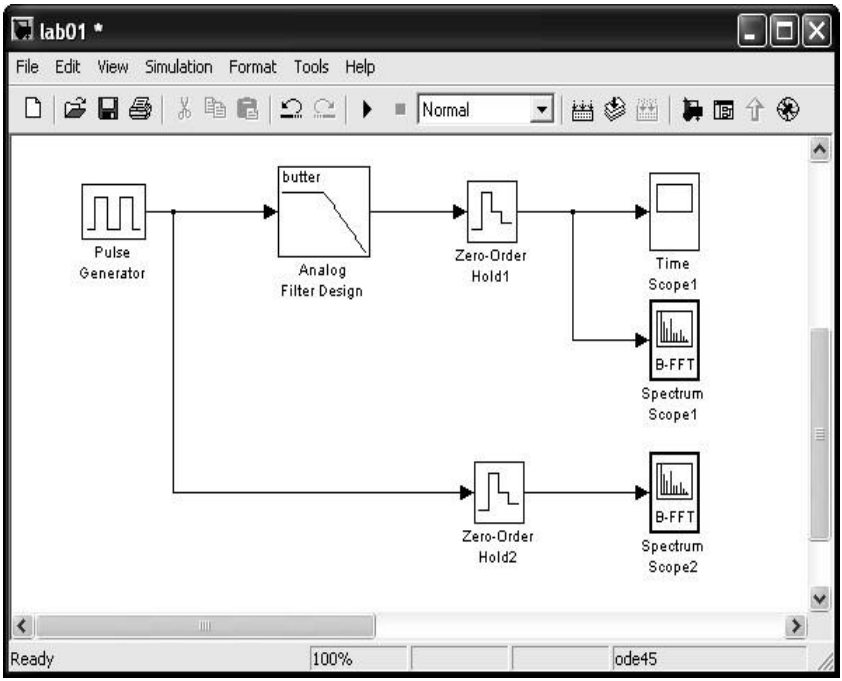


Рисунок 2.5 - Схема для вивчення впливу обмеження спектра аналогового сигналу при дискретизації

Налаштувати аналоговий протівомаскіровочнй фільтр (Analog Filter Design, рис2.6) вибравши в якості характеристики фільтра характеристику Батерворте, порядок 16 частоту зрізу 10 КГц.

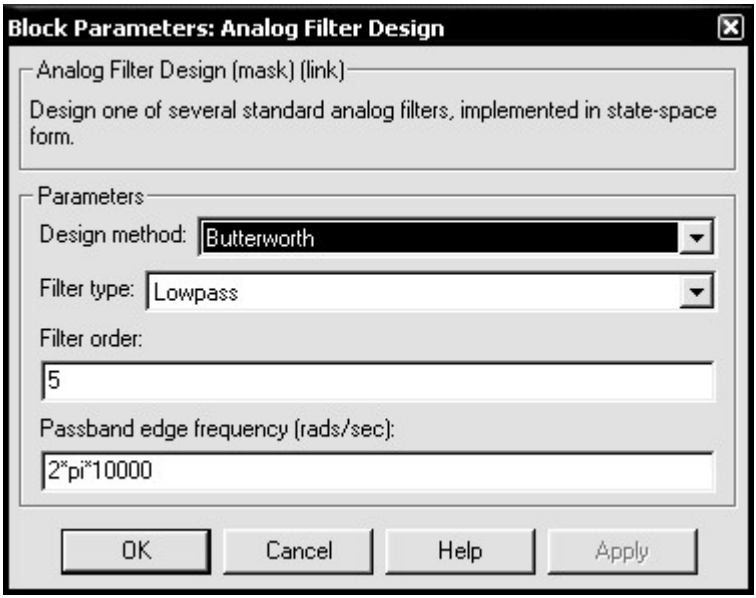


Рисунок 2.6 - Вікно налаштування аналогового фільтра

Запустити схему. Порівняти спектри, одержувані до аналогового фільтра і після нього. Зробити експеримент для різних частот зрізу фільтра, його характеристики і порядку. Як частоту зрізу вибрати значення 2, 5 і 10 кГц. Порядок фільтра вибрати рівним: 32, 16 і 2. Отримані результати з поясненнями відобразити в звіті. Привести теоретичний спектр сигналу, який використовується в експерименті. Привести формулу, по якій проводиться розрахунок його гармонік

2.4. Експеримент 3. Вивчення квантування методом усічення

Зібрати схему, наведену на рисунку 2.7. Провести настройку генератора сигналів, вибравши вид сигналу SIN, амплітуду вихідного сигналу 1 вольт і частоту 10Гц.

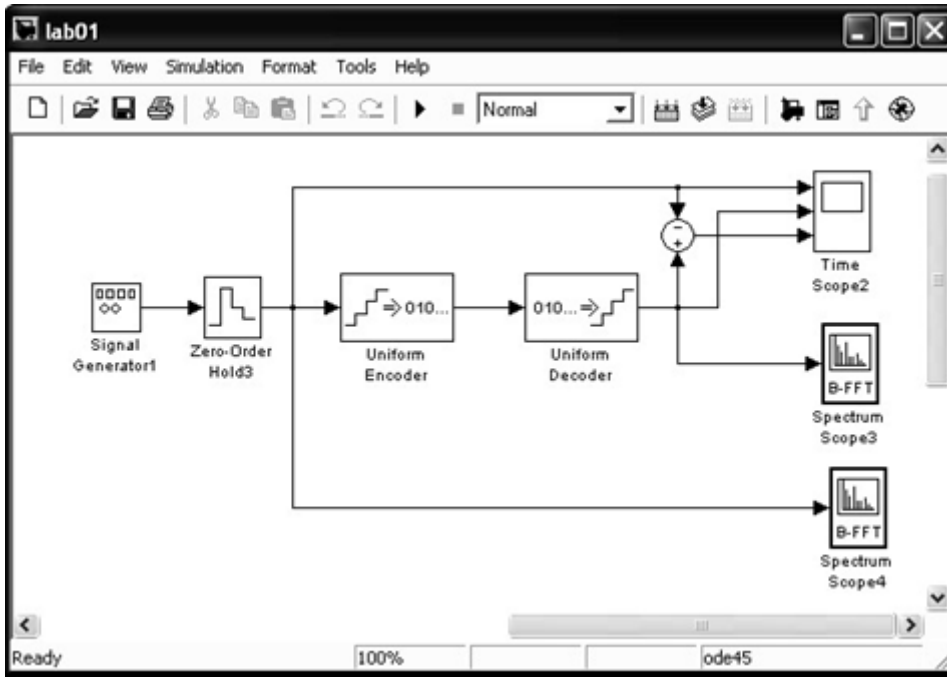
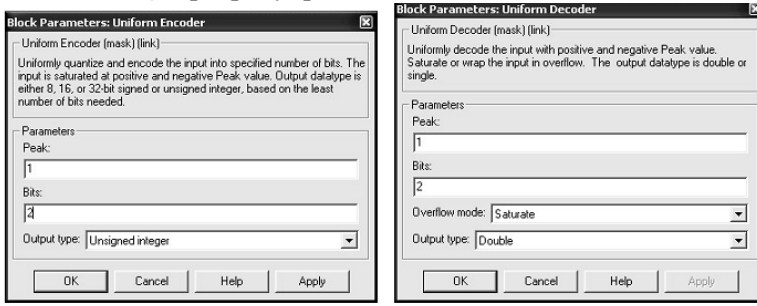


Рисунок 2.7 - Схема для вивчення квантування методом усічення

Далі необхідно встановити частоту дискретизації 1кГц. Зробити тривалість симуляції 10 секунд. Для проведення досліджень необхідно вибрати розрядність АЦП (Uniform Encoder) і ЦАП (Uniform Decoder) 2 розряду, рис.9.



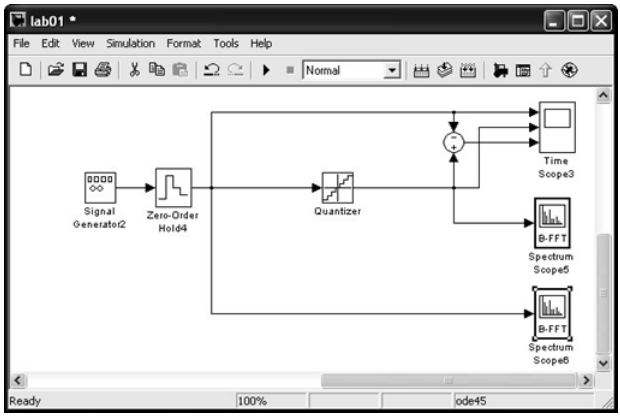
а) в)

Рисунок 2.8 Вікна настройки АЦП І ЦАП

Запустити модель. Проаналізувати отримані результати. Виконати експеримент для значень розрядності 4, 8, 16. Всі отримані результати з поясненнями привести в звіті.

2.5. Експеримент 4. Вивчення квантування методом округлення сигналу

Зібрати схему 4, наведену на рис.2.9. Провести настройку генератора сигналів, вибравши вид сигналу SIN, амплітуду вихідного сигналу 1 вольт і частоту 10Гц.



Рисунрк 2.9 - Схема 4 застосовується для вивчення квантування сигналів округленням

Встановити частоту дискретизації 1КГц. Зробити тривалість симуляції 10 секунд. Вибрати число рівнів квантування 4, як показано на рис.11.

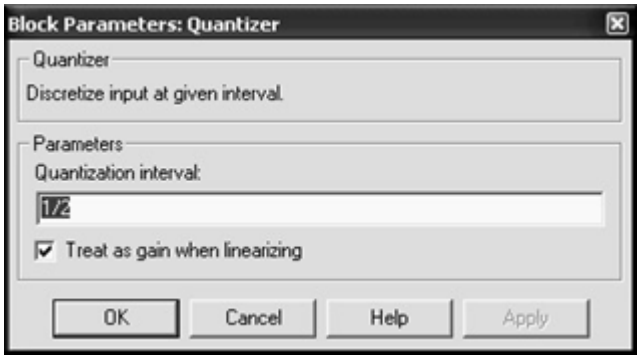


Рисунок 2.10 - Вікно налаштування квантувача

Запустити модель. Проаналізувати отримані результати. Виконати експеримент для значень числа рівнів квантування рівних 2, 8 і 16. Отримані результати з поясненнями привести в звіті.

2.6. Зміст звіту лабораторної роботи

1. Мета роботи.
2. Схеми моделей.
3. Результати експериментів з поясненнями.
4. Відповіді на контрольні питання5

2.7. Контрольні питання

1. Які перетворення сигналів мають місце в системі цифрової обробки аналогових сигналів?
2. Що таке Сигнал?
3. В чому полягають взаємозв'язок і відмінність спектрів дискретного і аналогового сигналів?
4. У чому полягає і як виявляється накладення спектрів при дискретизації сигналів?
5. З яких умов вибирається частота дискретизації аналогових сигналів?
6. Яким чином можуть бути зменшені спотворення, пов'язані з дискретизацією сигналу?
7. Як змінюється спектр сигналу при цифроаналогових перетворенні?
8. Як залежать спотворення відновлення сигналу від частоти дискретизації і максимальної частоти його спектра в основній смузі?

9. З яких умов вибирається необхідна розрядність АЦП?

Лабораторна робота 3.

Розрахунок низькочастотного фільтру

Мета роботи – освоїти технологію проектування цифрових фільтрів засобами пакету **MATLAB**.

3.1. Теоретичні відомості

У процесі цифрової обробки сигналів нерідко виникає задача фільтрації сигналу в заданому діапазоні частот. Прикладом такої задачі може служити ситуація, коли сигнал містить кілька гармонік, і необхідно виділити одну з них. Для цього необхідно спроектувати цифровий фільтр. У залежності від розташування складового сигналу це може бути фільтр нижніх, верхніх частот, смуговий або загороджувальний фільтр. Область частот, у якій фільтр пропускає сигнал, називається *смугою пропущення*, а область частот, у якій ослаблення вхідного сигналу велико, – *смугою затримування*.

Розглянемо приклад. Нехай є дискретний сигнал: $u(n) = sin(2\pi\ f_1 nT) + sin(2\pi\ f_2 nT) + e(n)$, де $n=1, 2,\ldots$, $f_1 = 200$ *Гц* і $f_2 = 1000$ *Гц* – частоти складового сигналу, $T = 1/8000$ с – період дискретизації, $F_s = 8000$ *Гц* – частота дискретизації; $e(n)$ – гауссівський шум з нульовим середнім та дисперсією $\sigma^2 = 0.1$. Потрібно виділити синусоїдальну складову з частотою $f_1 = 200$ *Гц*. Для цього необхідно побудувати фільтр нижніх частот із граничними значеннями частот смуги пропущення F_{pas} і смуги затримки F_{stop}, що задовольняє умові $f_1 < F_{pass} < F_{stop} < f_2$. Оскільки частота Найквіста $f_N = F_s/2 = 4000$ *Гц*, для нормалізованих значень $\overline{F}_{pass} = \overline{F}_{pass} / f_N$ та $\overline{F}_{stop} = \overline{F}_{stop} / f_N$ $\overline{F}_{stop}$ ця умова буде виглядати:

$$\frac{200}{4000} = 0.05 < \overline{F}_{pass} < \overline{F}_{stop} < \frac{1000}{4000} = 0.25 .$$

Отже, перехідна смуга Δf амплітудно-частотної характеристики фільтра (АЧХ) не повинна перевищувати величину $\Delta f < 0.25 - 0.05 = 0.2$. Ці вимоги схематично показані на рис.3.1.

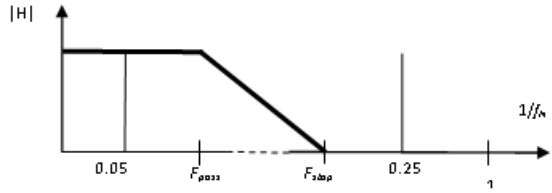


Рисунок 3.1

Для розв'язання поставленої задачі спроектуємо нерекурсивний фільтр зі смугою пропущення від **0** до **210** ***Гц*** і смугою затримування від **980** до **4000** ***Гц***, тобто до частоти Найквіста. Також потрібно, щоб АЧХ у смузі пропущення знаходилася в межах [**0.99, 1.01**], а в смузі затримування не перевищувала значення **0.0001**. Для цього скористаємося функціями пакета **MATLAB,** що оцінює за заданими вимогами порядок фільтра і розраховує коефіцієнти. Завантажимо графічну програму **FDAtool** за допомогою інструкції. На екрані з'явиться панель **Filter Design & Analysis Tool** (рис.3.2):

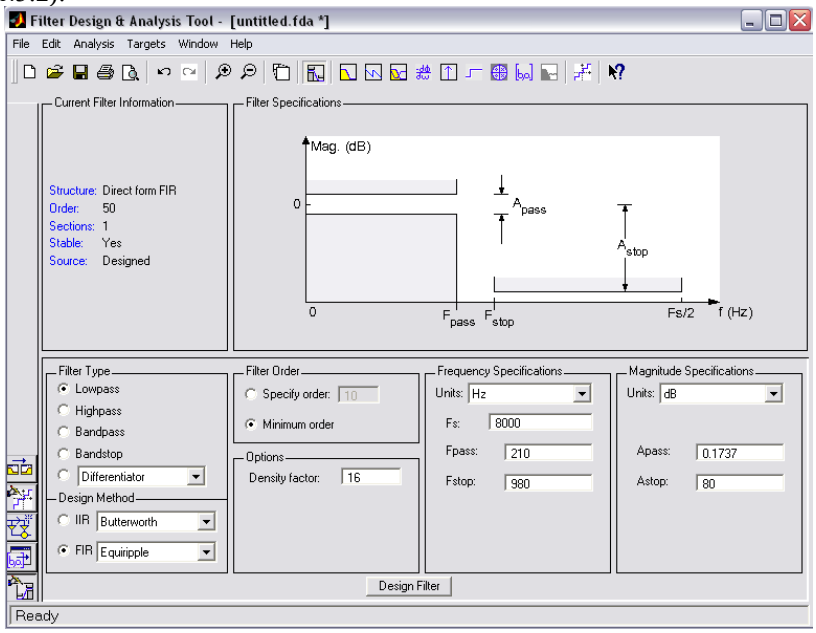


Рисунок 3.2

Щоб розрахувати фільтр необхідно виконати наступні дії:

Виконати команду меню **Analysis|Filter Specifications**;

У полі **Filter Type** установити опцію **Lowpass** (фільтр нижніх частот);

У полі **Design Method** установити опцію **FIR\Equiripple** (фільтр із кінцевою імпульсною характеристикою);

У полі **Filter Order** установити опцію **Minimum order** (мінімальний порядок фільтра);

У полі **Frequency Specifications** установити наступні опції:

Units: **Hz** (Гц);

F_S: **8000** (частота дискретизації);

***Fpass*: 210** (верхня частота смуги пропущення);

***Fstop*: 980** (нижня частота смуги затримування);

У полі **Magnitude Specifications** установити опції:

Units: d (логарифмічна АЧХ);

Apass=20*log(1.01)-20*log(0.99)=0.1737 (нерівномірність АЧХ у смузі пропущення);

Astop=20*log(0.0001)=80 (ослаблення в смузі затримування).

Після уведення усіх даних необхідно натиснути кнопку **Design Filter**, після чого з'явиться панель, що показана на рис.3.3.

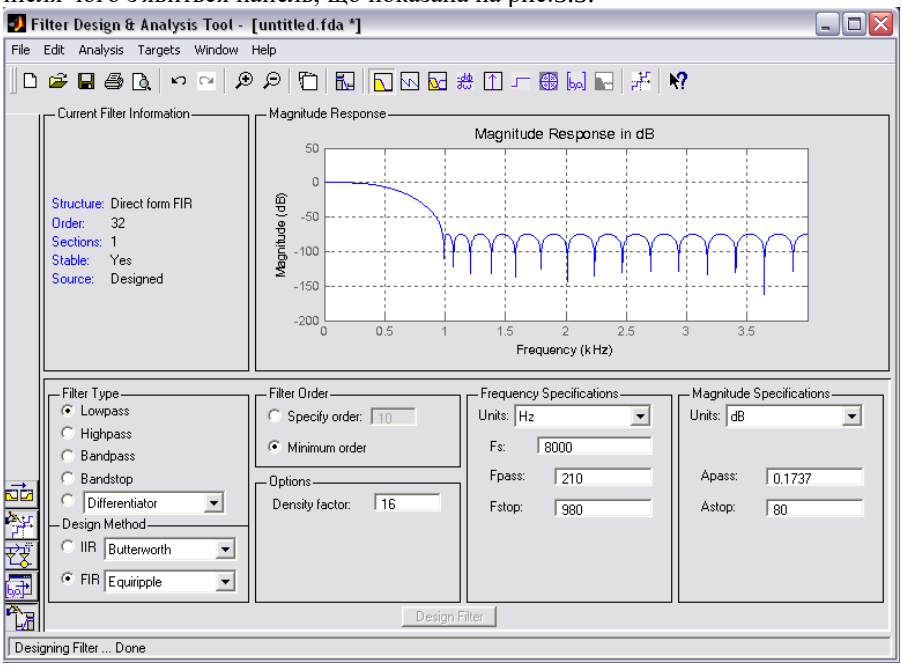


Рисунок 3.3

У верхній частині панелі в полі **Current Filter Information** дається коротка інформація про фільтр (тип фільтра, порядок фільтра, інформація про стійкість). У полі **Magnitude Response in d** приводиться графік АЧХ. З допомогу команд меню **Analysis** (рис.3.4) можна переглянути основні характеристики фільтра:

Magnitude Response – АЧХ;

Phase Response – ФЧХ;

Magnitude Response and Phase Response – АЧХ і ФЧХ;

`Impulse Response` – імпульсна характеристика фільтра;
`Step Response` – перехідна характеристика фільтра;
`Pole/zero Plot` – розподіл нулів і полюсів фільтра;
`Filter Coefficients` – коефіцієнти фільтра.

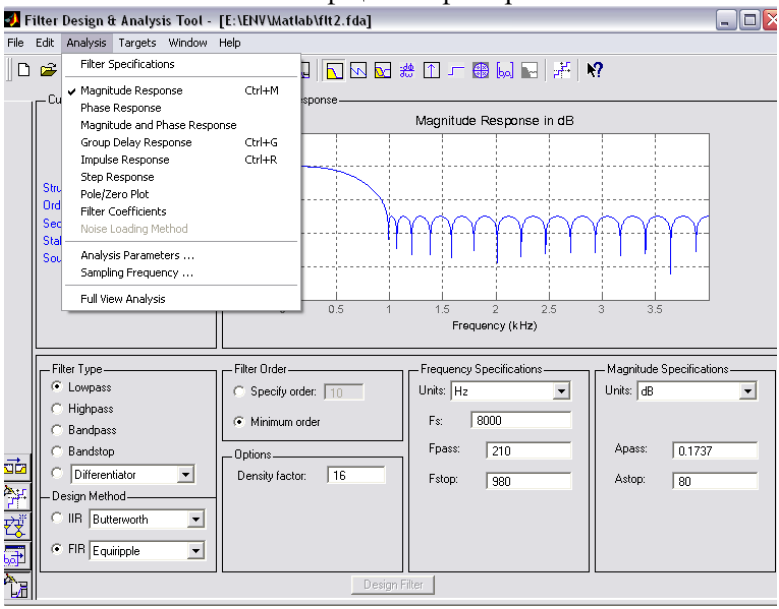


Рисунок 3.4

Після того як фільтр синтезовано, необхідно створити файл, у якому зберігається вся інформація про фільтр. Зробити це можна за допомогою наступної команди меню **`File|Save session|Name`**. Для запису коефіцієнтів фільтра в робочий простір MATLAB необхідно скористатися наступною командою **`File|Export`**.

3.2. Порядок виконання роботи

1. Виберіть згідно з таблицею 3.1 верхню границю смуги пропущення F_{pass}, нижню границю смуги затримування F_{stop}, частоту дискретизації F_s, максимальне H_{max} і мінімальне H_{min} значення АЧХ у смузі пропущення та коефіцієнт загасання $K_{заг}$ фільтра в смузі затримування;

Таблиця 3.1 № вар.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
F_{pass}	100	120	220	350	600	500	600	400	300	320
F_{stop}	200	180	280	500	900	1500	1000	500	500	1200

F_s	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000
H_{max}	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
H_{min}	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
$K_{заг}$	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}

2 Розра хуйте частоту Найквіста $f_N = F_s/2$;

3 Розрахуйте перехідну смугу фільтра по формулі: $\Delta f = (F_{stop} - F_{pass})/f_N$;
Розрахуйте величину нерівномірності АЧХ у смузі пропущення фільтра

A_{pass}=`20*log(`H_{max}`)-20*log(`H_{min}`)`;

4 Розрахуйте період дискретизації $T = 1/F_s$, с;

5 Розрахуйте величину загасання фільтра в смузі затримування

A_{stop}=`20*log(`$K_{заг}$`)`;

6 У командному вікні **MATLAB** введіть команду **`FDAtool`**;

7 Виконайте всі необхідні дії для розрахунку фільтра;

8 За допомогою меню **`Analysis`** перегляньте основні характеристики фільтра:

`Magnitude Response` – АЧХ

`Phase Response` – ФЧХ

`Impulse Response` – імпульсна характеристика фільтра

`Step Response` перехідна характеристика фільтра

`Pole/zero Plot` – розподіл нулів і полюсів фільтра

`Filter Coefficients` – коефіцієнти фільтра;

9 Запишіть значення коефіцієнтів фільтра в робочий простір **MATLAB.** Для цього скористуйтися наступною командою **`File|Export`**.

Створіть файл, у якому зберігається вся інформація про фільтр. Зробити це можна за допомогою наступної команди меню

`File|Save session|Name.`

10. Перегляньте коефіцієнти фільтра в робочому просторі **MATLAB**, для чого в командному вікні наберіть команду **`NUM`**;

11 Запишіть коефіцієнти фільтра у файл. Для цього наберіть в командному вікні наступні команди:

```
fid = fopen('B.dat','wt')
fprintf(fid,'%16.8f\n',Num)
```

3.3. Зміст звіту

1 Титульний лист, виконаний відповідно до СТП.

2 Мета роботи.

3 Протокол сеансу роботи з **MATLAB.**

4 Відповіді на контрольні питання.

3.4. Контрольні питання

1 Що таке смуга пропущення фільтра?
2 Що таке смуга затримування фільтра?
3 Як розрахувати величину нерівномірності АЧХ фільтра?
4 Які існують типи фільтрів?
5 Як визначити величину ослаблення фільтра в смузі затримування ?
6 Приведіть графік типової АЧХ фільтра нижніх частот.

Лабораторна робота 4.

Моделювання низькочастотного фільтра

Мета роботи – освоїти технологію моделювання цифрових фільтрів засобами надбудови **SIMULINK** системи **MATLAB**.

4.1. Теоретичні відомості

Надбудова **SIMULINK** системи **MATLAB** широко використовується для моделювання різних процесів. **SIMULINK** дозволяє моделювати роботу дуже складних систем і навіть в реальному часі. Прикладами може служити моделювання: синхронної машини, фіксатора гальма, генераторів, маятники з одним і двома шарнірами, гідравлічної системи, системи керування літака, термодинамічної моделі будинку й ін.

Запустити **SIMULINK** можна декількома способами:

– натисканням кнопки  **"Нова Simulink Модель"** на панелі інструментів вікна керування **MATLAB**;

– введенням у командний рядок робочої області вікна керування **MATLAB** команди **simulink;**

– через пункт меню **"Файл"** вікна керування **MATLAB**, вибравши послідовно опції **"Створити"=>"Модель".**

Після запуску **SIMULINK** з'являється вікно **Simulink Library Browser** (рис. 4.1) – вікно бібліотеки **SIMULINK**, що містить блоки для моделювання.

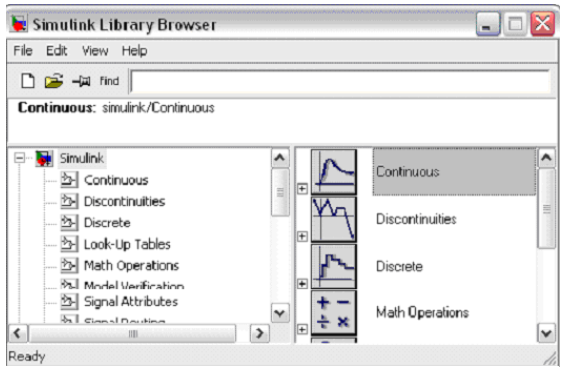


Рисунок 4.1 – Робоче вікно бібліотек **SIMULINK**

Бібліотека блоків **SIMULINK** містить великий набір стандартних блоків, за допомогою за допомогою яких можна скласти практично будь-яку модель. Бібліотеку блоків також можна поповнювати, склавши їх у системі **MATLAB**.

Для створення нової моделі необхідно в меню **File** скористатися опцією **New: Model** (створення нової моделі **SIMULINK**). Після натискання кнопки **New:Model** з'явитися робоче вікно **SIMULINK**: **untitled** (рис.4.2), у якому відбувається створення схем моделювання.

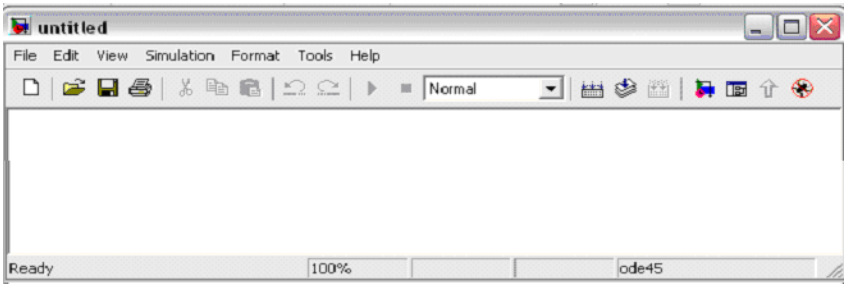


Рисунок 4.2 – Робоче вікно **SIMULINK**

Як приклад, розглянемо технологію побудови моделі для дослідження роботи низькочастотного фільтра, що має наступні параметри: F_{pass} **=100** ***Гц***, F_{stop}**=300** ***Гц***, F_s **=8000** ***Гц***, H_{max}**=1.01,** H_{min}**=0.99,** $K_{заг}$**=0.0001.**

Фільтр призначено для виділення низькочастотної складової з дискретного сигналу:

$$u(nT) = sin(2\pi f_1 nT) + sin(2\pi f_2 nT) + e(nT),$$

де $n=1, 2,\ldots$, $f_1 = 200$ *Гц* і $f_2 = 1000$ *Гц* – частоти складового сигналу, $T = 1/8000$ с – період дискретизації, $F_s = 800$ *Гц* – частота дискретизації сигналу; $e(n)$ – гауссівський шум з нульовим середнім та дисперсією $\sigma^2 = 0.1$.Потрібно виділити синусоїдальну складову з частотою $f_1 = 200$ *Гц*.

Для побудови моделі необхідні наступні блоки:

– блок дискретного часу **Digital Clock** (бібліотека **Simulink|Sources**);

– блок **Fcn** (бібліотека **Simulink|User-Defined-Functions**);

– два аналізатори спектра **Spectrum Scope** (бібліотека **Simulink|Sinks)**.

– осцилограф **Scope** (бібліотека **Simulink|Sinks**).

Для переносу блоків у модель їх необхідно скопіювати з бібліотеки. Для цього установите курсор на значок ліворуч від блоку, потім кнопкою миші перетягніть блок у вікно моделі. Копія блоку з'явиться у вікні моделі. У такий спосіб скопіюйте всі блоки у вікно для побудови моделі. Перемістити блоки усередині вікна можна, виділивши блок і перемістивши його мишею. Після того, як усі потрібні блоки скопійовані у вікно моделі (рис.3.3), для кожного блоку необхідно встановити значення параметрів. Для цього необхідно установити курсор на блок та двічі клацнути лівою кнопкою миші. З'явитися вікно настроювання блоку.

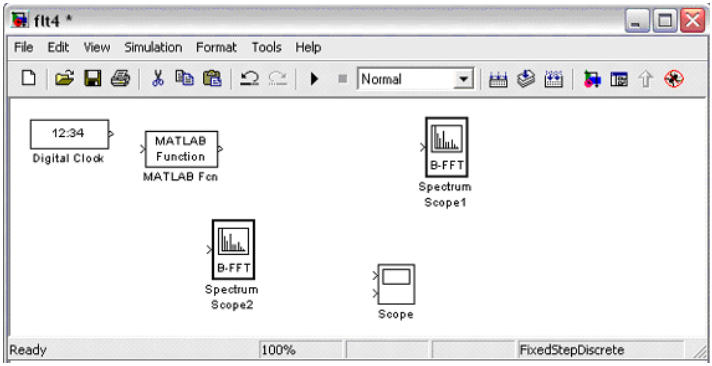


Рисунок 4.3

У вікні параметрів утримується вся інформація про ті або інші параметри і стандартні кнопки керування:

– **Apply** – зберегти внесені вами зміни параметрів у блоці;

– **Revert** – повернути попередні збережені параметри блоку;

– **Help** – вивести додаткові відомості про даний блок;
– **Close** – закрити поточне вікно параметрів.

У нашому прикладі блоки мають наступні параметри. Блок **Digital Clock** призначено для завдання періоду дискретизації в моделях дискретних систем та має єдиний параметр – величину кроку зміни модельного часу **Sample Time** (рис.3.4).

Блок **Fcn** дозволяє застосувати до вхідного сигналу будь-яку процедуру обробки, реалізовану у виді **M-функції**. Цей блок призначено у моделі для обчислення вхідного сигналу фільтра

$$u(nT) = sin(2\pi f_1 nT) + sin(2\pi f_2 nT) + e(nT).$$

Він має три параметри настроювання (рис.3.5):

– **MATLAB function** – ім'я **M-функції**;

– **Output with** тривалість вихідного сигналу. Якщо тривалість вихідного сигналу збігається з тривалістю вхідного, то значення параметра – 1. Якщо це не так, то необхідно заздалегідь визначити тривалість вихідного сигналу і вказати неї як параметр.

– **Output signal type** – тип вихідного сигналу.

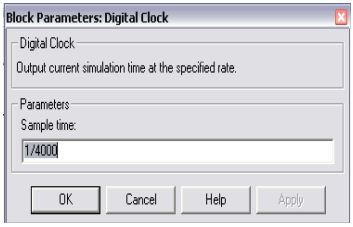


Рисунок 4.4

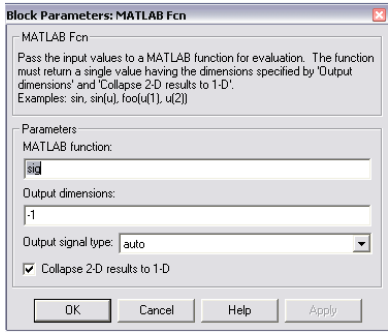


Рисунок 4.5

Відповідна М-функція має вигляд:

```
function u=sig(t);
f1=200;
f2=1000;
d=0.1;
u=sin(2*pi*f1*t)+sin(2*pi*f2*t)+d*randn(1);
```

Блок **Spectrum Scope** (рис.3.6) має наступні параметри настройки:

Length – 256,

Buffer size – 256,
Sample time of original time series – 1/8000.

Блок **Scope** (рис.3.7) має наступні параметри настройки:

Number of axis – кількість осей (каналів) осцилографа – 3;

Time range – межі часового інтервалу – **auto**;

Tick labels – виведення/приховання оцінок на осях;

Sampling – установка часових співвідношень:

Decimation (у десяткових частках часу, за замовчуванням **1**) або **Sample Time** (у тактах еталонного часу, за замовчуванням **0**).

Кожен блок має кілька символів "**>**", що входять і виходять із блоку. Символ "**>**",що вказує усередину блоку, називається вхідним *портом,* а символ, що вказує з блоку, називається *вихідним портом.* Сигнал передається з виходу одного блоку на вхід іншого блоку за допомогою *лінії зв'язку.* Якщо блок підключений, то символ порту зникає.

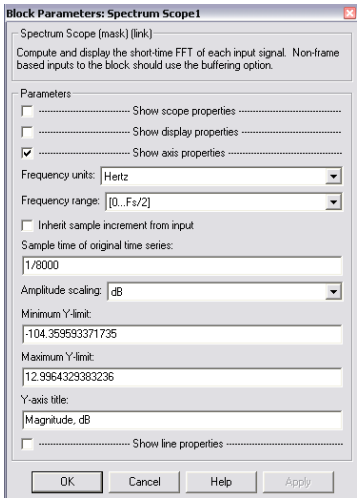


Рисунок 4.6

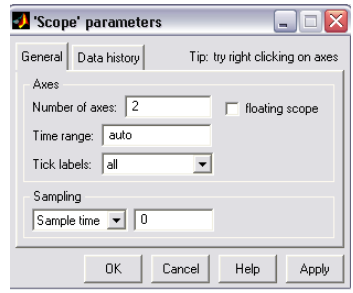


Рисунок 4.7

З'єднати вихід одного блоку з входом іншого можна в такий спосіб: установіть курсор усередину вихідного порту з правої сторони блоку. Стрілка заміниться перехрестям. Натисніть і, утримуючи кнопку миші, пересуньте курсор до вхідного порту іншого блоку. Перехрестя заміниться подвійним перехрестям. Якщо необхідно приєднати вхідний порт блоку до лінії зв'язку, то необхідно установити курсор на вхідний порт і, утримуючи кнопку миші,

пересуньте курсор до лінії зв'язку. Після виконання всіх з'єднань модель буде мати такий вигляд (рис 4.8).

У модель додані усі компоненти, необхідні для дослідження цифрового фільтра. Залишилося додати тільки блок, що реалізує заданий **FIR** фільтр. Для цього скористаємося програмою **FDAtool**. Завантажимо цю програму за допомогою інструкції **FDAtool** і розрахуємо фільтр із необхідними параметрами. Методика синтезу докладно описано в методичних матеріалах по лабораторній роботі №2. Після того як фільтр розрахований, необхідно клацнути по розташованій унизу вікна **FDATool** кнопці **Realize Model**. У поточну модель системи моделювання **Simulink** буде доданий новий компонент – модель заданого **FIR** фільтра (рис 4.9, блок **Filter**).

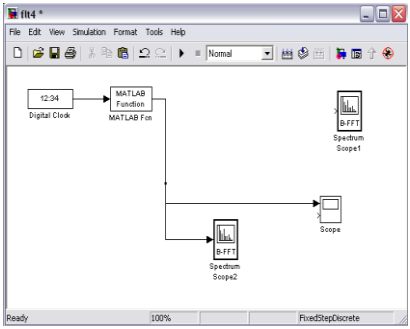


Рисунок 4.8

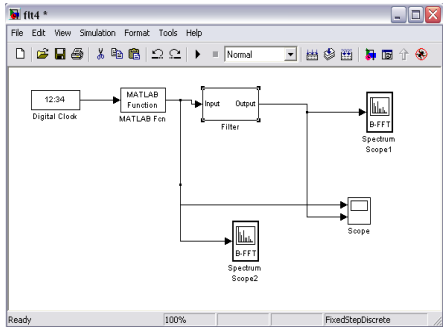


Рисунок 4.9

Двічі клацніть на блоці **Filter**. З'явиться вікно, що містить готову схему **FIR** фільтра. Для моделювання приклада спочатку необхідно відкрити блок осцилографа **Scope**, двічі клацнувши на ньому. Потім необхідно *установити параметри моделювання.* Для цього виберіть пункт меню **Parameters** з меню **Simulation**. Установите необхідний час закінчення моделювання **Stop time**.

Закрийте вікно установки параметрів, клацнувши по клавіші **Close, Simulink** установить задані параметри і закриє вікно. Виберіть пункт **Start** з меню **Simulation**. Результати моделювання виводяться у вікна **Spectrum Scope1** (рис. 4.10), **Spectrum Scope2** (рис. 4.11), **Scope** (рис. 4.12).

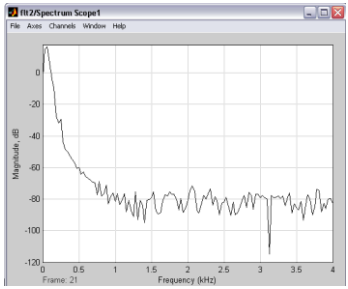


Рисунок 4.10

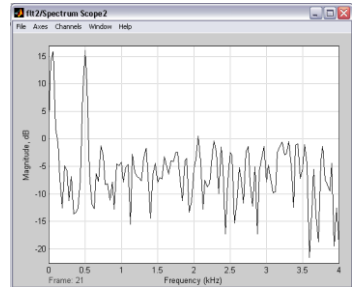


Рисунок 4.11

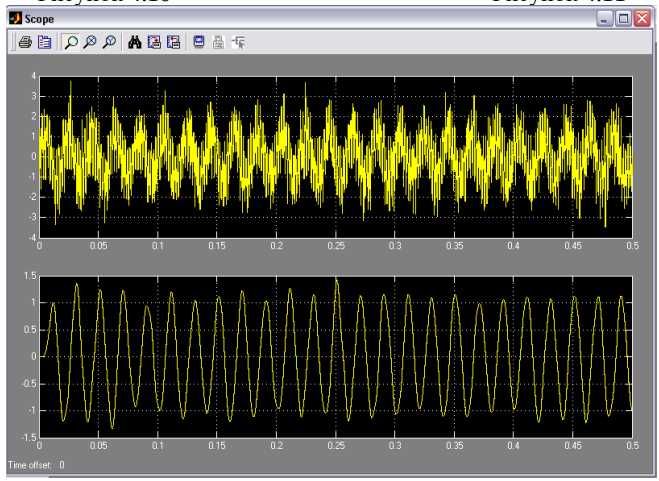


Рисунок 4.12

Моделювання закінчиться, коли досягнутий час, зазначене в параметрах моделювання або якщо обрано пункт **Stop** у меню **Simulation**. Для запису моделі виберіть пункт **Save** з меню **File**, введіть ім'я файлу і виберіть каталог, у якому зберегти файл.

4.2. Порядок виконання роботи

1 Запустіть надбудову **SIMULINK**.

2 Скористайтесь опцією **New: Model** для створення вікна моделі.

3 Перетягніть з вікна **Simulink Library Browser** у створене вікно необхідні для моделювання блоки.

4 Відповідно до таблиці 3.1 створіть **M-функцію** для розрахунку вхідного сигналу фільтра

$$u(nT) = sin(2\pi f_1 nT) + 0.4\,sin(2\pi f_2 nT) + 0.4\,sin(2\pi f_3 nT) + e(nT);$$

Таблиця 3.1

№ вар.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f_1	75	100	200	90	450	120	150	375	135	200
f_2	200	300	300	1200	1000	1600	1200	700	680	800
f_3	725	500	475	1500	1250	2000	2500	1225	2250	1000
σ^2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2

5 Для кожного блоку установіть необхідні значення параметрів.
6 З'єднайте всі блоки моделі відповідно до рис. 3.8.
7 За допомогою інструкції **`FDAtool`** завантажте програму **`FDAtool`**.
8 Використовуючи команду меню **`File|Open session|Name`**, завантажте інформацію про фільтр, що спроектований при виконанні лабораторної роботи №2.
9 Клацніть на розташованій унизу вікна **`FDATool`** кнопці **`Realize Model`** та додайте в поточну модель новий компонент – модель заданого **`FIR`** фільтра (блок **`Filter`**).
10 Проведіть моделювання.
11 Створіть файл, у якому зберігається вся інформація про модель.

4.3. Зміст звіту

1 Титульний лист, виконаний відповідно до СТП.
2 Мета роботи.
3 Блок схема моделі.
4 Результати моделювання.
5 Відповіді на контрольні питання.

4.4. Контрольні питання

1. Поняття цифрового фільтра. Рекурсивні і нерекурсивні фільтри.
2. Чотири основні форми реалізації фільтрів.
3. Основні характеристики цифрових фільтрів.
4. Класифікація методів проектування цифрових фільтрів.

Лабораторна робота 5.

Вивчення основ обробки зображень у системі MATLAB

Мета роботи – ознайомлення з основними процедурами обробки зображень в системі **MATLAB**.

5.1. Теоретичні відомості

Для розробки програм по обробці зображень для цифрових сигнальних процесорів, необхідно обрати метод й алгоритм обробки, провести їхній аналіз й визначити оптимальні параметри процедур. Ці процедури повинні бути реалізовані у вигляді програмних модулів, написаних на мові C/C++ або асемблері конкретного сигнального процесора. Найбільш доцільно проводити вищевказані дослідження в області обробки зображень із використанням системи **MATLAB**, що є зручним і досить простим інструментом дослідження. Крім того, у її склад входить компілятор мови **MATLAB** на мову C/C++ і бібліотека математичних функцій на C++. Таким чином, процедури обробки зображень, написані в системі **MATIAB**, без особливих труднощів перетворяться у виконуючі модулі для цифрових сигнальних процесорів сімей **7MS** й **ADSP**. Для дослідження методів обробки зображень у системі **MATIAB 5.2** розроблено пакет **Image Processing Toolbox**, що включає широкий набір функцій, і може бути доповнений написаними користувачем власними нетиповими функціями.

Розглянемо функції, алгоритми яких найбільш широко використовуються. З повним списком функцій й особливостями їхнього використання можна ознайомитися в середовищі **MATLAB Command Window** за допомогою команд **help images** та **help ім'я функції**.

Базовим типом даних в **MATIAB** є прямокутна матриця впорядкованої безлічі дійсних або комплексних елементів зображення. Тому доступ до любого піксіля зображення здійснюється як до елемента матриці.

MATLAB 5.2 підтримує чотири типи зображень: 1) кольорове індексоване; 2) напівтонове; 3) бінарне; 4) кольорове **RGB**.

Для індексованого кольорового зображення потрібне наявність двох матриць: таблиці кольоровості й матриці індексів. Таблиця кольоровості **(colormap)** – це набір використовуваних у зображенні кольорів, а матриця індексів – набір номерів (індексів) від **0** до n**–1** відповідних кольорів таблиці кольоровості. Розмір таблиці кольоровості залежить від кількості N_u , що використані у зображенні кольорів, і дорівнює N_u×**3**. Кожному індексу таблиці відповідають значення трьох **RGB** кольорів. Для відображення кожного із цих кольорів використаються дійсні числа з подвійною точністю в діапазоні від **0.0** (рівень чорного) до **1.0** (максимальна інтенсивність).

У напівтоновому типі зображення використається тільки одна матриця, що містить значення інтенсивностей (рівнів сірого) кожного з пікселів. Значення інтенсивності представляються реальними числами з подвійною точністю в діапазоні від **0.0** до **1.0**, де рівень **0.0** відповідає чорному а **1.0** – білому кольорам.

У бінарному (чорно-білому) зображенні використається тільки два рівні: **0** – чорний й **1** – білі кольори. У системі **MATIAB** ці рівні показуються числами з подвійною точністю. Бінарні зображення можна представляти логічною (булевою) матрицею. У цьому випадку при обробці зображення застосовуються логічні операції.

У кольоровому **RGB**-зображенні використаються три матриці інтенсивностей відповідних кольорів, які представляються реальними числами з подвійною точністю в діапазоні від **0,0** до **1,0**. Кольори й інтенсивність пікселів визначається комбінацією інтенсивностей відповідних їм кольорових компонентів.

Якщо тип зображення заздалегідь не відомий, то його можна визначити за допомогою функцій **ISBW**, **ISGRAY**, **ISIND** або **ISRGB**. Ці функції перевіряють на істинність відповідно бінарного чорно-білого, напівтонового, індексованого або кольорового **RGB** зображення.

Функція **FLAG=ISGRAY(A)**. повертає **1**, якщо це зображення інтенсивностей рівня сірого, у противному випадку повертається **0**.

При необхідності зображення деяких типів можуть бути конвертовані в інший тип, зокрема, можна перетворити індексовані зображення в напівтонові або кольорові **RGB** (функції **IND2GRAY**, **IND2RGB**), або зробити їхнє зворотне перетворення в індексовані (функції **GRAY2IND**, **RGB2IND**).

Наприклад, якщо прочитати з файлу **floers.tif** **RGB**-зображення й перетворити його в індексований тип з матрицею інтенсивностей **X** і картою кольоровості (палітрою) **map**, що містить 128 кольорів, варто включити в програму наступні фрагменти.

```
RGB=Hmread ('flowers.tif');
[X, map]= rgb2ind(RGB,128);
imshow (X, map).
```

Інструментальні засоби **MATLAB** дозволяють обробляти зображення окремими фрагментами (за термінологією **MATLAB-блоками**) з використанням віконних функцій. Це дозволяє знизити затрати пам'яті комп'ютера й підвищити швидкість обробки. При цьому немає необхідності програмно задавати переміщення вікна (блоку) по

зображенню. Задана функція автоматично виконується для кожного фрагмента всього зображення, що обробляється. Окремі блоки по черзі покривають матрицю зображення, починаючи з лівого верхнього кута. Якщо блоки не вкладаються в границі зображення, то відсутні пікселі автоматично заповнюються нулями. Аналогічне заповнення нулями виробляється при обробці пікселів, розташованих на краях зображення.

Для блокової обробки в **Image Processing Toolbox** використовується процедура **BLKPROC**, що виділяє окремий блок (фрагмент) зображення й передає його у функцію визначену користувачем. Потім **BLKPROC** збирає повернуті функцією блоки для створення вихідного зображення. Функція, використовувана для обробки кожного блоку, повинна повертати матрицю того ж розміру, що й блок. Функція **BLKPROC** описується в такий спосіб:

B=BLKPROC(A,[m n],'fun'),

де **A** – матриця вхідного зображення, **B** – матриця вихідного зображення; [**m n**] – розмір фрагмента в пікселях; **fun** – функція обробки фрагмента.

При блокової обробки зображення за допомогою функції **SIZ=BESTBLK([M N],K)** можна спочатку знайти оптимальний розмір блоку. Тут **M**, **N** – розмір вихідного зображення, а **K** – граничні розміри шуканого блоку.

Для читання зображення із графічного файлу застосовується функція **IMREAD**. Її синтаксичне подання:

A=IMREAD(FILENAME,FMT)

або **[X, MAP]=IMREAD(FILENAME,FMT).**

У першому випадку проводиться зчитування зображення з файлу. Якщо зображення напівтонове, то **A** – це двовимірний масив значень інтенсивностей елементів зображення. У випадку трикольорового (**RGB**) зображення **A** є тривимірним масивом розміром **M**x**N**x**3**. Параметр **FMT** специфіцірує формат подання зображення. Серед форматів подання графічних зображень використаються **.bmp**, **.Jpeg**, **.pcx**,**.tif** і ін.

Функція **[X, MAP]=IMREAD(FILENAME, FMT)** зчитує індексоване зображення з файлу в **X** , а карту кольорів – в **MAP**.

Функція **IMWRITE(A,FILENAME,FMT)** записує зображення **A** в файл з ім'ям **FILENAME.FMT**, а функція **IMWRITE(X, MAP,FILENAME, FMT)** записує у файл індексоване зображення **X** і пов'язану з ним карту кольорів **map**.

Функція **IMCROP(I,[XMIN YMIN WIDTH HEIGHT]**) видаляє фрагмент зображення заданого розміру. Параметри **XMIN**, **YMIN** задають початкові координати прямокутного фрагмента, а **WIDTH**, **HEIGHT** – кількість колонок і рядків фрагмента відповідно. Прямокутник можна визначити за допомогою миші.

Функція **IMHIST(I,N)** відображає гістограму розподілу інтенсивності пікселів зображення **I** із числом рівнів інтенсивності **N**. За замовчуванням **N** приймається рівним **256**.

Функція **EDGE** визначає границі інтенсивності зображення. При цьому як параметри функції вказується метод визначення границі, рівень порога й ряд інших.

Функція **HISTEQ** поліпшує контраст зображення за рахунок вирівнювання гістограми.

Функція **IMNOISE(I,TYPE,...)** додає до зображення інтенсивностей **I** адитивний гауссівський або мультиплікативний шум типу **TYPE**. Для кожного типу шуму додатково задаються його параметри.

Для фільтрації зображень використаються функції **FILTER2** й **FSPECIAL(TYPE)**. У другій функції специфікація **TYPE** задає тип використовуваного лінійного або нелінійного фільтра. Зокрема може бути заданий один з наступних типів фільтрів:

'gaussian' – гауссівський низькочастотний фільтр ;

'sobel' – виділення границь на основі оператора Собела;

'previt' – виділення границь на основі оператора Превіта;

'laplacian' – з використанням оператора Лапласа;

'average' – усредняючий фільтр .

Крім цього використаються й інші функції фільтрації. Такі, як **MEDFILT2** – медіанний фільтр; **WIENER2** – адаптивний вінерівський фільтр, що подавляє шуми.

У складі **MATLAB** є широкий набір функцій для різних перетворень зображень (**Image transforms**):

- **DCT2** й **IDCT2** – обчислення двовимірного прямого й зворотного дискретно-косинусного перетворення відповідно;
- **FFT2, FFTn, IFFT2, IFFTn** – прямі й зворотні, двовимірні й n-мірне швидке перетворення Фур'є;
- **RADON** – перетворення Радону й ряд інших.

Нижче приводяться приклади використання різних функцій системи MATLAB для обробки зображень.

Приклад 1. Використання різних функцій для обробки зображень.

```
>>I=imread('forest.tif); %Читання зображення з %файлу.
>>IC=imcrop(I,[50 100 40 60]; % Вирiз фрагмента iз вказiвкою
                              % координат його початку й розмiрiв
>>imshow(I),figure(2),imshow(IC) % вiдображення повного й
                                 % усiченого зображення
>>J=imnoise(I,'gaussian',0,0.05); % вплив на зображення
               % гауссовской перешкоди з нульовим середнiм i нормованим
               % среднеквадратическим вiдхиленням,рiвним 0.05
>>figure(3),imshow(J) % вiдображення на екранi зашумленого
                      %зображення
>>M=medfilt2(J);  % медiанна фiльтрацiя
>>W=wiener2(J,[3 3]);  % Винерiвська фiльтрацiя
>>figure(4),imshow(M),figure(5),imshow(W)
>>figure(6),imhist(I),figure(7),imhist(IC);
% Одержання й вiдображення на екранi гiстограм повного й
% усiченого зображень.
>>BWs=edge(I,'sobel'); % виділення контурів
% з використанням оператора Собела
>>BWp=edge(I,'prewitt');  % виділення контурів
% з використанням операторів Превіта
>>figure(1),imshow(PWs),figure(2),imshow(PWp)
% вiдображення гiстограм
```

Зверніть увагу на зміну якості зображення при варіації параметрів перешкоди й потім при проведенні різних видів фільтрації, а також на істотну відмінність гістограм повного зображення і його фрагмента.

Приклад 2. Скласти програму стиску зображення з використанням дискретно-косинусного перетворення й функцій блокової обробки.

```
>>load trees % завантаження  вхiдного зображення
>>I=ind2gray(X,map); % перетворення iндексованого зображення
                     % в напiвтонове
>>J=blkproc(I, [8   8],'dct2'); % поблочне дискретно-косинусне
                                % перетворення iз блоком 8x8
>>nz=find(abs(J)< 0.1); % знаходження iндексiв пикселiв,
                        % iнтенсивнiсть яких менше 0.1
>>J(nz)=zeros(size(nz)); % заміна цих пикселів нульовим значенням
                         % (властиво стиск)
>>K=blkproc(J,[8  8],'idct2'); % поблочне вiдновлення зображення
>>imshow(I), figure, imsho(K) % перегляд вихiдного й вiдновленого
                              % зображень
```

Приклад 3. Контрастування зображення

```
 % Приклад демонструє контрастування зображення
>>I=imread('pout.tif'); % Читання зображення з файлу
```

```
>>imshow(I);
 % Побудова гістограми вихідного зображення
>>figure, imhist(I);
 % Контрастування вихідного зображення
>>I=imadjust(I,[0 180]/255,[],2.5);
 % Виведення перетвореного зображення на екран
>>figure, imshow(I);
 % Виведення гістограми перетвореного зображення
>>figure, imhist(I);
```

5.2. Порядок виконання роботи

1 Використовуючи методичний матеріал, вивчіть основні функції для обробки зображень, які є в системі **MATLAB**.

2 Відповідно до таблиці 5.1 виберіть ім'я файлу, у якому перебуває тестове зображення.

Таблиця 5.1

Варіант	Ім'я файлу
1	pout.tif
2	rice.tif
3	saturn.tif
4	trees.tif
5	moon.ti'

Варіант	Ім'я файлу
6	circles.tif
7	cameraman.tif
8	testpat1.tif
9	forest.tif
10	trees.tif

3 Створіть **Файл-сценарій** для аналізу обраного зображення й виконайте його з командного рядка системи **MATLAB**. Перелік операцій, які треба виконати над зображенням наступний:

- завантаження вхідного зображення;
- виріз фрагмента із вказівкою координат його початку й розмірів (вихідні дані для цієї операції наведені в таблиці 5.2);
- відображення повного й усіченого зображення;
- одержання й відображення на екрані гістограм повного й усіченого зображень;
- вплив на зображення гауссівської перешкоди з нульовим середнім і нормованим середньоквадратичним відхиленням, рівним 0.05;
- відображення на екрані зашумленого зображення;
- медіанна фільтрація зашумленого зображення;
- відображення на екрані відфільтрованого зображення;
- Вінерівська фільтрація зашумленого зображення;

– відображення на екрані відфільтрованого зображення;
– виділення контурів з використанням оператора Собела;
– виділення контурів з використанням операторів Превіта.

4 Створіть **Файл-сценарій** стиску зображення з використанням дискретно-косинусного перетворення й функцій блокової обробки. Рівень інтенсивності пікселів, які відкидаються, прийняти рівним 0.2. Виконайте отриманий **Файл-сценарій** з командного рядка.

5 Створіть **Файл-сценарій** контрастування зображення. Виконайте отриманий **Файл-сценарій** з командного рядка системи **MATLAB**.

Таблиця 5.2

Варіант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
XMIN	150	40	120	220	180	200	35	1	2	10
YMIN	100	60	80	220	120	100	45	1	2	10
WIDTH	40	20	30	100	80	60	200	200	100	50
HEIGHT	60	30	20	100	90	60	200	200	120	60

5.3. Зміст звіту

1 Титульний лист, виконаний відповідно до СТП.
2 Мета роботи.
3 Тексти створених *Файлів-сценаріїв*
4 Протокол сеансу роботи в середовищі MATLAB.
5 Відповіді на контрольні питання.

5.4. Контрольні питання

1 Основні цілі й завдання обробки зображень.
2 Типи растрових зображень.
3 Дайте визначення бінарного, напівтонового, палітрового, повнокольорового зображень.
4 Операції над зображеннями; Рекурсивні й нерекурсивні операції.
5 Поняття околиці. 4-хточкова, 8-миточкова околиця.
6 Поняття вікна. Опорна крапка вікна.
7 Зміна контрастності зображення. Метод контрастного розтягування.
8 Лінійна фільтрація зображень. Рівняння лінійної фільтрації.
9 Вікна для придушення шумів і згладжування.
10 Нелінійна фільтрація. Медіанний фільтр.
11 Підкреслення перепадів яскравості. Вікна Превіта, Кирша, Собела.
12 Підкреслення ліній і фокусування зображень. Вікна Лапласа.

ЛІТЕРАТУРА

1. Visual DSP++ 3.0 Getting Started Guide for ADSP-21xx DSPs, 2002
2. Visual DSP++ Development Environment for Analog Devices DSPs, 2002.
3. Visual DSP++ 3.0 C Compiler and Library Manual for ADSP-218X DSPs, 2002.
4. FIR-filters without tears // Analog Dialogue #17-2, 1983.
5. Signals and Systems: A Practical Approach, 2nd edition Dr. D Sundararajan, Concordia University Springer International Publishing, 2022
6. Квєтний, Р. Н. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 2. // Р. Н. Квєтний, І. В. Богач, О. Р. Бойко, О. Ю. Софина, О. М. Шушура. – Вінниця: ВНТУ, 2013. – 235 с.
7. Заболотній С. В. Цифрове оброблення сигналів: Посібник для студентів напряму підготовки 6.050901 "Радіотехніка" усіх форм навчання [Електронний ресурс] / Авт.-укл. С. В.Заболотній ; За ред. проф. Ю. Г. Леги ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси: ЧДТУ, 2010. – 119 с.
8. Тотосько О.В. Цифрова обробка сигналів та зображень : навчальний посібник / О.В. Тотосько , П.Д. Стухляк. - Тернопіль : ТНТУ імені Івана Пулюя, 2016. - 132 с.
9. Gonzalez R. Digital Image Processing / R. Gonzalez, R. Woods. – Pearson Education Limited, 2018 (4th edition). – 1022 p. – ISBN 10: 1-292-22304-9.
10. Alessio. Digital Signal Processing and Spectral Analysis for Scientists / Alessio, Silvia Maria. Springer. – 2016. – 200 p. –ISBN 978-3-319-25468-5.
11. Бортник Г.Г. Цифрова обробка сигналів в телекомунікаційних системах підручник. Г.Г. Бортник / 2014. – Вінниця: ВНТУ 2014.— 231с.
12. Ушенко Ю.О. Основи та методи цифрової обробки сигналів: від теорії до практики навчальний посібник. Ю.О. Ущенко, В.В. Дрожак, М.С. Гавриляк, М.В. Талах / 2021. – Чернівці : Чернівецький національний університет імені Юрія Федьковича 2021. – 307 с.
13. What Is Digital Signal Processing? – URL: https://ch.mathworks.com/discovery/digital-signal-processing.html