

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

ЦИФРОВА ОБРОБКА СИГНАЛІВ

Лабораторний практикум

Вінниця ВНТУ 2014

Міністерство освіти і науки України
Вінницький національний технічний університет
Кафедра обчислювальної техніки

АРХІТЕКТУРА КОМП'ЮТЕРІВ
Лабораторний практикум, частина 1

Затверджено Методичною радою Вінницького національного технічного університету як методичні вказівки для студентів напряму підготовки 6.050102 Комп'ютерна інженерія

Протокол № від „ ” 2014 р.

Вінниця ВНТУ 2014

Цифрова обробка сигналів. Лабораторний практикум / Методичні вказівки для студентів напряму підготовки 6.050102 Комп'ютерна інженерія

/Уклад. Л.В.Крупельницький, М.А.Очкуров – Вінниця: ВНТУ, 2014 – с. /

Рекомендовано до видання Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України

Лабораторний практикум передбачає: ознайомлення з системою Matlab, способами формування та візуалізації сигналів, побудови їх спектрів на основі швидкого перетворення Фур'є, правилами розрахунку та дослідження нерекурсивних і рекурсивних цифрових фільтрів, дослідження кореляційних функцій.

Укладачі: Леонід Віталійович Крупельницький, к.т.н., доцент,
Микола Андрійович Очкуров, старший викладач.

Редактор

Відповідальний за випуск

Рецензенти: , доктор технічних наук, професор
, кандидат технічних наук, доцент

Зміст

Вступ	4
Наскрізне індивідуальне завдання на лабораторні роботи № 1-5	5
Лабораторна робота 1. Ознайомлення з системою Matlab	6
Лабораторна робота 2. Дискретизація і квантування сигналів при їх аналого-цифровому перетворенні	
Лабораторна робота 3. Спектральний аналіз сигналів методом швидкого перетворення Фур'є	18
Лабораторна робота 4. Застосування вікон при спектральному аналізі	
Лабораторна робота 5. Цифровий нерекурсивний фільтр низьких частот	20
Лабораторна робота 6. Цифровий рекурсивний фільтр високих частот	
Лабораторна робота 7. Кореляційний аналіз	
Лабораторна робота 8. Застосування ЦОС в інформаційно-обчислювальній техніці та в телекомунікаційних мережах	
Перелік рекомендованої літератури	25

Вступ

Метою дисципліни «Цифрова обробка сигналів» є формування знань і навичок, необхідних для проектування апаратних і програмних засобів цифрової обробки сигналів. Дисципліна дає математичні основи цифрової обробки сигналів, ознайомлює з основними методами перетворення цифрових сигналів, алгоритмами цифрової обробки сигналів та спеціалізованими процесорами, що їх реалізують.

Основні завдання вивчення дисципліни полягають у тому, щоб навчити студентів розумінню математичних моделей сигналів, процесів та систем обробки, ознайомити їх з основними методами перетворення цифрових сигналів і задачами, для розв'язання яких вони використовуються, а також галузями застосування цифрової обробки сигналів.

Згідно з вимогами освітньо-професійної програми студенти повинні:

- *знати*: методи, математичні, алгоритмічні та апаратні основи перетворення аналогових сигналів в цифрові, основи спектрального та кореляційного аналізу, цифрової фільтрації, апроксимації і децимації;
- *вміти*: розробляти апаратні та програмні засоби для розв'язання задач цифрової обробки сигналів у різних галузях науки та техніки, виконувати інженерні розрахунки систем обробки сигналів, аналізувати отримані результати та приймати рішення за результатами цих розрахунків, користуватись науковою та довідковою літературою, знаходити раціональні методи розв'язання практичних задач.

Для виконання лабораторних робіт з курсу «Цифрова обробка сигналів» можна скористатися студентською версією, Trial-версією або працювати з додатком для роботи з віддаленим сервером MATLAB, перейшовши на сайт компанії-розробника MathWorks за посиланням www.mathworks.com/academia/student_version/

Наскрізне індивідуальне завдання на лабораторні роботи № 1-5 з курсу ЦОС- «цифрова обробка сигналів»

В системі MATLAB або за допомогою інших програмних пакетів обробки сигналів чи за допомогою універсальних мов програмування виконати наступні задачі.

Лабораторна робота № 1

Освоїти систему Matlab в обсязі, достатньому для виконання лабораторних робіт з курсу.

Лабораторна робота № 2

Використати модифіковану функцію ШПФ системи Matlab **fftdemo.m**.

При цьому, змінити її відповідно до свого варіанта.

Здати функцію суми сигналів, залежну від часу $F(t)$ на проміжку $0 \dots 100$ мс з числом відліків 256, при цьому:

частота дискретизації – $(N+20 \cdot G)$ кГц,

де N – номер студента за списком _____, G - номер групи (1,2),

частота 1-го сигналу (N) кГц, амплітуда $(N/2)$ В, початкова фаза $45 \cdot G$ град,

частота 2-го сигналу $(N/3)$ кГц, амплітуда $(N/4)$ В початкова фаза $90 \cdot G$ град, додати шумовий сигнал амплітудою $(N/100)$ В.

Вивести і роздрукувати графік функції $F(t)$.

Лабораторна робота № 3

Побудувати амплітудний спектр сигналу методом швидкого перетворення Фур'є, виділити на ньому частоти вхідних сигналів, їх амплітуди, що повинні відповідати заданим.

Роздрукувати спектр.

Лабораторна робота № 4

Використати функцію системи Matlab **sptool.m**

Розрахувати і реалізувати програмно фільтр низьких частот ФНЧ з кінцевою імпульсною характеристикою КІХ порядку $(N+4) \cdot G$ з метою фільтрації сигналу 1 (повинен залишитись сигнал 2). Роздрукувати спектр після фільтрації.

Лабораторна робота № 5 Розрахувати і реалізувати програмно фільтр високих частот ФВЧ з нескінченною імпульсною характеристикою НІХ $(5-G)$ -го порядку з метою фільтрації сигналу 2 (повинен залишитись сигнал 1). Роздрукувати спектр після фільтрації.

Для кожної роботи Надати на роздруківках повний текст програми, початкові, проміжні значення, коефіцієнти і структури фільтрів. Пояснити хід виконання задачі і отримані результати. Підготувати теоретичний матеріал.

Лабораторна робота № 1

Ознайомлення з системою MatLAB

Мета роботи

Метою даної лабораторної роботи є ознайомлення з основами роботи в середовищі MatLAB: вивчення типів використовуваних даних, робота з масивами, побудова графіків, ознайомлення з вбудованим мовою програмування.

Вступ

Після запуску системи MatLAB на екрані з'явиться вікно, показане на рис. 1.1. Вікно містить три області: Command Window, призначене для введення команд і виведення результатів; Command History, що містить історію всіх виконаних в Command Window команд і вікно Workspace, де відображаються всі створені в робочій області змінні.

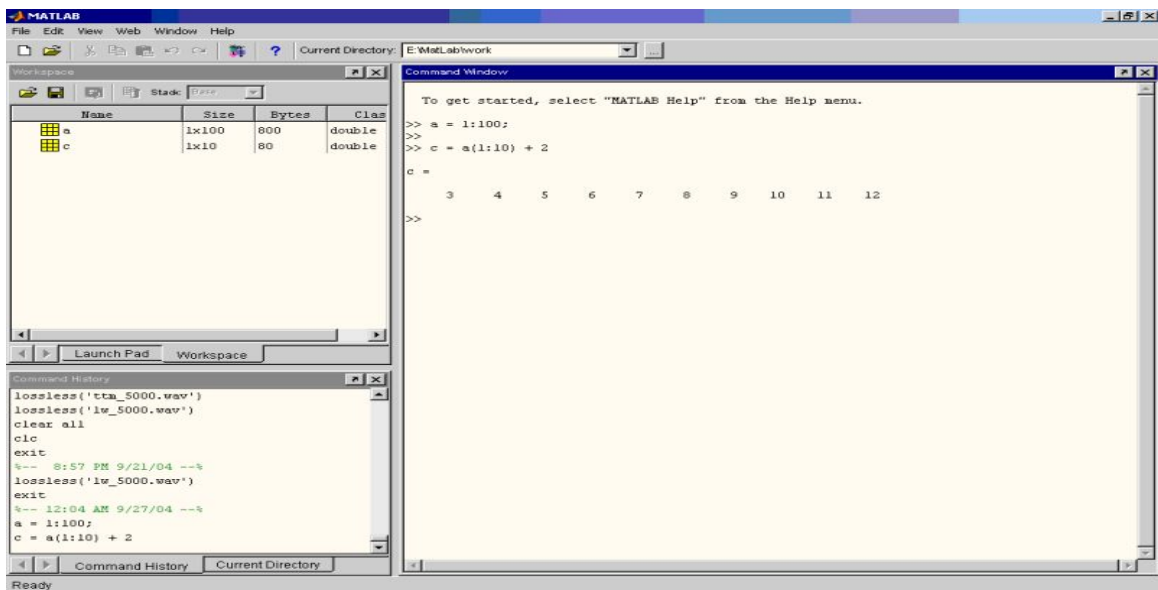


Рис . 1.1 . Командне вікно системи MatLAB

Знак « >> » показує готовність системи до виконання введених команд. Набравши найпростіші математичні вирази в природній формі запису , відразу ж обчислюється результат. Це вираз може бути записано в двох видах:

< Вираз > або <Ім'я змінної> = < Вираз > .

У другому випадку результат не тільки обчислюється , а й присвоюється зазначеній змінній. MatLAB не вимагає від користувача спеціальних команд для оголошення змінних , вони створюються автоматично при першій вказівці користувачем їхнього імені. У першому випадку насправді результат виразу присвоюється спеціальної службовій змінній, яка має ім'я **ans** , так само можна використовувати цю змінну в розрахунках. Якщо немає необхідності , щоб MatLAB виводив результати проміжних виразів на екран , то необхідно поставити в кінці виразу символ « ; ».

Типи даних

Фактично MatLAB містить один тип даних - масив або матриця . Масив це група комірок пам'яті , що мають одне ім'я. Масиви бувають одномірні - рядок або стовпець , прямокутні , квадратні (число рядків дорівнює числу стовпців) . Коли Ви вказуєте змінну і присвоюєте їй одне число , фактично MatLAB створює матрицю з одного рядка і одного стовпця (розмірність масиву відображається у вікні Workspace в поле Size) . Будь-яка змінна в MatLAB - це матриця .

Нижче наведені приклади стовпця - а , рядка - б , прямокутної матриці - в , квадратної матриці - г , матриці одиничної розмірності - е (простої змінної)

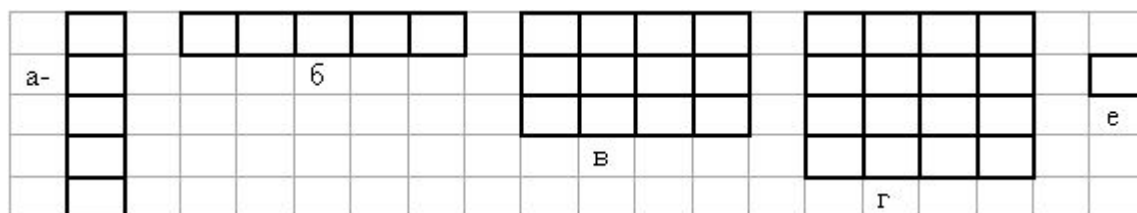


Рис . 1.2 . Види матриць

Крім одновимірних і двовимірних матриць MatLAB підтримує ряд інших типів даних. До них відносяться багатовимірні масиви , рядки , структури , масиви осередків , а також об'єкти .

Дії над матрицями

1 . Створення матриць

Найпростішою операцією з матрицею є її створення . Для створення рядка необхідно вказати його ім'я , знак рівності і в квадратних дужках через кому або через пробіл перерахувати значення елементів :

```
>> A = [1 2 3 4 5];
```

У випадку якщо необхідно створити стовпець чисел, то як роздільник виступає символ крапка з комою :

```
>> B = [ 1 ; 3 ; 5 ; 7 ] ;
```

Для створення квадратної або прямокутної матриці знадобиться чергувати обидва ці способи .

```
>> C = [ 1 2 3 ; 4 5 6 ; 7 8 9 ] ;
```

2 . Створення матриць спеціального виду

Для генерації векторів користувачеві надається наступна команда:

<Ім'я вектора > = < Початкове значення > : < Крок > : < Кінцеве значення >

```
>> X = 6:0.2:26 ;
```

У результаті буде створено вектор X наступного вигляду:

```
6 6.2 6.4 6.6 ... 25.6 25.8 26
```

У математиці часто зустрічаються матриці спеціального виду . Нижче наведено ряд з них:

Одинична матриця , рис. 1.3.а. У одиничній матриці всі елементи рівні нулю , крім елементів, які стоять на головній діагоналі (матриця є квадратною) . Для створення одиничної матриці необхідно подати команду <Ім'я матриці > = eye (< Розмір >) ;

```
>> A = eye ( 4 ) ;
```

Матриця з усіма одиницями , рис. 1.3.б. Ця матриця містить одиниці у всіх осередках. Для створення матриці необхідно вказати

<Ім'я матриці > = ones (< Кількість рядків > , < Кількість стовпців >) ;

```
>> B = ones ( 3 , 4 ) ;
```

Нульова матриця , рис. 1.3.в. Ця матриця містить у всіх своїх комірках одні нулі . Для створення необхідно виконати наступну команду:

```
<Ім'я матриці > = zeros ( < Кількість рядків > , < Кількість стовпців > ) ;
>> C = zeros ( 4 , 2 ) ;
```

Випадкова матриця , рис. 1.3.г. Всі значення цієї матриці виходять з генератора випадкових чисел. Для створення такої матриці необхідно дати таку команду:

```
<Ім'я матриці > = rand ( < Кількість рядків > , < Кількість стовпців > ) ;
>> D = rand ( 3 , 4 )
```

1	0	0	0	1	1	1	1	0	0	5	6				
0	1	0	0	1	1	1	1	0	0	4	1				
0	0	1	0	1	1	1	1	0	0	0	3				
0	0	0	1					0	0						
	а				б				в		г				

Рис . 1.3 . Спеціальні матриці .

3 . Доступ до комірок матриці

Для доступу до комірок матриці необхідно вказати ім'я матриці , номер рядка та номер стовпця. Нумерація рядків і стовпців ведеться з ОДИНИЦІ ! Номери пишуться в круглих дужках. Загальний формат запису:

```
<Ім'я масиву > ( < Номер рядка > , < Номер стовпця > ) ;
```

```
>> D ( 3,2 )
```

```
ans =
```

```
2
```

```
>> D ( 3,1 ) = 9 ;
```

4 . Скалярні операції

У математиці для всіх матриць визначена операція множення (ділення) матриці на скаляр (число) - « . *" (" . / »). Всі значення матриці в цьому випадку множаться (діляться) на це число. Всі скалярні (поелементні)

операції в MatLAB позначаються за допомогою крапки (тобто для складання « . + », Для вирахування « . - » і т.д.). Слід пам'ятати, що відсутність крапки перед знаком дії призводить до виконання матричної операції, яка може вимагати виконання спеціальних вимог до операндів (наприклад, збігу розмірів)

$$\begin{array}{|c|c|c|} \hline A(1,1) & A(1,2) & A(1,3) \\ \hline A(2,1) & A(2,2) & A(2,3) \\ \hline A(3,1) & A(3,2) & A(3,3) \\ \hline \end{array} \cdot r = \begin{array}{|c|c|c|} \hline A(1,1).\text{*}r & A(1,2).\text{*}r & A(1,3).\text{*}r \\ \hline A(2,1).\text{*}r & A(2,2).\text{*}r & A(2,3).\text{*}r \\ \hline A(3,1).\text{*}r & A(3,2).\text{*}r & A(3,3).\text{*}r \\ \hline \end{array}$$

5. Додавання, віднімання скаляра з матриці

Крім операції множення матриці на скаляр, для матриці і скаляра визначені операції додавання «+» і віднімання "-". Дії так само виконуються з кожною коміркою матриці окремо.

6. Додавання, віднімання матриць

Ця операція допустима тільки з матрицями однакового розміру. При виконанні операції дія виконується за відповідними один одному комірками.

приклад:

$$\begin{array}{|c|c|c|} \hline a(1,1) & a(1,2) & a(1,3) \\ \hline a(2,1) & a(2,2) & a(2,3) \\ \hline a(3,1) & a(3,2) & a(3,3) \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline b(1,1) & b(1,2) & b(1,3) \\ \hline b(2,1) & b(2,2) & b(2,3) \\ \hline b(3,1) & b(3,2) & b(3,3) \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a(1,1)+ b(1,1) & a(1,2)+ b(1,2) & a(1,3)+ b(1,3) \\ \hline a(2,1)+ b(2,1) & a(2,2)+ b(2,2) & a(2,3)+ b(2,3) \\ \hline a(3,1)+ b(3,1) & a(3,2)+ b(3,2) & a(3,3)+ b(3,3) \\ \hline \end{array}$$

7. Добуток матриць

При виконанні операції перемноження матриць виконується послідовне множення рядка на стовець. При цьому кількість стовпців в першу матриці повинна дорівнювати кількості рядків у другій матриці. Матриця результату буде мати стільки ж рядків скільки і в першу матриці, і кількість стовпців дорівнює кількості стовпців у другій матриці.

$$\begin{array}{|c|c|c|} \hline a(1,1) & a(1,2) & a(1,3) \\ \hline a(2,1) & a(2,2) & a(2,3) \\ \hline a(3,1) & a(3,2) & a(3,3) \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline b(1,1) & b(1,2) & b(1,3) \\ \hline b(2,1) & b(2,2) & b(2,3) \\ \hline b(3,1) & b(3,2) & b(3,3) \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a(1,1)+ b(1,1) & a(1,2)+ b(1,2) & a(1,3)+ b(1,3) \\ \hline a(2,1)+ b(2,1) & a(2,2)+ b(2,2) & a(2,3)+ b(2,3) \\ \hline a(3,1)+ b(3,1) & a(3,2)+ b(3,2) & a(3,3)+ b(3,3) \\ \hline \end{array}$$

Побудова графіків

Виведення одного графіка

MatLAB надає наступні функції для роботи з графікою:

`plot (<Масив>)` - побудова графіка значень з масиву X від номера відліку.

`plot (<Масив точок по осі X>, <Масив точок по осі Y>)` - побудова графіка значень з масиву Y від значень з масиву X .

При виклику команди створюється вікно з вказаним графіком.

Виведення декількох графіків

Для виведення декількох графіків на одному вікні Вам необхідно вказати їх послідовно, наприклад:

```
>> t=-10:0.1:10;
```

```
>> x1=sin(t);
```

```
>> x2=cos(t)./t;
```

```
>> plot(t,x1,t,x2)
```

або

```
>> t=-10:0.1:10;
```

```
>> x1=sin(t);
```

```
>> x2=cos(t)./t;
```

```
>> plot(t,x1)
```

```
>> hold on;
```

```
>> plot(t,x2)
```

```
>> hold off
```

Команда `hold` в даному прикладі служить для « фіксації » поточного вікна виведення графіків (`hold on` « включає утримання » , `hold off` - знімає) .

Програмування в системі MatLab

Програма в середовищі MatLAB може бути введена двома способами:

безпосередньо в Command Window , або заздалегідь в редакторі.

Програмування в системі MatLAB дуже близько до звичайного програмування . Програма створюється в будь-якому текстовому редакторі. Файл повинен мати розширення *. m.

Текст М- файлу функції повинен починатися з заголовка **function** , що має наступний вигляд:

```
function [ y1 , y2 , ...] = fname ( x1 , x2 , ...)
```

Тут **fname** - ім'я функції , **x1** , **x2** і т.д. - вхідні параметри , **y1** , **y2** і т.д. - вихідні параметри . Вхідні і / або вихідні параметри можуть бути відсутні.

Насправді ім'я функції визначається не рядком **fname** , а ім'ям , під яким збережений М- файл , воно має збігатися з ім'ям функції .

Як приклад створимо функцію **myfunc** , яка будуватиме тривимірних графік і приймати три вхідних параметра : точку початку побудови графіка , точку закінчення побудови , крок.

Для цього , відкривши вікно редактора командою меню **File -> New -> M-file** , вводимо текст:

```
function myfunc ( x1 , x2 , step )  
[X , Y ] = meshgrid ( [ x1 : step : x2 ] ) ;  
Z = X. * exp ( -X. ^ 2 - Y . ^ 2 ) ;  
mesh ( X , Y , Z ) ;
```

Ввівши текст , необхідно зберегти файл під ім'ям **myfunc** .

Іншим способом створення нового М- файлу служить команда **edit fname.m**.

За допомогою неї можна як редагувати вже наявні М- файли , так і створювати нові.

Для того щоб функція була доступна з системи MatLab , система повинна бути здатна знайти відповідний М -файл. Пошук файлів здійснюється наступним чином : спочатку проглядається поточний робочий каталог (його ім'я показано в панелі інструментів головного вікна , рис 1.1) , а потім каталоги входять до шлях пошуку (MatLab search path) .

Для виклику М файлу необхідно набрати його ім'я в командному рядку MatLAB , і якщо необхідно його аргументи. Важливим елементом полегшує

програмування є коментарі. Рядок коментаря починається в MatLAB символом '%' .

Перевірка умови

Оператор перевірки умови дозволяє організувати розгалуження виконання програми . Зовнішній вигляд оператора представлений на рис. 2.3 .

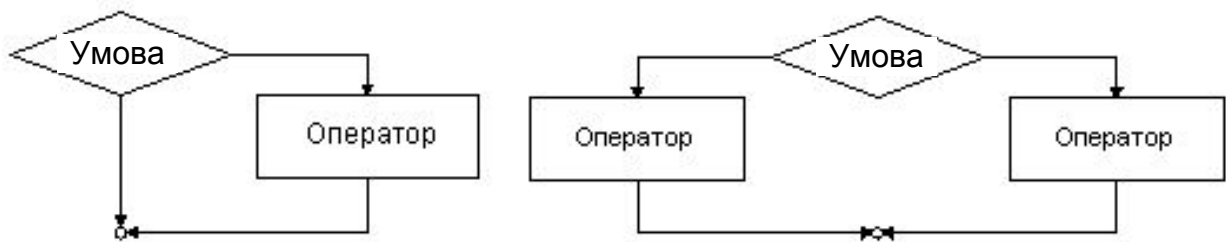


Рис . 2.3 . Блок - схема умовного оператора: редукована і повна форми

Формат запису оператора , редукована форма :

```
if умова  
оператори  
end
```

повна форма :

```
if умова  
оператори  
elseif умова  
оператори  
else  
оператори  
end
```

Введення з клавіатури

```
x = input ( ' рядок підказки ' )
```

```
x = input ( ' рядок підказки ' , ' s ' )
```

Функція input виводить на екран рядок підказки і чекає введення змінної.

Функція x = input (' рядок підказки ' , ' s ') повертає введений користувачем рядок.

При введенні змінних допустимо користуватися стандартними функціями.

Цикли.

MatLAB надає користувачеві два способи організації циклу . Перший з них цикл з відомою кількістю повторень. Блок - схема цього циклу представлена на рис. 2.4 .



Рис. 2.4. Блок-схема цикла for

Формат оператора:

```
for змінна = початкове значення : [ крок : ] кінцеве значення  
оператори  
end
```

Поле «крок» в конструкції оператора не є обов'язковим. наприклад:

```
for i = 1:100  
x ( i ) = sin ( 2 * PI * i/100 ) ;  
end
```

Тіло циклу обов'язково закінчується службовим словом **end** .

При роботі з циклом **for** допустимо використання оператора переривання циклу **break** . При виконанні оператора **break** робота циклу завершується і управління передається на наступний після кінця циклу оператор . Блок - схема програми , ілюструє використання оператора **break** , наведена на рис. 2.5 .

Текст програми, що відповідає рис. 2.5 .

```
for i = 1:100
оператори
if a ( i ) == 0
break
end
оператори
end
```

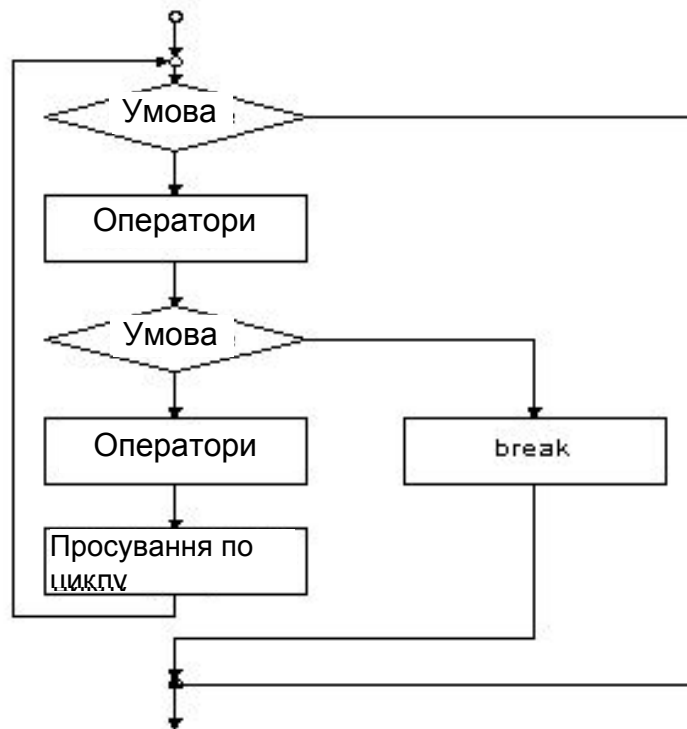


Рис. 2.5. Ілюстрація використання оператора break

Крім циклу for в програмуванні на MatLAB використовується цикл while. На відміну від циклу for у структурі циклу while не передбачені елементи для зміни змінної, по якій організовано цикл. Ці елементи повинен передбачити сам користувач в операторах циклу. Блок-схема циклу представлена на рис. 2.6.



Рис . 2.6 . Блок - схема циклу `while`

Формат оператора циклу з невідомою кількістю повторень :

`while умова`

`оператори`

`end`

Тіло циклу обов'язково закінчується службовим словом `end` .

Наприклад , програма визначення точності обчислень :

`a = 1 ;`

`while a + 1! = 1`

`a = a / 2 ;`

`end`

`a`

Останній рядок програми виведе на екран значення змінної `a` .

Завдання на лабораторну роботу

У ході виконання лабораторної роботи потрібно написати функцію , яка виконує задану послідовність дій . Виконання лабораторної роботи у вигляді файла- скрипта не допускається.

Загальна частина

- 1 . Згенерувати вхідний сигнал , як суму 3 -х синусоїд різної частоти (`a` , `b` , `c`)
- вводяться з клавіатури за допомогою функції `input ()` .

2 . Зберегти отриманий результат у файлі (функція `save ()`) .

3 . Вивести на екран графіки вихідного і перетвореного сигналу.

варіант 1

Зробити підписи осей графіків , оформити заголовок графіка .

варіант 2

При збереженні файлу задіяти стандартний діалог `uiputfile ()` . У діалозі повинні бути описані необхідні формати , змінений заголовок вікна .

варіант 3

Для вихідного сигналу повинен бути автоматично доданий шум , згенерований по рівномірному закону розподілу (функція `rand ()`) .

Амплітуда шуму $\sim 10\%$ від амплітуди сигналу.

варіант 4

Оформити графіки вихідного і перетвореного сигналу в одному вікні (функція `subplot ()`) . Колір графіків повинен бути різний .

Примітка . Для виконання лабораторної роботи може знадобитися використання додаткових функцій або властивостей об'єктів MatLAB . Для їх пошуку необхідно звертатися до вбудованої довідкової системи (`Help -> MATLAB Help`) . Система містить описи різних вбудованих функцій і операторів , згрупованих за категоріями . Слід пам'ятати , що більшість об'єктів в MatLAB мають додаткові властивості (звертатися до яких можна за допомогою функцій `set` і `get`) , які також описані в довідковій системі . Так, наприклад , для пошуку властивостей графіків можна у вкладці Index довідкової системи ввести `figure` або `axes` .

Лабораторна робота № 3
Спектральний аналіз сигналів методом
швидкого перетворення Фур'є

Теоретичні відомості

Дискретне перетворення Фур'є для обмеженого в часі сигналу

Дискретне перетворення Фур'є (ДПФ) застосовується при визначенні спектральної функції з дискретних значень амплітуди сигналу, тобто для побудови спектру сигналу, який дискретизовано.

Звичайна форма запису ДПФ має вигляд:

$$F_d(j\omega k) = \sum_{n=0}^{N-1} f(nT_a) e^{-j2\pi kn/N} \quad (1)$$

де ω - кругова частота дискретизації,
 k - номер вибірки,
 N - кількість вибірок,
 n - індекс дискретних проміжків часу T_a ,
 T_a - період дискретизації,
 $f(nT_a)$ - функція сигналу.

Існує деякий сигнал, що описується функцією $f(t)$. Його дискретизують з періодом дискретизації T_a . Для спостереження беруть певну кількість вибірок N . Час спостереження набуває значення NT_a . В результаті ДНФ отримуємо N комплексних спектральних складових. Номера спектральних складових задаються значенням k .

Отриманий спектр має дійсну та уявну частини. Спектральна функція періодична відносно аргументу ω з періодом $2\pi/T_a$ та періодична до

аргументу f з періодом $1/T_a$. Спектральна функція повинна обчислюватись лише для дискретних значень ω .

Дискретне перетворення Фур'є необмеженого в часі сигналу

В цьому випадку при обчисленні ДПФ труднощі виникають через те, що при $t \rightarrow \infty$ інтеграл Фур'є не сходиться. Для подолання цієї колізії необмежений в часі сигнал розглядають в межах вікна, тобто множать на функцію вікна. В межах вікна сигнал існує, а за межами ні. Так сигнал перетворюється в обмежений по часі. Множення на функцію вікна в часовій області відповідає згортці сигналу зі спектром вікна. Це призводить до розширення спектру, тобто до зменшення спектральної роздільної здатності.

Збіжність інтегралу досягається через таке припущення:

$$w(t) = 1 \quad \text{для} \quad -\frac{N-1}{2}T_a \leq nT_a \leq \frac{N-1}{2}T_a \quad (2)$$

Формула для ДПФ має вигляд:

$$F_d(j\omega) = \sum_{n=-(N-1)/2}^{(N-1)/2} f(nT_a)w(nT_a)e^{-j\omega nT_a} \quad (3)$$

Рівняння дає огинаючу амплітуд, що належать частотам ω_k .

Відновлення обмеженого в часі сигналу

Зворотне ДПФ проводиться за формулою:

$$g(t) = \frac{1}{N} \sum_{n=0}^{N-1} F_d(j\omega n) e^{j2\pi n t} \quad (4)$$

Лабораторна робота № 5

Цифровий нерекурсивний фільтр низьких частот

Теоретичні відомості

В цифровій обробці сигналів використовуються цифрові фільтри. За їх допомогою в сигналі покращують співвідношення сигнал/шум, виділяючи або придушуючи певні частоти.

Фільтр характеризується ваговими коефіцієнтами a_k . Дискретизований вхідний сигнал $x_{n-k} = x([n-k]T_a)$ (де n - номер елемента в послідовності значень вхідного сигналу, k - індекс вагових коефіцієнтів фільтра, змінюється від $-N$ до N , N - порядок фільтра, T_a - період дискретизації, x - функція вхідного сигналу) проходить через фільтр і отримується вихідний

сигнал $y_n = \sum_{k=-N}^N a_k x_{n-k}$. Отже вихідний сигнал y_n відомо лише тоді, коли

стане відомо вхідний сигнал x_{n+N} .

Властивість **нерекурсивний фільтр** означає, що вихідний сигнал залежить лише від значень вхідного сигналу.

Під час синтезу нерекурсивного фільтра виникає задача визначення вагових коефіцієнтів. Для отримання передаточної функції реалізованого фільтра з передаточної функції ідеального фільтра, що має вигляд

$$G_w(j\omega) = 1 \quad \text{при } |\omega| \leq \omega_g \text{ та рівна } 0 \text{ інакше,} \quad (1)$$

її можна апроксимувати методом найменших квадратів. Спочатку вона визначається як ряд Фур'є. У цьому випадку вагові коефіцієнти фільтра є коефіцієнтами ряду Фур'є при розкладі передаточної функції. Для зручності краще користуватись граничною круговою частотою:

$$\Omega_g = \frac{\omega_g}{f_a} = 2\pi \frac{f_g}{f_a}, \quad (2)$$

де f_a - частота виборок, f_g - гранична частота.

Вагові коефіцієнти фільтра обчислюються за формулою:

$$a_0 = \frac{\Omega_g}{\pi}$$
$$a_k = a_{-k} = \frac{1}{\pi k} \sin k\Omega_g . \quad (3)$$

Передаточна функція має вигляд:

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

Хід роботи

1. Розрахувати вагові коефіцієнти для ФНЧ. (Формула (3))
2. Побудувати графік ідеальної передаточної функції.(Формула (1))
3. Побудувати графік передаточної функції розрахованого ФНЧ.
(Формула (4))
4. Зробити висновки.

Приклад

1. Ідеальний ФНЧ необхідно апроксимувати фільтром при $N=3$, граничній частоті $f_g = 25\text{Гц}$ і частоті виборок $f_a = 100\text{Гц}$.

Обчислимо граничну кругову частоту (2).

$$\Omega_g = \frac{\omega_g}{f_a} = 2\pi \frac{f_g}{f_a} = 2\pi \frac{25\text{Гц}}{100\text{Гц}} = \frac{\pi}{2}$$

Обчислимо вагові коефіцієнти.

$$a_0 = \frac{\pi}{2\pi} = \frac{1}{2}$$
$$a_1 = a_{-1} = \frac{1}{\pi} \sin\left(\frac{\pi}{2}\right) = \frac{1}{\pi}$$
$$a_2 = a_{-2} = \frac{1}{2\pi} \sin\left(2 \frac{\pi}{2}\right) = 0$$

$$a_3 = a_{-3} = \frac{1}{3\pi} \sin\left(3 \frac{\pi}{2}\right) = -\frac{1}{3\pi}$$

Рівняння фільтра має вигляд:

$$y_n = -\frac{1}{3\pi} x_{n+3} + \frac{1}{\pi} x_{n+1} + \frac{1}{2} x_n + \frac{1}{\pi} x_{n-1} - \frac{1}{3\pi} x_{n-3}$$

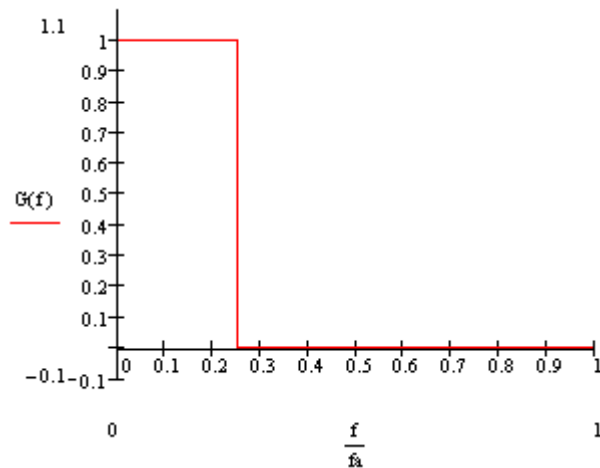
2. Побудуємо графік ідеальної передаточної функції.

$$G_w(j\omega) = 1 \quad \text{при } |\omega| \leq \omega_g \text{ та рівна } 0 \text{ інакше}$$

або

$$G_w(jf) = 1 \quad \text{при } |f| \leq 25 \text{ та рівна } 0 \text{ інакше}$$

Графік віднормуємо відносно частоти вибірки.

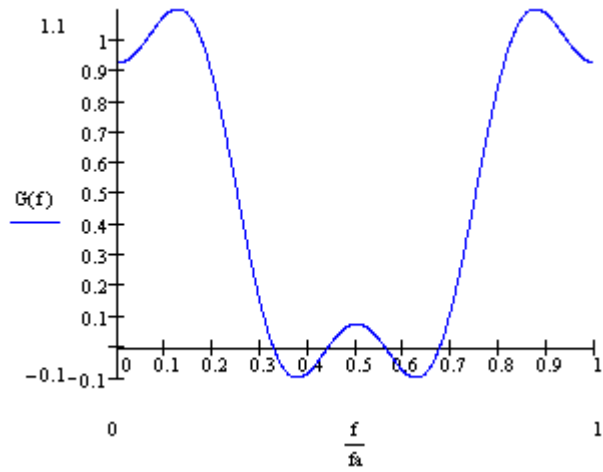


3. Побудуємо графік передаточної функції розрахованого ФНЧ.

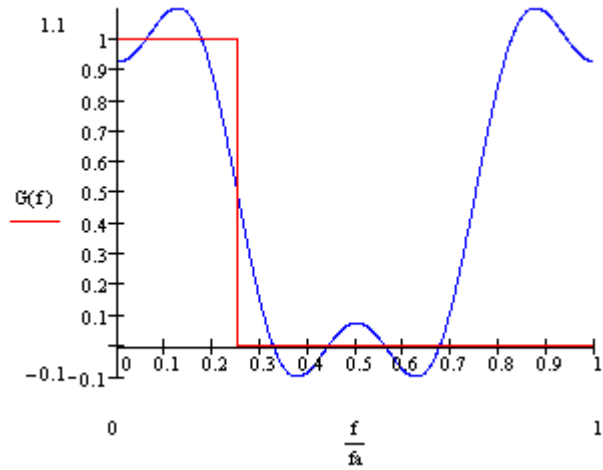
$$G_d(j\omega) = \frac{1}{2} + \frac{2}{\pi} \cos(\omega T_a) - \frac{2}{3\pi} \cos(3\omega T_a)$$

або

$$G_d(f) = \frac{1}{2} + \frac{2}{\pi} \cos\left(\frac{2\pi f}{f_a}\right) - \frac{2}{3\pi} \cos\left(3 \frac{2\pi f}{f_a}\right)$$



4. Для того, щоб зробити висновки, варто побудувати ідеальну і розраховану передатні характеристики на одному графіку у відносних одиницях.



Додаток А Формули для визначення коефіцієнтів фільтра

Тип фільтра	М парне N=(M+1) непарне	М непарне N=(M+1) парне
ФНЧ	$a_0 = 2 \frac{f_c}{F_s}$ $a_i = \frac{1}{\pi i} \sin(2\pi i \frac{f_c}{F_s})$	$a_i = \frac{1}{\pi(i-\frac{1}{2})} \sin(2\pi(i-\frac{1}{2}) \frac{f_c}{F_s})$
ФВЧ	$a_0 = 1 - 2 \frac{f_c}{F_s}$ $a_i = -\frac{1}{\pi i} \sin(2\pi i \frac{f_c}{F_s})$	$a_i = \frac{1}{\pi(i-\frac{1}{2})} [\sin(\pi(i-\frac{1}{2})) - \sin(2\pi(i-\frac{1}{2}) \frac{f_c}{F_s})]$
ПФ	$a_0 = 2 \frac{f_{c2} - f_{c1}}{F_s}$ $a_i = \frac{1}{\pi i} [\sin(2\pi i \frac{f_{c2}}{F_s}) - \sin(2\pi i \frac{f_{c1}}{F_s})]$	$a_i = \frac{1}{\pi(i-\frac{1}{2})} [\sin(2\pi(i-\frac{1}{2}) \frac{f_{c2}}{F_s}) - \sin(2\pi(i-\frac{1}{2}) \frac{f_{c1}}{F_s})]$
РФ	$a(0) = 1 - 2 \frac{f_{c2} - f_{c1}}{F_s}$ $a(i) = \frac{1}{\pi i} [\sin(2\pi i \frac{f_{c1}}{F_s}) - \sin(2\pi i \frac{f_{c2}}{F_s})]$	$a_i = \frac{1}{\pi(i-\frac{1}{2})} [\sin(\pi(i-\frac{1}{2})) - \sin(2\pi(i-\frac{1}{2}) \frac{f_{c2}}{F_s}) + \sin(2\pi(i-\frac{1}{2}) \frac{f_{c1}}{F_s})]$

Перелік рекомендованої літератури

1. Айфичер Э. С., Джервис Б. У. Цифровая обработка сигналов.: Пер. с англ. – М.: “Издательский дом “Вильямс”, 2004. – 992 с.
2. Бабак В. П., Хандецкий В. С., Шрюфер Е. Обробка сигналів. – К.: Либідь, 1999.- 496 с.
3. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов .: Пер. с англ. – М.: «Мир», 1978.
4. Бортник Г.Г., Кичак В.М. Цифрова обробка сигналів. Навчальний посібник – Вінниця: УНІВЕРСУМ, 2006. – 167 с.
5. Обробка сигналів: навчальний посібник / В. П. Майданюк, А. М. Петух. – Вінниця : ВНТУ, 2012. – 144 с.
6. Применение цифровой обработки сигналов // Пер. с англ. / Под ред. Э. Оппенгейма. – М.: Мир, 1980.- 382 с.
7. Цифровая обработка сигналов: Учеб. пос-е для высш. учеб. завед-й – 2-е изд., перераб. и доп. / Гольденберг Л.М., Матюшкин Б.Д., Поляк М.Н. – М.: Радио и связь, 1990. – 256 с..3.
8. Шрюфер Э. Обработка сигналов: цифровая обработка дискретизированных сигналов: Учебник / Под ред. проф. В.П.Бабака. – К.: Либідь, 1995. – 320 с.4.
9. Потемкин В.Г. Система Matlab для студентов. Справочное пособие.- М.: Диалог-МИФИ, 1998 – 314 с.
10. Дьяконов В.П. Matlab 6: учебный курс. – Сп-б: Питер, 2001.- 592.
11. Разевиг В.Д. MICROCAP-5 для студентов. - М.:Радио и связь, 1999.