

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

О.І. Толочко

ТЕОРІЯ АВТОМАТИЧНОГО КЕРУВАННЯ

**Математичні методи та програмування в *MATLAB* для
теорії керування**

Комп'ютерний практикум

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
освітньою програмою «Електромеханічні системи автоматизації, електропривод
та електромобільність»
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»,*

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2025

Рецензент

Островерхов М.Я., доктор технічних наук,
професор, завідувач кафедри теоретичної
електротехніки

Відповідальний редактор

Ковбаса С.М., доктор технічних наук, доцент, завідувач
кафедри автоматизації електромеханічних систем та
електроприводу

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 7 від 8.05.2025 р.)
за поданням факультету електроенерготехніки та автоматики (протокол №11 від 28.04.2025 р.)*
Електронне мережне навчальне видання **Реєстр 24/25-492**

Толочко О.І.

Теорія автоматичного керування. Математичні методи та програмування в *MATLAB* для теорії керування. [Електронний ресурс]: комп'ютерний практикум: навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Електромеханічні системи автоматизації, електропривод та електромобільність» спец. 141 «Електроенергетика, електротехніка та електромеханіка» / О. І. Толочко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл). – Київ: КПІ ім. Ігоря Сікорського, 2025. – 77 с.

Практикум присвячено питанням застосування студентами спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» математичних методів для розв'язання інженерних задач в галузі теорії автоматичного керування, електротехніки та електромеханіки в середовищі системи програмування *MATLAB*. Практикум складається з методичних вказівок до 8 практичних занять, присвячених придбанню навичок з основ програмування в *MATLAB*, основ математичного моделювання в додатку Simulink пакету *MATLAB*, основ роботи з інструментами додатків *Extended Symbolic Toolbox* та *Control Toolbox*, а також власно математичних методів розв'язання прикладних інженерних задач в галузі теорії автоматичного керування.

Практикум призначений для використання бакалаврами спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» при вивченні дисципліни «Теорія автоматичного керування» та може бути використаним студентами навчальної освітньої програми «Електромеханічні системи автоматизації, електропривод та електромобільність» при вивченні дисциплін «Моделювання електромеханічних систем в *МАТЛАБ*», «Теоретичні основи електротехніки», «Інтелектуальне керування та оптимізація в електромеханічних системах», а також при виконанні курсових, бакалаврських та магістерських робіт.

© О.І. Толочко, 2025
© КПІ ім. Ігоря Сікорського, 2025

ВСТУП

В теорії автоматичного керування (ТАК) та в її застосуваннях в електротехніці та електромеханіці існує безліч задач, що не можуть бути розв'язані аналітичними методами або такий розв'язок є занадто складним [1-7]. Вони потребують застосування чисельних математичних методів [8, 9], пов'язаних з наявністю ітераційних циклів, або громіздких алгоритмів, які можуть бути успішно реалізовані тільки на ЕОМ при використанні сучасного програмного забезпечення.

До таких задач належать задачі математичного аналізу, задачі лінійної алгебри, операції зі степеневими поліномами, пошук визначених інтегралів, апроксимація нелінійних функцій, розв'язання диференціальних рівнянь, алгебраїчних та трансцендентних рівнянь, які широко використовуються в теорії автоматичного керування.

Усі ці задачі можна вирішувати як за допомогою програм, написаних будь-якою алгоритмічною мовою, що потребує від дослідника достатньо високої кваліфікації в галузі програмування та обчислювальної математики, так і за допомогою спеціалізованого програмного забезпечення, що дозволяє користувачу значно легше і скоріше отримувати результати у зручній формі завдяки наявності великої кількості різноманітних інструментів.

Серед спеціалізованих систем програмування однією з найбільш розповсюджених натеper є система *MATLAB* фірми *Mathworks* [10-15], що отримує у своєму складі об'єктно-орієнтовану алгоритмічну мову, розвинутий графічний інтерфейс, засоби розв'язання задач лінійної алгебри, математичного аналізу, оптимізації, обчислювальної математики, цифрової обробки сигналів, структурного математичного та віртуального фізичного моделювання, аналізу і синтезу систем автоматичного керування та багато інших інструментів (*Toolboxes*) [3-7, 16-22].

Особливістю пакета *MATLAB* є можливість застосування математичних методів не тільки шляхом написання програми на алгоритмічній мові, але й шляхом застосування блоків бібліотек додатку для структурного математичного моделювання *Simulink* [8, 23-25].

У навчальному посібнику наведені завдання до 8 практичних занять, виконання яких дозволить студентами надбати практичні навички з основ програмування в *MATLAB*, основ математичного моделювання в *Simulink*, застосування базових інструментів пакету *MATLAB* для розв'язання задач теорії керування, електротехніки та електромеханіки, що вирішуються чисельними методами, застосування інструментів додатку *Control Toolbox* для виконання аналізу лінійних стаціонарних систем автоматичного керування.

До кожного практичного заняття в комп'ютерному практикумі додається довідкова інформація та методичні рекомендації до його виконання. Більш детальну інформацію можна знайти у відповідному навчальному посібнику з лекційними матеріалами, в якому наведено багато прикладів та контрольні питання і завдання.

Навички, надбані студентами при виконанні комп'ютерного практикуму «Автоматизація розрахунків та досліджень в середовищі пакету *MATLAB*» дисципліни «Теорія автоматичного керування», стануть їм в нагоді при вивченні дисциплін «Моделювання систем автоматичного керування», «Моделювання електромеханічних систем», «Адаптивне та робастне керування», «Інтелектуальне керування та оптимізація в електромеханічних системах», а також при курсовому і дипломному проектуванні.

Практичне заняття № 1

ОСНОВИ РОБОТИ В MATLAB

1.1. Довідкова інформація

Термінологія:

- *Desktop* – робочий стіл (комбіноване вікно, що відкривається після запуску MATLAB);
- *Command Window* – командне вікно;
- *Command History* – історія команд;
- *Workspace* – робочий простір (оперативна пам'ять);
- *Current Folder* – поточна папка.

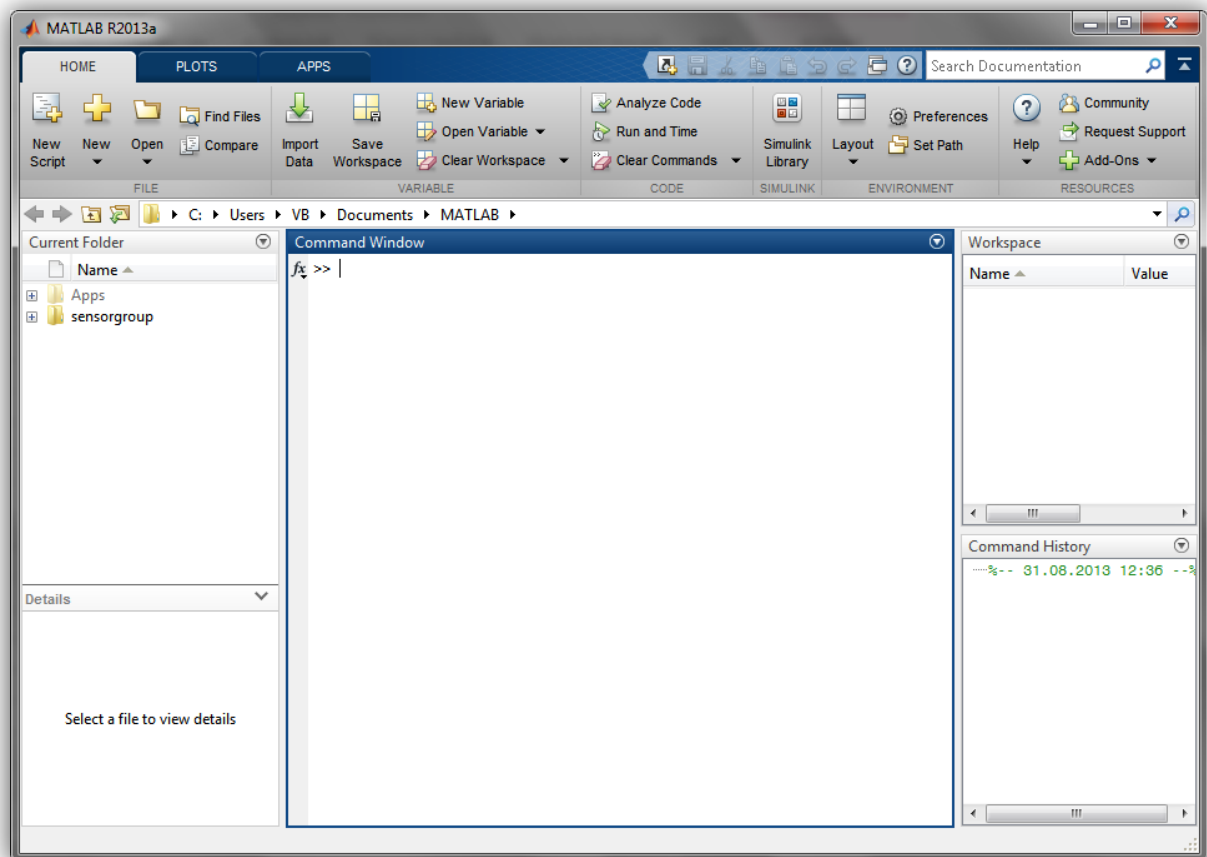


Рис. 1.1. Робочий стіл системи програмування MATLAB 13a

>> (*prompt*) – запрошення на введення команди;

% – ознака початку коментаря;

Правила введення та виконання команд та/або операторів у командному рядку:

- команди або оператори, розташовані в одному рядку, відокремлюються один від одного символами ",", " або ";"
- якщо команда або оператор не поміщається в одному рядку, її можна продовжити в наступному рядку, використовуючи у якості оператору переносу три або більше точок поряд без пробілів ("...");
- для виконання набраної в командному рядку команди натисніть клавішу *Enter*;
- деяка кількість введених команд запам'ятовується в буфері, з якого їх можна викликати по черзі у зворотному порядку клавішею "↑".

Деякі команди:

- *bench* – еталонний тест *MATLAB*;
- *ver* – інформація про версії *MATLAB* та його інструментів;
- *demo* – виконання демонстрацій;
- *clc* (*clear screen*) – очистка командного вікна;
- *clear* – очистка змінних та функцій з робочого простору;
- *echo on/off* – керування режимом виведення виконуваних операторів на екран;
- *format* – визначення або зміна формату виведення інформації;
- *debug* – список команд відладчика;
- *diary FileName, diary off* – створення файлу для збереження сеансу роботи в *MATLAB*.
- *type FileName* – виведення текстового файлу у вікно *Command Window*;
- *help* – перелік доступних папок;
- *help folder* – перелік функцій, що входять до складу даної папки, з коротким (в один рядок) описом призначення кожної функції (складається з коментарів файлу *Contents.m*, розташованого в поточній папці);

- `help fun` – докладний опис призначення функції, який складається з коментарів, розташованих одразу після заголовку цієї функції у відповідному файлі;
- `doc fun` – документація *MATLAB* щодо обраної функції;
- `type fun` – відображення тексту функції у командному вікні;
- `which fun` – визначення локації функції у файловій системі *MATLAB*;
- `path` – визначення шляху до файлів;
- `addpath` – приєднання папки за визначеним шляхом.

Основні папки базових інструментів MATLAB (ядро пакету):

`elfun` – елементарні математичні функції;

`elmat` – елементарні матриці та вектори та маніпуляції над ними;

`datafun` – аналіз даних та перетворення Фур'є;

`matfun` – операції лінійної алгебри;

`ops` – математичні операції та спеціальні символи;

`general` – команди загального призначення;

`lang` – конструкції алгоритмічної мови;

`helptools` – функції допомоги;

`graph2d` – двовимірна графіка;

`graph3d` – тривимірна графіка;

`graphics` – обробка графіків (допоміжні графічні функції);

`specgraph` – спеціальна графіка;

`polyfun` – операції над степеневими поліномами;

`funfun` – функції над функціями;

`optimfun` – функції оптимізації та пошук коренів;

`strfun` – функції над рядками символів.

Склад імен змінних (ідентифікаторів): латинські букви, цифри та знак підкреслення, перший символ – буква.

Арифметичні константи:

- цілі (-16, +5, 283);
- дійсні з фіксованою точкою (25.6, -138.54);
- дійсні з плаваючою точкою $M \times 10^p$ – Мер (0.25e-8, -35e12);
- комплексні (2.5-5i, -0.2+1e-3j).

Таблиця 1.1. *Зарезервовані константи*

Ім'я константи	Значення константи
ans	answer – результат останньої операції
i, j	sqrt(-1) – уявна одиниця
pi	число π , обчислюється як $4 * \text{atan}(1)$ або $\text{imag}(\log(-1))$
eps	машинна точність – 2.2204e-16
realmax	модуль максимального дійсного числа – 1.7977e+308
realmin	модуль мінімального дійсного число – 2.2251e-308
intmax	максимальне ціле число – 2147483647
intmin	мінімальне ціле число – -2147483648
namelengthmax	максимальна довжина імені – 63
Inf	Infinity – нескінченність (number/0)
NaN	Not a Number – невизначеність (0/0, inf/inf, inf-inf, 0 ^{inf} , ...)
end	найбільше значення індексу масиву

Таблиця 1.2. *Формати виведення чисел*

Формат	Подання
format short e	Експоненціальна форма (з плаваючою точкою) з мантиєю із 5 цифр і показником степені із 3 цифр
format short	Природна форма (з фіксованою точкою) з 4 цифрами після крапки або в форматі <i>short e</i> , якщо формат <i>short</i> не підходить для відображення значення числа (за замовчанням)
format long e	Експоненціальна форма з мантиєю із 16 цифр і показником степені із 3 цифр
format long	Природна форма з 16 цифрами після крапки або в форматі <i>long e</i>
format loose	Інформація виводиться просторо – з пропуском одного рядка між рядками (за замовчанням)
format compact	Інформація виводиться щільно (без пропуску рядків)

Операція присвоєння – вираз expr розраховується, а його значення запам'ятовується у змінній з ім'ям VarName :

$\text{VarName} = \text{expr}$

Неявне присвоєння – вираз expr розраховується, а його значення запам'ятовується у змінній з ім'ям ans :

expr

Арифметичні операції зі скалярними числовими даними:

- $+$ – додавання або уточнення знаку;
- $-$ – віднімання або зміна знаку;
- $*$ – множення;
- $/$ – ділення;
- $\wedge$ – підвищення у степінь.

В результаті отримуємо інформацію, з якої випливає призначення таких функцій:

- **тригонометричні з аргументом у радіанах:** $\sin(x) - \sin x$, $\cos(x) - \cos x$, $\tan(x) - \operatorname{tg} x$, $\cot(x) - \operatorname{ctg} x$, $\sec(x) - \sec x$, $\csc(x) - \operatorname{cosec} x$;
- **тригонометричні з аргументом у градусах (degree):** $\operatorname{sind}(x)$, $\operatorname{cosd}(x)$, $\operatorname{tand}(x)$, $\operatorname{cotd}(x)$, $\operatorname{secd}(x)$, $\operatorname{cscd}(x)$;
- **зворотно тригонометричні зі значенням у радіанах:** $\operatorname{asin}(x) - \arcsin x$, $\operatorname{acos}(x) - \arccos x$, $\operatorname{atan}(x) - (-\pi/2) \leq \operatorname{arctg} x \leq \pi/2$, $\operatorname{atan2}(y, x) - (-\pi) \leq \operatorname{arctg} y/x \leq \pi$, $\operatorname{acot}(x) - \operatorname{arcctg} x$, $\operatorname{asec}(x) - \operatorname{arcsec} x$, $\operatorname{acsc}(x) - \operatorname{arccosec} x$;
- **зворотно тригонометричні зі значенням у градусах:** $\operatorname{asind}(x)$, $\operatorname{acosd}(x)$, $\operatorname{atand}(x)$, $\operatorname{atand2}(y, x)$, $\operatorname{acotd}(x)$, $\operatorname{asecd}(x)$, $\operatorname{acscd}(x)$;
- **гіперболічні:** $\sinh(x) - \operatorname{sh} x = (e^x - e^{-x})/2$, $\cosh(x) - \operatorname{ch} x = (e^x + e^{-x})/2$, $\tanh(x) - \operatorname{th} x = (e^x - e^{-x})/(e^x + e^{-x})$, $\coth(x) - \operatorname{cth} x = (e^x + e^{-x})/(e^x - e^{-x})$, $\operatorname{sech}(x) - \operatorname{sech} x = 2/(e^x + e^{-x})$, $\operatorname{csch}(x) - \operatorname{cosech} x = 2/(e^x - e^{-x})$;

- **експоненціальні, логарифмічні та степеневі:** $\exp(x) = e^x$, $\log(x) = \ln x$, $\log_{10}(x) = \lg x$, $\log_2(x) = \log_2 x$, $\text{pow2}(x) = 2^x$, $\text{sqrt}(x) = \sqrt{x}$, $\text{nthroot}(x,n) = \sqrt[n]{x}$;
- **функції комплексних аргументів** $z = x + jy = Ae^{j\varphi}$: $\text{abs}(z) = |z| = \sqrt{x^2 + y^2} = A$, $\text{angle}(z) = \varphi = \arctg(y/x)$, $\text{real}(z) = x$, $\text{imag}(z) = y$, $\text{complex}(x,y) = z = x + jy$, $\text{conj}(z) = x - jy = Ae^{-j\varphi}$.
- **округлювання та залишки:** $\text{fix}(x)$ – округлювання у напрямку до 0 (дробова частина відкидається); $\text{floor}(x)$ – округлювання у напрямку до $-\infty$ (до найближчого меншого цілого); $\text{ceil}(x)$ – округлювання у напрямку до $+\infty$ (до найближчого більшого цілого); $\text{round}(x)$ – округлювання до найближчого цілого; $\text{rem}(x,y)$ – залишок від цілочислового ділення x на y ; $\text{sign}(x)$ – знакова функція (дорівнює +1 при $x > 0$, 0 при $x = 0$ та -1 при $x < 0$).

До цього переліку можна додати ще декілька функцій цілочислових аргументів із папки `specfun`: $\text{factor}(n) = n!$; $\text{gcd}(m,n)$ – найбільший загальний дільник (*Greatest Common Divisor*); $\text{lcm}(m,n)$ – найменший загальний множник (*Least Common Multiple*).

Формати функцій:

`FunName (arg1, arg2, ..., argN) ,`

`[out1, out2, ..., outN] = FunName (arg1, arg2, ..., argN),`

де `FunName` – ім'я функції; `arg` – аргументи функції, які передаються до функції за значеннями; `out` – імена вихідних величин.

Формат команди:

`FunName arg1, arg2, ..., argN,`

де `arg` – аргументи функції, які обробляються як словесна інформація або як рядки символів.

Наприклад,

```
>> disp (aaa)
```

```
Undefined function or variable 'aaa'.
```

```
>> disp ('aaa')
aaa
>> disp aaa
aaa
```

Генерація векторів та матриць:

```
» x=[2 3 -8];      % Вектор-рядок
» y=[5; 6.5; -2.23; 0]; % Вектор-стовпець
» A=[1,2,3; 4 5 6]; % Матриця 2*3
» B=[1 2; 3 4; 5 6]; % Матриця 3*2
```

Вектори-рядки з рівномірним розподіленням елементів:

```
VarName = InitValue : Step : EndValueh
VarName = InitValue : EndValue → Step=1
VarName = linspace (InitValue, EndValue, Number)
VarName = linspace (InitValue, EndValue) → Number=100
```

Вектори-рядки, рівномірно розподілені впродовж логарифмічної осі:

```
VarName = logspace (pInit, pEnd, Number) →
→ InitValue =10pInit, EndValue =10pEnd
VarName = logspace (pInit, pEnd) → Number=50
```

Деякі спеціальні матриці:

zeros (m,n) – матриця розміром $m \times n$, у якої всі елементи дорівнюють 0;
ones (m,n) – матриця розміром $m \times n$, у якої всі елементи дорівнюють 1;
rand (m,n) – матриця розміром $m \times n$, складена із випадкових чисел з рівномірним розподілом;
randn (m,n) – матриця розміром $m \times n$, складена із випадкових чисел з нормальним розподілом;
eye(n) – формує квадратну одиничну діагональну матрицю розміром $n \times n$;
diag(A) – формує з елементів головної діагоналі матриці **A** вектор-стовпець;
diag(X) – формує діагональну матрицю, застосовуючи у якості головної діагоналі вектор **X**.

При відсутності параметру **n** (кількість стовбців) за замовчанням призначається $m = n$, тобто матриця стає квадратною.

Звернення до елементів масивів:

$C(i, j)$ – елемент i -го рядка та j -го стовпця матриці C ;

$C(i, \text{end})$ – останній елемент i -го рядка матриці C ;

$C(\text{end}, j)$ – останній елемент j -го стовпця матриці C ;

$X(i)$ – i -ий елемент вектору X ;

$X(\text{end})$ – останній елемент вектору X ;

$C(i, :)$ – всі елементи i -го рядка матриці C ;

$C(:, j)$ – всі елементи j -го стовпця матриці C .

Пуста (порожня) матриця []:

$A(2,:) = []$ викидає з матриці A другий рядок.

1.2. Завдання

1. В текстовому редакторі розробити програму, що передбачає виконання таких дій:
 - а) виведення значень зарезервованих констант;
 - б) виконання довільних операторів явного і неявного присвоєння;
 - в) створення довільної матриці A розміром 3×5 , вектору-рядка B із 5 елементів та вектору-стовпця C із 3 елементів; формування **нової** матриці D приєднанням (конкатенація) вектору B до матриці знизу; формування ще однієї матриці E вставкою між 2-м та 3-м стовпцями матриці A вектору- стовпця C ; створення матриці G , усуненням із матриці E 1-го рядка та останнього стовпчика; заміна елементу, розташованого у 3-му стовпчику 2-го рядка матриці G нулем;
 - г) створення спеціальних векторів та матриць, описаних в довідковій частині методичних вказівок.
 - д) розрахувати значення виразів, заданих у табл. 1.2.
2. В **режимі командного інтерпретатора** апробувати дію команд bench, demo, help, help general, help polyfun, help polyval, doc polyval, which polyval, lookfor minimum, version, ver, clc, clear, beep.

Таблиця 1.2

№	Вираз	Значення змінних
1	$y = \sin \frac{a-x}{c} + 10^4 \sqrt[3]{\frac{a-kx^2}{2b}} + \frac{\cos kx^2}{\operatorname{tg} 3} - \frac{bc}{ax}$	$a=-1.3; b=0.91;$ $c=0.75; x=2.32; k=8.$
2	$y = -\frac{(x-d)(x^2+b^2)}{\sqrt[3]{x^2+b^2-cd}} + 10^{-3} \operatorname{tg} kn - \frac{\cos kx}{\sin 5}.$	$d=1.25; b=0.75; n=4;$ $c=2.2; x=0.32; k=2.$
3	$y = \operatorname{tg} ik + 10^3 e^{-5} + \sqrt[3]{\frac{10^2 xk }{(a+b)^2}} - \frac{ax^3 - b}{(a+b)^2}.$	$i=5; b=2.35;$ $a=25.2; x=0.1; k=-2.$
4	$y = \frac{\sqrt{ c-d +(a+c)^2}}{\sin 2i} + 10^{-3} e^{ix} - \frac{ c-d +a^2}{\sqrt[3]{(a+c)^2}}.$	$a=-1.25; d=2.5; i=5;$ $c=0.05; x=1.35.$
5	$y = \frac{\ln kx }{2\sin(\pi/3)} - \sqrt{ x-a^2 } - \frac{10^4 a-b}{\cos kx} + \sqrt[3]{x-a^2} + c^3 x.$	$a=0.93; b=5.61;$ $c=0.31; x=-2.5;$ $k=2.$
6	$y = 10^4 \frac{ax}{b^2} - \left \frac{a-b}{kx} \right + \frac{\ln 3}{\sqrt[3]{ax^2+b^2}} - e^{-kx}$	$b=0.35; a=3.5;$ $x=1.523; k=-2.$
7	$y = -\frac{ b-a }{kx} + 10^4 \sqrt[5]{ \cos kx } + \sqrt{\frac{abc}{2.4}} - \frac{0.7abc}{\sin(32^\circ)}.$	$a=1.7; b=-1.25;$ $c=-0.3; x=2.5; k=3.$
8	$y = \frac{ a^2-b^2 }{\sin kx} + 10^4 \sqrt[5]{ \sin kx - bc } - \frac{k^2 + \operatorname{tg} 3k}{e^{kx}}.$	$a=1.3; b=2.42;$ $c=0.83; x=1.5; k=2.$
9	$y = \frac{\sqrt[3]{\ln x + a^2}}{0.47x^2} - \left 0.47x^2 - \frac{10^4}{7} \cos^2 k \right - \frac{c}{e^x}$	$c=1.52; a=-2.4;$ $x=0.29; k=3.$
10	$y = \frac{1.5(a-b)^2}{ a-b c} + \frac{e^{-x/2}}{5} + 10^3 \sqrt{ a-b } - \frac{(a+x^2)\cos(72^\circ)}{ix^2 + a^2 bc}$	$a=-2.5; b=1.35; i=3;$ $c=-0.72; x=2.75.$
11	$y = 10^4 \sin^2 i - \frac{0.32x^3 + 4x + b}{\cos ia} \sqrt[6]{0.32x^3 - b} + b $	$a=3.5; b=-0.7; i=2;$ $x=0.8.$
12	$y = -\frac{\cos i}{\sin kx} + \frac{ax^2 + d }{(a+b)^2} - 10^4 \sqrt[6]{\frac{kx}{(a+b)^2}}.$	$d=-0.01; b=1.25;$ $a=4.72; i=2;$ $x=2.25; k=3.$
13	$y = \cos k(x-a) + 10^{-4} \frac{(x+a)^3 + x^4 d}{k(x-a)^3} + \frac{\sqrt[5]{ x+a }}{2.4b}.$	$d=0.95; b=0.05;$ $a=-3.25; x=8.2; k=4.$
14	$y = \sqrt[5]{ ax^2 - b^3 } + \ln kx - \frac{e^{kx} + c^2}{\sin kx} - 10^{-3} \sqrt{2157}.$	$c=1.72; b=-0.31;$ $a=2.01; x=0.48; k=3.$

Сеанс зберегти у файлі. Результати подати в скороченому вигляді (залишайте ту інформацію, яка є для вас зрозумілою)

1.3. Порядок виконання завдання

1. Відкрийте Провідник за шляхом *C:\Users\UserName\Documents\MATLAB* , де зберігаються документи *MATLAB*. Створіть за цим шляхом нову папку користувача, призначену для збереження власних файлів, наприклад, *ivanov24*. Важливо, що ім'я користувача *UserName* не повинно отримувати символів кирилиці.
2. Приєднайте створену папку до списку дозволених в *MATLAB* папок командою меню *Set Path*→*Add Folder...*→*Save*.
3. Завантажте текстовий редактор *edit.com* командою меню *New*→*Script*.
4. Збережіть поки ще пустий файл командою *Save as...*, замінивши ім'я *untitled* на інше ім'я, яке може складатися, як і імена змінних, з латинських букв, цифр та знаків підкреслення (перший символ – буква), наприклад, *LR1_Ivanov_ep31*.
5. Створіть *script*-файл, який виконуватиме пункт 1 завдання.
6. Виконайте в командному вікні пункт 2 завдання. Сеанс збережіть у файлі.

1.4. Методичні рекомендації

1. Після опробування команди *format* поверніться у стан *format short*, *format compact*.
2. Доповніть програму коментарями, командою *echo on* і збережіть фінальний варіант командою *Save*.
3. **При роботі з масивами списки індексів та параметрів функцій оточуйте круглими дужками та відокремлюйте один від одного комою «,».**
4. При виконанні завдання 5 не забувайте про пріоритети виконання арифметичних операцій та за необхідністю змінюйте цей порядок за допомогою круглих дужок: слідкуйте за тим, щоб кількість відкриваючих та закриваючих дужок була однаковою. **Для перевірки правильності**

розрахунку значення виразу з табл. 2.1 розбийте вираз на декілька частин та виконайте розрахунок поетапно.

5. Після написання імені функції одразу записуйте в дужках її аргументи і тільки потім виконуйте зі вже розрахованою функцією будь-які операції.
6. Виконайте програму командою Run (F5) та скопіюйте саму програму та результати її виконання у звіт з лабораторної роботи.
7. При будь яких непорозуміннях користуйтеся командами help, doc та demo.
8. Результати виконання завдання 2 подайте в скороченому вигляді (залишайте тільки ту інформацію, яка є для вас зрозумілою).

Практичне заняття № 2

ПОБУДОВА ТА ОФОРМЛЕННЯ БАЗОВИХ ДВОВИМІРНИХ ГРАФІКІВ ФУНКЦІЙ

2.1. Довідкова інформація

Термінологія:

figure – графічне вікно;

plot – будувати графік;

grid – координатна сітка;

hold – затримка, фіксація зображення в графічному вікні;

axis – вісь;

axes – система координат;

gca – *graphic current axes* (поточна система координат);

gcf – *graphic current figure* (поточне графічне вікно);

title – заголовок графіка.

Початкову інформацію про основні функції двовимірної графіки можна отримати виконанням команди **help graph2d**

```
>> help graph2d
```

Two dimensional graphs.

Elementary X-Y graphs.

plot - Linear plot.

loglog - Log-log scale plot.

semilogx - Semi-log scale plot.

semilogy - Semi-log scale plot.

polar - Polar coordinate plot.

plotyy - Graphs with y tick labels on the left and right.

Axis control.

axis - Control axis scaling and appearance.

grid - Grid lines.

hold - Hold current graph.

subplot - Create axes in tiled positions.

Graph annotation.

title - Graph title.

xlabel - X-axis label.

ylabel - Y-axis label.
text - Text annotation.
gtext - Place text with mouse.

Домовимося, що в робочому просторі (*Workspace*) пакета *MATLAB* існують значення таких перемінних: x , y , – вектори-рядки із n елементів; u – вектор-стовбець із m елементів; A , B , C – матриці розміром $m \times n$; z – комплексне число, Z – вектор комплексних чисел. Тоді

`plot (y)` – будує графік функції $y(i) = f(i)$ по точках з абсцисами $i = [1, 2, \dots, n]$ та ординатами $[y(1), y(2), \dots, y(n)]$;

`plot (x,y)` – будує графік функції $y(i) = f(x(i))$;

`plot (Z)` – те ж саме, що і `plot(real(Z),imag(Z))`;

`plot (A)` – будує в одній системі координат n графіків, кожний з яких уявляє собою залежність елементів одного стовпчика матриці A від порядкового номеру рядка;

`plot (u,A)` – те ж саме, але у функції елементів вектору-стовбця u .

`plot (x,A)` – будує в одній системі координат m графіків, кожний з яких уявляє собою залежність елементів одного рядка матриці A від елементів вектору-рядка x .

Одним графічним оператором можна вивести в одній системі координат декілька графіків у такий спосіб:

`plot (x1,y1, x2,y2, ..., xN,yN)`

Кожну пару векторів, що застосовуються для виводу одного з графіків, можна доповнити *параметром* *Style*, який є рядком символів (може мати у своєму складі *від одного до чотирьох символів*), які визначатимуть колір та способи зображення маркера вузлових точок таблиці та лінії, що поєднує сусідні точки:

`plot (x1,y1,Style1, x2,y2,Style2,..., xN,yN,StyleN)`

У пакеті *MATLAB* кольори можна задавати вектором-рядком із 3-х чисел в діапазоні від 0 до 1, які визначають інтенсивність змішуваних основних

кольорів: [red green blue]. 1 означає максимальну інтенсивність, а 0 – нульову.

Таблиця 2.1. Значення символів у параметрі Style графічних функцій

Колір лінії	Тип маркера	Тип лінії
b – <i>blue</i> (синій)	. – крапка	- – суцільна (<i>solid</i>)
g – <i>green</i> (зелений)	o – коло	: – дрібний (точковий) пунктир (<i>dotted</i>)
r – <i>red</i> (червоний)	x – хрестик	-- – штрихова (<i>dashed</i>)
y – <i>yellow</i> (жовтий)	+ – плюс	-. – штрих-пунктирна (<i>dashdotted</i>)
m – <i>magenta</i> (пурпурний)	* – зірка	
c – <i>cyan</i> (блакитний)	s – квадрат (<i>square</i>)	
w – <i>white</i> (білий)	d – ромб (<i>diamond</i>)	
k – <i>black</i> (чорний)	v – трикутник вершиною вниз	
	^ – трикутник вершиною угору	
	< – трикутник вершиною вліво	
	> – трикутник вершиною вправо	
	p – п'ятикутна зірка (<i>pentagram</i>)	
	h – шестикутна зірка (<i>hexagram</i>)	

Кольорам з табл. 2.1 відповідають такі значення **rgb**-векторів:

[0 0 1] – blue	[1 0 1] – magenta
[0 1 0] – green	[1 1 0] – yellow
[1 0 0] – red	[0 0 0] – black
[0 1 1] – cyan	[1 1 1] – white

Значення **rgb**-векторів, що призначаються в *MATLAB* за замовчанням при зображенні декількох графіків в одній системі координат одним оператором plot:

$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ – blue
 $\begin{bmatrix} 0 & 0.5 & 0 \end{bmatrix}$ – dark green
 $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ – red
 $\begin{bmatrix} 0 & 0.75 & 0.75 \end{bmatrix}$ – dark cyan
 $\begin{bmatrix} 0.75 & 0 & 0.75 \end{bmatrix}$ – violet
 $\begin{bmatrix} 0.75 & 0.75 & 0 \end{bmatrix}$ – light brown
 $\begin{bmatrix} 0.25 & 0.25 & 0.25 \end{bmatrix}$ – dark grey

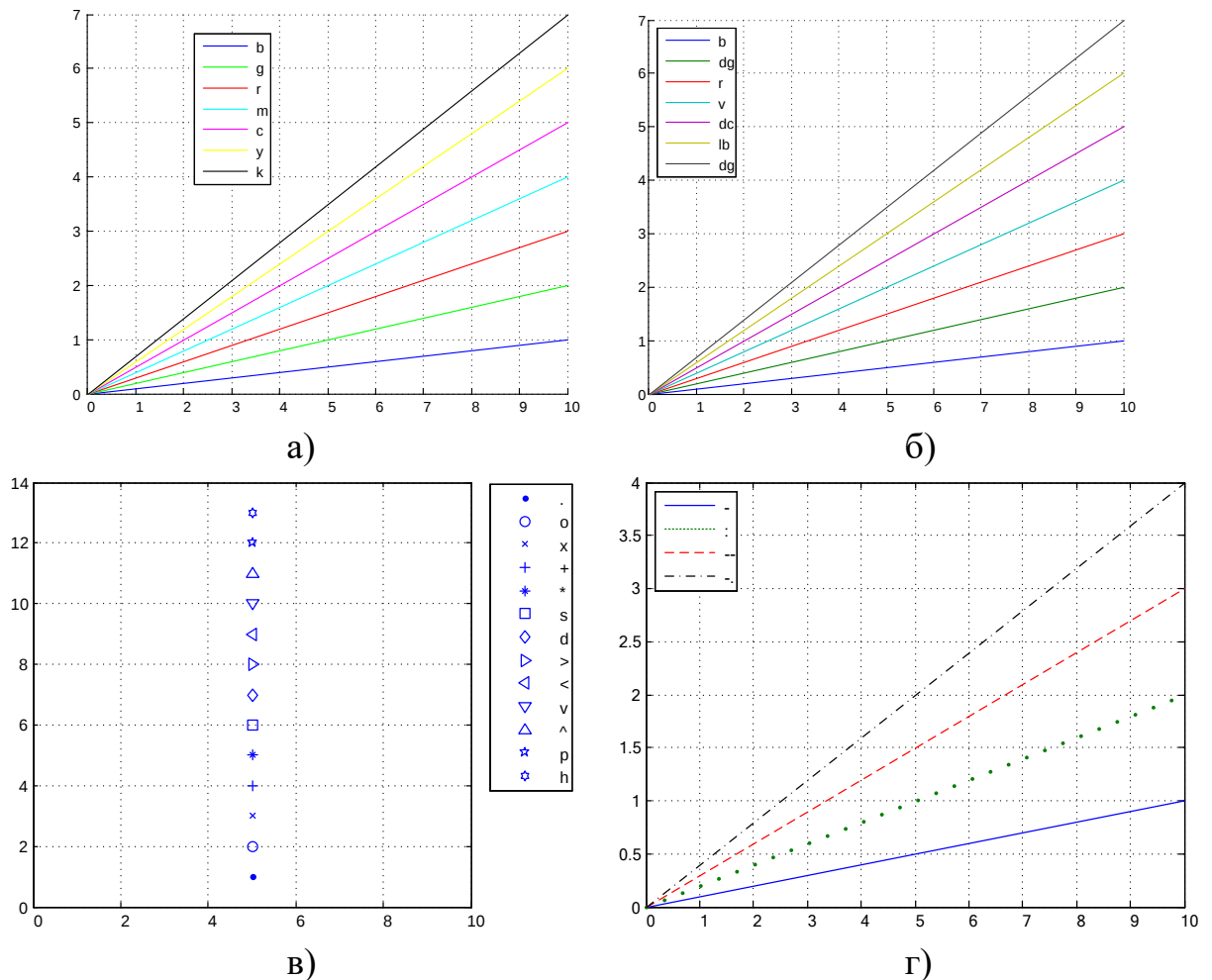


Рис. 2.1. Кольори та стилі зображення графіків

Визначення та зміна окремих властивостей (Properties) графіків:

`plot (x,y, PropName1,PropValue1, PropName2,PropValue2, ...)`

Отримання інформації про імена властивостей (PropertyName) та їх значення за замовчанням (PropertyValue) :

`get (gca)`

де gca означає *Graphic Current Axes* (поточна система координат) .

Деякі властивості, що виводяться при виконанні функції get (gca):

Box = on
Color = [1 1 1]
ColorOrder = [(7 by 3) double array]
FontName = Helvetica
FontSize = [10]
FontWeight = normal
GridLineStyle = :
LineStyleOrder = -
LineWidth = [0.5]
NextPlot = replace
XLim = [0 10]
XScale = linear
XTick = [(1 by 11) double array]
YAxisLocation = left
YLim = [1 2.8]
YScale = linear
YTick = [1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8]

Конкретизація значення обраного параметру:

get (gca, PropName)

Зміна кольору і товщини лінії:

plot(x1,y1, 'LineWidth', 2, 'Color', [0.8 0.8 0])

або

h = plot (x1,y1), set (h , 'LineWidth', 2, 'Color', [0.8 0.8 0])

Виведення текстової інформації в графічних вікнах:

title (string) – виведення заголовку графіка;

xlabel (string), ylabel (string) – позначення осей;

text (string, location), gtext (string) – виведення інших текстів у графічне вікно (коментарів, позначень графіків, тощо); локація тексту (рядку символів) при використанні функції визначається графічним курсором;

legend (string1, string2, ..., stringN, position) – виведення легенди; кількість рядків символів збігається з кількістю графіків у поточній системі координат графічного вікна; параметр position визначає позицію розташування легенди і може приймати такі значення: 1, 2, 3, 4 – номер квадранта, -1 – найбільш зручне місце у полі графіків, 0 – поза полем графіків.

Визначення координат виділених за допомогою миші точок:

$$[x,y] = \text{ginput} (n)$$

де n – кількість точок.

Нанесенням на графік координатної сітки керує команда `grid` у таких модифікаціях: `grid on` – увімкнути режим зображення сітки, `grid off` – вимкнути цей режим, `grid` – змінити стан команди.

За аналогією команда `hold` керує ***режимом затримки графіка***, побудованого попереднім оператором `plot`, при виконанні наступного оператора `plot`, тобто `hold on` вмикає цей режим, `hold off` вимикає його, а `hold` змінює поточний стан команди.

Зміна меж системи координат:

$$\begin{aligned} &\text{xlim} ([x_left \ x_right]), \\ &\text{ylim} ([y_bottom \ y_top]) \end{aligned}$$

та

$$\text{axis} ([x_left \ x_right \ y_bottom \ y_top])$$

Керування графічними вікнами:

`figure` – відкриває нове графічне вікно з черговим номером;

`figure (n)` – відкриває або активізує графічне вікно з заданим номером;

`close` – закриває активне графічне вікно;

`close (n)` – закриває графічне вікно з заданим номером;

`close all` – закриває усі графічні вікна;

`shg` – показує активне графічне вікно на передньому плані.

Розбиття графічного вікна на підвікна:

$$\text{subplot} (m,n,p)$$

де m – кількість підвікон по вертикалі (кількість рядків у матриці підвікон), n – кількість підвікон по горизонталі (кількість стовпців у матриці підвікон), p ($1 \leq p \leq m \cdot n$) – номер поточного активного підвікна (накрізна нумерація здійснюється по рядках), у якому будуть виконуватися дії, передбачені наступним графічним оператором.

Полярна система координат (СК) – це двовимірна система координат, в якій кожна точка на площині визначається двома числами: **полярним кутом ϕ** та **полярним радіусом ρ** :

$$\rho = f(\phi). \quad (2.1)$$

Полярна СК задається **нульовим променем** або **полярною віссю**. Точка, з якої виходить цей промінь називається началом координат або **полюсом**. Радіальна і кутова координати можуть приймати значення від 0 до ∞ .

Для побудови графіків у полярних координатах використовують функцію

polar (fi, ro)

При застосуванні параметричних формул

$$\begin{cases} x = f_x(R, \alpha), \\ y = f_y(R, \alpha) \end{cases} \quad (2.2)$$

графік функції можна побудувати як у Декартових координатах (plot (x,y)), так і в полярних (polar (alpha, ro)) координатах. Полярні координати розраховують за формулами

$$\begin{cases} \alpha = \arctg \frac{y}{x} - \text{полярний кут}, \\ \rho = \sqrt{x^2 + y^2} - \text{полярний радіус}. \end{cases} \quad (2.3)$$

В **MATLAB** ці операції можна виконати декількома способами (виконай команду help elfun):

alpha = atan2 (y,x);

ro = sqrt (x.^2+y.^2);

або

ro = hypot (x,y);

або

z = complex (x, y); alpha = angle (z); ro = abs (z);

2.2. Завдання

- Для заданих чотирьох математичних функцій відповідно до варіанту у табл. 2.1 написати програми для побудови їх графіків:
 - у чотирьох різних графічних вікнах;
 - в одному графічному вікні;
 - у двох підвікнах, розташованих по горизонталі (непарні варіанти) та по вертикалі (парні варіанти) по 2 графіки у кожному;
 - в одному графічному вікні, розбитому на 4 підвікна, по одному графіку в кожному підвікні.
- Побудувати графік кола з радіусом R і центром у точці з координатами $(0, 0)$ у полярних координатах за рівнянням

$$\rho(\varphi) = R = \text{const}, \quad \varphi \in [0, 2\pi] \quad (2.3)$$

та у Декартових координатах за рівняннями

$$\begin{cases} x = R \cos \varphi, \\ y = R \sin \varphi. \end{cases} \quad (2.4)$$

Таблиця 2.2. Функції для завдання 1

№ вар.	Функція 1	Функція 2	Функція 3	Функція 4
1,9,17	$y_1 = x - \sin x$	$y_2 = -\cos x$	$y_3 = x + 1/x$	$y_4 = \cos x + \cos x $
2,10,18	$y_1 = \arctan x + 2^x$	$y_2 = x + \log x$	$y_3 = x^2 - 5 x + 6 $	$y_4 = 1 - \cos x$
3,11,19	$y_1 = x^3 - 3x$	$y_2 = e^{-x} \sin x$	$y_3 = 1/2(\sin x - \sin x)$	$y_4 = \sin x + \cos x$
4,12	$y_1 = x^2 - x^4$	$y_2 = xe^{-x}$	$y_3 = \arcsin(x-1)$	$y_4 = \sqrt{ x }$
5,13	$y_1 = x^2 + x - 2$	$y_2 = e^x / x$	$y_3 = x^3 + 3x$	$y_4 = 1/\sqrt[3]{(1-x^2)^2}$
6,14	$y_1 = x^2 - 1 $	$y_2 = e^{-x^2}$	$y_3 = \tan(x/3)$	$y_4 = \sqrt[3]{(1-x)/(1+x)}$
7,15	$y_1 = \cos^{2/3} x$	$y_2 = e^{-x} / x^2$	$y_3 = -x^2 + x + 2$	$y_4 = \arcsin x-1 $
8,16	$y_1 = \tan^{2/3} x$	$y_2 = 3x/(1+x^3)$	$y_3 = 1/2(\arctan x + \arctan x)$	$y_4 = \sqrt{(2-x)^3}$

- За даними табл. 2.3 побудувати графіки у полярних координатах та декартових координатах.
- На довільному прикладі продемонструвати вміння використовувати деякі функції редагування графічних вікон, описані в додатку А посібника.

Таблиця 2.3. Дані до завдання 3

Вар.	Рівняння	Назва поверхні	φ	Параметри
1	$\rho = ae^{k\varphi}$	Логарифмічна спіраль	$0 \div 4\pi$	$a=4, k=0.2$
2	$\begin{cases} x = R \cos^3 \varphi \\ y = R \sin^3 \varphi \end{cases}$	Астроїда	$0 \div 2\pi$	$R = 3$
3	$\rho = 1 - \sin \varphi$	Кардіоїда	$0 \div 2\pi$	
4	$\rho = 2 - 4 \sin \varphi$	Равлик Паскаля	$0 \div 2\pi$	
5	$\begin{cases} x = k \cos \varphi - \cos(k\varphi) \\ y = k \sin \varphi - \sin(k\varphi) \end{cases}$	Епіциклоїда	$0 \div 2\pi$	$k = 4$
6	$\rho = \frac{1}{1 - \cos \varphi}$	Парабола	$0 \div 2\pi$	
7	$\begin{cases} x = 2R \operatorname{ctg} \varphi \\ y = R(1 - \cos^2 \varphi) \end{cases}$	Локон Анґезі	$\frac{\pi}{20} \div \pi - \frac{\pi}{20}$	$R = 2$
8	$\rho = \sin k\varphi$	Полярні рози	$0 \div 2\pi$	$k = 6$
9			$0 \div 8\pi$	$k = 7/4$
10			$0 \div 8\pi$	$k = 3/4$
11	$\begin{cases} x = R(k-1) \left(\cos \varphi + \frac{\cos(k-1)\varphi}{k-1} \right) \\ y = R(k-1) \left(\sin \varphi - \frac{\sin(k-1)\varphi}{k-1} \right) \end{cases}$	Гіпоциклоїди	$0 \div 6\pi$	$k = 5/3$
12			$0 \div 4\pi$	$k = 11/2$
13	$\rho = 2 - 2 \sin(\varphi) + \sin(\varphi) \frac{\sqrt{ \cos(\varphi) }}{\sin(\varphi) + 1,4}$	Серце	$0 \div 2\pi$	
14	$\rho = e^{\sin \varphi} - 2 \cos(4\varphi) + \sin^5 \frac{2\varphi - \pi}{24}$	Метелик	$0 \div 2\pi$	

15	$\begin{cases} x = 16 \sin^3 \varphi \\ y = 13 \cos \varphi - 5 \cos(2\varphi) - 2 \cos(3\varphi) - \cos(4\varphi) \end{cases}$			
16	$\rho = a \sqrt{2 \cos(2\varphi) }$	Лемніската Бернуллі	$-\frac{\pi}{4} \div \frac{\pi}{4},$ $\frac{3\pi}{4} \div \frac{5\pi}{4}$	$a=5$

2.3. Методичні рекомендації

1. При виконанні пункту 1 можна замість заданих в табл. 2.1 функцій використовувати будь-які інші функції (наприклад такі, які описують статичні або динамічні властивості об'єктів з дисциплін, що вивчаються паралельно).
2. При виконанні підпунктів пункту 1 у кожному випадку підписуйте осі, графіки в одному підвікні зображуйте різними кольорами, різними типами ліній та (або) різними маркерами, а також додавайте до них легенду. Перші 2 графіки будуйте тільки відрізками прямих між точками, третій графік – тільки маркерами, а четвертий – і маркерами і відрізками прямих між ними.
Для двох останніх графіків кількість точок в масивах абсцис та ординат підібрати так, щоб можна було б побачити тип маркера. На одному з графіків апробуйте дію функцій масштабування системи координат.
3. Не забувайте ставити «;» після операторів присвоєння, які обчислюють багато значень.
4. Не забувайте застосовувати поелементні операції «.*», «./», «.^», якщо обидва операнди є масивами однакового розміру, наприклад,
$$x = \text{linspace}(-2,2); \quad y = x.^2.*\sin(x)./ \exp(x);$$
5. Підбирайте діапазон зміни аргументу, враховуючи властивості функції, наприклад, запобігаючи тих значень аргументу, в яких функція прагне до нескінченості. Можна поєднувати декілька діапазонів зміни аргументів в 1, наприклад,

$$x = [\text{linspace}(-5,-0.5), \text{linspace}(0.5,5)]; \quad y = 1./x;$$

6. При зображенні в одній системі координат декількох функцій з суттєво різними діапазонами ординат, вирівнюйте ці діапазони множенням деяких функцій на постійний коефіцієнт:

`plot (x1,y1, x2,5*y2)`

7. Якщо деяка полярна функція задана параметричними рівняннями (2.2), то спочатку розрахуйте її полярні координати за формулами (2.3), а потім будуйте графік в полярних координатах.
8. Для того, щоб впевнитися у правильності зображення графіка в полярних координатах, продублюйте той же графік у Декартових координатах. Якщо обидва графіки однакові, то завдання виконано правильно. Якщо вони відрізняються, шукайте помилку.

Практичне заняття № 3

МОДЕЛЮВАННЯ ЛІНІЙНИХ ЕЛЕКТРИЧНИХ КІЛ

3.1. Довідкова інформація

Математичний опис динамічної системи довільного виду n -го порядку з k входами та r виходами у вигляді системи диференціальних рівнянь першого порядку в нормальній формі Коші:

$$\begin{cases} \frac{dx_1(t)}{dt} = f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ \frac{dx_2(t)}{dt} = f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ \dots, \\ \frac{dx_n(t)}{dt} = f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \end{cases} \quad (3.1)$$

де

$$\begin{cases} y_1 = g_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k), \\ y_2 = g_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k), \\ \dots, \\ y_r = g_r(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k), \end{cases} \quad (3.2)$$

$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ – вектор змінних стану;

$\mathbf{y} = [y_1 \ y_2 \ \dots \ y_r]^T$ – вектор вихідних сигналів;

$\mathbf{u} = [u_1 \ u_2 \ \dots \ u_k]^T$ – вектор вхідних сигналів

з початковими умовами

$$\mathbf{x}(t_0) = \mathbf{x}_0 = [x_{10} \ x_{20} \ \dots \ x_{n0}]^T. \quad (3.3)$$

Перехід до операторної форми:

$\frac{d}{dt} \rightarrow s$ – заміна оператора диференціювання оператором Лапласа;

$x_i(t) \rightarrow x_i(s), \ u_i(t) \rightarrow u_i(s), \ y_i(t) \rightarrow y_i(s)$ – заміна сигналів у просторі часу їх зображеннями у функції оператора Лапласа.

ДР (3.1) в операторній формі:

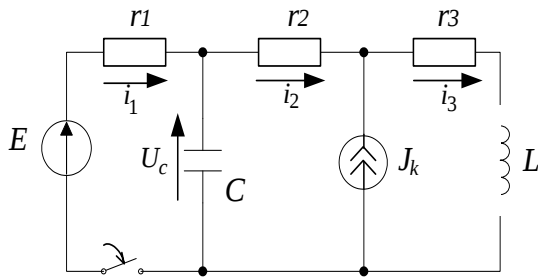
$$\begin{cases} sx_1(s) = f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ sx_2(s) = f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ \dots, \\ sx_n(s) = f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r). \end{cases} \quad (3.4)$$

У такий спосіб *операція диференціювання у просторі часу замінюється операцією множення зображення сигналу на змінну Лапласа*. Відповідно *операції інтегрування відповідає операція ділення зображення сигналу на оператор Лапласа*.

Порядок розробки деталізованої Simulink-моделі, її налаштування, симуляції та фіксації результатів:

- розробка структурної схеми за рівняннями (3.1)-(3.4) **методом неявних сигналів**: формування перших похідних від змінних стану за рівняннями (3.1), не звертаючи уваги на те, що деякі з них є невідомими; формування змінних стану шляхом інтегрування їх похідних (встановлення інтеграторів $W_i(s) = 1/s$, $i = 1, 2, \dots, n$ з початковими умовами (3.3)); формування вихідних сигналів за рівняннями (3.2);
- створення за розробленою структурною схемою Simulink-моделі: вибір із - бібліотек потрібних блоків, з'єднання їх між собою, доповнення моделі блоками формування вхідних сигналів та реєстрації вихідних сигналів, встановлення параметрів блоків (краще задавати їх у вигляді змінних, а не констант);
- забезпечення ініціалізації моделі;
- розробка плану модельного експерименту і створення у разі необхідності програмних файлів для його реалізації;
- налаштування моделі шляхом вибору методу розв'язання диференціальних рівнянь та його параметрів (Simulation → Model Configuration ;
- виконують власне симуляцію, фіксують, візуалізують та аналізують отримані результати.

3.2. Приклад математичного моделювання лінійного електричного кола в *Simulink*



ключа

Як приклад, розглянемо розгалужену схему, зображену на рис. 3.1, для якої треба розрахувати перехідні процеси, тобто знайти $i_1(t)$, $i_2(t)$, $i_3(t)$, $U_C(t)$ при замиканні

Рис. 3.1. Досліджуване електричне коло при таких параметрах: $E = 100$ В, $J_k = 5$ А, $C = 200$ мкФ, $L = 0,3$ Гн, $r_1 = 20$ Ом, $r_2 = 50$ Ом, $r_3 = 70$ Ом.

Математичний опис лінійних електричних кіл створюють, використовуючи закони Ома та Кірхгофа з урахуванням тієї обставини, що напруги і струми на резисторі індуктивності та конденсаторі пов'язані між собою співвідношеннями:

$$U_R(t) = RI_R(t), \quad U_L(t) = L \frac{dI_L(t)}{dt}, \quad I_C(t) = C \frac{dU_C(t)}{dt}. \quad (3.5)$$

Математичний опис схеми при замкненому стані ключа, згідно з законами Кірхгофа, має вигляд:

$$E = i_1 r_1 - U_c, \quad (3.6)$$

$$E = i_1 r_1 + i_2 r_2 + i_3 r_3 + L \frac{di_3}{dt}, \quad (3.7)$$

$$i_1 + C \frac{dU_c}{dt} = i_2, \quad (3.8)$$

$$i_2 + J_k = i_3. \quad (3.9)$$

Рівняння (3.7), (3.8) – диференційні; (3.6), (3.9) – алгебраїчні рівняння, які часто називають рівняннями зв'язку.

Перетворимо диференційні рівняння у нормальну форму Коші:

$$\begin{cases} \frac{dU_c}{dt} = (i_2 - i_1)/C, \\ \frac{di_3}{dt} = (E - i_1 r_1 - i_2 r_2 - i_3 r_3)/L. \end{cases} \quad (3.10)$$

Об'єднаємо змінні стану, вхідні та вихідні сигнали у відповідні вектори:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} U_c \\ i_3 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} E \\ J_k \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} U_c \\ i_3 \\ i_1 \\ i_2 \end{bmatrix}. \quad (3.11)$$

Як бачимо, два перших вихідних сигналів співпадають зі змінними стану, а два останніх можна виразити через змінні стану з рівнянь зв'язку:

$$\begin{cases} i_1 = (E + U_c)/r_1, \\ i_2 = i_3 - J_k. \end{cases} \quad (3.12)$$

Записуємо ДР (3.10) в операторній формі:

$$\begin{cases} sU_c(s) = (i_2(s) - i_1(s))/C, \\ si_3(s) = (E(s) - i_1(s)r_1 - i_2(s)r_2 - i_3(s)r_3)/L. \end{cases} \quad (3.13)$$

Рівняння виходу (3.12) у зображеннях залишаються такими ж, як і у просторі часу.

Розрахуємо початкові умови для заданої схеми, які збігаються з усталеними значеннями відповідних сигналів при розімкненому стані ключа. Така схема зображена на рис. 3.2.

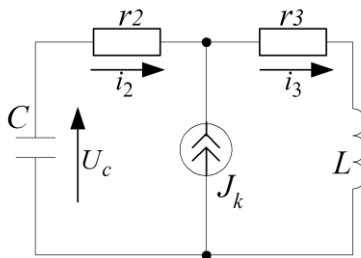


Рис. 3.2. Схема для визначення початкових умов

За законами Кірхгофа вона описується рівняннями

$$\begin{cases} i_2 = C \frac{dU_c}{dt}, \\ i_2 + J_k = i_3, \\ U_c + i_2 r_2 + i_3 r_3 + L \frac{di_3}{dt} = -J_k r_3. \end{cases}$$

В усталеному режимі усі похідні дорівнюють 0, звідси

$$\begin{cases} i_3(0) = J_k, \\ U_c(0) = -i_3 r_3 = -J_k r_3. \end{cases} \quad (3.14)$$

Для складання структурної моделі лінійного неперервного динамічного об'єкта завжди можна обійтися тільки трьома типами блоків: інтеграторами, алгебраїчними суматорами та блоками множення на постійний коефіцієнт, які у *Simulink* називаються відповідно: *Integrator*, *Sum* та *Gain*. Ще декілька блоків знадобляться для запам'ятовування або для візуалізації результатів моделювання.

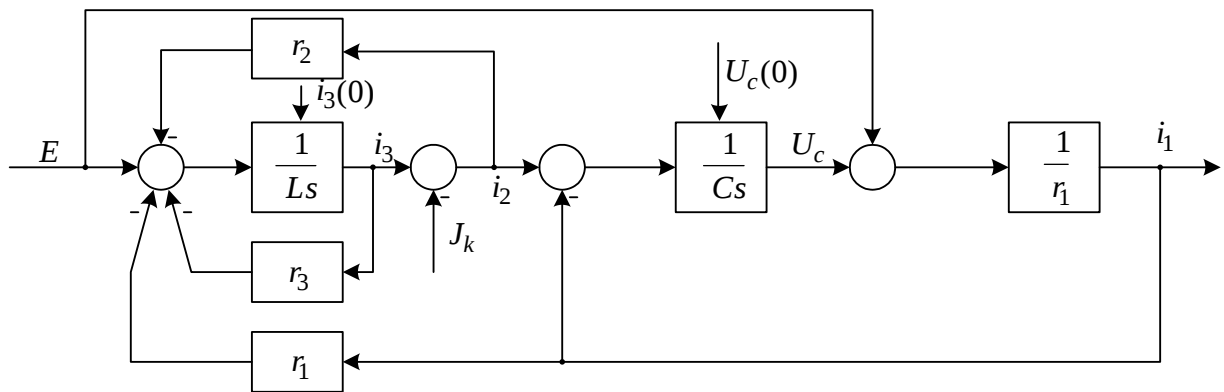


Рис. 3.3. Структурна схема досліджуваного електричного кола

Відповідна схема в блоках *Simulink* показана на рис. 3.4, на якій вхідні сигнали формуються блоками *Constant*, а вихідні запам'ятовуються блоками *To Workspace*. Початкові умови встановлюються на інтеграторах.

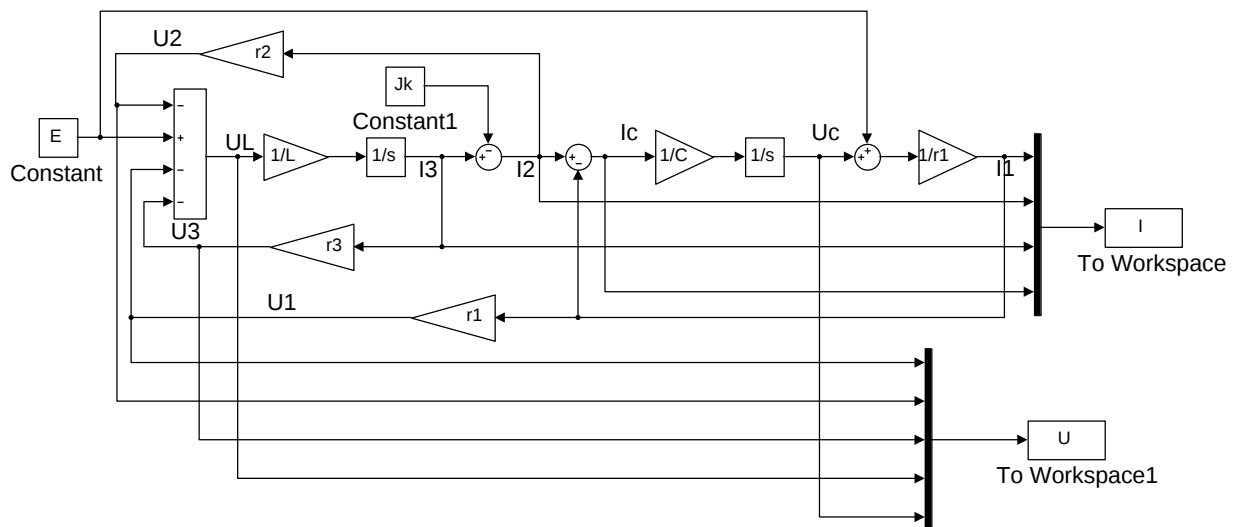


Рис. 3.4. Структурна *Simulink*-модель досліджуваного лінійного електричного кола

За результати симуляції за допомогою операторів *MATLAB* побудовані графіки перехідних процесів струмів та напруг, зображені на рис. 3.5.

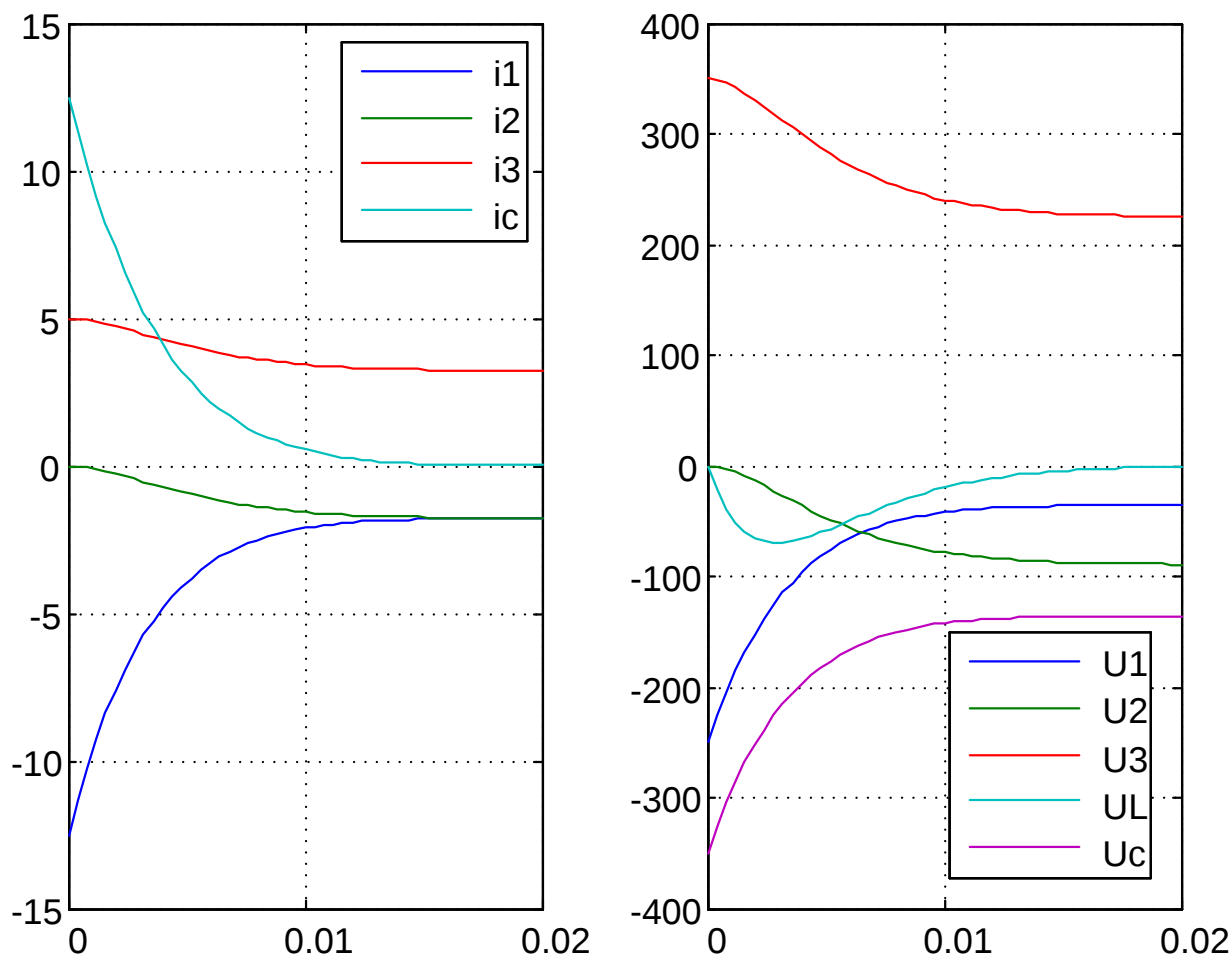


Рис. 3.5. Перехідні процеси у досліджуваному лінійному електричному колі

3.3. Завдання

Розрахувати перехідні процеси напруги на конденсаторі та струмів у гілках однієї з електричних схем, зображених на рис. 3.6, методом структурного математичного моделювання для $U=U_m$ – джерело постійної напруги.

За результатами розрахунків побудувати графіки перехідних процесів. Параметри схем надані у табл. 3.1

3.4 Порядок виконання завдання

1. Ознайомитися з методами складання математичного опису електричних кіл та розробки їх структурних моделей.

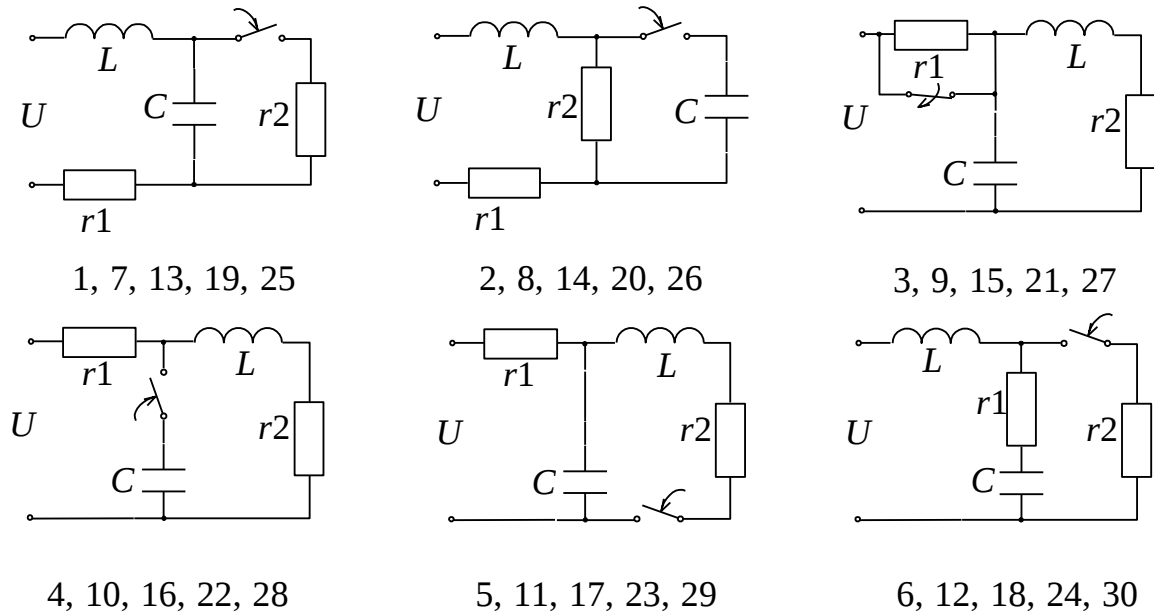


Рис. 3.6. Варіанти розгалужених лінійних електричних кіл

Таблиця 3.1

№ вар.	$U_m, \text{В}$	$r1, \text{Ом}$	$r2, \text{Ом}$	$\omega, \text{с}^{-1}$	$L, \text{Гн}$	$C, \text{мкФ}$
1-6	200	50	100	125	0,1	10
7-12	100	100	120	100	0,2	5
13-18	300	60	120	75	0,15	15
19-24	400	150	100	150	0,25	2
25-30	500	100	200	100	0,05	10

2. Виконати математичний опис досліджуваних електричних кіл при замкненому і розімкненому ключі, перетворити диференційні рівняння до нормальної форми Коші та записати їх в операторній формі.
3. Враховуючи, що змінними стану в заданих схемах будуть струми, що протікають через котушку індуктивності, і напруги на конденсаторах, за отриманими операторними рівняннями у початковому стані ключового

- елементу визначити усталені значення цих сигналів, які є початковими умовами для режиму перемикання ключа.
4. Розробити структурну схему для кінцевого стану ключового елемента.
 5. Перейти від структурних схем до структурних моделей із застосуванням блоків *Simulink*-бібліотек, ввести їх у комп'ютер, встановити параметри блоків, обрати метод розв'язання ДР та його параметри.
 6. Забезпечити програмно ініціалізацію моделей.
 7. Виконати розрахунок перехідних процесів досліджуваних електричних кіл за допомогою створених моделей, візуалізувати та проаналізувати результати.

3.5. Методичні рекомендації

1. Параметри моделей задавайте іменами змінних, а не константами.
2. Для попередньої візуалізації використовуйте блоки *Scope*, а для запам'ятовування чисельних результатів з метою подальшої побудови графіків перехідних процесів – вихідні порти *Out* та/або блоки *ToWorkspace*. Усі струми виводьте на один графік, а напруги – на інший.
3. Ініціалізацію моделей та побудову графіків виконайте за допомогою виконання послідовності опцій меню *File: Model Properties → Callbacks → InitFn/StopFn* (див. підрозділ 5.2 навчального посібника). Попередньо позиції *InitFn* та *StopFn* вкладки *Callbacks* вікна *Model Properties* треба заповнити командами або операторами, які Ви хочете виконати перед симуляцією та після її закінчення відповідно.
4. Час розрахунку перехідного процесу обирайте таким, щоб усталений режим був досягнутий, але не тривав дуже довго.

Практичне заняття №4

ПОБУДОВА ЧАСТОТНИХ ХАРАКТЕРИСТИК

4.1. Довідкова інформація

Частотна передавальна функція (ЧПФ):

$$W(j\omega) = W(s)|_{s=j\omega}, \quad (4.1)$$

де ω – частота, $j = \sqrt{-1}$ – уявна одиниця.

Алгебраїчна форма ЧПФ:

$$W(j\omega) = W_{re}(\omega) + jW_{im}(\omega) \quad (4.2)$$

Експоненціальна форма ЧПФ:

$$W(j\omega) = A(\omega)e^{j\varphi(\omega)}. \quad (4.3)$$

Зв'язок між алгебраїчною та експоненціальною формами ЧПФ:

$$A = |W(j\omega)| = \sqrt{W_{re}^2 + W_{im}^2}; \quad (4.4)$$

$$\varphi = \arg(W(j\omega)) = \arctg(W_{im} / W_{re}); \quad (4.5)$$

$$W_{re} = A \cos \varphi; \quad (4.6)$$

$$W_{im} = A \sin \varphi. \quad (4.7)$$

Тригонометрична форма ЧПФ:

$$W = A(\cos \varphi + j \sin \varphi). \quad (4.8)$$

Види частотних характеристик:

$W_{re}(\omega)$ – дійсна частотна характеристика (ДЧХ);

$W_{im}(\omega)$ – уявна частотна характеристика (УЧХ);

$A(\omega)$ – амплітудно- частотна характеристика (АЧХ);

$\varphi(\omega)$ – фазо- частотна характеристика (ФЧХ);

$A(\varphi)$ – амплітудно-фазова частотна характеристика (АФЧХ);

$20 \lg A(\lg \omega)$ – логарифмічна амплітудно- частотна характеристика (ЛАЧХ);

$\varphi(\lg \omega)$ – фазо-частотна характеристика (ЛФЧХ).

АФЧХ називають *годографом або діаграмою Найквіста (Nyquist Plot)*.

ЛАЧХ та ЛФЧХ називають також *діаграмами Бode (Bode Plots)*.

Відношення амплітуди вихідної синусоїди до амплітуди вхідної синусоїди є *амплітудою частотних характеристик*, а різниця фаз (фазовий зсув) – *фазою частотних характеристик*:

$$A(\omega) = \frac{A_{out}}{A_{in}}, \quad \varphi(\omega) = \varphi_{out} - \varphi_{in}. \quad (4.9)$$

Амплітуда в децибелах (скорочено дБ або dB):

$$L(\omega) = 20 \lg A(\omega) \quad (4.10)$$

Розрахунок ЧПФ W в за відомими коефіцієнтами поліномів чисельника num та знаменника den і частотою omega:

H = polyval (num, p);

G = polyval (den, p);

W = H ./ G;

або

W = freqs (num, den, omega);

де

omega = linspace (0, wk);

або

omega = 0:dw:wk;

для ДЧХ, УЧХ, АЧХ і ФЧХ,

omega = logspace (d0, dk);

для ЛАЧХ і ЛФЧХ,

p = j*omega;

Розрахунок складових комплексних чисел W за формулами (4.4)-(4.7):

U = real (W); V = imag (W);

A = abs (W); Phi = angle (W);

Phi_deg = Phi ./pi * 180;

Останній оператор перетворює фазу в радіанах у фазу в градусах.

Побудова лінійних ЧХ:

```
figure (1), plot (omega, U, omega, V), legend ('ДЧХ','МЧХ')  
figure (2), subplot (2,1,1), plot (omega, A), grid on, title ('АЧХ')  
subplot (2,1,2), plot (omega, Phi_deg), grid on, title ('ФЧХ')
```

Побудова АФЧХ в Декартових і в полярних координатах:

```
plot (U, V), axis equal  
polar (A, Phi)
```

Побудова діаграм Бодє (ЛЧХ):

```
L = 20*log10(A);  
figure, subplot (2,1,1), semilogx (omega, L), grid on, title ('ЛАЧХ')  
subplot (2,1,2), semilogx (omega, Phi_deg), grid on, title ('ЛФЧХ')
```

Перетворення амплітуди в абсолютних одиницях в децибели
($L = 20 \cdot \log_{10}(A)$;) можна здійснити функцією

```
L = mag2db (A);
```

Створення LTI-об'єкту sys для використання функцій аналізу та синтезу інструментів Control System Toolbox:

```
sys = tf (num, den)
```

Побудова діаграми Бодє і Найквіста без збереження у пам'яті результатів їх розрахунку:

```
bode (sys)                                nyquist (sys)  
bode (sys, omega)                        nyquist (sys, omega)
```

Частота ω у списку вхідних аргументів може бути задана вектором рівномірно розподіленим у логарифмічному масштабі як $\{w0, wk\}$. За замовчанням вона призначається автоматично на підставі аналізу параметрів моделі. Функції

```
bodemag (sys)    bodemag (sys, omega)
```

будують тільки ЛАЧХ (без ЛФЧХ).

Розрахунок параметрів ЧХ функціями додатку Control System Toolbox:

```
[A, phase, omega] = bode (sys)            [U, V, omega] = nyquist (sys)
```

[A, phase] = bode (sys, omega)

[U, V] = nyquist (sys, omega)

З АЧХ замкненої системи можна визначити смугу пропускання – діапазон частот, у якому амплітуда вихідного сигналу є не меншою, ніж $\sqrt{2} / 2 \approx 0,707$ від амплітуди вхідного сигналу

$$A(\omega_n) = \sqrt{2} / 2. \quad (4.11)$$

В *MATLAB* цю частоту w_{prop} можна знайти за допомогою функції *bandwidth*:

$$W_{prop} = \text{bandwidth}(\text{sys})$$

Екстремум АЧХ називають *резонансним максимумом*. Його можна знайти, застосовуючи оператор

$$Mr = \max(A)$$

або

$$Mr = \text{norm}(\text{sys})$$

Величина резонансного максимуму M_p впливає на перерегулювання перехідної функції, на коливальність перехідних процесів та на запас стійкості за фазою. Зокрема, перерегулювання можна оцінити шляхом розрахунку його за емпіричною формулою

$$\sigma \approx 1,3 \frac{M_p - 1}{M_p}. \quad (4.12)$$

4.2. Завдання

1. За даними табл. 4.1 відповідно до свого варіанту побудувати

- ДЧХ і УЧХ системи з заданою передавальною функцією в одному вікні;
- АЧХ і ФЧХ системи у двох підвікнах одного вікна (одне під іншим); визначити полосу пропускання і резонансного максимуму
- Bode-діаграми (ЛАЧХ і ЛФЧХ) системи у двох підвікнах одного вікна (одне під іншим) функцією *semilogx* і функцією *bode*;
- діаграму Найквіста (АФЧХ) у полярних координатах, у Декартових координатах та за допомогою функції *nyquist*.

2. Визначити полосу пропускання і величину резонансного максимуму (якщо він існує).
3. Методом математичного моделювання отримати реакцію досліджуваної системи на одиничний стрибок вхідного сигналу. Порівняти початкові та кінцеві значення перехідної функції та АЧХ.
4. Методом математичного моделювання отримати реакцію досліджуваної системи на синусоїдальний сигнал з частотою пропускання та її подвоєним значенням; пояснити зв'язок між частотними характеристиками та перехідними процесами. Порівняти амплітуди і фази вихідних сигналів, отриманих з АЧХ та при моделюванні.

4.3. Методичні рекомендації

1. ЧХ можете розраховувати і будувати будь-яким методом, що дозволяють забезпечити інструменти пакету *MATLAB*.
2. На усі графіки нанесіть координатні сітки, титули, позначення осей та характеристик.
3. Діаграми Найквіста будуйте у режимі *axis equal*.
4. Амплітуду і фазу реакцій системи на синусоїдальні вхідні сигнали визначаєте в усталеному режимі.
5. Фазовий зсув реакції системи на синусоїдальні вхідні сигнали визначаєте за формулою

$$\varphi(\omega) = 360 \frac{\tau}{T} = 360 \frac{\tau \omega}{2\pi}, \quad (4.12)$$

де T, ω, τ – період і частота вхідної синусоїди, та час запізнення вихідної синусоїди в усталеному режимі.

Таблиця 4.1

Вар.	Передавальна функція	Параметри
1	$W(p) = \frac{k}{T_2 T_1 p^2 + T_1 p + 1}$	$k = 10, T_1 = 1, T_2 = 0.5$
2		$k = 5, T_1 = 1, T_2 = 1$
3	$W(p) = \frac{T_3 p}{T_2 T_1 p^2 + T_1 p + 1}$	$T_1 = 1, T_2 = 2, T_3 = 0.5$
4		$T_1 = 1, T_2 = 1, T_3 = 1$
5	$W(p) = \frac{k(T_3 p + 1)}{T_2 T_1 p^2 + T_1 p + 1}$	$k = 10, T_1 = 1, T_2 = 2, T_3 = 1$
6		$k = 2, T_1 = 1, T_2 = 1, T_3 = 2$
7	$W(p) = \frac{k(T_4 T_3 p^2 + T_3 p + 1)}{T_2 T_1 p^2 + T_1 p + 1}$	$k = 0.5, T_1 = 1, T_2 = 2, T_3 = 0.5, T_4 = 1$
8		$k = 1, T_1 = 1, T_2 = 1, T_3 = 2, T_4 = 1$
9	$W(p) = \frac{T_3^2 p^2}{T_2 T_1 p^2 + T_1 p + 1}$	$k = 0.25, T_1 = 1, T_2 = 2, T_3 = 0.75$
10		$k = 2, T_1 = 1, T_2 = 1.5, T_3 = 2$
11	$W(p) = k \frac{(T_3^2 p^2 + 1)}{T_2 T_1 p^2 + T_1 p + 1}$	$k = 4, T_1 = 1, T_2 = 0.8, T_3 = 0.5$
12		$k = 0.4, T_1 = 1, T_2 = 2, T_3 = 0.5$
13	$W(p) = \frac{T_2 p}{T_1^2 p^2 + T_3 p + 1}$	$T_1 = 1, T_2 = 2, T_3 = 0.5$

Практичне заняття № 5
РОЗРАЗХУНОК УСТАЛЕНИХ РЕЖИМІВ
У ЕЛЕКТРИЧНИХ КОЛАХ МАТРИЧНИМ МЕТОДОМ

5.1. Довідкова інформація

Математичний опис усталених режимів в лінійних електричних колах постійного та змінного струмів утворюється *методом законів Кірхгофа*. Він являє собою систему лінійних алгебраїчних рівнянь, в якій кількість рівнянь і кількість невідомих струмів дорівнює кількості гілок.

При дослідженні статичних властивостей кіл постійного струму система лінійних алгебраїчних рівнянь матиме тільки дійсні коефіцієнти, а при дослідженні кіл змінного струму частина коефіцієнтів буде мати комплексні значення із-за залежності комплексних опорів котушок індуктивності та конденсаторів від частоти.

В обох випадках для визначення струмів гілок необхідно розв'язувати *систему n лінійних рівнянь з n невідомими*, яка має вигляд

- *у скалярній формі*

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n = b_1, \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n = b_2, \\ \dots \\ a_{n1} \cdot x_1 + a_{n2} \cdot x_2 + \dots + a_{nn} \cdot x_n = b_n, \end{cases} \quad (5.1)$$

- *у матричній формі*

$$\mathbf{AX} = \mathbf{B}, \quad (5.2)$$

де $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$ – квадратна матриця коефіцієнтів;

$$\mathbf{B} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} - \text{вектор вільних членів}; \quad \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} - \text{шуканий вектор коренів}.$$

Способи рішення систем лінійних алгебраїчних рівнянь (СЛАР):

- точні методи (метод обернення матриці коефіцієнтів, правило Крамера, метод Гауса та інші);
- ітераційні методи (Ньютона, Зейделя, простих ітерацій та інші).

Розв'язок СЛАР методом обернення матриці коефіцієнтів (якщо матриця $\mathbf{A}$ неособлива, тобто її визначник не дорівнює нулю):

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}, \quad (5.3)$$

де $\mathbf{A}^{-1}$ – матриця, обернена до матриці $\mathbf{A}$.

Розв'язок системи методом Крамера:

$$x_1 = \frac{\Delta_1}{\Delta}, x_2 = \frac{\Delta_2}{\Delta}, \dots, x_n = \frac{\Delta_n}{\Delta}, \quad (5.4)$$

де Δ – визначник матриці $\mathbf{A}$;

Δ_i – визначники матриць, отримані з матриці $\mathbf{A}$ шляхом заміни її i -го стовпця вектором вільних членів $\mathbf{B}$.

Визначник матриці або **детермінант матриці** розраховується з усіх перестановок елементів матриці розміром $n \times n$ як сума добутків із n елементів. Кожний доданок складається з добутку елементів у такий спосіб, щоб у ньому були присутні тільки по одному елементу з кожного рядка і з кожного стовпця. Він може позначатися як Δ , $\Delta(\mathbf{A})$, $\det(\mathbf{A})$, $|\mathbf{A}|$.

Визначники матриць другого та третього порядків:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12},$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} - a_{31}a_{22}a_{13} - a_{21}a_{12}a_{33} - a_{32}a_{23}a_{11}.$$

Оберненою матрицею $\mathbf{X}=\mathbf{A}^{-1}$ у відношенні до вихідної квадратної матриці $\mathbf{A}$ називається така квадратна матриця яка, будучи помноженою на вихідну, дає одиничну діагональну матрицю $\mathbf{E}$ того ж розміру:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{E}, \quad (5.5)$$

або у розгорнутій формі:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{12} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad (5.6)$$

Розрахунок оберненої матриці:

$$\mathbf{A}^{-1} = \frac{\text{Adj}(\mathbf{A})}{\det(\mathbf{A})}, \quad (5.7)$$

$$\text{Adj}(\mathbf{A}) = \tilde{\mathbf{A}}^T, \quad (5.8)$$

$$\tilde{A}_{ij} = (-1)^{i+j} M_{ij}. \quad (5.9)$$

де $\text{Adj}(\mathbf{A})$ – приєднана матриця, що дорівнює транспонованій союзній матриці $\tilde{\mathbf{A}}^T$; M_{ij} – **мінор** елементу a_{ij} визначника квадратної матриці n -го порядку, що являє собою називають визначник матриці $(n-1)$ -го порядку, отриманої із вихідної матриці викреслюванням з неї елементів i -го рядка та j -го стовпчика.

М-функції, що виконують основні операції лінійної алгебри:

$d = \det(\mathbf{A})$ – розраховує визначник матриці;

$\mathbf{A}_o = \text{inv}(\mathbf{A})$ – розрахунок оберненої матриці;

$\mathbf{X} = \text{linsolve}(\mathbf{A}, \mathbf{B})$ – розв’язання системи лінійних алгебраїчних рівнянь.

Розв’язати систему лінійних алгебраїчних рівнянь у пакеті MATLAB можна ще виконати за допомогою **операції лівобічного ділення**

$$\mathbf{X} = \mathbf{A} \backslash \mathbf{B}$$

або безпосередньо за формулою (5.3):

$$\mathbf{X} = \text{inv}(\mathbf{A}) * \mathbf{B}$$

Розрахунок вектору нев'язок для перевірки точності визначення коренів:

$$F = A \cdot X - B$$

Розрахунок матриці нев'язок для перевірки точності визначення оберненої матриці:

$$F_0 = \text{eye}(\text{size}(A)) - \text{inv}(A) \cdot A,$$

що випливає з формули (5.5):

$$F_0 = E - A^{-1}A, \quad (5.10)$$

При вірному виконанні розглянутих операцій нев'язки (неузгодженості) повинні бути близькими до нуля.

5.2. Приклад застосування методів лінійної алгебри при розрахунку струмів і напруг у гілках кіл постійного струму

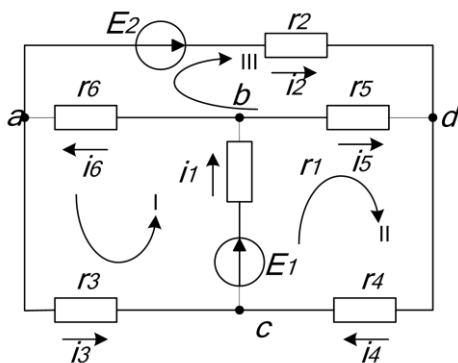


Рис. 5.1. Розгалужене електричне коло

Знайти струми в гілках розгалуженого електричного кола, зображеного на рис. 5.1 при відомих значеннях опорів резисторів та параметрів джерел струмів та ЕРС: $r_1=8 \text{ Ом}$; $r_2=10 \text{ Ом}$; $r_3=5 \text{ Ом}$; $r_4=7 \text{ Ом}$; $r_5=12 \text{ Ом}$; $r_6=3 \text{ Ом}$; $E_1=100 \text{ В}$; $E_2=30 \text{ В}$. Правильність результату перевірити методом балансу потужностей.

Досліджувана система має 6 гілок та 4 вузли, з них незалежними є 3 вузли. Позначимо їх буквами а, b, с. До незалежних вузлів додаємо 3 незалежних контури: I, II, III. Виберемо напрямки струмів у гілках та додатні напрямки у незалежних контурах. Після цього складаємо систему 6 лінійних рівнянь з 6 невідомими: 3 за першим законом Кірхгофа і 3 за другим:

$$\begin{aligned}
 a: & \begin{cases} i_6 = i_2 + i_3, \\ \end{cases} \\
 b: & \begin{cases} i_1 = i_6 + i_5, \\ \end{cases} \\
 c: & \begin{cases} i_3 + i_4 = i_1, \\ \end{cases} \\
 \text{I:} & \begin{cases} E_1 = i_1 r_1 + i_6 r_6 + i_3 r_3, \\ \end{cases} \\
 \text{II:} & \begin{cases} E_1 = i_1 r_1 + i_4 r_4 + i_5 r_5, \\ \end{cases} \\
 \text{III:} & \begin{cases} E_2 = i_2 r_2 + i_6 r_6 - i_5 r_5. \\ \end{cases}
 \end{aligned} \tag{5.11}$$

Формалізуємо систему (5.11) згідно з загальним виглядом системи лінійних рівнянь:

$$\begin{cases}
 0 \cdot i_1 - 1 \cdot i_2 - 1 \cdot i_3 + 0 \cdot i_4 + 0 \cdot i_5 + 1 \cdot i_6 = 0, \\
 1 \cdot i_1 + 0 \cdot i_2 + 0 \cdot i_3 + 0 \cdot i_4 - 1 \cdot i_5 - 1 \cdot i_6 = 0, \\
 -1 \cdot i_1 + 0 \cdot i_2 + 1 \cdot i_3 + 1 \cdot i_4 + 0 \cdot i_5 + 0 \cdot i_6 = 0, \\
 r_1 \cdot i_1 + 0 \cdot i_2 + r_3 \cdot i_3 + 0 \cdot i_4 + 0 \cdot i_5 + r_6 \cdot i_6 = E_1, \\
 r_1 \cdot i_1 + 0 \cdot i_2 + 0 \cdot i_3 + r_4 \cdot i_4 + r_5 \cdot i_5 + 0 \cdot i_6 = E_1, \\
 0 \cdot i_1 + r_2 \cdot i_2 + 0 \cdot i_3 + 0 \cdot i_4 - r_5 \cdot i_5 + r_6 \cdot i_6 = E_2.
 \end{cases} \tag{5.12}$$

З (5.12) складаємо матрицю коефіцієнтів та вектор вільних членів

$$\mathbf{A} = \begin{pmatrix} 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & -1 \\ -1 & 0 & 1 & 1 & 0 & 0 \\ r_1 & 0 & r_3 & 0 & 0 & r_6 \\ r_1 & 0 & 0 & r_4 & r_5 & 0 \\ 0 & r_2 & 0 & 0 & -r_5 & r_6 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ E_1 \\ E_1 \\ E_2 \end{pmatrix}. \tag{5.13}$$

Тоді програма для розрахунку струмів у гілках схеми рис 5.2 матиме вигляд:

```

clc, clear all
r1=8; r2=10; r3=5; r4=7; r5=12; r6=3;
e1=100; e2=30;
A=[ 0 -1 -1 0 0 1;
    1 0 0 0 -1 -1;
    -1 0 1 1 0 0;
    r1 0 r3 0 0 r6;
    r1 0 0 r4 r5 0;
    0 r2 0 0 -r5 r6];
B=[0; 0; 0; e1; e1; e2];
i=A\B
E=[e1;e2;0;0;0;0];

```

```

P=sum(E.*i)
R=[r1; r2; r3; r4; r5; r6];
dP=sum(i.^2.*R)

```

В результаті її виконання отримуємо:

```

i =
    7.5975
    2.4584
    3.9806
    3.6169
    1.1585
    6.4390
P =
    833.5034
dP =
    833.5034

```

Як бачимо, сумарна потужність джерел ЕРС співпадає з потужністю втрат електроенергії на активних опорах, тобто баланс потужностей виконується.

5.3. Приклад застосування методів лінійної алгебри при розрахунку усталених режимів у колах змінного струму

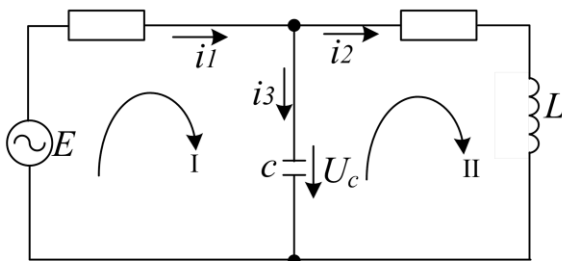


Рис. 5.2. Розгалужене електричне коло змінного струму

Побудувати графіки струмів та напруги на конденсаторі в гілках розгалуженого електричного кола змінного струму, зображеного на рис. 5.2, на одному періоді усталеного режиму при $E = E_m \sin(\omega t)$, $E_m = 220$ В, $\omega = 2\pi f$, $f = 50$ Гц, $r_1 = 10$ Ом; $r_2 = 20$ Ом; $L = 0.1$ Гн; $C = 30$ мкФ.

Щоб побудувати графіки усталених синусоїдальних сигналів, необхідно розрахувати їх амплітуди і фазові зсуви відносно ЕРС джерела. Ці величини розраховуються, як і у попередньому прикладі шляхом розв'язання системи лінійних алгебраїчних рівнянь, складених за законами Кірхгофа. Різниця полягає в тому, що значення опорів гілок будуть комплексними числами:

$$z_1=R_1, \quad z_2=R_2+j\omega L, \quad z_3=\frac{1}{j\omega C}. \quad (5.14)$$

З урахуванням (5.14) згідно з законами Кірхгофа досліджуване електричне коло в усталеному режимі описується системою лінійних рівнянь 3-го порядку з комплексними коефіцієнтами:

$$\begin{cases} i_1=i_2+i_3, \\ E_m=i_1z_1+i_2z_2, \\ i_2z_2=i_3z_3. \end{cases} \quad (5.13)$$

Формалізуємо систему (5.13) згідно з загальним виглядом системи лінійних рівнянь:

$$\begin{cases} 1 \cdot i_1 - 1 \cdot i_2 - 1 \cdot i_3 = 0, \\ z_1 \cdot i_1 + z_2 \cdot i_2 + 0 \cdot i_3 = E_m, \\ 0 \cdot i_1 + z_2 \cdot i_2 - z_3 \cdot i_3 = 0. \end{cases} \quad (5.14)$$

З (5.14) складаємо матрицю коефіцієнтів та вектор вільних членів

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & -1 \\ z_1 & z_2 & 0 \\ 0 & z_2 & -z_3 \end{pmatrix}. \quad (5.15)$$

Після розв'язання системи рівнянь (5.15) усталені струми отримаємо також у вигляді комплексних чисел, що складаються з дійсної та уявної частин:

$$i(j\omega) = i_{re} + j i_{im}. \quad (5.16)$$

Їх треба перетворити до експоненціальної форми

$$i(j\omega) = A_i e^{j\varphi_i}, \quad (5.17)$$

де

$$A_i = \sqrt{i_{re}^2 + i_{im}^2}, \quad \varphi_i = \arctg \frac{i_{im}}{i_{re}}. \quad (5.18)$$

Після цього можна записати рівняння усталеного періодичного режиму у функції часу:

$$i_{ycm}(t) = A_i \sin(\omega t + \varphi_i). \quad (5.19)$$

Тоді програма для розрахунку струмів у гілках схеми рис 5.2 матиме вигляд:

```
clc, clear all
R1 = 10; R2 = 20; L = 0.1; C = 30e-6;
f = 50; T = 1/f; w = 2*pi*f;
z1 = R1; z2 = R2+j*w*L; z3 = 1/(j*w*C);
Em = 220;
A = [ 1 -1 -1;
      z1 z2 0;
      0 z2 -z3]
B = [0; Em; 0]
i = A\B
im = abs(i), phi = angle(i)
Uc = Em-i(2)*R2
Ucm = abs(Uc), phi_c = angle(Uc)
t = linspace(0,T);
E = Em*sin(w*t);
Uc = Ucm*sin(w*t+phi_c);
plot(t,[E;Uc]), legend('E(t)','Uc(t)')
i1 = im(1)*sin(w*t+phi(1));
i2 = im(2)*sin(w*t+phi(2));
i3 = im(3)*sin(w*t+phi(3));
figure, plot(t,[i1;i2;i3]), legend('i1(t)','i2(t)','i3(t)')
```

В результаті її виконання отримуємо:

```
A =
  1.0e+02 *
    0.0100 + 0.0000i -0.0100 + 0.0000i -0.0100 + 0.0000i
    0.1000 + 0.0000i  0.2000 + 0.3142i  0.0000 + 0.0000i
    0.0000 + 0.0000i  0.2000 + 0.3142i  0.0000 + 1.0610i
B =
     0
    220
     0
i =
  3.0261 - 2.1932i
  3.2328 - 3.9815i
 -0.2067 + 1.7882i
im =
    3.7373
    5.1287
    1.8002
phi =
```

-0.6272
-0.8888
1.6859

$U_c =$
 $1.5534e+02 + 7.9630e+01i$
 $U_{cm} =$
174.5637
 $\phi_c =$
0.4737

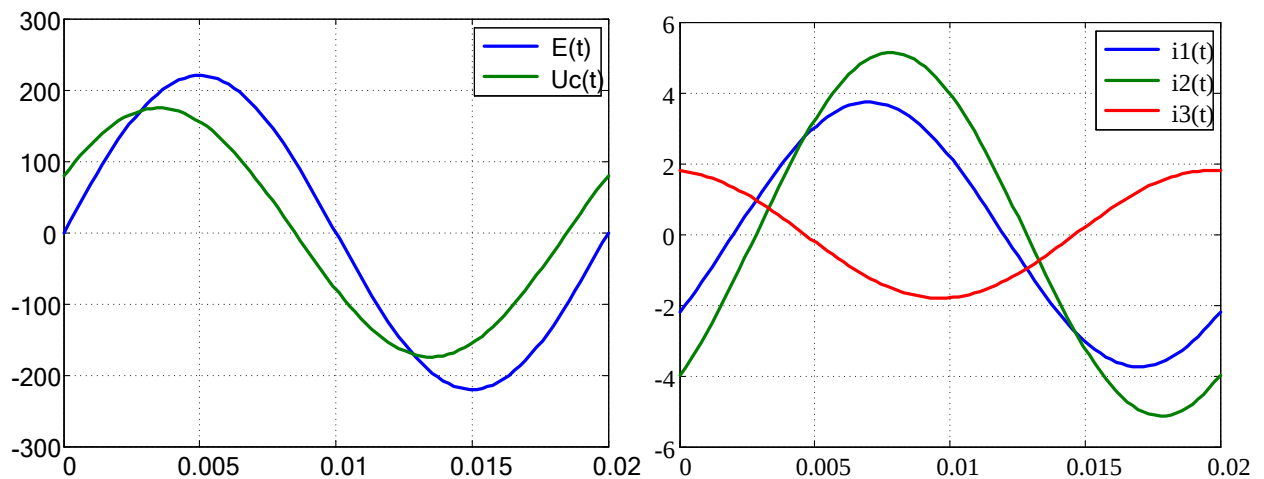


Рис. 5.3. Усталені перехідні процеси у схемі рис. 5.2.

5.4. Завдання

1. Розрахувати струми і напруги в гілках електричних кіл, наданих на рис. 5.1. Параметри схем надані в табл. 5.1. Дані щодо вибору однієї з електричних схем для кожного варіанту наведені в табл. 5.2.
2. Для схем, наведених на рис. 5.2, розрахувати і побудувати графіки усталених струмів і напруг гілок при $U(t) = U_m \sin(\omega t + \phi)$, $\omega = 2\pi f$. Параметри синусоїдальної напруги і реактивних елементів наведені у табл. 5.3.

За бажанням можна змінювати запропоновані у завданні електричні схеми та їх параметри.

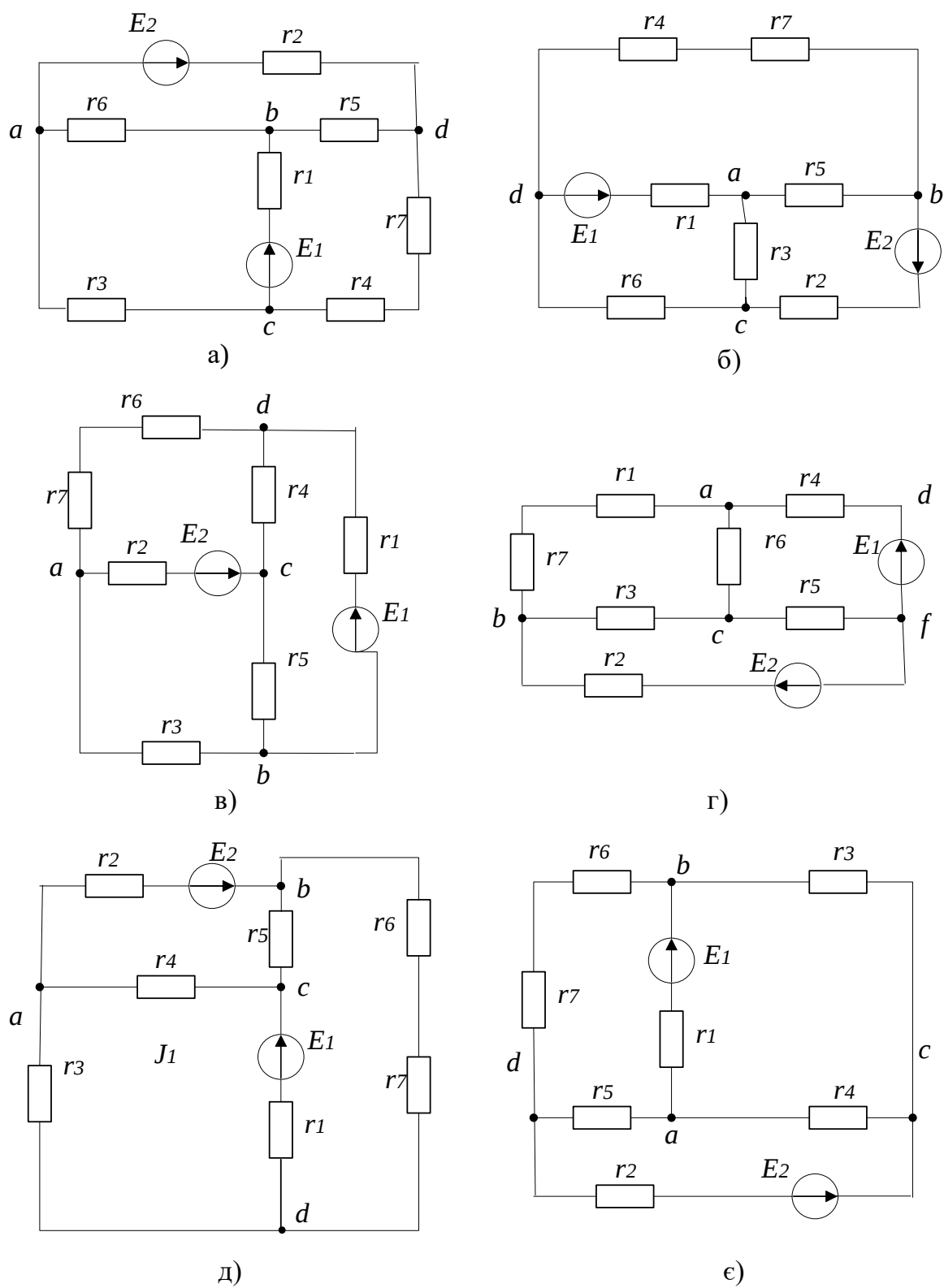


Рис. 5.3. Електричні кола постійного струму для завдання 1

Таблиця 5.1

№ вар.	E_1 , В	E_2 , В	r_1 , Ом	r_2 , Ом	r_3 , Ом	r_4 , Ом	r_5 , Ом	r_6 , Ом	r_7 , Ом
1-6	50	-	8	10	10	4	6	20	4
7-12	-	10	8	14	20	12	40	12	6
13-18	40	-	10	5	8	12	6	15	5
19-24	-	40	12	10	16	6	14	16	4
25-30	-	20	8	4	5	8	10	6	4

Таблиця 5.2

Схема	а	б	в	г	д	е
Варіант	1, 7, 13, 19, 24	2, 8, 14, 20, 25	3, 9, 15, 21, 26	4, 10, 16, 22, 27	5, 11, 17, 23, 29	6, 12, 18, 24, 30

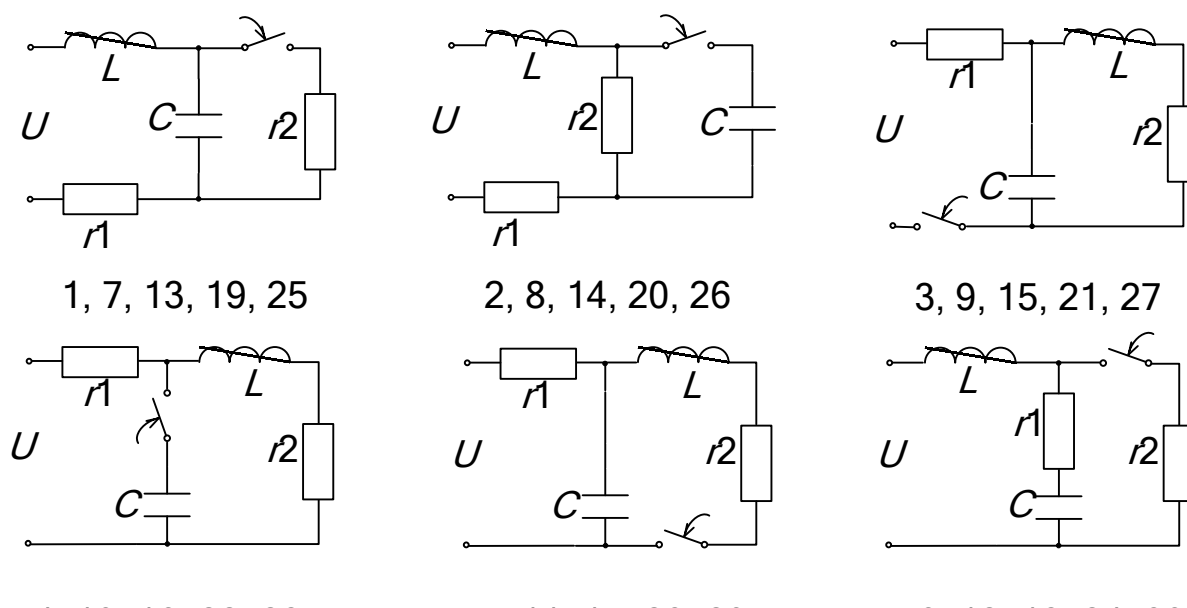


Рис. 5.2. Електричні кола змінного струму для завдання 2

Таблиця 5.3

№ вар.	E_m , В	f , Гц	φ , град	L , Гн	C , мкФ
1-6	50	25	15	0.4	80
7-12	100	10	30	1	100
13-18	150	40	45	0.6	200
19-24	200	50	60	2	600
25-30	250	20	90	0.5	400

5.5. Порядок виконання завдання

1. Намалюйте задані електричні кола, позначте стрілками напрями струмів у гілках.
2. Позначте на схемах вузли, підрахуйте їх кількість та оберіть з них незалежні вузли, кількість яких на одиницю менше від загальної кількості.
3. Складіть рівняння за першим законом Кірхгофа для незалежних вузлів.
4. Оберіть на схемах незалежні контури так, щоб сума кількості незалежних вузлів та незалежних контурів дорівнювала кількості гілок розгалуженого кола; позначте умовні напрями напруг в незалежних контурах.
5. Складіть рівняння за другим законом Кірхгофа для незалежних контурів.
6. Об'єднайте рівняння, складені за законами Кірхгофа в одну систему рівнянь і формалізуйте її запис за зразком 5.2.
7. Розробіть програмні модулі для розв'язання завдань 1 і 2, в яких визначте вхідні дані, розрахуйте матриці коефіцієнтів і вектори вільних членів, знайдіть розв'язки систем лінійних рівнянь та перевірте баланс потужностей.
8. Для кола змінного струму перетворіть усталені комплексні значення струмів і напруг з алгебраїчної форми в експоненціальну і побудуйте графіки одного періоду синусоїдальних струмів і напруг в усталеному режимі.

5.6 Методичні рекомендації

При виконанні завдань скористуйтеся прикладами, наведеними в пунктах 5.2 та 5.3.

Практичне заняття № 6

АПРОКСИМАЦІЯ ТА ІНТЕРПОЛЮВАННЯ ТАБЛИЧНИХ ФУНКЦІЙ СТЕПЕНЕВИМИ ПОЛІНОМАМИ

6.1. Довідкова інформація

В ЕОМ інформацію про табличні функції $y_i(x_i)$ зберігають у вигляді масивів даних

$$\mathbf{X} = [x_1 \ x_2 \ \dots \ x_n], \quad \mathbf{Y} = [y_1 \ y_2 \ \dots \ y_n]. \quad (6.1)$$

Табличні значення аргументів x_i називають **вузлами таблиці**, а відповідні значення функції y_i – **значеннями функції у вузлах**.

Оскільки будь-яка таблична функція $y_i(x_i)$ за своїм характером дискретна, то виникає **задача розрахунку значення такої функції в довільній точці інтервалу її існування, яка в загальному випадку не співпадає з вузлом таблиці**. Така задача може бути розв’язана шляхом **заміни табличної функції деякою аналітичною функцією, достатньо близькою до вихідної табличної**. Таку аналітичну функцію $y_a(x)$ називають **апроксимуючою**. **Апроксимація** (від лат. *approximare* – **наближатися**) – це приблизне вираження будь-яких залежностей через інші, звичайно більш простіші.

Апроксимація методом найменших квадратів (МНК) забезпечує мінімальну суму квадратів відхилень апроксимуючої функції від табличної у вузлових точках:

$$\Phi = \sum_{i=1}^n (y_a(x_i) - y_i)^2. \quad (6.2)$$

Апроксимуюча функція, що відповідає умові

$$y_a(x_i) = y_i, \quad i = 1, 2, \dots, n. \quad (6.3)$$

називається **інтерполюючою**. Процес пошуку такої функції на заданому інтервалі зміни аргументу $[x_0; x_n]$ називають **інтерполяцією** (від лат. *inter* – **внутри**, *pole* – **узел**), а зовні цього інтервалу – **екстраполяцією** (від лат. *extra* – **зовні**).

Для неперіодичних функцій найважливішим класом апроксимуючих функцій є *алгебраїчні степеневі поліноми*.

Розрізняють *глобальну і локальну інтерполяцію*. *Глобальний поліноміальний інтерполянт (англ. global fit) уявляє собою степеневий поліном, порядок якого є на одиницю меншим за кількість вузлів табличної функції з такими коефіцієнтами, що задовольняють умовам:*

$$P_{n-1}(x_i) = C_0 x_i^{n-1} + C_1 x_i^{n-2} + \dots + C_{n-2} x + C_{n-1} = y_i, \quad i=1, 2, \dots, n. \quad (6.4)$$

Система лінійних рівнянь (5.4) має рішення, при тому тільки одне. При зменшенні порядку полінома система стає недовизначеною, а при збільшенні порядку – перевизначеною. В обох випадках вона немає розв'язку.

При розв'язанні інженерних задач звичайно використовують *інтерполяцію рухомими поліномами* невисокого порядку (зазвичай від 0-го до 3-го), коли задану нелінійну залежність замінюють сукупністю сполучених між собою лінійних відрізків, квадратичних або кубічних парабол. *Таку інтерполяцію ще називають кусковою, локальною, або кусково-поліноміальною*. При застосуванні рухомих поліномів 0-го порядку отримують *ступінчасту апроксимацію*, а при застосуванні поліномів 1-го порядку – *кусово-лінійну апроксимацію*.

Задача локальної інтерполяції розбивається на 2 етапи: 1) пошук початкового вузла інтерполяції; 2) конструювання локального полінома за обраною інтерполяційною формулою. Недоліком локальної апроксимації є наявність зламів (розривів похідних) апроксимуючої функції у вузлових точках.

Цього недоліку позбавлений метод *інтерполяції кубічними сплайнами*, при застосуванні якого не тільки сусідні кубічні параболи, а ще й їх перші та другі похідні з'єднуються між собою неперервно.

Апроксимація табличної функції степеневими поліномами методом найменших квадратів полягає у визначенні коефіцієнтів степеневого поліному

$$y_a(x) = P_k(x) = c_0 + c_1x + c_2x^2 + \dots + c_kx^k = \sum_{j=0}^k c_j x^j, \quad (6.5)$$

що забезпечують мінімізацію функціоналу

$$\Phi = \sum_{i=1}^n (P_k(x_i) - y_i)^2 = \sum_{i=1}^n \left(\sum_{j=0}^k c_j x_i^j - y_i \right)^2. \quad (6.6)$$

Порядок апроксимуючого поліному повинен відповідати умові

$$k \leq n-1. \quad (6.7)$$

Для визначення потрібного вектору коефіцієнтів $\mathbf{c} = [c_0 \ c_1 \ c_2 \ \dots \ c_k]$ необхідно розв'язати систему лінійних рівнянь $(k+1)$ -го порядку:

$$\frac{\partial \Phi}{\partial c_0} = 0; \quad \frac{\partial \Phi}{\partial c_1} = 0; \quad \dots; \quad \frac{\partial \Phi}{\partial c_k} = 0, \quad (6.8)$$

яка після перетворень набуває вигляду:

$$\begin{cases} c_0 n + c_1 \sum_{i=1}^n x_i + c_2 \sum_{i=1}^n x_i^2 + \dots + c_k \sum_{i=1}^n x_i^k = \sum_{i=1}^n y_i; \\ c_0 \sum_{i=1}^n x_i + c_1 \sum_{i=1}^n x_i^2 + c_2 \sum_{i=1}^n x_i^3 + \dots + c_k \sum_{i=1}^n x_i^{k+1} = \sum_{i=1}^n y_i x_i; \\ \dots \\ c_0 \sum_{i=1}^n x_i^k + c_1 \sum_{i=1}^n x_i^{k+1} + c_2 \sum_{i=1}^n x_i^{k+2} + \dots + c_k \sum_{i=1}^n x_i^{2k} = \sum_{i=1}^n y_i x_i^k. \end{cases} \quad (6.9)$$

В *MATLAB* для знаходження коефіцієнтів апроксимуючого поліному методом найменших квадратів використовується функція *polyfit*:

$$\mathbf{C} = \text{polyfit}(\mathbf{Xt}, \mathbf{Yt}, k),$$

де $\mathbf{C}$ – вектор-рядок коефіцієнтів апроксимуючого полінома, упорядкований за зменшенням степені x ;

$\mathbf{Xt}$, $\mathbf{Yt}$ – вектори аргументів та значень табличної функції;

k – порядок апроксимуючого полінома.

Задача інтерполяції полягає у формуванні поліноміальної функції k -ої степені, яка приймає на деякому інтервалі $[x_j, x_{j+k}]$ значення, що співпадають

зі значеннями табличної функції $y_i(x_i)$ ($i=j, j+1, \dots, j+k$) у $(k+1)$ -й вузлових точках:

$$P_k(x_j)=y_j, P_k(x_{j+1})=y_{j+1}, \dots, P_k(x_{j+k})=y_{j+k} \quad (6.10)$$

та в розрахунку значень знайденої функції у точках, відмінних від вузлових ($x \neq x_j$).

Точку x_j звать початковим вузлом інтерполяції. Якщо $i=0,1,2,\dots,n$, а $k=n$, то знайдена функція є **глобальним інтерполантом**, бо в цьому випадку його значення співпадають зі значеннями початкової функції в усіх вузлах:

$$P_n(x_i)=a_0+a_1x_i+a_2x_i^2+\dots+a_nx_i^n=\sum_{j=0}^na_jx_i^j, \quad i=0,1,2,\dots,n. \quad (6.11)$$

При локальній інтерполяції коефіцієнти поліномів залишаються невідомими, а формування поліномів відбувається за інтерполяційними формулами, в яких вихідними даними є координати вузлових точок табличної функції.

Інтерполяційна формула Лагранжа:

$$\begin{aligned} y(x) &\approx L_k(x) = \\ &= \sum_{m=j}^{j+k} y_m \frac{(x-x_j)(x-x_{j+1})\dots(x-x_{m-1})(x-x_{m+1})\dots(x-x_{j+k})}{(x_m-x_j)(x_m-x_{j+1})\dots(x_m-x_{m-1})(x_m-x_{m+1})\dots(x_m-x_{j+k})} \end{aligned} \quad (6.12)$$

Для інтерполювання в пакеті MATLAB використовується функція `interp1`:

$$Y=\text{interp1}(Xt, Yt, X, \text{method}),$$

де Xt, Yt – вектори аргументів та значень табличної функції;

X – точка або масив точок, у яких необхідно обчислити значення інтерпольованої функції;

`method` – метод інтерполювання, може приймати наступні значення:

- 'linear' – лінійна інтерполяція (кусково-лінійна апроксимація, інтерполяція рухомими поліномами 1-го порядку); використовується за замовчанням;
- 'pchip', 'cubic' – кубічна інтерполяція (кусково-кубічна апроксимація);

- 'spline' – кубічна сплайн-інтерполяція;
- 'nearest' – інтерполяція за найближчим сусіднім вузлом (*nearest neighbour*), або інтерполяція рухомими поліномами 0-го порядку (ступінчата апроксимація).

Щоб включити *режим екстраполяції*, треба додати до списку вхідних параметрів в кінці ще один параметр: 'extrap'.

У Simulink кусково-лінійну інтерполяцію табличної функції за формулою (11.18) виконує блок **1D-Look-Up Table (Одномірний Функціональний Перетворювач)** з параметрами *Breakpoints* (табличні значення вхідного сигналу) та *Table data* (табличні значення вихідного сигналу). Іконка блоку **1D-Look-Up Table** та вікно введення його параметрів зображені на рис. 11.8 і 11.9 відповідно.

6.2. Завдання

1. Апроксимувати табличні функції, що приведені в табл. 6.1, степеневими поліномами k -го порядку методом найменших квадратів ($k=2, 3, 4, n-1$). Обчислити значення функціоналу Φ , що підлягає мінімізації. Проілюструвати результати графіками.
2. Розрахувати приблизні значення табличних функцій, які завдані у табл. 6.1, для аргументів, котрі змінюються від x_{min} до x_{max} в 100 точках інтерполяційними методами і побудувати їх графіки.

6.3. Методичні рекомендації

1. Апроксимуючий степеневий поліном (СП) виводьте на екран у вигляді рядків символів.
2. Зміну порядку СП виконуйте у циклі. По завершенні циклу виведіть на екран у двох стовпцях або у двох рядках вектори порядків СП і відповідних значень мінімізованих функціоналів.

3. Для порівняння результатів інтерполяції та екстраполяції дещо розширите діапазон зміни аргументу інтерполянтів в обидві сторони порівняно з діапазоном зміни аргументу табличної функції.

Таблиця 6.1

№ вар.	Табличні функції									
1,2	x_t	-1	1	3	5	7	9	11	13	15
	y_t	8.71	109,8	124.4	122.5	112.1	96.6	80.2	63	57.9
3,4	x_t	2	3.2	4.4	6.2	7.8	9.5	10.9	11.5	12.7
	y_t	19.9	22	30	42.1	65	99.5	120	126.8	133.4
5,6	x_t	-3.5	-1.5	0.5	2.5	4.5	6.5	8.5	10.5	12.5
	y_t	0.45	-3.09	-4.01	-3.9	-3	-1.62	-0.18	0.99	1.72
7,8	x_t	1.25	2.59	4.4	6.54	8.5	11.5	13.5	14.5	15
	y_t	3.0	5.0	7.0	8.5	9.3	9.9	10.6	11.2	11.64
9,10	x_t	-2	0	2	4	6	8	10	12	14
	y_t	7.84	7.13	6.31	5.29	4.03	2.5	0.87	-0.68	-0.79
11,12	x_t	-1.5	1	2.7	5.5	6.5	8.3	9.6	11.2	12.75
	y_t	2.45	1.12	-1	-2.1	-2.3	-1.9	-1	2	3.5
13,14	x_t	0,67	1,5	2,5	3,5	5	6,5	10	12,4	14
	y_t	110	118,7	124,5	125,2	122,5	115,1	88,3	70	61,2
15,16	x_t	0,5	2,5	4,5	6,5	8,5	10,5	12,5	14,5	16,5
	y_t	23,7	20,1	27,8	45,3	79,2	115,4	132,9	141,1	147
17,18	x_t	-2,77	-0,5	1	2	3,5	7	10	11,5	12,5
	y_t	-1,5	-3,65	-4,03	-4,0	-3,54	-1,58	0,73	1,4	1,83
19,20	x_t	0,5	2,0	3,5	5,0	6,5	8,5	9,5	11,0	12,5
	y_t	1,23	0,92	0,78	0,68	0,6	0,53	0,49	0,47	0,45
21,22	x_t	-1	1	3	5	7	9	11	13	15
	y_t	1,02	2,57	5,51	7,52	8,69	9,38	9,79	10,35	11,64
23,24	x_t	-3	0,5	1,5	2,5	4,3	6,2	7,7	9,0	11
	y_t	9,4	7,52	6,75	5,8	3,6	0,53	-1,5	-2,94	-4,4
25,26	x_t	-4	-2	0	2	4	6	8	10	12
	y_t	3,1	2,66	1,74	0,35	-1,26	-2,28	-2,07	-0,54	2,53

4. При візуалізації результатів апроксимації та інтерполювання виведіть на екран у графічному режимі табличну функцію і її наближення у різній формі чи різним кольором. Наприклад, табличну функцію можна вивести на екран у вигляді “кружечків” або функцією `stem`, а інтерпольовані значення – у вигляді відрізків прямих між двома сусідніми точками.
5. Вибір метода інтерполювання організуйте у вигляді меню. Для цього використайте функцію `menu`.

Практичне заняття № 7

ГАРМОНІЧНИЙ АНАЛІЗ І СИНТЕЗ ПЕРІОДИЧНИХ ФУНКЦІЙ

7.1. Довідкова інформація

Функція часу $f(t)$ називається періодичною, якщо для неї справедливі умови:

$$f(t) = f(t + mT), \quad m = 1, 2, 3, \dots, \quad (7.1)$$

де T – період.

Розкладання періодичних функцій в ряд Фур'є:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(\omega k t) + b_k \sin(\omega k t)) = \frac{a_0}{2} + A_k \sum_{k=1}^{\infty} \cos(\omega k t + \varphi_k), \quad (7.2)$$

де k – порядковий номер гармоніки;

кругова частота першої гармоніки:

$$\omega = 2\pi/T;$$

коефіцієнти ряду Фур'є:

$$a_0 = \frac{2}{T} \int_0^T f(t) dt, \quad a_k = \frac{2}{T} \int_0^T f(t) \cos(\omega k t) dt, \quad b_k = \frac{2}{T} \int_0^T f(t) \sin(\omega k t) dt; \quad (7.3)$$

амплітуди і фази гармонік:

$$A_k = \sqrt{a_k^2 + b_k^2}, \quad \varphi_k = \arctg \frac{b_k}{a_k} \quad (7.4)$$

Апроксимуюча полігармонічна функція (ряд Фур'є з обмеженою кількістю гармонік m):

$$Q_m(t) \approx \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos(\omega k t) + b_k \sin(\omega k t)) = \frac{a_0}{2} + A_k \sum_{k=1}^m \cos(\omega k t + \varphi_k). \quad (7.5)$$

Визначення абсцис та ординат табличної періодичної функції:

$$t_i = i \cdot \Delta t, \quad \Delta t = T/n, \quad y_i = f(t_i), \quad i = 0, 1, 2, \dots, n, \quad (7.6)$$

де n – кількість відрізків ділення періоду (на 1 менше кількості значень табличної функції).

Приблизні значення коефіцієнтів тригонометричного полінома (7.5),
 знайдені при чисельному інтегруванні виразів (7.3) методом прямокутників:

$$\begin{cases} a_0 \approx \frac{2}{n} \sum_{i=0}^{n-1} y_i, \\ a_k \approx \frac{2}{n} \sum_{i=0}^{n-1} y_i \cos \frac{2\pi k i}{n}, \\ b_k \approx \frac{2}{n} \sum_{i=0}^{n-1} y_i \sin \frac{2\pi k i}{n}. \end{cases} \quad (7.7)$$

Визначення коефіцієнтів a_0, a_k, b_k тригонометричного полінома та/або амплітуд і фаз A_k, φ_k називають **гармонічним аналізом** періодичних функцій, а розрахунок тригонометричного поліному (7.5) у заданих точках при відомих коефіцієнтах окремих гармонік називають **гармонічний синтезом**.

Множину абсолютних величин амплітуд A_k гармонічних складових періодичних полігармонічних сигналів називають **спектром амплітуд**, сукупність фаз φ_k — **спектром фаз**, а множину величин A_k^2 — **спектром потужності**. Аналіз частотних спектрів сигналів називають **спектральним аналізом**.

В обчислювальній математиці доведено, що функція $Q_m(t)$ з коефіцієнтами (7.7) апроксимує вихідну періодичну функцію (7.1) **методом найменших квадратів**.

При $n=2m$ функція $Q_m(t)$ стає **тригонометричним інтерполянтом**, тобто для того, щоб тригонометрична функція у вузлових точках мала такі ж значення, як і вихідна таблична функція, **треба мати, як мінімум 2 точки табличної функції на 1 гармоніку**.

Приклад m-функції, що виконує гармонічний аналіз за формулами (7.7):

```
function [a0, a, b] = k_fur (y, m);
% гармонічний аналіз:
y(end) = []; n = length(y);
a0 = 2/n*sum(y);
for k = 1:m
    Sa = sum(y.*cos(2*pi*k*(0:n-1)/n));
    Sb = sum(y.*sin(2*pi*k*(0:n-1)/n));
end
```

```

a = 2/n*Sa;
b = 2/n*Sb
end

```

Приклад m-функції, що виконує гармонічний синтез за формулою (7.5):

```

function Q = f_fur (t, T, a0, a, b);
% гармонічний синтез:
f=2*pi/T; % частота основной гармоники
m = length (a); % кол-во гармоник
n = length (t);
for l = 1:n
    Q(l) = a0/2+sum (a.*cos(f*[1:m]*t(l)) + b.*sin(f*[1:m]*t(l)));
end
end

```

7.2. Завдання

1. Розрахувати коефіцієнти тригонометричних поліномів, які апроксимують табличну функцію із табл. 7.1 методом найменших квадратів, при різній кількості гармонік.
2. Розрахувати значення знайдених тригонометричних поліномів у вузлах таблиці та значення мінімізованих функціоналів.
3. Розрахувати значення знайдених тригонометричних поліномів у 100-200 рівномірно розташованих на періоді точках і побудувати графіки апроксимуючих полігармонічних функцій; зобразити у тому ж вікні вихідну табличну функцію
4. Побудувати в одному вікні графіки мінімальної кількості складових гармонік, що роблять апроксимуючу функцію глобальним інтерполянтом.
5. Побудувати графіки амплітудних і фазових спектрів, а також графік спектру потужності досліджуваної періодичної функції.

Таблиця 7.1

№ вар.	Табличні значення функції																					
1	2																					
1	1,00	1,803	3,085	4,776	6,434	7,347	7,027	5,652	3,897	2,381	1,347	0,422	0,419	0,256	0,176	0,142	0,136	0,155	0,209	0,324	0,554	1,00
2	7,38	6,76	5,22	3,47	2,07	1,16	0,64	0,36	0,23	0,16	0,13	0,13	0,16	0,23	0,37	0,64	1,16	2,08	3,48	5,22	6,76	7,38
3	-1,24	-1,17	-1,08	-0,96	-0,84	-0,79	-0,8	-0,9	-1,1	-1,21	-1,02	-1,28	-1,32	-1,34	-1,36	-1,37	-1,37	-1,36	-1,35	-1,33	-1,30	-1,24
4	1,0	1,05	90,6	520,4	1714,7	2915,0	2439,2	1020,6	230,7	32,17	3,29	0,3	0,03	0,004	0,001	0,0003	0,0006	0,002	0,01	0,09	0,9	1,0
5	2980,1	2089,3	742,4	146,6	18,6	1,8	0,16	0,02	0,003	0,001	0,001	0,001	0,002	0,003	0,018	0,9	1,22	18,6	146,6	742,5	2089,7	2980,1
6	1	1,34	1,75	2,18	2,53	2,71	2,65	2,37	1,97	1,54	1,16	0,86	0,64	0,5	0,42	0,37	0,36	0,39	0,45	0,56	0,74	1
7	2,71	2,6	2,28	1,86	1,44	1,07	0,8	0,46	0,42	0,4	0,37	0,37	0,4	0,48	0,6	0,8	1,07	1,44	1,86	2,28	2,6	2,71
8	-1,32	-1,28	-1,26	-1,24	-1,25	-1,25	-1,25	-1,26	-1,27	-1,29	-1,29	-1,33	-1,34	-1,37	-1,37	-1,37	-1,37	-1,36	-1,34	-1,3	-1,2	-1,1
9	-4,0	-4,2	-4,5	-4,7	-4,9	-5,0	-4,9	-4,8	-4,6	-4,4	-4,1	-3,8	-3,5	-3,1	-3,0	-3,0	-3,0	-3,1	-3,2	-3,4	-3,7	-4,0
10	1,0	2,4	5,4	10,4	16,3	19,9	18,6	13,4	7,7	3,6	1,6	0,64	0,27	0,13	0,07	0,05	0,05	0,06	0,09	0,18	0,4	1,0
11	20,0	17,5	11,9	6,4	2,9	1,2	0,5	0,2	0,1	0,06	0,05	0,05	0,06	0,1	0,5	1,0	1,2	2,9	6,4	11,9	17,5	20,0
12	-1,1	-0,8	-0,3	0,3	0,7	0,8	0,7	0,5	0,04	-0,6	-0,9	-1,1	-1,27	-1,32	-1,35	-1,37	-1,37	-1,36	-1,34	-1,3	-1,2	-1,1
13	-2,0	-2,8	-3,7	-4,3	-4,7	-4,9	-4,9	-4,5	-4,1	-3,3	-2,4	-1,5	-0,6	-0,04	0,6	0,7	0,99	0,79	0,34	0,3	-1,1	-2,0
14	1,1	3,2	9,5	22,8	41,4	53,9	49,4	31,9	15,2	5,7	1,8	0,55	0,17	0,06	0,03	0,02	0,01	0,02	0,04	0,1	0,3	1,1
15	-0,78	-1,22	-1,34	-1,39	-1,42	-1,43	-1,42	-1,41	-1,37	-1,3	-1,1	-0,1	1,1	1,2	1,33	1,36	1,37	1,35	1,3	1,17	0,65	-0,78

Продовження таблиці 7.1

1	2
16	54,5 45,7 21,2 12,1 4,3 1,3 0,4 0,13 0,05 0,03 0,02 0,02 0,03 0,05 0,13 0,41 1,3 4,3 12,1 21,2 45,7 54,5
17	-0,78 0,18 0,89 1,13 1,21 1,24 1,23 1,18 1,04 0,63 -0,38 -1,01 -1,22 -1,3 -1,35 -1,36 -1,37 -1,36 -1,33 -1,27 -1,1 -0,78
18	-1,0 -2,1 -3,2 -4,1 -4,7 -4,9 -4,8 -4,4 -3,7 -2,7 -1,6 -0,4 0,7 1,7 2,4 2,9 3,0 2,7 2,1 1,2 0,2 -1,0
19	1,0 4,36 16,7 49,8 105,0 146,3 130,9 75,9 30,0 8,75 2,1 0,47 0,11 0,03 0,01 0,007 0,006 0,009 0,02 0,05 0,2 1,0
20	148,4 118,8 62,6 22,5 6,21 1,45 0,33 0,08 0,02 0,01 0,007 0,007 0,01 0,02 0,08 0,32 1,45 6,2 22,6 62,2 119,0 148,4
21	0,0 0,97 1,23 1,32 1,36 1,37 1,36 1,34 1,28 1,13 0,64 -0,64 -1,13 -1,28 -1,34 -1,37 -1,36 -1,32 -1,23 -0,9 -0,2 0,0
22	-0,0001 -1,47 -2,8 -3,9 -4,65 -4,98 -4,87 -4,33 -3,4 -2,16 -0,74 0,74 2,17 3,14 4,33 4,87 4,98 4,65 3,9 2,8 1,4 -0,0001
23	1,0 5,8 29,3 108,9 266,4 396,7 347,1 180,5 59,2 13,5 2,4 0,4 0,07 0,01 0,005 0,003 0,002 0,004 0,009 0,03 0,1 1,0
24	403,4 309,0 142,2 42,1 8,9 1,56 0,26 0,05 0,01 0,0044 0,0026 0,0026 0,0044 0,01 0,05 0,263 1,56 8,95 42,1 142,2 309,9 403,4
25	0,78 1,22 1,34 1,39 1,42 1,43 1,42 1,41 1,37 1,3 1,1 0,1 -1,1 -1,2 -1,33 -1,36 -1,37 -1,35 -1,3 -1,17 -0,65 0,78
26	1,0 -0,77 -2,3 -3,6 -4,6 -4,9 -4,8 -4,1 -3,1 -1,6 0,1 1,9 3,6 5,1 6,2 6,84 6,98 6,58 5,69 4,4 2,7 1,0
27	1,0 7,8 51,5 238,1 675,9 1075,4 920,1 429,3 110,8 20,8 2,83 0,35 0,04 0,01 0,002 0,001 0,001 0,001 0,001 0,004 0,02 0,12 1,0
28	1,10 1,32 1,40 1,43 1,45 1,46 1,46 1,44 1,42 1,37 1,25 0,76 -0,8 -1,22 -1,33 -1,36 -1,37 -1,35 -1,29 -1,1 -0,1 1,10
29	2,0 -0,06 -1,9 -3,4 -4,9 -4,8 -4,0 -2,7 -1,1 0,95 3,0 5,0 6,7 8,1 8,8 8,9 8,5 7,47 5,94 4,06 2,1 2,0

7.3 Порядок виконання завдання

1. Введіть задані значення табличної функції y_i ; визначить кількість інтервалів на періоді; задайте довільним періодом і розрахуйте вектор аргументів табличної функції.
2. Знайдіть мінімальну кількість гармонік m_g , які перетворюють апроксимуючу полігармонічну функцію у глобальний інтерполянт.
6. Для 3-х значень кількості гармонік (наприклад, $m=[2, 5, m_g]$) виконайте гармонічний аналіз досліджуваної табличної функції; синтезуйте її апроксимуючі та глобальну інтерполуючу функції;
7. Розрахуйте їх при табличних значеннях аргументів і розрахувати мінімізовані функціонали.
8. Розрахуйте апроксимуючі функції у 100 точках, рівномірно розташованих на періоді; побудуйте в одному вікні різними стилями і кольорами графіки заданої табличної функції і варіанти її апроксимацій;
9. Визначите амплітуди і фази гармонік при $m=m_g$ і побудуйте їх графіки.
10. Побудуйте графіки амплітудних і фазових спектрів, а також графік спектру потужності.

7.4 Методичні рекомендації

1. Результати гармонічного аналізу перевіряйте за допомогою гармонічного синтезу. Якщо синтезована періодична функція не збігається, з вихідною табличною функцією у вузлових точках при $n \geq 2m_g$, знайдіть та виправте помилки у відповідних програмних модулях.
2. Вихідну табличну функцію зображуйте як решітчасту (функція stem).
3. Для наочності щільність точок на графіку апроксимуючої функції повинна в 5-10 раз перевищувати щільність точок, що ділять період T на частки для чисельного інтегрування.

Практичне заняття № 8

ЧИСЕЛЬНЕ РІШЕННЯ ТРАНСЦЕНДЕНТНИХ ТА АЛГЕБРАЇЧНИХ РІВНЯНЬ

8.1. Довідкова інформація

Постановка задачі: Знайти приблизні значення коренів рівняння

$$F(x)=0 \tag{8.1}$$

де $F(x)$ – нелінійна функція однієї змінної загального вигляду.

Якщо ця функція є степеневим поліномом ($F(x)=P_n(x)$), то рівняння (8.1) стає *алгебраїчним рівнянням відповідного степеню*. У *трансцендентних рівняннях* функція $F(x)$ уявляє собою сукупність будь-яких математичних операцій над будь-якими функціями: степеневими, тригонометричними, експоненціальними, логарифмічними і т.п.

Числове розв’язання нелінійних рівнянь поділяється на два етапи: *відділення коренів* та *уточнення їх початкових наближень ітераційними методами*.

Відділення коренів полягає у пошуку початкового наближення шуканого кореня x_0 або інтервалу існування кореня $[a, b]$. на якому функція, нуль якої треба знайти, змінює знак:

$$F(a) \cdot F(b) < 0. \tag{8.2}$$

Уточнення коренів виконується *ітераційними методами*, тобто *методами поступових наближень*. При використанні ітераційних методів розв’язок знаходять приблизно, з *заданою точністю*, яка задається у вигляді *максимально припустимої абсолютної або відносної похибки*.

Найбільш розповсюдженими *методами уточнення коренів* є методи *бісекцій, хорд, дотичних та метод простих ітерацій*.

Метод бісекцій (метод діхотомії, метод половинного ділення) полягає у послідовного поділу відрізка, який містить корінь, навпіл:

$$x = \frac{a + b}{2}. \tag{8.3}$$

Для кожного наступного ділення обирається та половина відрізка, на кінцях якої функція має протилежний знак. При цьому інтервал існування кореня звужується за рахунок зміни однієї з його меж: лівої ($a=x$) або правої ($b=x$).

Ітераційний процес закінчується, якщо виконується умова:

$$b - a \leq \varepsilon_x, \quad (8.4)$$

де ε_x – задана точність обчислення кореня ($\varepsilon_x \ll 1$). Часто вимагають, щоб одночасно виконувалася умова:

$$|F(x)| \leq \varepsilon_y. \quad (8.5)$$

Якщо на відрізку $[a,b]$ знаходиться кілька коренів, то процес збігається до одного з них. Метод непридатний для пошуку кратних коренів парного порядку.

Метод хорд (метод пропорційних відрізків) полягає у послідовному поділенні відрізка $[a,b]$, який має корінь, на частини, пропорційні значенням функції на кінцях відрізка:

$$x = \frac{aF(b) - bF(a)}{F(b) - F(a)}. \quad (8.6)$$

Геометрично це еквівалентно заміні графіка функції $F(x)$ хордою, яка пройде через точки $(a, F(a))$ та $(b, F(b))$.

Для закінчення ітераційного процесу замість умови (8.4) використовують умову:

$$|x_{i+1} - x_i| \leq \varepsilon, \quad (8.7)$$

де x_{i+1} , x_i – відповідно, останнє обчислене і попереднє до нього наближення кореня.

Розв'язання трансцендентних та алгебраїчних рівнянь у середовищі пакета MATLAB:

$$x = \text{fzero}(\text{Fun}, x_0)$$

де *Fun* – або ідентифікатор функції (*function handle*), що утворюється за форматом *@FunName*, або ім'я функції у вигляді рядка символів '*FunName*', або звернення до *inline*-функції у вигляді *inline ('expr')*, або так звана анонімна функція (*anonymous function*) у форматі *@(x) expr*; *x0* – початкове наближення кореня або діапазон існування коренів (*[a b]*), на якому функція хоча б один раз змінює знак.

Ця функція знаходить єдиний нуль заданої функції в околі заданого початкового наближення або на заданому інтервалі.

Щоб мати можливість втручатися у процес рішення, змінюючи алгоритм, точність розв'язку, максимальну кількість ітерацій, тощо, список вхідних параметрів можна доповнити параметром *options*:

$$x = fzero (Fun, x0, Options)$$

елементи якого треба попередньо визначити за допомогою функції *optimset*:

$$options = optimset ('param1', value1, 'param2', value2, ...)$$

Щоб визначитися з іменами параметрів, які можна змінювати, їх можливими значеннями та значеннями, що призначаються «за замовченням» (*default*), треба звернутися до функції *optimset* без вхідних та вихідних параметрів. Значення за замовчанням подаються у круглих дужках.

Щоб прослідкувати процес збіжності розв'язку, тобто побачити результати послідовності ітерацій, слід встановити параметр *Display* у стан *Iter*, наприклад,

$$x = fzero (Fun, x0, optimset ('Display', 'Iter'))$$

Кількість вихідних параметрів функції можна збільшити:

$$[x, Fval, ExitFlag, Output] = fzero (Fun, x0, Options),$$

де

Fval – дійсне значення функції, яке буде близьким до нуля але не нульовим;

ExitFlag – діагностичний параметр, який може приймати такі значення:

- 1 – «нуль» функції знайдено з заданою точністю;
- -1 – при заданій кількості ітерацій задана точність не досягнута;

- -3 – при пошуку інтервалу існування нуля функція отримала значення NaN або Inf;
- -4 – при пошуку інтервалу існування нуля функція отримала комплексне значення;
- -5 – процес пошуку нуля не збігається в одну точку;
- -6 – не знайдено інтервал, на якому функція змінює знак;

Output – параметр, що керує виведенням додаткової інформації: використані чисельні методи пошуку нуля, кількість ітерацій, кількість обчислень значень функцій.

Рішення алгебраїчних рівнянь в пакеті *MATLAB*:

$$X = \text{roots}(P)$$

де P – вектор коефіцієнтів степеневого полінома; X – вектор коренів алгебраїчного рівняння (вектор нулів полінома).

8.2. Завдання

1. Розрахувати перший додатний корінь трансцендентного рівняння з точністю 10^{-3} і 10^{-5} за допомогою функції *fzero* та власної функції, що знаходить корені нелінійного рівняння методом, заданим у табл. 8.1. Порівняти результати та кількість ітерацій.
2. Розрахувати усі дійсні та комплексно спряжені корені алгебраїчного рівняння з табл. 7.1 за допомогою функції *roots*. Виконати перевірку точності розв'язків за допомогою функції *poly*. Зобразити знайдені корені на комплексній площині.

8.3 Порядок виконання завдання

1. Складіть програму для розрахунку значень заданої нелінійної функції у певному діапазоні зміни аргументу та побудови її графіка.
2. Методом проб і помилок знайдіть такий діапазон зміни аргументу, в якому на графіках чітко просліджується перетин або торкання функцією осі абсцис.

3. Складіть програму, що уточнює значення дійсних коренів за допомогою Вашої власної функції та стандартної *MATLAB*-функції *fzero*.
4. Апробуйте можливості функцій *fzero* та *optimset* і різні способи визначення функції користувача, що входить у параметри функції *fzero*.
5. Складіть блок-схему алгоритму та функцію користувача для уточнення коренів нелінійного рівняння заданим в табл. 8.1 методом.
6. Складіть програму, що уточнює значення дійсних коренів заданим методом за допомогою Вашої власної функції. Порівняйте результати, точність розв'язків та кількість ітерацій при використанні функції *fzero* та функції, складеної Вами.
7. Знайдіть корені алгебраїчного рівняння за допомогою функції *roots*. Для оцінювання точності отриманих результатів розрахуйте значення заданої поліноміальної функції при знайдених значеннях її нулів за допомогою функції *poly*.
8. Зобразите знайдені корені на комплексній площині за допомогою функцій *pzmap (X)* та/або *roots (real (X), imag (X), 'x')*.

Таблиця 8.1

№ вар.	$F(x)$		Метод рішення
1	2	3	4
1	$2 \cdot \sin\left(x + \frac{\pi}{3}\right) - 0,5 \cdot x^2 + 1 = 0$	$4,2x^3 - 31,92x^2 + 74,3x - 51,87$	Бісекцій
2	$\cos(x + 0,3) - x^2 = 0$	$3,6x^3 - 172,8x^2 + 5,184x - 237,32$	Хорд
3	$\sin^3(x) - x + 1 = 0$	$5,8x^3 - 47,56x^2 + 121,2x - 97,02$	Бісекцій
4	$2 \arctg(x) - x + 3 = 0$	$6,1x^3 - 90,28x^2 + 388,2x - 506,2$	Хорд
5	$(x + 3)\cos(x) - 1 = 0$	$3,6x^3 - 39,96x^2 + 12,17x + 426,4$	Бісекцій
6	$0.25 + \arctg(0,58x + 0,1) - x^2 = 0$	$2.7x^3 - 37,26x^2 + 16,71x - 202,7$	Хорд
7	$\ln(x) - \frac{7}{2x + 6} = 0$	$1,3x^3 - 5,98x^2 - 1,09x + 13,76$	Бісекцій

8	$\frac{1}{\cos(1,05-x)} - x^2 = 0$	$4,5x^3 - 26,1x^2 + 176,6x - 112,4$	Хорд
---	------------------------------------	-------------------------------------	------

Продовження табл. 8.1

1	2	3	4
9	$2\ln(x) - \frac{x}{2} + 1 = 0$	$5,1x^3 - 62,22x^2 + 142,7x + 109,2$	Бісекцій
10	$\ln(x) - \frac{1}{x^2} = 0$	$1,6x^3 - 3,04x^2 - 29,18x + 8,98$	Хорд
11	$4,3\sin(4x) - 3,5x = 0$	$-2,3x^3 + 0,23x^2 + 17,05x + 13,48$	Бісекцій
12	$2^x - 2^{x-2} - 1 = 0$	$1,6x^3 - 14,24x^2 + 38,13x - 29,02$	Хорд
13	$\cos(15,6x) + 0,5 = 0$	$5,3x^3 - 36,04x^2 + 12,25x + 28,05$	Бісекцій
14	$0,5^x + 1 - (x-2)^2 = 0$	$-2,6x^3 + 4,68x^2 + 14,38x + 3,822$	Хорд
15	$3^{x-1} - 2 - x = 0$	$-1,5x^3 - 14,25x^2 - 37,98x - 22,03$	Бісекцій
16	$x^2 \cos(2x) + 1 = 0$	$3,4x^3 - 46,58x^2 + 127,3x - 60,34$	Хорд
17	$x^2 - 2^{x-1} = 0$	$2,8x^3 - 25,76x^2 + 6,18x + 107,4$	Бісекцій
18	$5\sin(x) - x = 0$	$-1,4x^3 - 10,78x^2 - 22,54x - 11,85$	Хорд
19	$\operatorname{arctg}(x-1) + 2x = 0$	$3,1x^3 - 62,6x^2 + 414,7x - 898,9$	Бісекцій
20	$(x-2)^2 - 2^x = 0$	$1,6x^3 - 12,48x^2 + 25,04x - 8,12$	Хорд
21	$x^2 - 20\sin(x) = 0$	$5,4x^3 - 54x^2 + 140,6x - 73,8$	Бісекцій
22	$2e^x - 5x - 2 = 0$	$2,7x^3 - 17,6x^2 - 45,4x + 123$	Хорд
23	$\cos(x+0,5) - x^3 = 0$	$-1,8x^3 - 5,58x^2 + 1,5x + 119$	Бісекцій
24	$2\operatorname{arctg}(x) - 0,5x^3 = 0$	$-2,5x^3 + 8,25x^2 + 61,9x - 117$	Хорд
25	$e^{-x} + x^2 - 2 = 0$	$1,7x^3 - 16,43x^2 + 21,4x - 12$	Бісекцій
26	$e^x - e^{x+2} + 6 = 0$	$3,7x^3 - 17,9x^2 - 55,1x + 105$	Хорд
27	$\sin(0,5x) - x^2 = 0$	$-3,1x^3 + 9,55x^2 + 71x - 111$	Бісекцій
28	$\operatorname{arctg}(x) - \cos(2x)\sin(x) = 0$	$3,4x^3 - 46,58x^2 + 127,3x - 60,34$	Хорд
29	$e^{-x} - 0,5x^3 = 0$	$5,3x^3 - 36,04x^2 + 12,25x + 28,05$	Бісекцій
30	$\cos^2(x) - 3x^5 = 0$	$4,2x^3 - 31,92x^2 + 74,3x - 51,87$	Хорд

8.4 Методичні рекомендації

1. Начальне наближення першого додатного кореня x_0 та діапазону його існування $[a, b]$ визначте візуально, побудувавши графік функції $F(x)$.
2. Для оцінки швидкості збігу різних методів виведіть на екран послідовність наближень коренів та значення функції у цих точках.

ЛІТЕРАТУРА

1. Попович М.Г., Ковальчук О.В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К.: Либідь, 2007. – 656 с.
2. Phillips C., Harbor R. Feedback control systems, Prentice-Hall, 2000, 658 p.
3. Lutz H., Wendt W. Taschenbuch der Regelungstechnik. – Tun und Frankfurt am Main: Verlag Harry Deutsch, 2000. – 1148 S.
4. Ogata K. Modern control engineering, Prentice-Hall, 2010, 905 p.
5. Толочко О.І. Аналіз та синтез електромеханічних систем зі спостерігачами стану. – Донецьк: НОРД-ПРЕС, 2004. – 298 с.
6. Dingyü Xue, YangQuan Chen, Derek P. Atherton. Linear Feedback Control Analysis and Design with MATLAB. – Philadelphia: Siam, 2007. – 356 p.
7. Теорія автоматичного керування. Курсова робота: навчальний посібник / КПІ ім. Ігоря Сікорського; уклад.: О.І. Толочко, С.М. Пересада, Б.І. Приймак. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 163 с.
8. Фельдман Л.П., Петренко А.І., Дмитрієва О.А. Чисельні методи в інформатиці. – К.: Видавнича група BVH, 2006. – 480 с.
9. Задачин В.М., Конюшенко І.Г. Чисельні методи. Навчальний посібник. – Харків. Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с.
10. Лазарєв Ю.Ф. Основи програмування в середовищі MatLAB: Навч. посібник. – К.: НТУУ "КПІ", 2003. – 424 с.
11. Лазарєв Ю.Ф. Довідник з MATLAB / Електронний навчальний посібник з курсового і дипломного проектування. – К.: НТУУ "КПІ", 2013. – 132 с.
12. Brian D. Hahn, Deniel T. Valentine. Essentiel MATLAB vor Engineers and Scientists. MATLAB examples: 7-th editions. – Elsevier Ltd, Academic Press, 2019. – 412 p.
13. Manassah Jamal T. Elementary mathematical and computational tools for electrical and computer engineers using MATLAB. – CRC Press LLC, 2001. – 349 p.

14. Helmut Bode. MATLAB-Simulink. Analyse und Simulation dynamischer Systeme: 2. Auflage. – Wiesbaden: Teubner Verlag, 2006. – 319 S.
15. Pietruszka W.D. MATLAB und Simulink in der Ingenieurpraxis. Modelbildung, Berechnung und Simulation: 2. Auflage. – Wiesbaden: Teubner Verlag, 2006. – 402 S.
16. Гаєв Є.О., Нестеренко Б.М. Універсальний математичний пакет MATLAB і типові задачі обчислювальної математики / Навчальний посібник. – К.: НАУ, 2004. – 176 с.
17. Математичні методи в електромеханіці: навч. посіб. для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / Укл. О.І. Толочко. – Київ: КПІ ім. Ігоря Сікорського, 2020. – 212 с.
18. Математичні методи в електромеханіці та теорії керування. Методичні вказівки до лабораторних робіт для студентів спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» спеціалізації «Електромеханічні системи автоматизації та електропривод» / Укл.: О.І.Толочко. – Київ: КПІ ім. Ігоря Сікорського, 2016. – 81 с.
19. Control System Toolbox. User's Guide, The MathWorks, Release 2015b, 2015.
20. Краснопрошина А.А., Репнікова Н.Б., Ільченко А.А. Сучасний аналіз систем керування з використанням MATLAB, Simulink, Control System: Навчальний посібник. – К.: "Корнійчук", 1999. – 144 с.
21. Лозинський А., Мороз В., Паранчук Я. Розв'язування задач електромеханіки в середовищі пакетів MathCAD і MATLAB: навч. посібник. – Львів: Видавництво Державного університету «Львівська політехніка», 2000. – 166 с.
22. Толочко О.І. Аналіз та синтез електромеханічних систем зі спостерігачами стану. – Донецьк: НОРД-ПРЕС, 2004. – 298 с.

23. Островерхов М.Я., Пижов В.М. Моделирование электромеханических систем в Simulink: Навч. посібник для студентів вищих навчальних закладів. – К.: ВД «Стилос», 2008. – 528 с.

24. Моделирование систем автоматического управления: підручник / О.І. Толочко; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2024. – 328 с.

25. Моделирование систем автоматического управления. Методичні вказівки до лабораторних робіт: навч. посіб. / О. І. Толочко; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2023. – 250 с.

ЗМІСТ

ВСТУП	3
Практичне заняття № 1. ОСНОВИ РОБОТИ В MATLAB	5
1.1. Довідкова інформація	5
1.2. Завдання	12
1.3. Порядок виконання роботи	14
1.4. Методичні рекомендації	14
Практичне заняття № 2. ПОБУДОВА ТА ОФОРМЛЕННЯ БАЗОВИХ ДВОВИМІРНИХ ГРАФІКІВ ФУНКЦІЙ	16
2.1. Довідкова інформація	16
2.2. Завдання	23
2.3 Методичні рекомендації	25
Практичне заняття № 3. МОДЕЛЮВАННЯ ЛІНІЙНИХ ЕЛЕКТРИЧНИХ КІЛ	27
3.1. Довідкова інформація	27
3.2. Приклад математичного моделювання лінійного електричного кола в <i>Simulink</i>	29
3.3. Завдання	31
3.4. Порядок виконання роботи	33
3.5. Методичні рекомендації	34
Практичне заняття № 4. ПОБУДОВА ЧАСТОТНИХ ХАРАКТЕРИСТИК	35
4.1. Довідкова інформація	35
4.2. Завдання	38
4.3. Методичні рекомендації	39
Практичне заняття № 5. РОЗРАЗХУНОК УСТАЛЕНИХ РЕЖИМІВ У ЕЛЕКТРИЧНИХ КОЛАХ МАТРИЧНИМ МЕТОДОМ	43
5.1. Довідкова інформація	43

5.2. Приклад застосування методів лінійної алгебри при розрахунку струмів і напруг у гілках кіл постійного струму	44
5.3. Приклад застосування методів лінійної алгебри при розрахунку струмів і напруг у гілках кіл змінного струму	46
5.4. Завдання	49
5.5. Порядок виконання роботи	49
5.6. Методичні рекомендації	52
Практичне заняття № 6. АПРОКСИМАЦІЯ ТА ІНТЕРПОЛЯЦІЯ	
ТАБЛИЧНИХ ФУНКЦІЙ СТЕПЕНЕВИМИ ПОЛІНОМАМИ	53
6.1. Довідкова інформація	53
6.2. Завдання	57
6.3. Методичні рекомендації	57
Практичне заняття № 7 ГАРМОНІЧНИЙ АНАЛІЗ ТА СИНТЕЗ	
ПЕРІОДИЧНИХ ФУНКЦІЙ	61
7.1. Довідкова інформація	61
7.2. Завдання	62
7.3. Порядок виконання роботи	65
7.4. Методичні рекомендації	65
Практичне заняття № 8. ЧИСЕЛЬНЕ РІШЕННЯ ТРАНСЦЕНДЕНТНИХ	
ТА АЛГЕБРАЇЧНИХ РІВНЯНЬ	66
8.1. Довідкова інформація	66
8.2. Завдання	68
8.3. Порядок виконання роботи	69
8.4. Методичні рекомендації	72
ЛІТЕРАТУРА	73