

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗІНА
НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ
«УКРАЇНСЬКА ІНЖЕНЕРНО ПЕДАГОГІЧНА АКАДЕМІЯ»

**НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ СИСТЕМ ТА
ОБ'ЄКТІВ»**

ХАРКІВ 2025

ЗМІСТ

Лабораторна робота № 1. Знайомство з Simulink	5
1 Теоретичні відомості	5
1.1 Загальні відомості про Simulink	5
1.2 Елементи інтерфейсу Simulink	6
2 Завдання до лабораторної роботи	10
3 Хід виконання роботи	11
4 Коментарі	15
Контрольні запитання	17
Рекомендована література	18
Лабораторна робота № 2. Приклади моделювання пасивних електричних кіл	19
1 Теоретичні відомості	19
2 Завдання	20
3 Методичні вказівки	21
Контрольні запитання	22
Рекомендована література.....	23
Лабораторна робота № 3. Підсистеми в Simulink	24
1 Теоретичні відомості	24
2 Завдання	24
3 Методичні вказівки	25
Контрольні запитання	34
Рекомендована література	34
Лабораторна робота № 4. Аналіз експериментальних даних у фазовому просторі	35
1 Короткі теоретичні відомості	35
2 Завдання	38
2.1 Генерація даних	38
2.2 Аналіз даних	39

3	Протокол лабораторної роботи	40
4	Методичні рекомендації до виконання роботи	48
	Контрольні запитання	53
	Рекомендована література	53
	Лабораторна робота № 5. Моделювання дивного атрактора Лоренца та Рьюсслера в системі Simulink	55
1	Короткі теоретичні відомості	55
2	Завдання	57
3	Методичні вказівки	58
	Контрольні запитання	58
	Рекомендована література	59
	Лабораторна робота № 6. Основи об'єктно-орієнтовного програмування	60
1	Теоретичні відомості	60
2	Завдання	65
	Контрольні запитання	66
	Рекомендована література	67
	Лабораторна робота № 7. Моделювання поширення акустичних хвиль у середовищах методом скінченних різниць.....	68
1	Теоретичні відомості	68
1.1	Основні класичні типи рівнянь математичної фізики	68
1.2	Базові положення МСР	70
1.3	Хвильове рівняння д'Аламбера	73
1.4	Граничні умови.....	76
2	Завдання	77
3	Методичні вказівки	79
	Контрольні запитання	81
	Рекомендована література	81

ЗМІСТ

Лабораторна робота № 1. Знайомство з Simulink	5
1. Теоретичні відомості	5
1.1. Загальні відомості про Simulink	5
1.2. Елементи інтерфейсу Simulink	6
2. Завдання до лабораторної роботи	10
3. Хід виконання роботи	11
4. Коментарі	15
Контрольні запитання.....	17
Рекомендована література.....	18
Лабораторна робота № 2. Приклади моделювання пасивних електричних кіл ..	19
1. Теоретичні відомості	19
2. Завдання	20
3. Методичні вказівки	21
Контрольні запитання.....	22
Рекомендована література.....	23
Лабораторна робота № 3. Підсистеми в Simulink.....	24
1. Теоретичні відомості	24
2. Завдання	24
3. Методичні вказівки	25
Контрольні запитання.....	34
Рекомендована література.....	34
Лабораторна робота № 4. Аналіз експериментальних даних у фазовому просторі	35
1. Короткі теоретичні відомості	35
2. Завдання	38
2.1. Генерація даних.....	38
2.2. Аналіз даних	39
3. Протокол лабораторної роботи	40
4. Методичні рекомендації до виконання роботи.....	48

Контрольні запитання.....	53
Рекомендована література.....	53
Лабораторна робота № 5. Моделювання дивного атрактора Лоренца та Р'юсслера в системі Simulink	55
1. Короткі теоретичні відомості	55
2. Завдання.....	57
3. Методичні вказівки	58
Контрольні запитання.....	58
Рекомендована література.....	59
Лабораторна робота № 6. Основи об'єктно-орієнтовного програмування	60
1. Теоретичні відомості	60
2. Завдання.....	65
Контрольні запитання.....	66
Рекомендована література.....	67
Лабораторна робота № 7. Моделювання поширення акустичних хвиль у середовищах методом скінченних різниць.....	68
1. Теоретичні відомості	68
1.1. Основні класичні типи рівнянь математичної фізики	68
1.2. Базові положення МСР.....	70
1.3. Хвильове рівняння д'Аламбера.....	73
1.4. Граничні умови.....	76
2. Завдання	77
3. Методичні вказівки	79
Контрольні запитання.....	81
Рекомендована література.....	81

ЛАБОРАТОРНА РОБОТА № 1. ЗНАЙОМСТВО З SIMULINK

Мета роботи: ознайомитись та набути базових практичних навичок створення моделей у системі Simulink.

1. Теоретичні відомості

Simulink – середовище динамічного міждисциплінарного моделювання складних технічних систем.

Simulink – інтерактивний інструмент (програмне забезпечення) для моделювання, імітації та аналізу динамічних систем, включаючи дискретні, неперервні та гібридні, нелінійні та розривні системи. Розроблене компанією The MathWorks.

1.1. Загальні відомості про Simulink

Simulink – середовище динамічного міждисциплінарного моделювання складних технічних систем, заснований на інструментарії модельно-орієнтованого проектування.

Для побудови функціональної блок-схеми моделей Simulink має велику бібліотеку блокових компонентів та зручний редактор блок-схем. Він заснований на графічному інтерфейсі користувача та по суті є типовим засобом *візуально орієнтованого програмування*.

Simulink пропонує тісну інтеграцію з іншим середовищем MATLAB і може використовувати MATLAB, або створювати сценарії з нього.

У поєднанні з іншими продуктами Simulink може автоматично генерувати вихідний код мовою С для реалізації систем в режимі реального часу, вбудовування моделей в інші програмні рішення.

Simulink дозволяє інженерам моделювати системи з різних галузей знань (таких, як механічні, електричні, гідравлічні та інші системи) шляхом створення моделей в інтерактивному графічному середовищі. Симуляція, розробка, генерація коду, тестування та верифікація – все це здійснюється в єдиному середовищі, дозволяючи різним групам інженерів сфокусуватися більше на своїй роботі та менше – на загальній платформі для симуляції.

1.2. Елементи інтерфейсу Simulink

Вікно браузера бібліотеки блоків Simulink Library Browser наведено на рис. 1.1, де виділено основну бібліотеку Simulink (у лівій частині вікна) та показано її розділи (у правій частині вікна).

Основна бібліотека системи Simulink містить такі розділи:

- Commonly Used Blocks – блоки, що часто використовуються;
- Continuous – блоки аналогових елементів;
- Discontinuities – нелінійні блоки;
- Discrete – дискретні блоки;
- Logic and Bit Operations – блоки логічних та бітових операцій;
- Look-Up Tables – блоки таблиць;
- Math Operations – блоки математичних операцій;
- Model Verification – блоки верифікації сигналів;
- Model-Wide Utilities – розділ допоміжних утиліт;
- Ports & Subsystems – порти та підсистеми;
- Signal Attributes – блоки отримання властивостей сигналів;

- Signals Routing – блоки перетворення сигналів та допоміжні блоки;
- Sinks – приймачі сигналів;
- Sources – джерела сигналів;
- User-defined Function – функції користувача.

В залежності від технічної задачі моделювання існують багато інших бібліотек. Деякі з них сумісні між собою, деякі – ні.

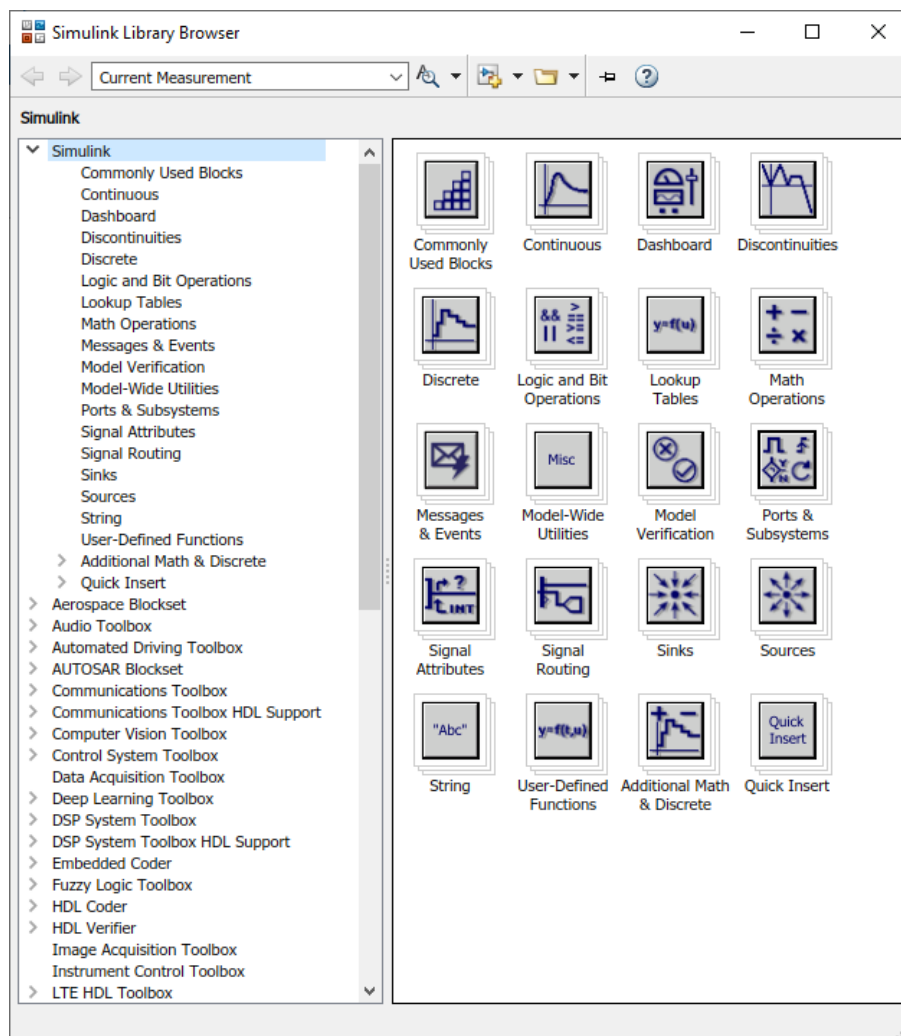


Рис. 1.1. Оглядач (браузер) бібліотеки блоків «Simulink Library Browser»

Запуск процесу моделювання складеної S-моделі (моделі Simulink) відбувається кнопкою «Start» (рис. 1.2).

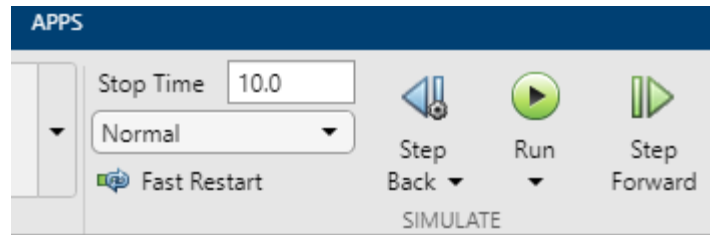


Рис. 1.2. Панель запуску процесу моделювання

На цій же панелі зазначається час моделювання Simulation time. Воно задається початковим часом Start time (зазвичай 0) та кінцевим часом Stop time. Рівність Stop time нескінченності (**inf**) означає, що моделювання буде відбуватися нескінченно довго, доки ми не перервемо його. Однак у цьому випадку важко отримати помітні осцилограми роботи пристрою, тому рекомендується задавати кінцеві значення Stop time.

Слід зазначити, що час моделювання – величина досить умовна. Не слід думати, що Stop time ~50 означає моделювання протягом 50 секунд. Точної відповідності між часом моделювання в секундах та заданим значенням немає. Реальний час моделювання залежить від швидкодії комп'ютера, на якому виконується моделювання.

Першорядне значення мають дві опції вирішувача у полі Solver options: тип рішення та метод рішення. Можливі два типи рішення (рис. 1.3):

- Variable-step solvers – рішення зі змінним кроком;
- Fixed-step solvers – рішення з фіксованим кроком.

Як правило, найкращі результати дає рішення зі змінним кроком (зазвичай за часом, але не завжди).

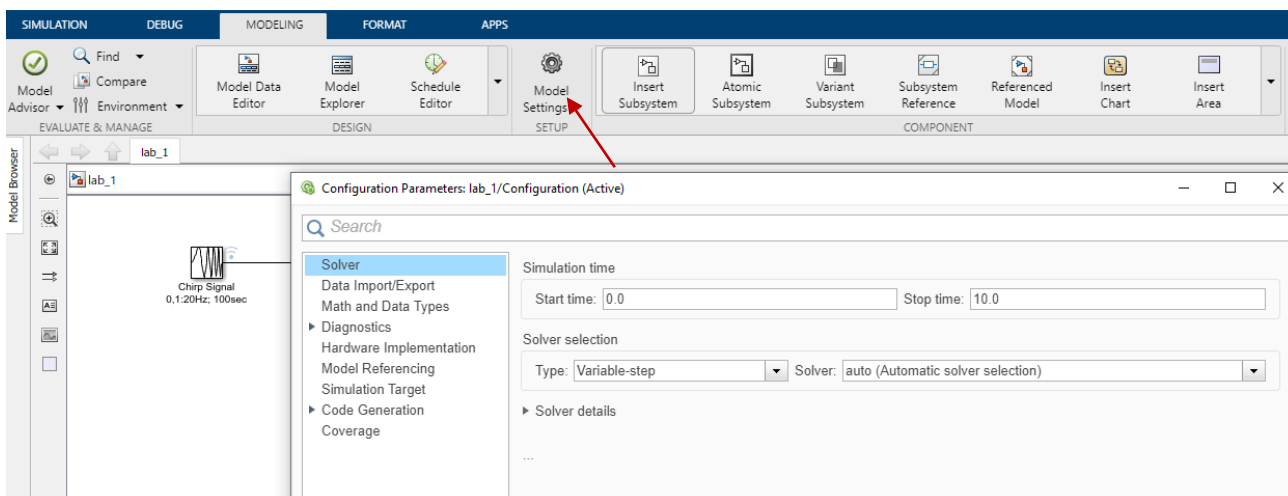


Рис. 1.3. Налаштування вирішувача Solver options

У разі необхідності можна більш детально налаштувати вирішувач, наприклад, задавши мінімальний та максимальний крок інтегрування (рис. 1.4):

- Max step size – максимальний крок інтегрування системи однорідних диференціальних рівнянь;
- Min step size – мінімальний крок інтегрування;
- Initial step size – початковий крок інтегрування.

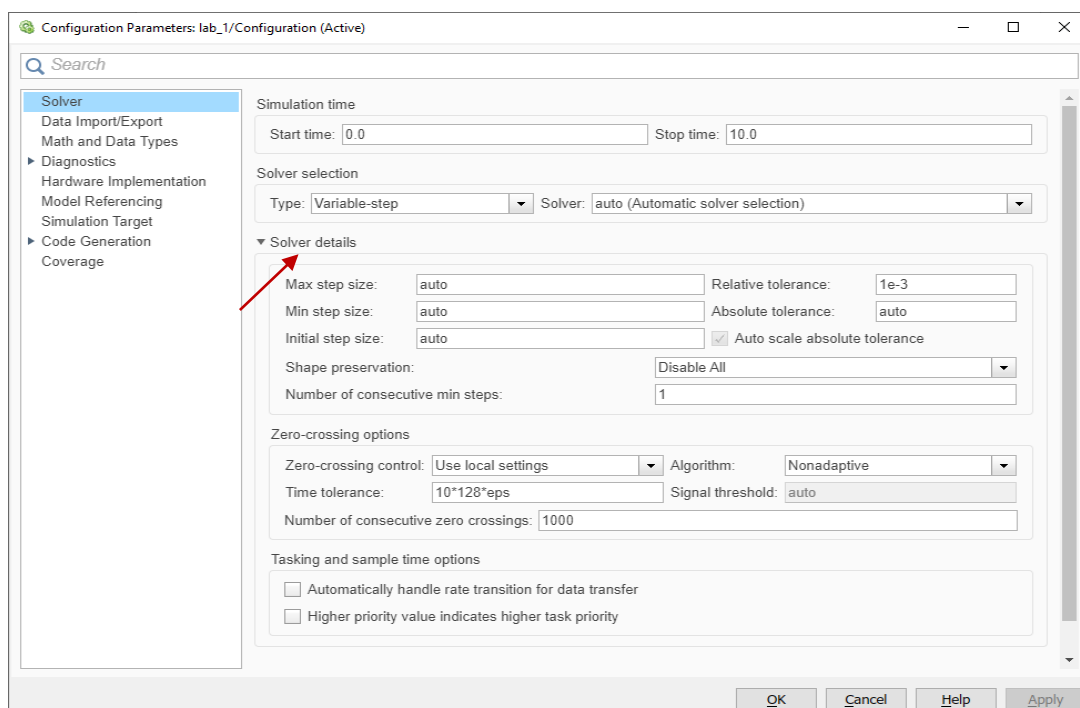


Рис. 1.4. Налаштування додаткових параметрів вирішувача

Важливим є і такий параметр моделювання, як точність інтегрування:

- Relative tolerance – відносна похибка інтегрування;
- Absolute tolerance – абсолютна похибка інтегрування.

За замовчуванням вони задані відповідно рівними 10^{-3} і 10^{-6} . Якщо, наприклад, графіки результатів моделювання виглядають складеними з окремих фрагментів, це вказує на необхідність зменшення зазначених значень похибки. Однак надто малі похибки можуть спричинити значне збільшення часу обчислень.

2. Завдання до лабораторної роботи

1. Створіть Simulink-модель як на рис. 2.1.

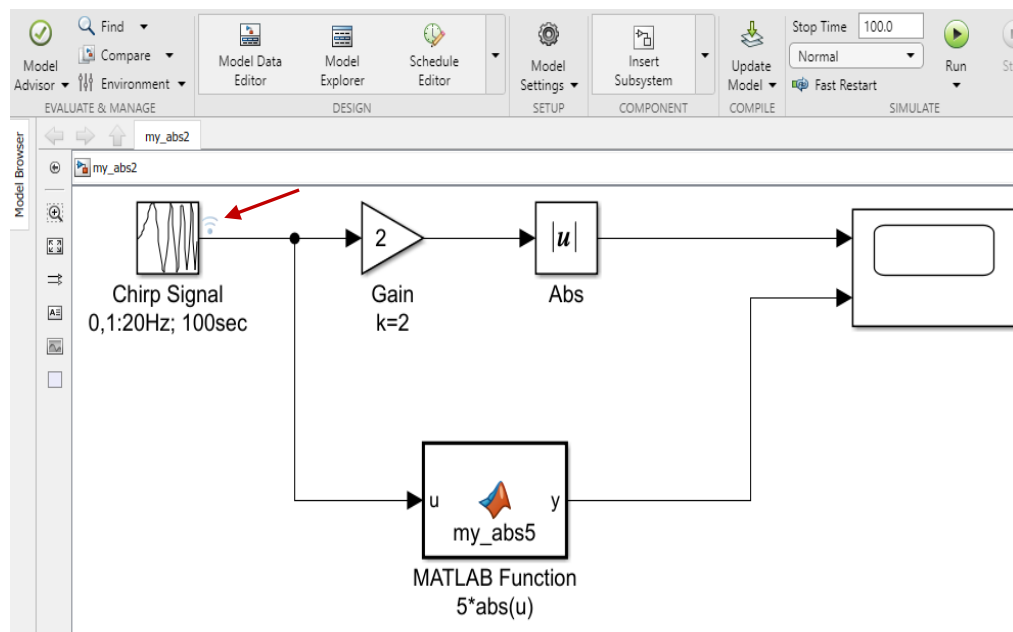


Рис. 2.1. S-модель, що створюється та досліджується

2. Порівняйте результати моделювання при параметрах розрахунків:

- Max Step size = auto;
- Max Step size = 0.1;
- Max Step size = 0.05;
- Max Step size = 0.001.

3. Для найкращого результату моделювання системи побудуйте графік функцією plot в Matlab для вихідної змінної з блоку «ABS» (див. рис. 1.1).
4. При умовах п.3 проведіть моделювання для п'ятих різних параметрів зміни частоти генератора «Chirp Signal». Для цього в полі властивість «Frequency at target time (Hz:)» цього блоку введіть, наприклад, 1:1:5. Відобразіть на графіку фрагмент результатів моделювання для одного з вихідних сигналів.

3. Хід виконання роботи

1. Створіть новий бланк моделі.
2. Натисніть «Library Browser» (рис. 3.1).

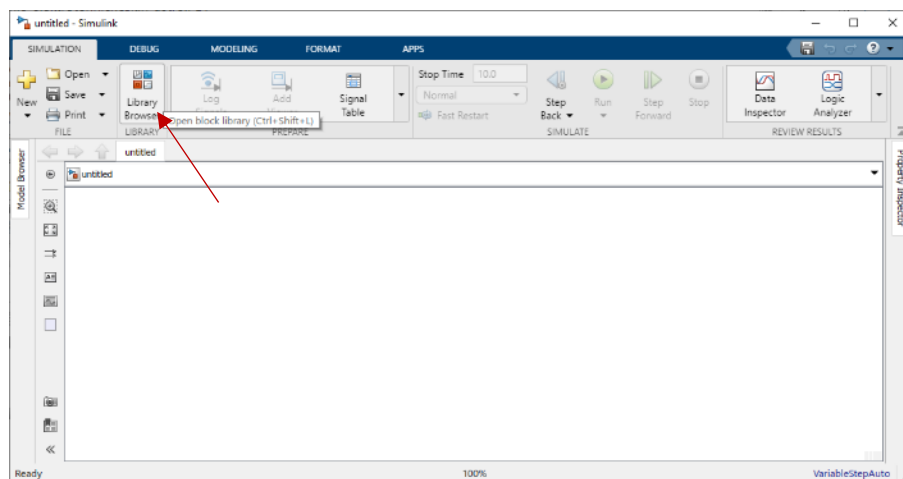


Рис. 3.1

3. В бібліотеці Simulink виберіть джерела стандартних сигналів «**Sources**» та необхідний генератор. Перетягніть його на робоче поле (рис. 3.2).

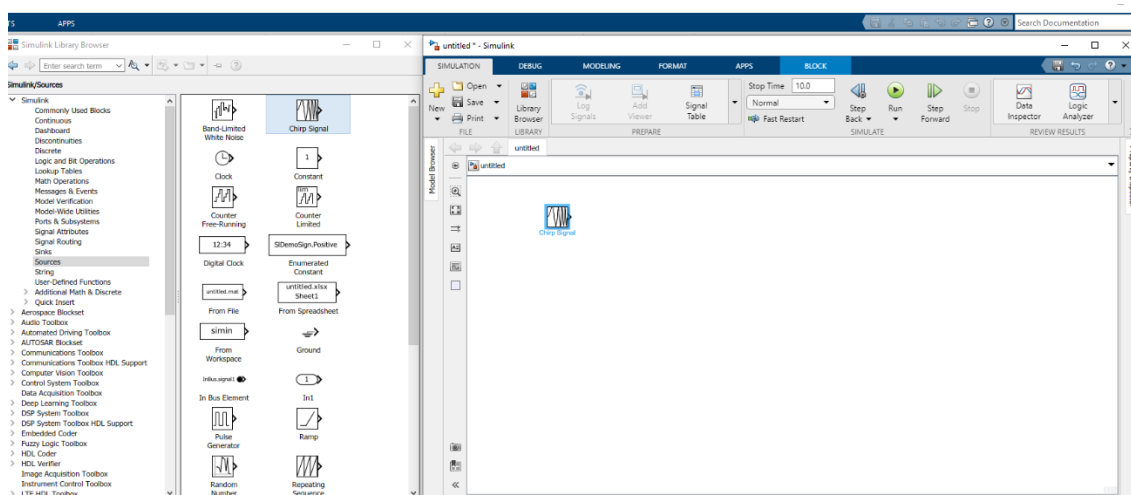


Рис. 3.2

4. В бібліотеці «Math Operation» виберіть блок «Abs» та «Gain» (рис. 3.3).
Перетягніть їх на робоче поле.

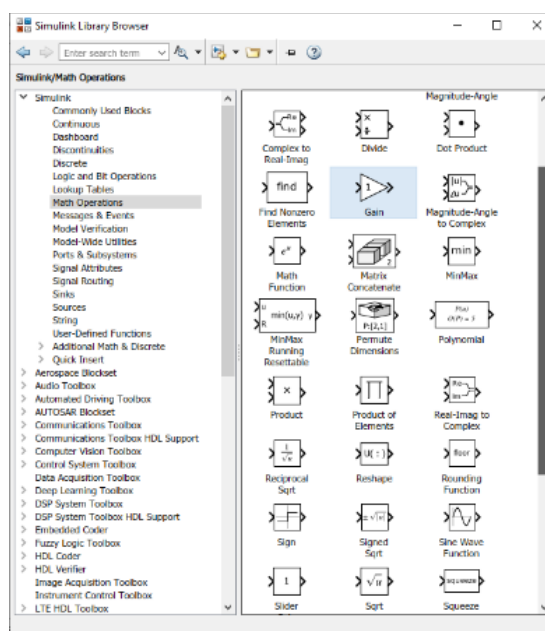


Рис. 3.3

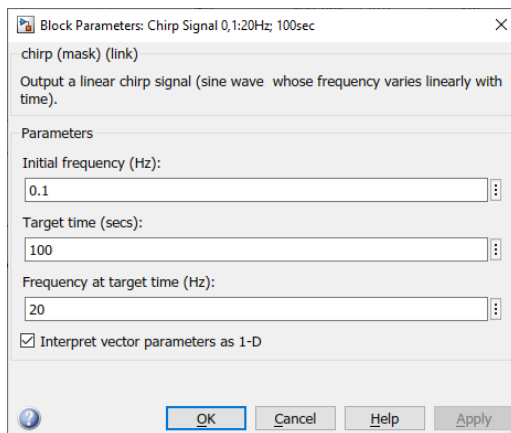
5. В бібліотеці «Sinks» виберіть осцилоскоп «Scope».
6. В підрозділі/бібліотеці «User-Defined Function» виберіть «MATLAB Function».
7. Збережіть модель. При цьому в кінцевій папці не допускаються назви кирилицею, перший символ повинен бути літерою.

8. Два рази клацніть лівою кнопкою миші по блоку «MATLAB Function» в робочій області. Відкриється вікно для редагування matlab-коду. Створить функцію «my_abs» чи іншу функцію у відповідності до завдання. Збережіть її.

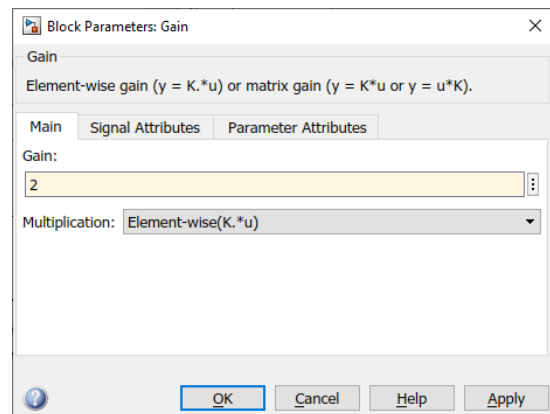
```
function y = my_abs5(u)
```

```
y = 5*abs(u);
```

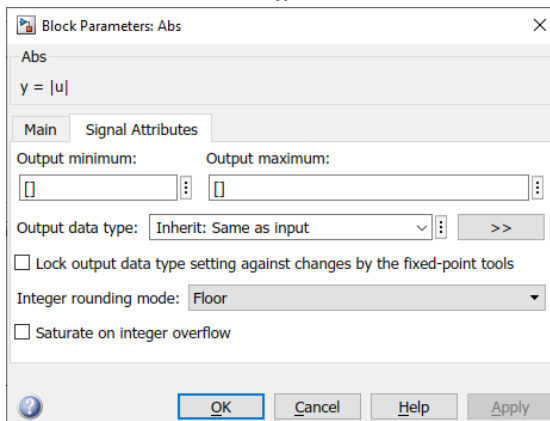
9. З'єднайте блоки у відповідності до завдання.
10. Налаштуйте кожен з блоків моделі (див. рис. 3.4) та підпишіть блоки. Для налаштування блоку два рази клацніть по ньому лівою кнопкою миші.



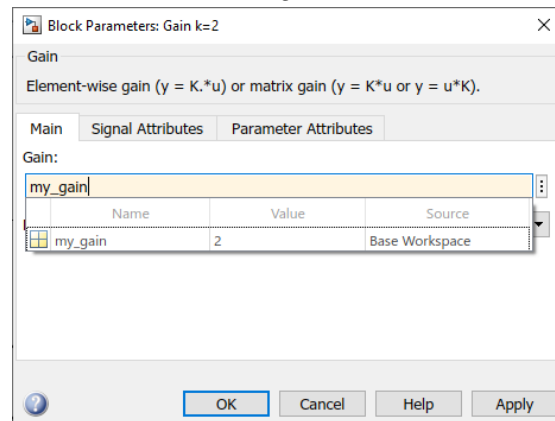
a



б



в



г

Рис. 3.4. Налаштування блоків моделі: *a-в* – явно, у відповідності до завдання; *г* – через змінну у Workspace (на прикладі блоку Gain). Значення “my_gain” повинна бути створена заздалегідь у робочій області Matlab

11. Запустіть код. За умовчанням результат симуляції може бути як на рис.3.5.

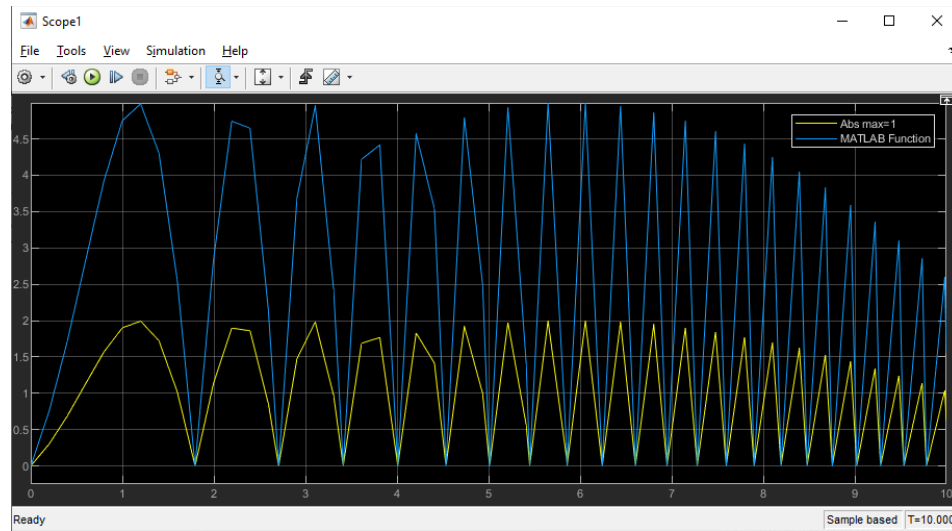


Рис. 3.5. Приклад результату моделювання

12. Для більш точних розрахунків налаштуйте параметри «Model Settings» (рис. 3.6, 3.7).

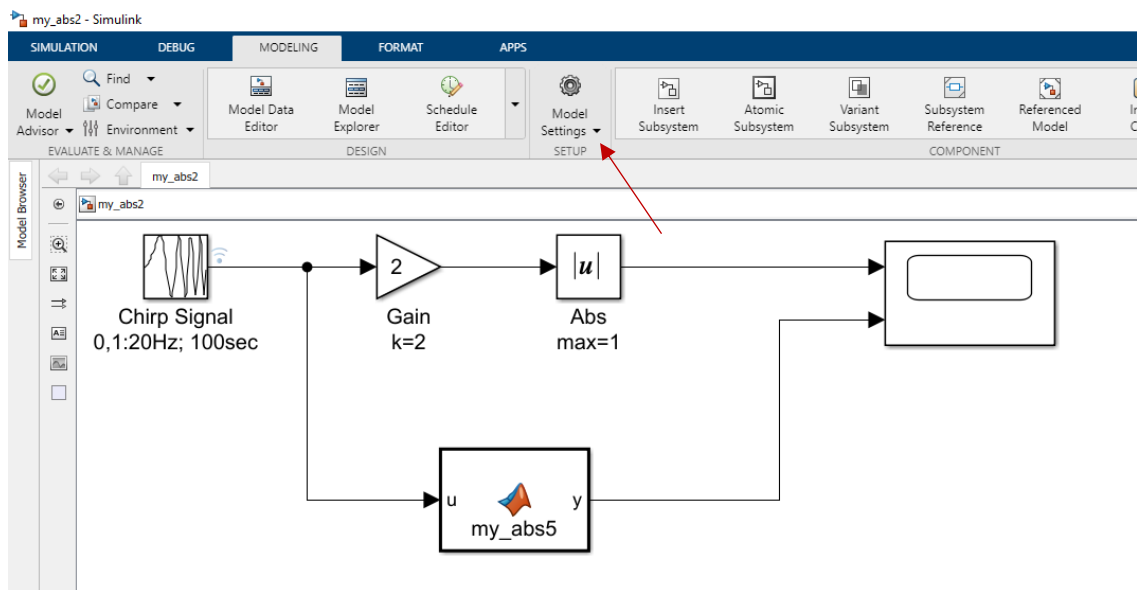


Рис. 3.6. Кнопка налаштування параметрів розрахунків

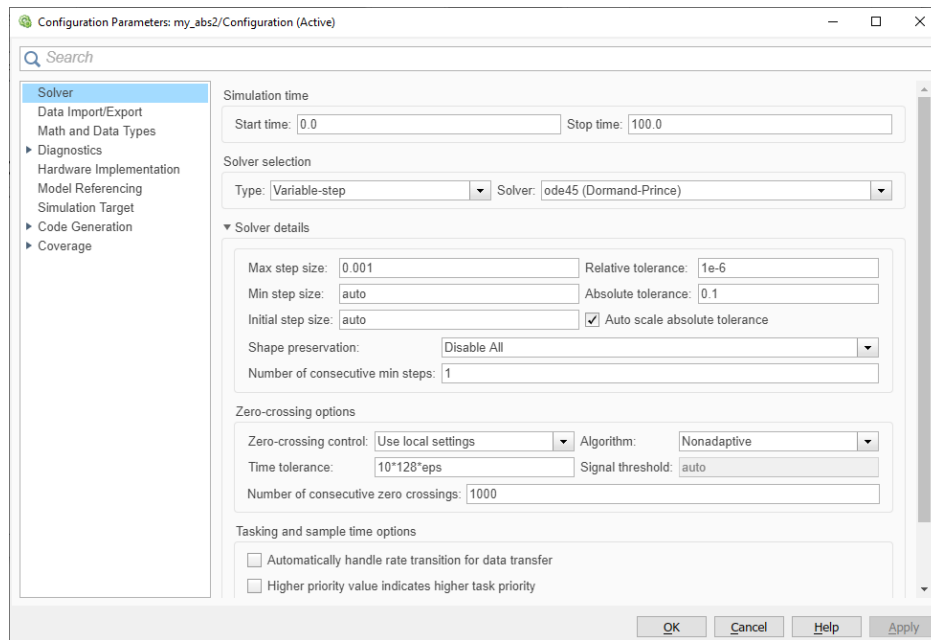


Рис. 3.7. Екранна форма параметрів розрахунку

13. Проведіть повторне симулювання. Оскільки амплітуда синусоїдального сигналу є незмінною, то вихідний результат повинен бути як на рис.3.8.

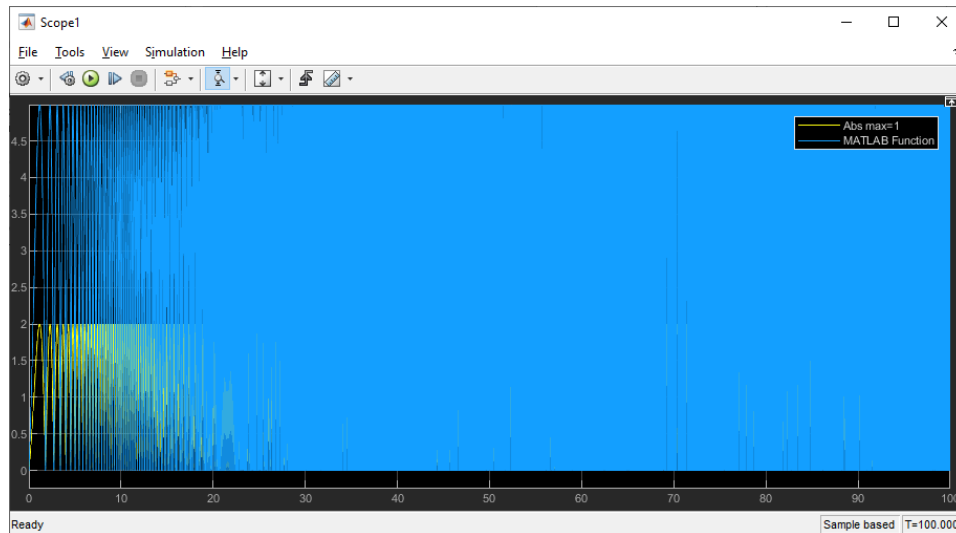


Рис. 3.8. Результат моделювання моделі на рис. 2.1

4. Коментарі

Для порівняння сигналів при різних параметрах моделі необхідно записати сигнал, наприклад, за допомогою властивостей лінії з'єднання. Якщо

обрати «Properties» виділеної лінії, необхідно активувати «Log signal data» (рис. 4.1), де бажано також задати «Signal name».

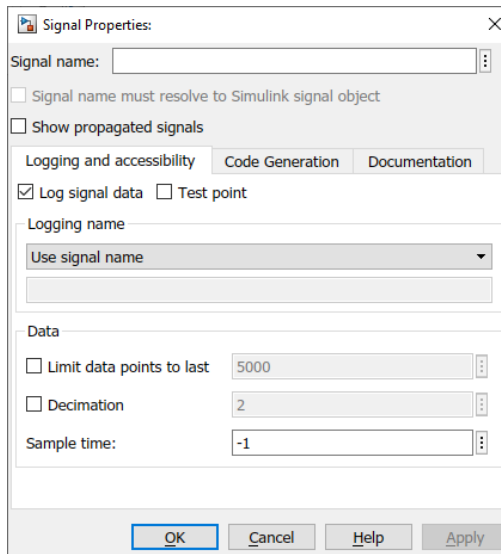


Рис. 4.1

Ознакою, що сигнал буде збережено, є «антена», на яку вказує червона стрілка на рис. 2.1. За замовченням у Workspace створюється структурна змінна «out» де зберігаються параметри та сигнали симуляції. Назву відповідної змінної можна налаштувати в «Model Setting\ Data Import/Export». Наприклад, змінимо це ім'я на «out_result» (рис. 4.2).

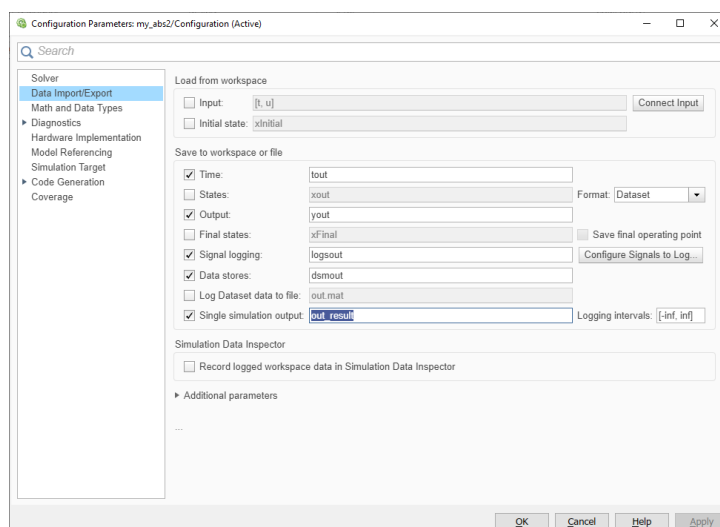
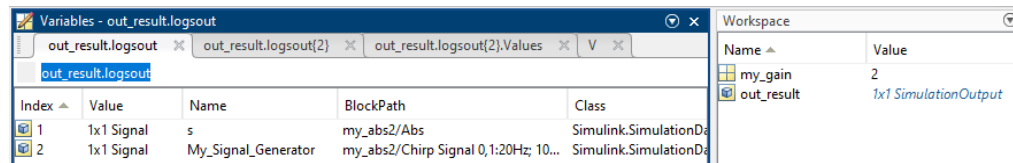
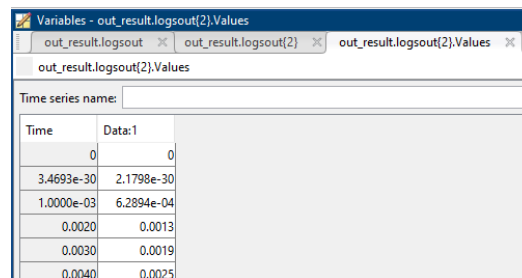


Рис. 4.2. Встановлення ім'я вихідних даних моделювання

Якщо задати ім'я сигналу, наприклад з виходу «Chirp Signal» як «My_Signal_Generator» у Properties значку , то доступ до результатів обчислень у Workspace буде наступний: out_result.logout (рис. 4.3).



a



Time series name:	
Time	Data:1
0	0
3.4693e-30	2.1798e-30
1.0000e-03	6.2894e-04
0.0020	0.0013
0.0030	0.0019
0.0040	0.0025

б

Рис. 4.3. Фрагменти вікон візуалізації даних у Workspace

Для зменшення ієрархічного дерева структурної вихідної змінної для певних даних і полегшення доступу до них доцільно користуватися блоком «Simulink\Sinks\To Workspace».

Контрольні запитання

1. Що таке моделювання?
2. Які переваги інженерам дає Simulink?
3. Як похибки інтегрування впливають на результат моделювання в Simulink?
4. В яких випадках в налаштуваннях «Solver options» краще вказувати «Variable-step solvers»?

Рекомендована література

1. Відеокурс: введення до Simulink:
https://www.youtube.com/playlist?list=PLmu_y3-DV2_k0FqQSqWVKE0cW-eSPUSTq
2. Гераїмчук, М. Д. Моделювання систем у середовищі MATLAB-SIMULINK [Електронний ресурс] : комп'ютерний практикум / М. Д. Гераїмчук, Ю. Ф. Лазарєв, Т. О. Толочко ; НТУУ «КПІ». – Київ : НТУУ «КПІ», 2006. – 175 с.
3. Моделювання систем керування в пакеті MATLAB SIMULINK, методичні вказівки до виконання лабораторної роботи по курсу «Комп'ютерні методи дослідження систем автоматичного управління»/ укл. : І.Р. Козбур, Г.В. Козбур , Р.І. Михайлишин. – Тернопіль : ТНТУ, 2019. - 23 с.
4. Теорія автоматичного керування. Частина І. Теорія лінійних неперервних систем автоматичного керування. Лабораторний практикум / укл. проф. В.Ш. Хісматулін. – Харків. – 2010. – 54 с. – Режим доступу: <http://lib.kart.edu.ua/bitstream/123456789/4593/1/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%BD%D1%96%20%D0%B2%D0%BA%D0%B0%D0%B7%D1%96%D0%B2%D0%BA%D0%B8.pdf>
5. Черних І.В. Моделювання електротехнічних пристроїв MATLAB, SimPowerSystems и Simulink. – 2008. – 288 с.

ЛАБОРАТОРНА РОБОТА № 2. ПРИКЛАДИ МОДЕЛЮВАННЯ ПАСИВНИХ ЕЛЕКТРИЧНИХ КІЛ

Мета роботи: закріплення навичок моделювання в системі Simulink та знайомство з елементами моделювання електричних кіл.

1. Теоретичні відомості

Будь-яку динамічну систему можна описати системою диференціальних рівнянь.

Розв'язавши таку систему, з'являється можливість охарактеризувати її передаточною функцією.

Передаточна функція являє собою відношення зображень по Лапласу вихідного параметра до вхідного при нульових початкових умовах: $W(s) = \frac{Y(s)}{X(s)}$, де s – оператор Лапласа.

Розглянемо диференційне рівняння:

$$a_1 \dot{y}(t) + a_2 y(t) = b_1 x(t).$$

Застосуємо до нього перетворення Лапласа, тобто оператор диференціювання заміниться множенням на s . Отримуємо:

$$s a_1 Y(s) + a_2 Y(s) = b_1 X(s).$$

Після спрощення отримуємо передаточну функцію

$$W(s) = \frac{Y(s)}{X(s)} = \frac{b_1}{a_1 s + a_2}. \quad (1)$$

Такий вид передаточної функції притаманний, наприклад, RC-ланцюгу (рис. 1.1).

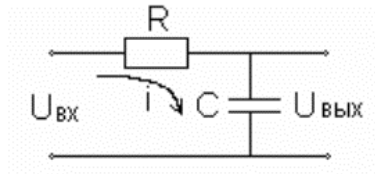


Рис. 1.1. Схемна модель RC-ланцюга

Розглянемо, яким чином отримується математична модель схеми на рис. 1.1.

$$U_{in}(t) = i(t)R + U_{out}(t),$$

$$i(t) = C \frac{dU_c(t)}{dt} = C \frac{dU_{out}(t)}{dt},$$

$$U_{in}(t) = RC \frac{dU_{out}(t)}{dt} + U_{out}(t).$$

Після перетворення Лапласа отримуємо:

$$U_{in}(s) = RCsU_{out}(s) + U_{out}(s),$$

$$W(s) = \frac{Y(s)}{X(s)} = \frac{1}{RCs + 1}. \quad (2)$$

Легко помітити, що для (2) $a_1=RC$, $a_2=1$, $b_1=1$.

2. Завдання

1. Побудуйте передаточну функцію системи на рис 1.1 за допомогою блоку «Transfer Fcn» в групі Simulink/Continuous (рис. 2.1) при $R=10\text{кОм}$, $C=5\text{мкФ}$.

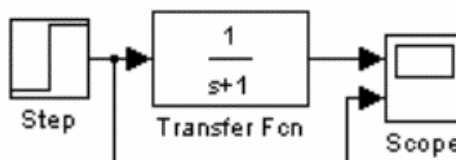


Рис. 2.1. Simulink-модель системи на рис. 1.1

2. Промоделуйте передаточну функцію системи на рис.1.1 за допомогою «SimPowerSystems» чи, теж саме, модулів з Simcape/Electrical/Passive (рис. 2.2).

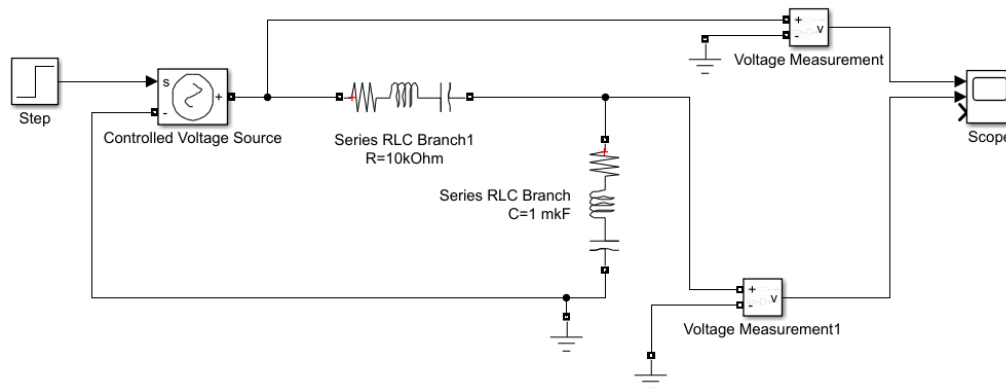


Рис. 2.2. Simscape Simulink-модель системи на рис. 1.1

3. Промоделуйте RLC-електричне коло (рис. 2.3) $R=10\text{k}\Omega$, $L=10\text{мГн}$, $C=5\text{мкФ}$.

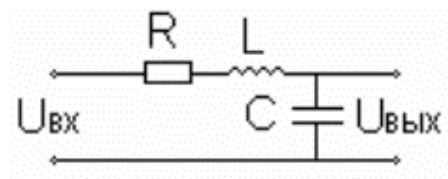


Рис. 2.3. RLC-електричне коло

3. Методичні вказівки

1. SPS-моделі мають і деякі особливості:

- Входи і виходи SPS-блоків, на відміну від блоків Simulink, не показують напрямок передачі сигналу, оскільки фактично є еквівалентами електричних контактів. Таким чином, електричний струм може через вхід або вихід блоку протікати в двох напрямках: як всередину блоку, так і назовні.
- Сполучні лінії між блоками є, по суті, моделями електричних проводів, по яким струм може протікати також в двох напрямках. У

Simulink-моделях же інформаційний сигнал поширюється тільки в одному напрямку – від виходу одного блоку до входу іншого.

- Simulink-блоки і SimPowerSystems-блоки не можуть бути безпосередньо з'єднані один з одним. Сигнал від S -блока можна передати до SPS-блоку через керовані джерела струму або напруги, а навпаки – за допомогою вимірювачів струму або напруги.
 - При розрахунку схеми, що містить нелінійні блоки, слід використовувати методи: ode15s – багатокроковий метод змінного порядку (від 1 до 5), використовує формули чисельного диференціювання; ode23t – метод трапецій з інтерполяцією. Ці методи дають найкращі результати по швидкодії.
 - На схемі повинен бути присутнім хоча б один вимірювальний блок (Current Measurement, Voltage Measurement, Three-Phase VI Measurement або Multimeter). Це пов'язано з особливостями перетворення SimPowerSystems-моделі в еквівалентну розрахункову Simulink-модель.
2. Щоб знайти блоки цифровий вольтметр Voltage Measurement і амперметр Current Measurement, необхідно зайти в бібліотеку Simscape, далі розділ *Electrical / Specialized Power Systems / Fundamental Blocks / Measurements*, значення задати за номером варіанта. Якщо ім'я блоку відоме, можна використати строку пошуку.

Контрольні запитання

1. Які є способи підвищення швидкості розрахунків в Simulink та SPS?
2. За рахунок чого здійснюється сумісність між Simulink та SPS?

3. Як отримується вираз $W(s)$ для передаточної функції на рис. 2.3

$$W(s) = \frac{1}{LCs^2 + RCs + 1} ?$$

4. Наведіть особливості Powergui.

Рекомендована література

1. Гармонічний аналіз та операційне числення: Перетворення Лапласа і його застосування: текст лекцій для студентів Інституту прикладного системного аналізу денної форми навчання освітньо-кваліфікаційного рівня «бакалавр» / Укладач: В.Г.Бондаренко – К.: НТУУ «КПІ», 2017. –70 с.
2. Куц, Ю. В. Спеціальні розділи математики. Курс лекцій [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні» спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології / Ю. В. Куц, Ю. Ю. Лисенко ; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 180 с.
3. Операційне числення. Навчальний посібник для інженерних спеціальностей : навч. посіб. для студ. спеціальності 131 «Прикладна механіка»/ КПІ ім. Ігоря Сікорського ; уклад.: Г. В. Журавська, Т. О. Карпалюк, І. М. Копась, Н. В. Рева. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 79 с.
4. Черних І.В. Моделювання електротехнічних пристроїв MATLAB, SimPowerSystems и Simulink. – 2008. – 288 с.

ЛАБОРАТОРНА РОБОТА № 3. ПІДСИСТЕМИ В SIMULINK

Мета роботи. Отримати навички створення підсистем моделі та керування їх варіантами.

1. Теоретичні відомості

Якщо схема виходить дуже громіздкою і складною, то окремі частини її можна згорнути в підсистеми (Subsystem). Для цього на виділеному шматку схеми треба натиснути праву клавішу миші, потім вибрати Create subsystem. У цьому випадку, виділена множина елементів буде представлена одним елементом, якому можна придати необхідний вигляд та за допомогою процедури «маскування» задати інтерфейс керування його параметрами.

2. Завдання

1. Для моделі рис. 2.3 з лабораторної роботи № 2 «Приклади моделювання пасивних електричних кіл» створіть підсистему, яка містить «Series RLC Branch1 R=10kOhm» та «Series RLC Branch C=1 mkF».
2. Для отриманої підсистеми створіть маску для інтерактивного налаштування опору (R) та ємності (C).
3. Створіть копію отриманої підсистеми та підключіть їх паралельно одна одній. Змініть параметри однієї з підсистем за допомогою маски.
4. Порівняйте результати моделювання.
5. Створіть Simulink модель, де є можливість вибору підсистеми з використанням «Variant Subsystem».

6. Порівняйте результати розрахунків при різних вибраних моделях в «Variant Subsystem».

3. Методичні вказівки

1. Виділіть множину елементів моделі для створення підсистеми/subsystem. В контекстному меню виберіть «Create Subsystem from Selection» (рис. 3.1).

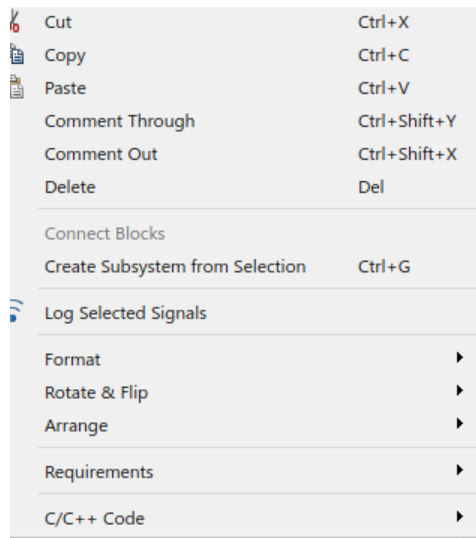


Рис. 3.1. Контекстне меню для вибору «Create Subsystem from Selection»

2. На створеній підсистемі необхідно викликати контекстне меню мишою та вибрати «Mask/Create Mask» (рис. 3.2). У вікні, що з'явиться, можна задати:

- Зовнішній вигляд підсистеми – вкладка «Icon drawing commands»;
- Параметри підсистеми – вкладка «Parameters & Dialog»;
- Задати ініціалізуючий код – вкладка «Initialization»;
- довідкову документацію – вкладка «Documentation».

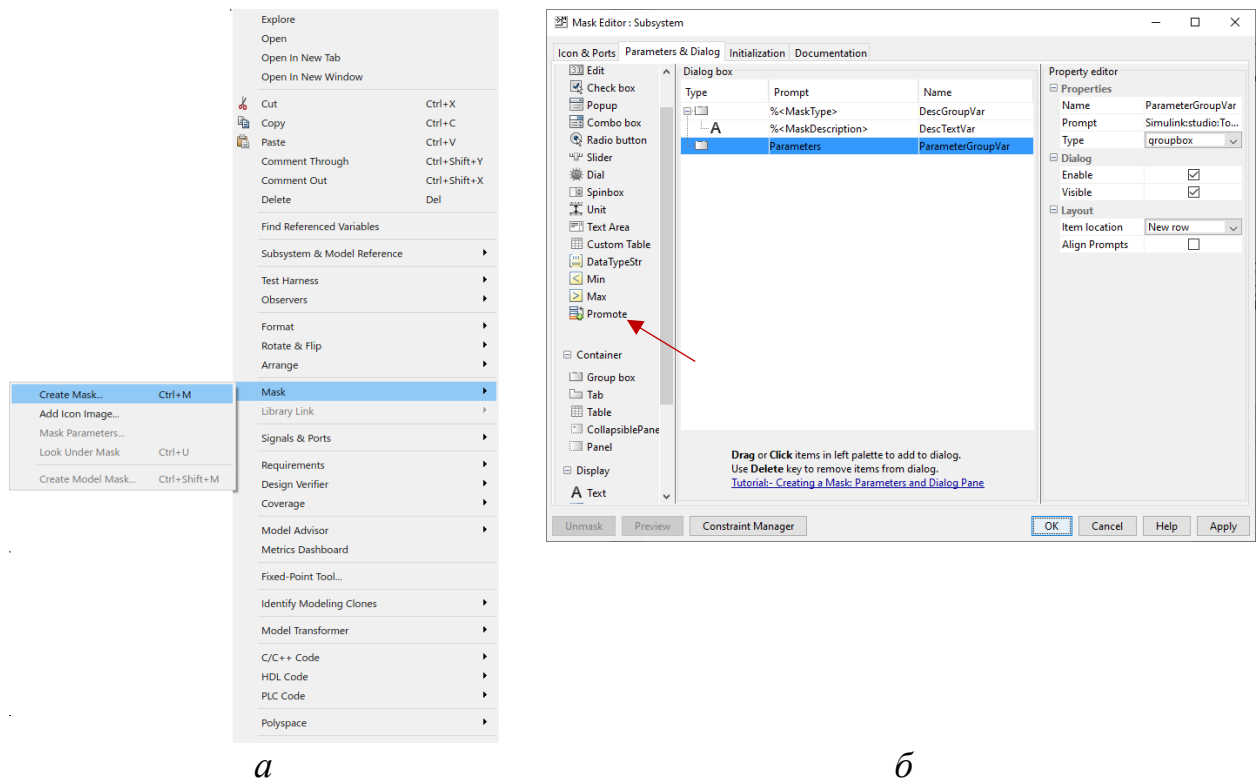


Рис. 3.2. Копія екранних форм для створення «маски» підсистеми: *а* – контекстне меню; *б* – пункт меню «Promote» для безпосереднього налаштування параметрів маски

- Задамо параметри підсистеми, які можна змінювати в інтерактивному режимі за допомогою «Mask». Для цього перейдіть на вкладку «Parameters & Dialog». Виберіть «Promote» (рис. 3.2, б). Далі виберіть необхідний параметр, який буде налаштовуватися за допомогою маски: «Type option\Promoted parameters list» (рис. 3.3).

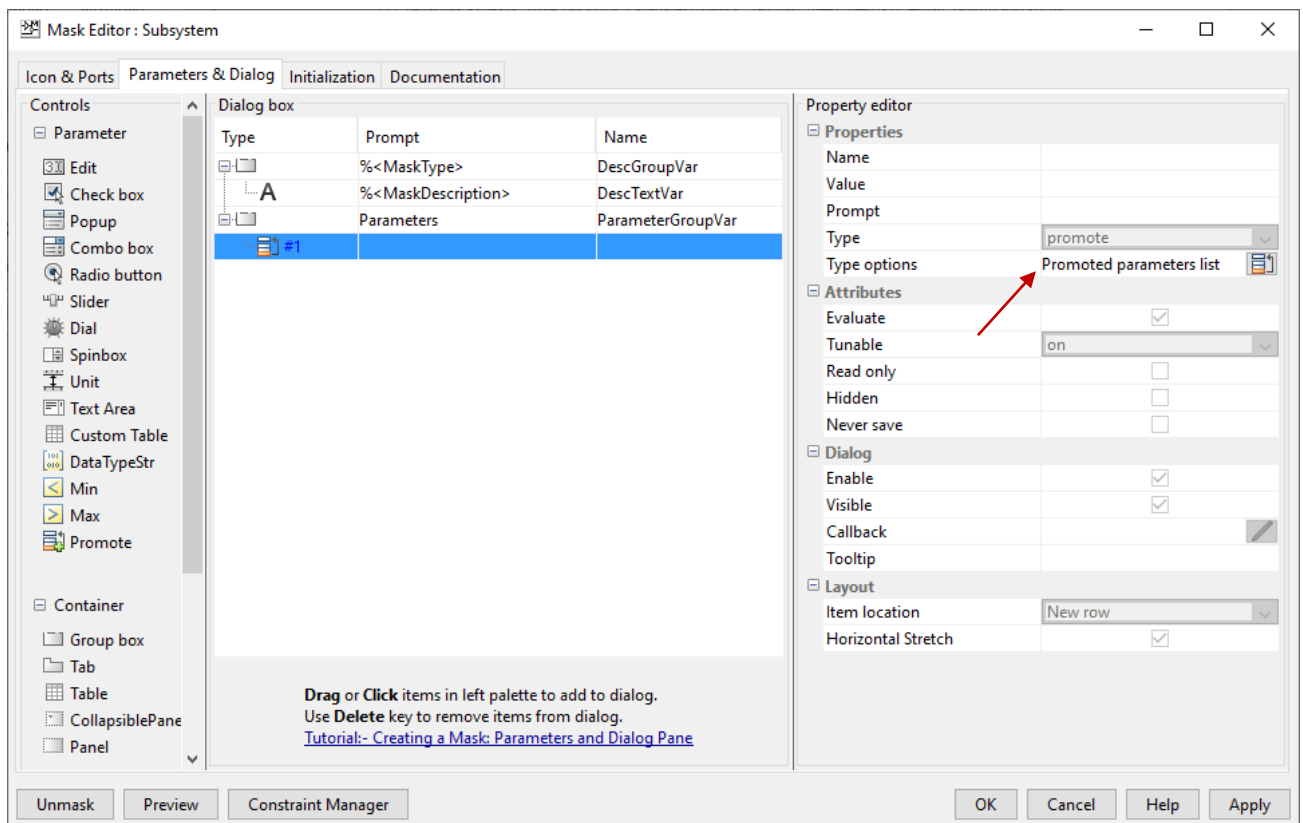


Рис. 3.3. Вибір «Type option\Promoted parameters list»

У новому вікні (рис. 3.4) виберіть об'єкт підсистеми, наприклад, «Series RLC Branch1 R=10 kOhm» (стрілка 1), елемент об'єкта для маскування, наприклад «Resistance (Ohms)» (стрілка 2), та за допомогою стрілки (3) додайте елемент у список.

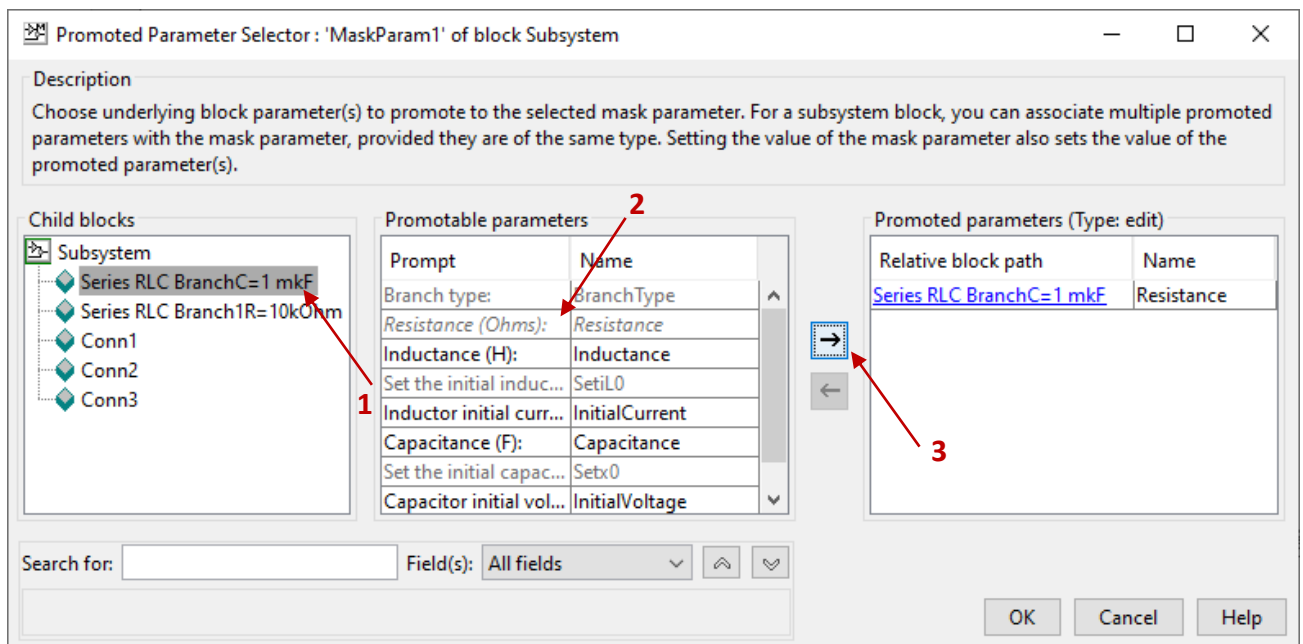


Рис. 3.4. Налаштування параметра підсистеми для маскування

Очікуваний кінцевий результат: подвійний клік миші на підмоделі визиває вікно на рис. 3.5.

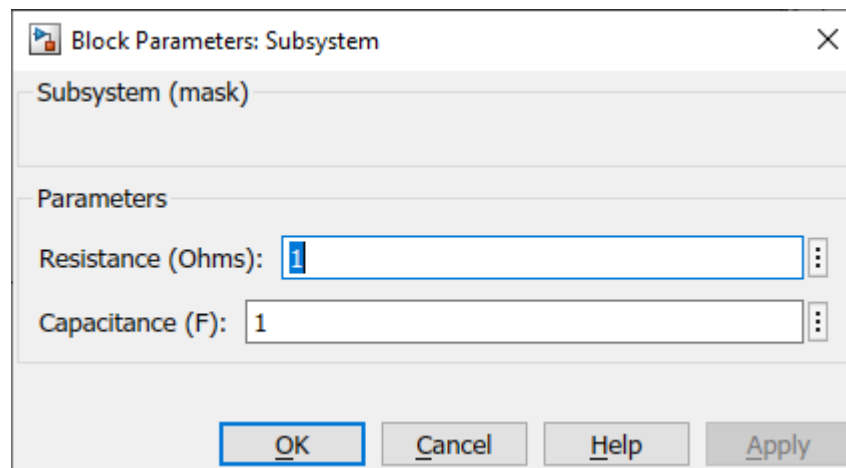


Рис. 3.5. Кінцевий очікуваний результат маскування

- Для створення «Variant Subsystem», блок якого містить декілька різних варіантів моделей підсистеми, виберіть його у Library Simulink «Simulink\Port & Subsystem\Variant Subsystem» (рис. 3.6).

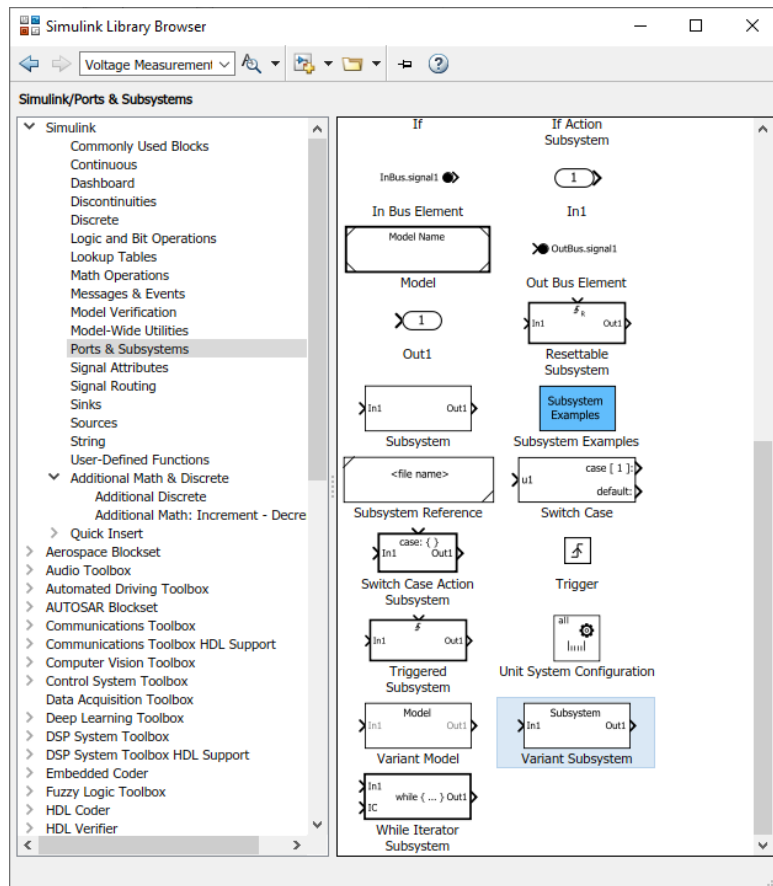


Рис. 3.6. Створення «Variant Subsystem»

5. Створіть різні підсистеми/submoduls та розташуйте їх у «Variant Subsystem». При перетягуванні вхідних та вихідних портів з'являється стрілка, яка означає, що даний порт еквівалентний відповідному входу чи виходу підсистеми (рис. 3.7).

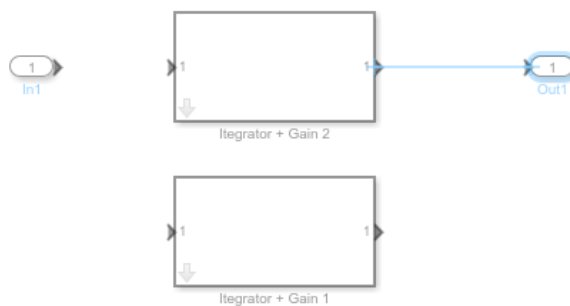


Рис. 3.7. Підсистеми у «Variant Subsystem». Вихід підсистеми «1» відповідає вихідному порту «1»

6. Для того, щоб задати активну підсистему, необхідно вказати «Condition». В робочій області Matlab прописується, **наприклад**, наступний код:

```
>> Subsystem_1_Gain_1=Simulink.Variant('cond_Variant1 == 1')
Subsystem_1_Gain_1 =
Simulink.Variant
    Condition: 'cond_Variant1 == 1'

>> Subsystem_1_Gain_2=Simulink.Variant('cond_Variant1 == 0')

Subsystem_1_Gain_2 =
Simulink.Variant
    Condition: 'cond_Variant1 == 0'

>> cond_Variant1 = 0

cond_Variant1 =
    0
```

7. Для налаштування «Variant Subsystem» станьте на відповідний модуль на базовій схемі моделі. В головному меню з'явиться пункт «Variant Subsystem» (рис. 3.8).

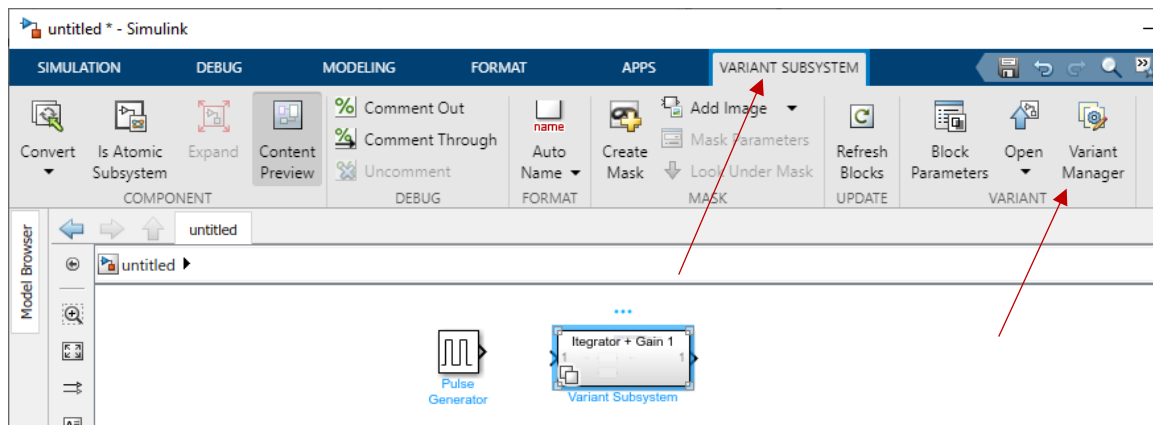


Рис. 3.8. Меню «Variant Manager»

Виберіть «Variant Manager». Відкриється вікно управління всіма «Variant Subsystem» (рис. 3.9).

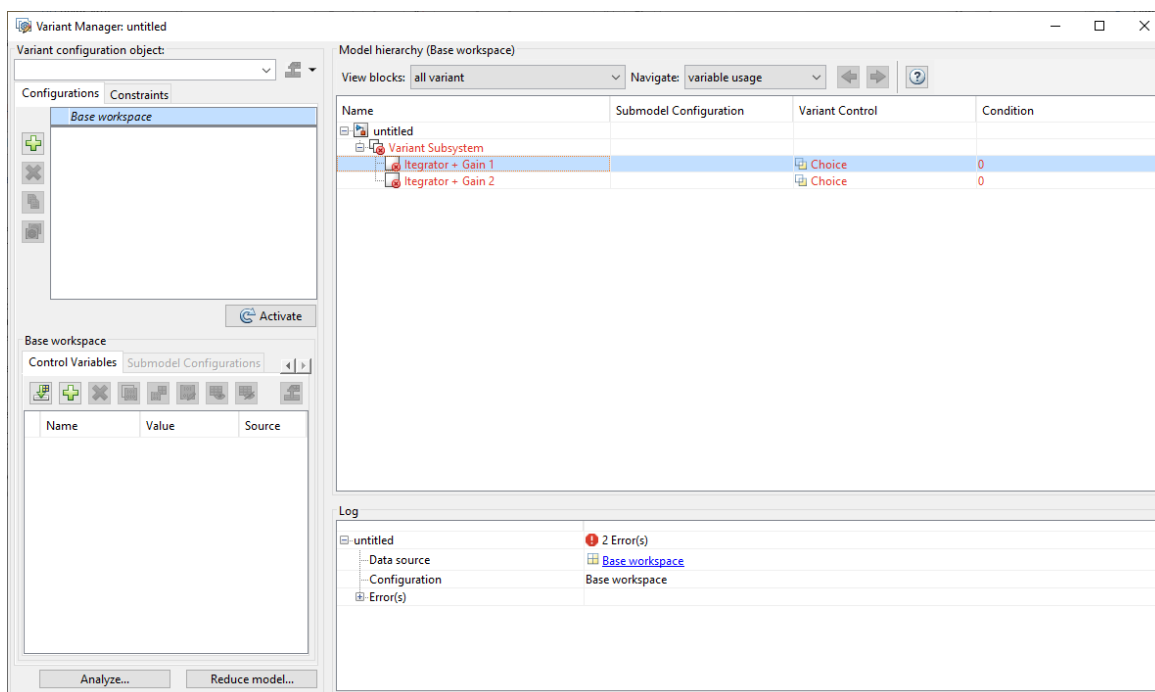


Рис. 3.9. Вікно налаштування всіх «Variant Subsystem» – «Variant Manager»

8. Для налаштування «Variant Subsystem» необхідно визвати контекстне меню та вибрати «Open Parent Block Parameters» (рис. 3.10).

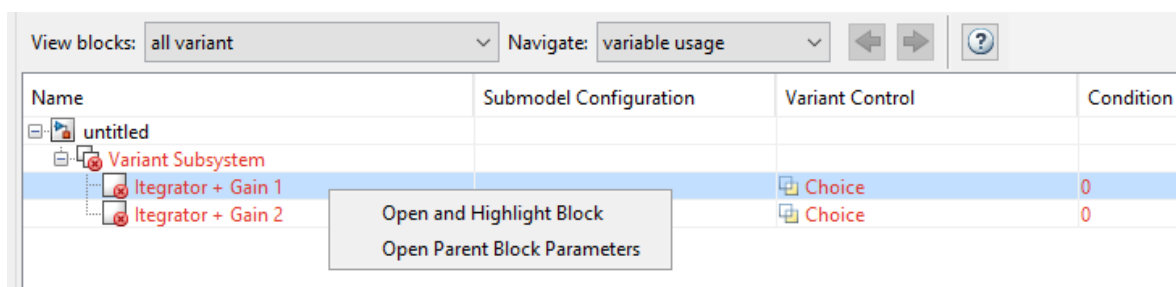


Рис. 3.10

9. У вікні «Open Parent Block Parameters» замість «Choice» введіть ім'я відповідного об'єкту з параметрами, що було створено в робочому вікні Matlab: *Subsystem_1_Gain_1* чи *Subsystem_1_Gain_2* (рис. 3.11 та 3.12).

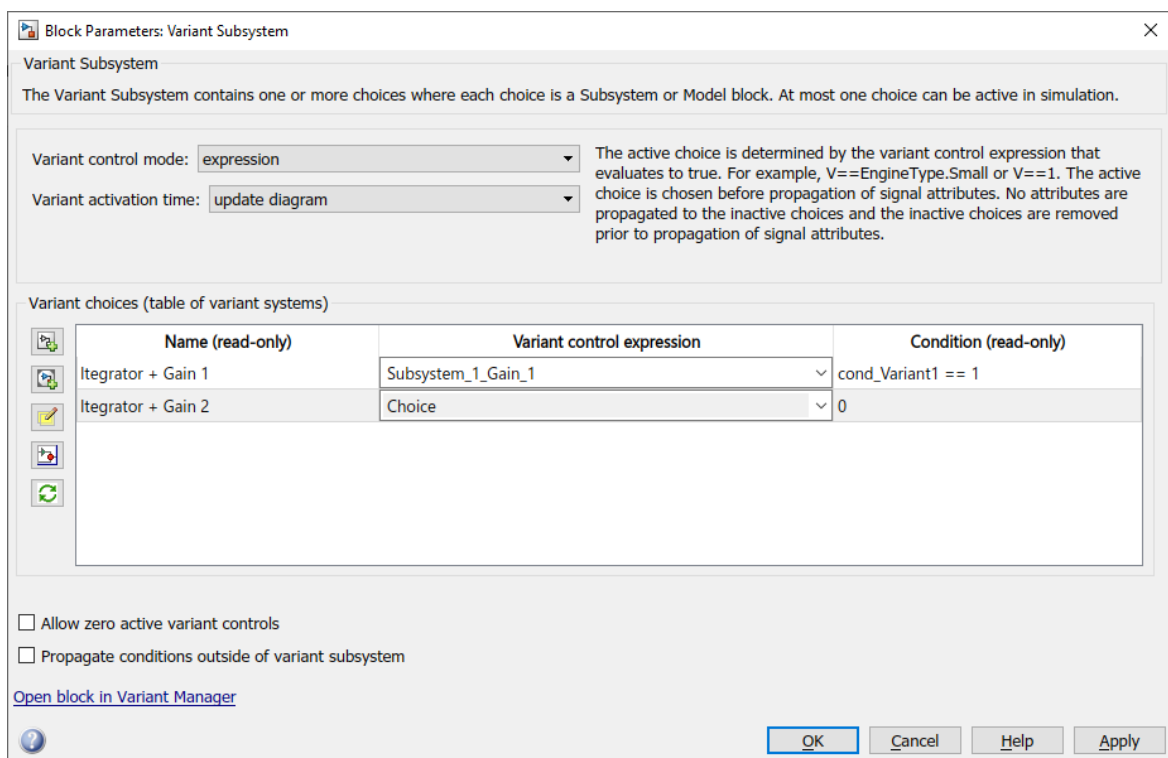


Рис. 3.11

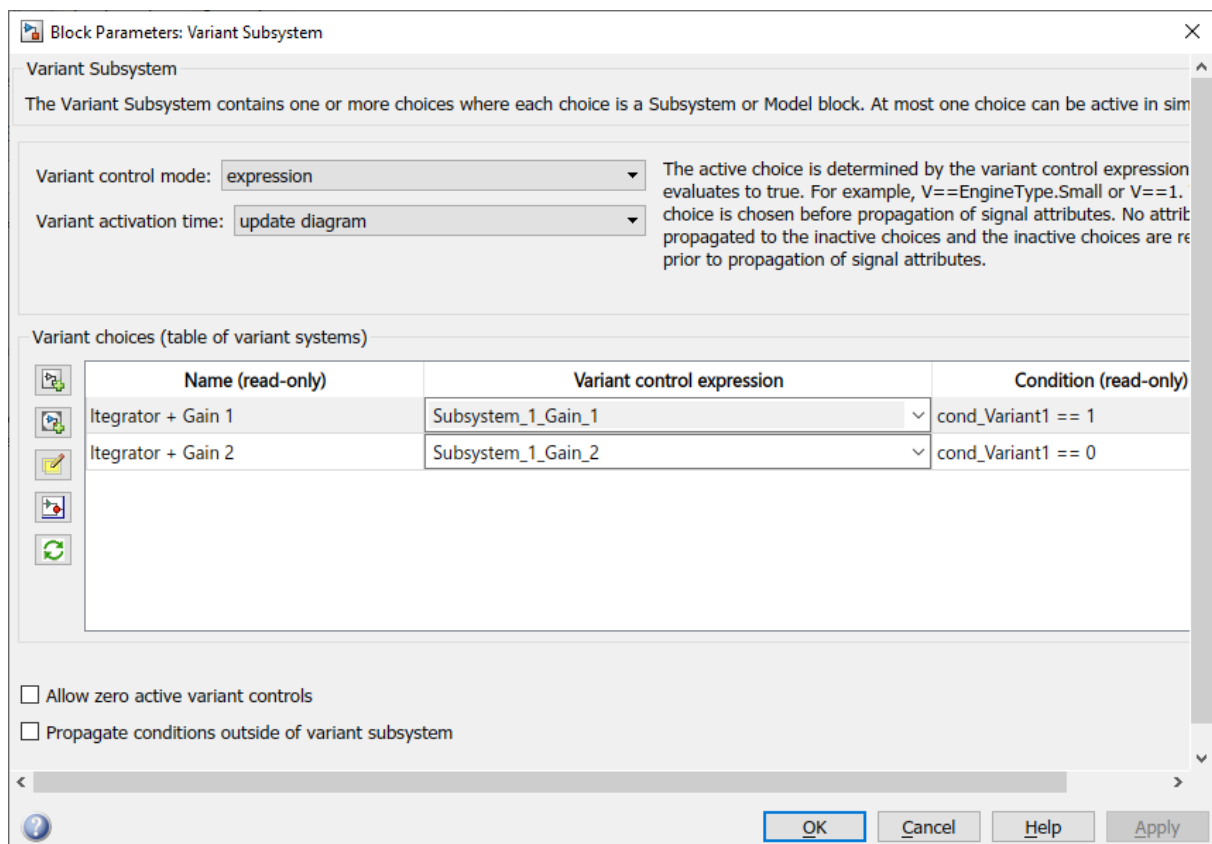


Рис. 3.12

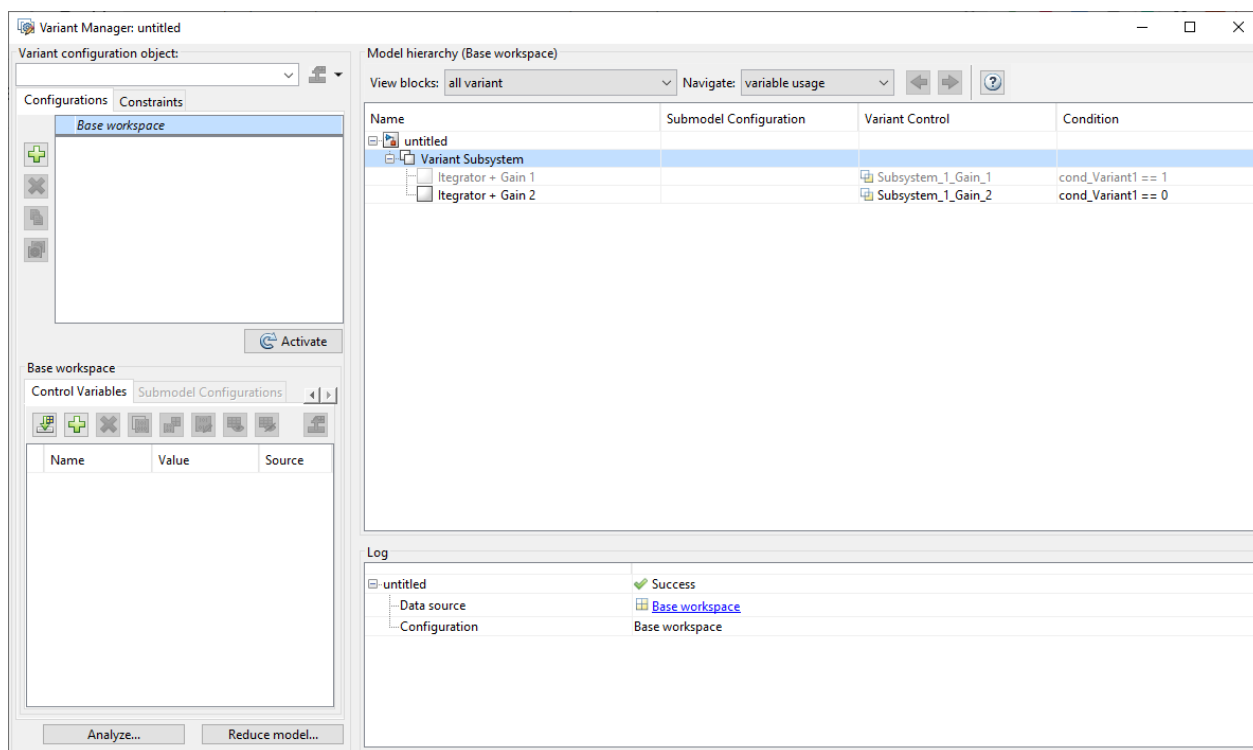


Рис. 3.13. Налаштування параметрів/(задання конфігурації) для вибору активної підсистеми

Іноді, для того щоб налаштування застосувались і коректно відображались, необхідно закрити та заново відкрити «*Variant Manager*».

10. Для активації тієї чи іншої підсистеми у «*Variant Subsystem*» (рис. 3.13) необхідно задати значення параметру, в нашому випадку `cond_Variant1 = 1` чи `0`.
11. Для одночасного порівняння підмоделей можна використати схему підключення як на рис. 3.14.

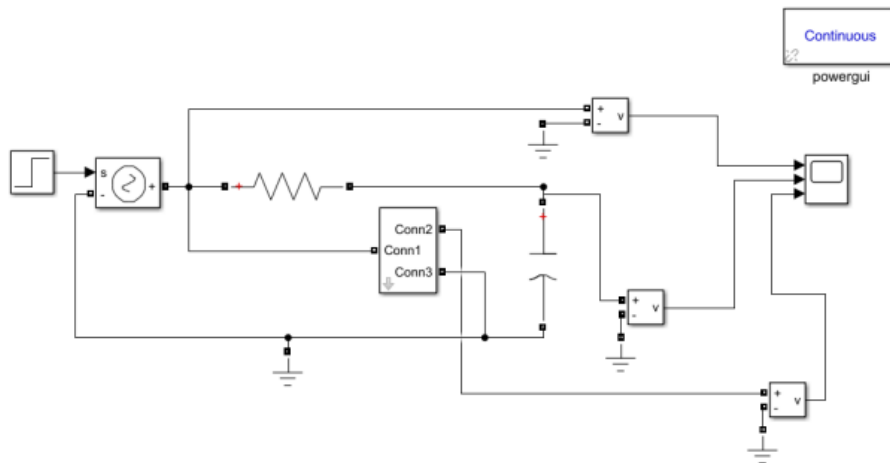


Рис. 3.14. Приклад схеми підключення двох підмоделей, для одночасного порівняння результатів моделювання

Контрольні запитання

1. Навіщо потрібні підмоделі/підсистеми?
2. Яким чином видалити підсистему, а її елементи повернути в модель?
3. Як створити власну бібліотеку моделей?
4. Як керувати параметрами моделі з командної строки Matlab?

Рекомендована література

1. Черних І.В. Моделювання електротехнічних пристроїв MATLAB, SimPowerSystems и Simulink. – 2008. – 288 с.
2. Гераїмчук, М. Д. Моделювання систем у середовищі MATLAB-SIMULINK [Електронний ресурс] : комп'ютерний практикум / М. Д. Гераїмчук, Ю. Ф. Лазарєв, Т. О. Толочко ; НТУУ «КПІ». – Київ : НТУУ «КПІ», 2006. – 175 с.
3. Моделювання систем керування в пакеті MATLAB SIMULINK, методичні вказівки до виконання лабораторної роботи по курсу «Комп'ютерні методи дослідження систем автоматичного управління»/ укл. : І.Р. Козбур, Г.В. Козбур , Р.І. Михайлишин. – Тернопіль : ТНТУ, 2019. – 23 с.

ЛАБОРАТОРНА РОБОТА № 4. АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ У ФАЗОВОМУ ПРОСТОРИ

Мета роботи – ознайомитись з нелінійними рівняннями для яких властива хаотична динаміка та отримати навички якісного та кількісного аналізу сигналів.

1. Короткі теоретичні відомості

Фазовий портрет – сукупність фазових траєкторій, що характеризує стан і рух динамічної системи.

Метод аналізу рівнянь у фазовому просторі – графоаналітичний метод дослідження динамічних систем, які описуються рівняннями виду:

$$\begin{aligned}\frac{dx}{dt} &= P(x, y), \\ \frac{dy}{dt} &= Q(x, y),\end{aligned}$$

де x та y – змінні стану системи, $P(x, y)$ та $Q(x, y)$ – функції, що задовольняють умові теорем існування та єдиності розв'язку, t – час (незалежна змінна). Поведінку такої системи можна представити геометрично на площині в прямокутних декартових координатах. При такому зображенні кожному стану динамічної системи однозначно відповідає точка на площині з координатами x, y і, навпаки, кожній точці площини відповідає один, і тільки один стан динамічної системи, що досліджується.

Площина Oxy називається фазовою площиною. Зміна стану системи відображається на фазовій площині рухом точки, яку називають фазовою.

Траєкторія, по якій рухається ця точка, називається фазовою траєкторією; швидкість і напрямок її руху визначаються вектором фазової швидкості $\{P, Q\}$. Суттєво, що через кожен точку фазової площини проходить тільки одна фазова траєкторія. Сукупність фазових траєкторій називається фазовим портретом системи і відображає сукупність всіх можливих сполучень системи і типи можливих рухів у ній.

Типи фазових рухів зазвичай визначають в околі так званих особливих точок, точок рівноваги, в яких

$$P(x, y) = 0, Q(x, y) = 0.$$

Різні типи руху системи у фазовій площині відокремлюються сепаратрисами.

Аналіз рівнянь у фазовому просторі полягає в побудові фазового портрета системи і подальшому аналізі цього портрета. Метод дозволяє визначити число, типи і характер особливих точок, наявність ізольованих замкнутих траєкторій і сепаратрис та дає можливість по вигляду фазових траєкторій наочно представити всю сукупність рухів, що виникають в динамічній системі при різних початкових умовах.

Особливі точки класифікують за характером фазових траєкторій в їх околі. Основні типи особливих точок зображені на рис. 1.1. Ізольовані замкнуті траєкторії (граничні цикли) класифікують за характером їх стійкості (рис. 1.2).

Теоретичні основи аналізу рівнянь у фазовому просторі розроблені А. Пуанкаре. Аналіз рівнянь у фазовому просторі – один з методів якісної теорії динамічних систем, і широко використовується в теорії коливань, теорії автоматичного управління, в електротехніці та механіці.

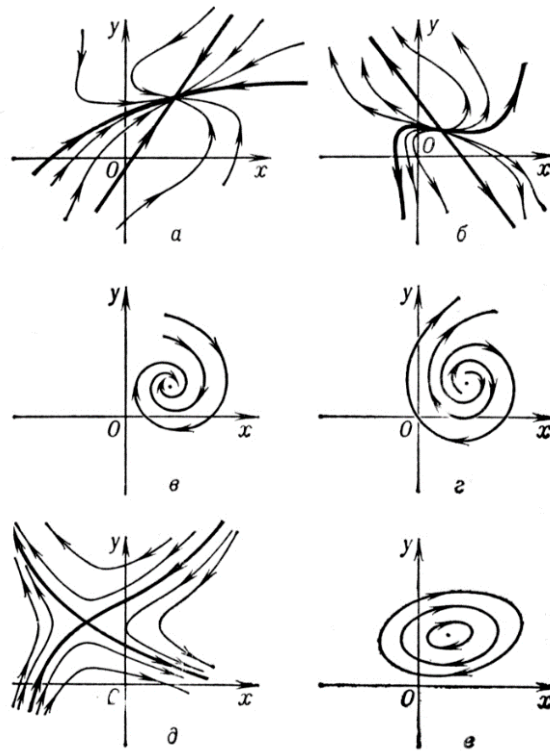


Рис. 1.1. Фазові траєкторії в околі особових точок: a – стійкий вузол; b – нестійкий вузол; c – стійкий фокус; d – нестійкий фокус; e – сідло; f – центр

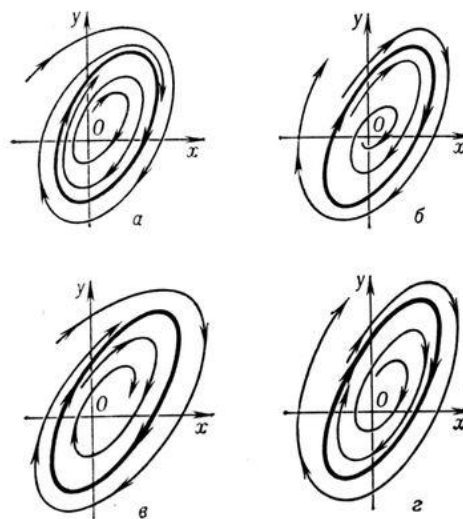


Рис. 1.2. Фазові траєкторії в околі різних граничних циклів, що зображені у вигляді замкнутих кривих: a – стійкий граничний цикл; b – нестійкий граничний цикл; c, d – напівстійкий граничний цикл

При комп'ютерному аналізі сигналів, останні представлені часто у вигляді дискретних відліків. Крім того, якщо сигнал є результатом фізичних вимірювань, майже завжди присутні шуми, в тому числі високочастотні. Це вкрай ускладнює оцінку похідної (швидкості). В цьому випадку будують так званий псевдофазовий портрет, опираючись на визначення похідної:

$$\text{похідна} \quad \frac{dx}{dt} = \lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t};$$

$$\text{похідна для дискретного ряду} \quad \frac{dx}{dt} \approx \frac{x_{i+1} - x_i}{\Delta t}.$$

Тобто, похідна при фіксованому Δt визначається x_{i+1} та x_i . Тож псевдофазовий портрет оцінюють як функціональну залежність $x_{i+1} = f(x_i)$. Однак, при досить мілкому кроці Δt сигнал між x_{i+1} та x_i змінюється квазілінійно. Тому необхідно підбирати інтервал Δt щоб коректно відобразити фазовий портрет: $x_{i+j} = f^j(x_i)$, де $j=1, 2, \dots, n$.

2. Завдання

2.1. Генерація даних

1. Розв'язати рівняння Лоренца (Edward N. Lorentz)

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(r - z) - y \\ \frac{dz}{dt} = xy - bz \end{cases} \quad (1)$$

при фіксованих $\sigma=10$, $b=8/3$ та при варіації параметра r ($r=3,7$; $r=10$; $r=28$; $r=30+N$, де N – номер варіанта).

2. Побудувати інтегральні криві рівняння (1) в просторі (x, y, z) та порівняти їх.

- а. Для варіантів розв'язків рівняння (1) при $r=3,7$ та $r=28$ побудувати графік функції $x=f(t)$ – сигнали A_1 та A_2 , відповідно.
- б. Для рівняння (1) побудуйте функцію $x=f(t)$ при $r=28$, але при різних значеннях похибки чисельного інтегрування.

3. За допомогою стандартних функцій Matlab згенерувати масиви даних C_1 та C_2 , що відображають випадковий гаусів закон розподілення цілих чисел від 1 до 21. Масив C_1 – 100 значень, C_2 – 10000 значень.

4. За допомогою стандартних функцій Matlab згенерувати синусоїдальний сигнал (D_0) 10 періодів.

5. За допомогою стандартних функцій Matlab згенерувати масиви D_1 та D_2 : додати до синусоїдального сигналу (D_0) випадкову складову інтенсивністю 5 % та 50% (рівень шуму повинен бути суттєвим) від максимальної амплітуди D_0 .

2.2. Аналіз даних

1. Побудувати гістограми функцій A_1 , A_2 , C_1 , C_2 , D_0 , D_1 , D_2 .
2. Побудувати псевдофазовий портрет модельних сигналів (A_2 , C_2 , D_0 , D_1 , D_2):

$$A_i = f(A_{i+1}), A_i = f(A_{i+5}), A_i = f(A_{i+10}), A_i = f(A_{i+j});$$

$$C_i = f(C_{i+1}), C_i = f(C_{i+5}), C_i = f(C_{i+10}), C_i = f(C_{i+j});$$

$$D_i = f(D_{i+1}), D_i = f(D_{i+5}), D_i = f(D_{i+10}), D_i = f(D_{i+j}).$$

Значення j підібрати таким, щоб фазовий портрет якісно відрізнявся від портретів з $j=1$ та $j=3$.

3. Побудувати автокореляційні функції для A_2 , C_2 , D_0 , D_1 , D_2 .
4. Оцінити математичне очікування, дисперсію, асиметрію та ентропію сигналів.

3. Протокол лабораторної роботи

Протокол включає:

- Назва роботи;
- Мета;
- Короткі теоретичні відомості (*необов'язкова вимога*);
- Завдання;
- Результати;
- Висновки.

Результати аналізу отриманих даних зводяться до таблиць та рисунків:

$r=3.7$	$r=10$
$r=28$	$r=30+N$

Рис. 3.1. Фазові траєкторії системи Лоренца при різних значеннях параметра r

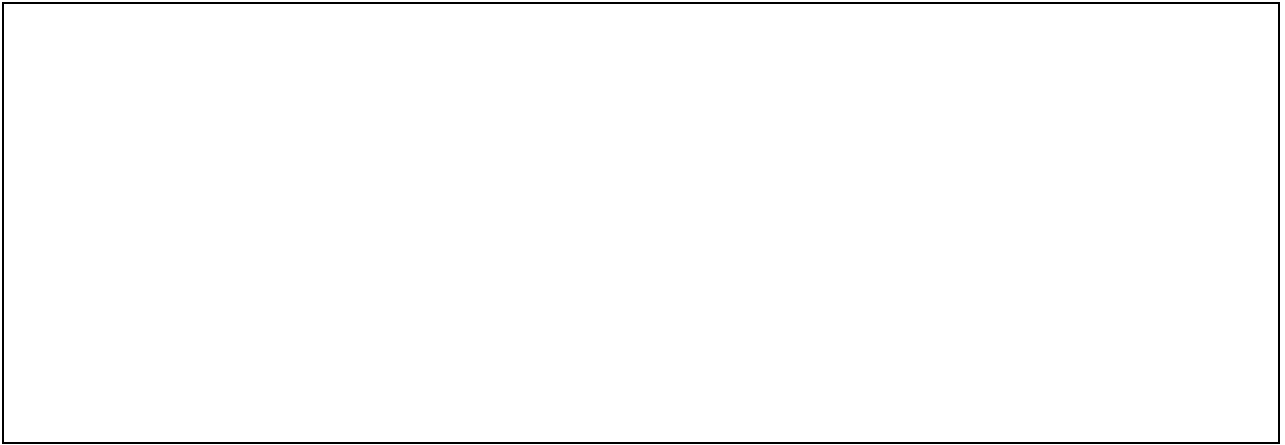


Рис. 3.2. Порівняння двох варіантів розв’язків системи Лоренца у часі для однієї із змінних (наприклад, x) при різних значеннях параметра r , де: зелена крива – $r=3.7$ (сигнал A_1), червона крива – $r=28$ (сигнал A_2)



Рис. 3.3. Порівняння двох варіантів розв’язків системи Лоренца у часі для однієї із змінних (наприклад, y) при однакових значеннях параметра $r=28$, при різних значеннях похибки чисельного інтегрування, де: зелена крива – $TOL=10^{-4}$, червона крива – $TOL=10^{-6}$

C_1	C_2

Рис. 3.4. Випадкові модельні сигнали

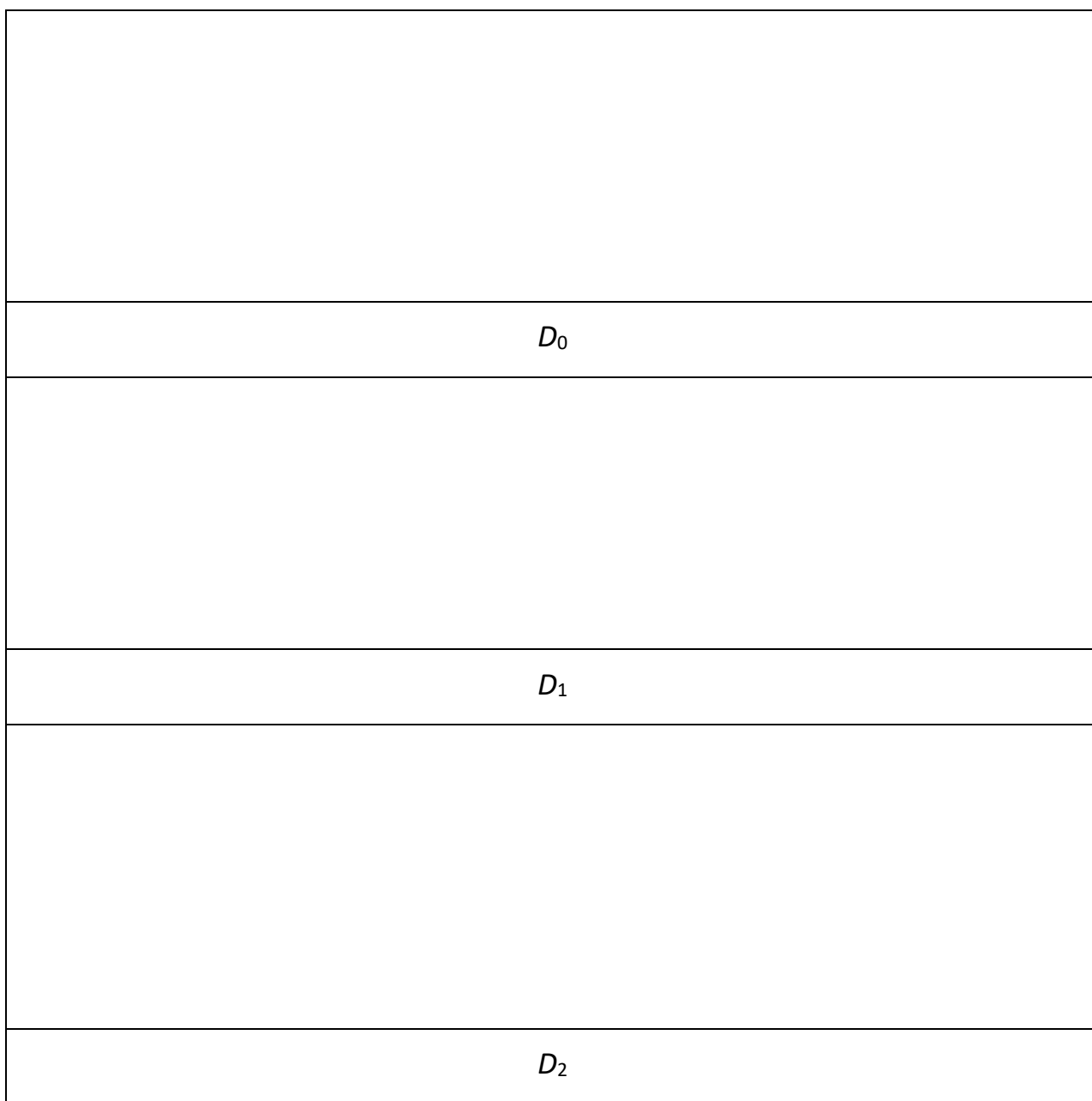


Рис. 3.5. Синусоїдальні модельні сигнали з шумом та без шуму

A_1	A_2

Рис. 3.6. Гістограми модельних сигналів A (системи Лоренца)

C_1	C_2

Рис. 3.7. Гістограми випадкових модельних сигналів C (випадкового сигналу)

D_0	D_1	D_2

Рис. 3.8. Гістограми випадкових модельних сигналів D (синусоїдального сигналу)

$j=1$	$j=5$
$j=$	

Рис. 3.9. Псевдофазовий портрет модельного сигналу A_2 для змінної x

$j=1$	$j=5$
$j=10$	$j=$

Рис. 3.10. Псевдофазовий портрет модельного сигналу C_2

$j=1$	$j=5$
$j=10$	$j=$

Рис. 3.11. Псевдофазовый портрет модельного сигнала D_0

$j=1$	$j=5$
$j=10$	$j=$

Рис. 3.12. Псевдофазовый портрет модельного сигнала D_1

$j=1$	$j=5$
$j=$	

Рис. 3.13. Псевдофазовий портрет модельного сигналу D_2

Таблиця 3.1. Результати статистичного аналізу даних

Модельний сигнал	Математичне очікування	Дисперсія	Ентропія	Асиметрія
A_1				
A_2				
C_1				
C_2				
D_0				
D_1				
D_2				

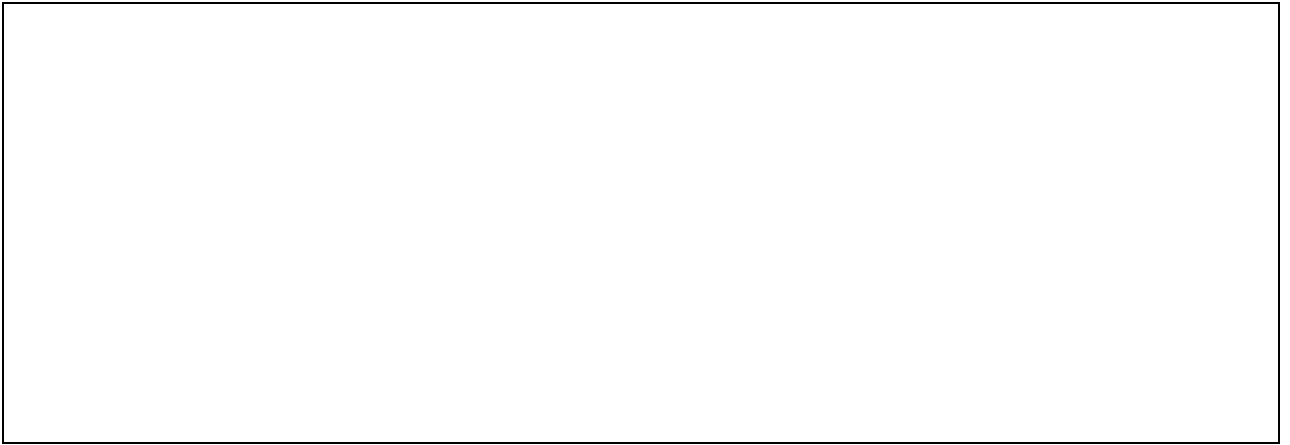


Рис. 3.14. Автокореляційна функція модельного сигналу A_1

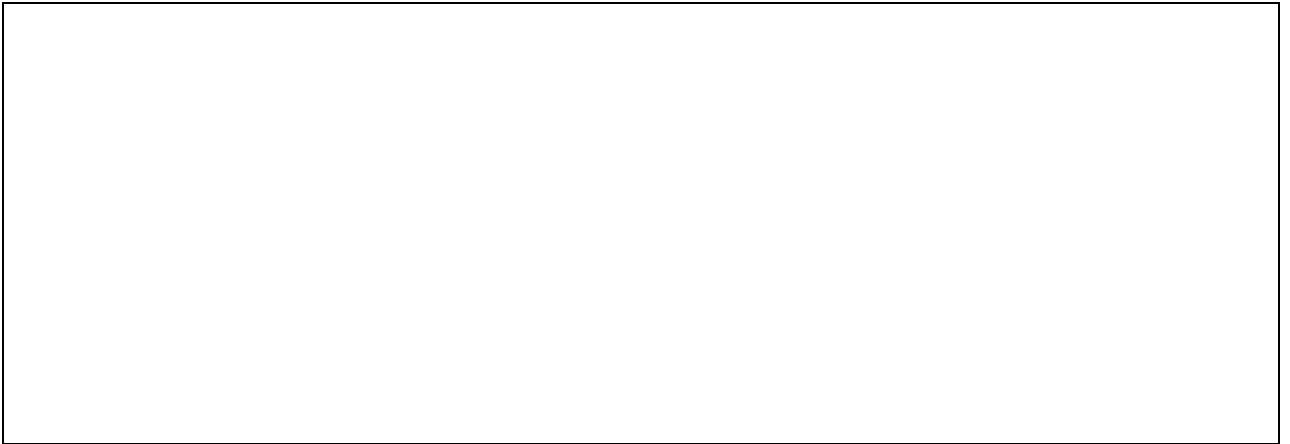


Рис. 3.15. Автокореляційна функція модельного сигналу C_2

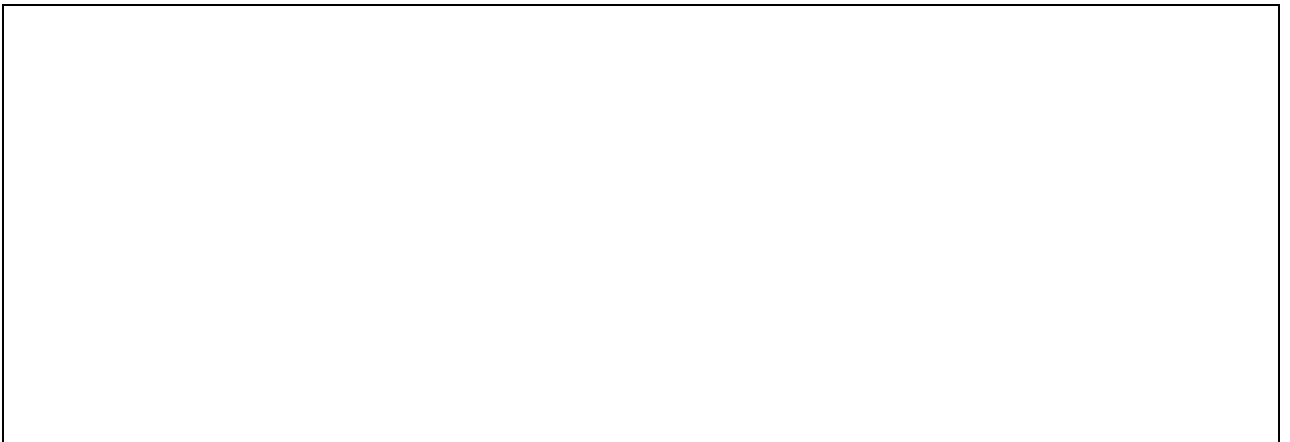


Рис. 3.16. Автокореляційна функція модельного сигналу D_0

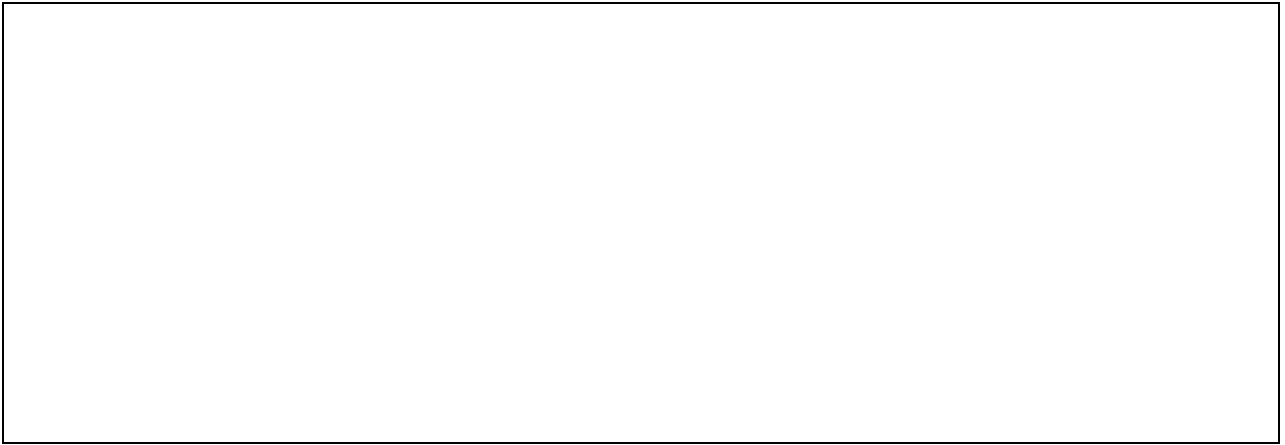


Рис. 3.17. Автокореляційна функція модельного сигналу D_1

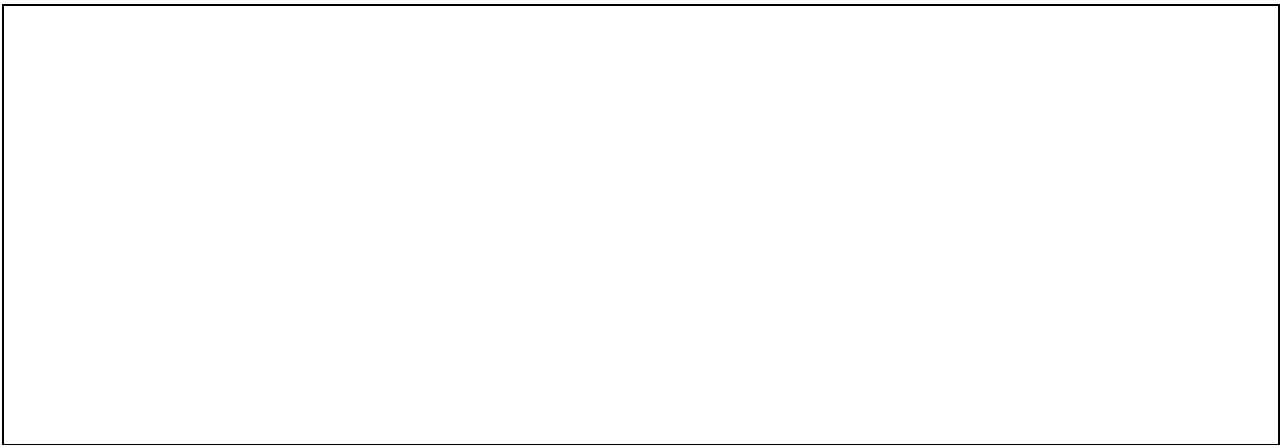


Рис. 3.18. Автокореляційна функція модельного сигналу D_2

4. Методичні рекомендації до виконання роботи

1. Рекомендації щодо програмного коду вирішування рівняння (1):

```
global S R B
S=5; B=1;
Rn=[0.3 1.8 3.7 10 16 24.06 28];
options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);

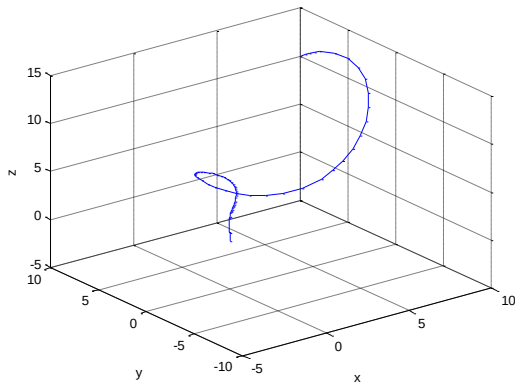
for i = 1:length(Rn);
    R=Rn(i);
    [t DD]=ode45(@F2,[0 100],[10 10 10], options);
    x=DD(:,1); y=DD(:,2); z=DD(:,3);
    figure
    plot3(x,y,z); grid on;
    title(R)
    xlabel('x')
    ylabel('y')
    zlabel('z')
end
```

F2 – ім'я m-файлу функції рівняння (1) (F2.m).

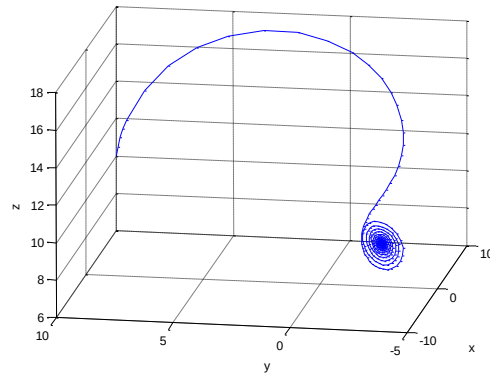
Програмний код функції F2.m:

```
function dsdt=F2(t,s)
global S R B % параметри рівняння
n=length(s)/3;
X=s(1:n);
Y=s(n+1:2*n);
Z=s(2*n+1:3*n);
dXdT=S.*(Y-X);
dYdt=X.*(R-Z)-Y;
dZdt=X.*Y-B.*Z;
dsdt=[dXdT;dYdt;dZdt];
```

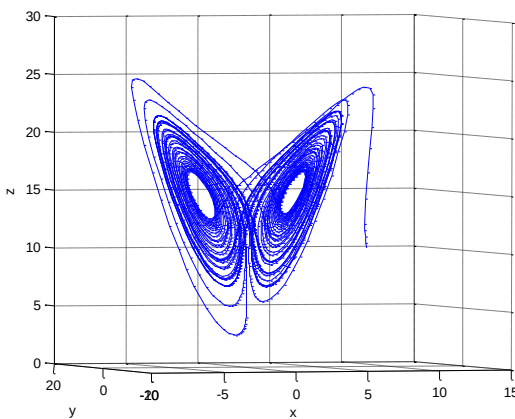
Типові результати розв'язання системи (1) наведені на рис. 4.1.



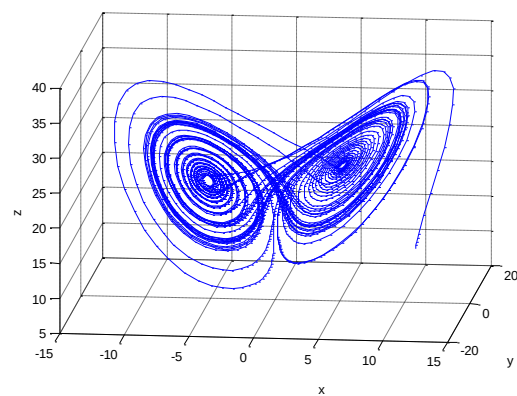
$r = 0,3$



$r = 10$



$r = 16$



$r = 24,06$

Рис. 4.1. Варіанти розв'язання системи диференціальних рівнянь Лоренца для завдання 2 при різних значеннях параметра r

Слід звернути увагу що за замовчуванням вирішувач диференціальних рівнянь, такий як `ode45`, що наведено в прикладі, інтегрує систему зі змінним кроком. Тобто Δt змінюється в залежності від ступеня нелінійності кривих на інтервалі обчислень. Це вкрай ускладнює подальший спектральний, кореляційний та статистичний наліз даних. Тому, після отримання інтегральних кривих доцільно застосувати сплайн-інтерполяцію вже зі сталим кроком. В системі Matlab цю процедуру можна здійснити за допомогою команд *spline* або *csaps* з *ppval*. Інакше діапазон інтегрування необхідно задати вектором, з вказанням Δt , наприклад, `[0: 0.01:100]`. Але це може бути неоптимально з точки зору обчислювальних ресурсів і похибки інтегрування.

2. Команда генерації випадкового нормального розподілення в системі Matlab:

`A=randn (n1, n2),`

де A – масив даних, n – розмірність масиву A ($n1$ – кількість строк, $n2$ – кількість стовпчиків).

3. Приклад побудови гістограми та розрахунку статистичних даних:

```
%% Функція розподілу ймовірності
N=20; % кількість стовпчиків
[h, x]=hist(A, N);
p=h./length(A); % перехід від абсолютних значень на гістограми до відносних,
тобто до розподілу ймовірності
hist(A, N); % візуалізація гістограми чи bar(x,h)

%% Розрахунок статистичних показників
```

Майже для кожного стандартного статистичного показника є окремі функції. Наприклад `mean` – команда для розрахунку середнього значення, `std` – середньоквадратичне відхилення. Але в деяких задачах для мінімізації машинних ресурсів бажано розрахунки проводити за гістограмою.

```

% Розрахунок середнього значення
% Розрахунок середнього значення за гістограмою
tmpSum = 0;
for i = 1:N
    tmpSum = tmpSum + x(i)*p(i);
end
Avarage_0 = mean(A); % стандартна функція в Matlab
Avarage_p= tmpSum; % якщо p – ймовірність події
% якщо нормування не було проведено, тобто є тільки h, тобто відображає
% абсолютну кількість подій, то %Avarage_p= tmpSum/length(A);

% розрахунок середньоквадратичного відхилення
tmpSum = 0;
for i = 1 : N
    tmpSum = tmpSum + p(i)*(M_av - x(i))^2;
end
Sigma_0=std(A) % стандартна функція в Matlab
Sigma_p = (tmpSum)^0.5

% розрахунок асиметрії

```

За визначенням, асиметрія розраховується за формулою:

$$A_s = \frac{\mu_3}{\sigma^3},$$

де σ – середньоквадратичне відхилення (за кодом вище воно позначалось як Sigma_0 чи Sigma_p), μ_3 – третій центральний момент:

$$\mu_3 = E[(X - EX^3)],$$

де E – оператор, що позначає середню величину.

```

tmpSum = 0;
for i = 1:N
    tmpSum = tmpSum + p(i)*(x(i)-M_av)^3;
end
if Sigma_h~=0
    Asimmetry_p = tmpSum/Sigma_p^3;
else
    Asimmetry_p =0;
end

Asymmetry_0= skewness(A) % стандартна функція в Matlab
Asimmetry_p

```

Як можна побачити з наведених прикладів розрахунку цих статистичних показників, для кожного з них застосовується цикл `for i = 1:n`.

Для задач, де необхідно одночасно розраховувати відповідні значення в реальному часі, представлені фрагменти коду доцільно об'єднати, що суттєво зменшує машинні ресурси. Це ж стосується і розрахунку показника ентропії за Шенноном. Для розрахунку ентропії за Шенноном використовують формулу:

$$E = - \sum_{i=1}^n p_i \ln p_i .$$

Безпосередній розрахунок ентропії в Matlab окремою командою здійснюється так:

```
entrop=abs(wentropy(p,'shannon')) .
```

Розрахунок дисперсії:

```
disp=std(p)^2 .
```

4. Приклад програмного коду побудови автокореляційної функції Пірсона:

```
k=length(A);
R(1)=1; % при нульовому зміщенні автокореляційна функція завжди =1
for i=1:k-2
    a1=A(1:k-i);
    a2=A(i+1:k);
    tem=corrcoef(a1,a2);
    R(i+1)=tem(1,2);
end
```

В інтегральній формі для нескінченного сигналу автокореляційна функція має вигляд:

$$R(\tau) = \int f(t) \cdot f(t - \tau) dt .$$

Автокореляція Пірсона визначається за формулою:

$$r_k = \frac{\sum_{t=k+1}^n (y_t - \overline{y_{2k-1}})(y_{t-k} - \overline{y_{2k}})}{\sqrt{\sum_{t=k+1}^n (y_t - \overline{y_{2k-1}})^2 \sum_{t=k+1}^n (y_{t-k} - \overline{y_{2k}})^2}},$$

де

$$\overline{y_{2k-1}} = \frac{1}{n-k} \cdot \sum_{t=k+1}^n y_t,$$

$$\overline{y_{2k}} = \frac{1}{n-k} \cdot \sum_{t=k+1}^n y_{t-k}.$$

Значення автокореляційної функції Пірсона знаходиться в межах $[-1;1]$, оскільки $|\text{cov}_{xy}| \leq \sqrt{D_x D_y}$, де D – дисперсія відповідної вибірки.

Контрольні запитання

1. Що таке фазовий простір?
2. Як трактувати функцію автокореляції і навіщо необхідно нормування функції на дисперсію вибірки?
3. Класифікація особливих точок системи.
4. Що означають: математичне очікування, середнє квадратичне відхилення, дисперсія, асиметрія, ексцес, ентропія, мода, медіана?
5. Принцип Пригожина для біологічних систем.
6. Основні властивості атрактора Лоренца.
7. Надайте визначення й пояснення терміну «емерджентність», «синергетичність».
8. Яким чином можна передати параметри/коефіцієнти у функцію при розв'язуванні диференціальних рівнянь?
9. Якими кількісними критеріями характеризуються хаотичні, стохастичні/випадкові сигнали?

Рекомендована література

1. James J. Scully, Alexander B. Neiman, Andrey L. Shilnikov; Measuring chaos in the Lorenz and Rössler models: Fidelity tests for reservoir computing. Chaos 1 September 2021; 31 (9): 093121. <https://doi.org/10.1063/5.0065044>

2. Лошицький П.П. Моделювання біофізичних процесів. Вступ до синергетики.[текст]: Навчальний посібник /П.П.Лошицький, М.О.Ніколов – Київ.: НТУУ»КПІ», 2014. – 412 с.
3. Лазарєв Ю. Ф. Моделювання динамічних систем у Matlab. Електронний навчальний посібник. – Київ: НТУУ "КПІ", 2011. – 421 с. – Режим доступу: https://elprivod.nmu.org.ua/files/mathapps/mds_matlab.pdf
4. В.Б. Хоцкіна, І.Н. Вдовиченко Робота в пакеті MATLAB: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. – 130 с. – Режим доступу: <https://dSPACE.duet.edu.ua/jspui/bitstream/123456789/851/1/%d0%9d%d0%9f%20MatLab.pdf>
5. Сплайн-функції та їх застосування / Б.П. Довгий, А.В. Ловеїкін, Є.С. Вакал, Ю.Є. Вакал. – К.:Видавничо-поліграфічний центр “Київський університет”, 2016. – 117 с. – Режим доступу: https://mp.mechmat.knu.ua/images/library/Dovhyi_Loveikin_Vakal_Vakal_2016_Spline_Tutor.pdf.

ЛАБОРАТОРНА РОБОТА № 5. МОДЕЛЮВАННЯ ДИВНОГО АТРАКТОРА ЛОРЕНЦА ТА РЬОССЛЕРА В СИСТЕМІ SIMULINK

Мета роботи – отримання навичок розв’язування систем диференціальних рівнянь в системі Simulink на прикладі моделювання хаотичної динаміки систем Лоренца та Рьосслера.

1. Короткі теоретичні відомості

В даному підрозділі зупинимося на особливостях дивного атрактора Рьосслера (нім. Otto Rössler). Відповідна система диференціальних рівнянь має вигляд:

$$\begin{cases} \frac{dx}{dt} = -y - x \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases}$$

де a, b, c – додатні постійні. При значеннях параметрів $a=b=0.2$ і $2.6 \leq c \leq 4.2$ рівняння Рьосслера мають стійкий граничний цикл. За цих значень параметрів у системі відбувається каскад подвоєння періоду. При значенні 4.2 виникає хаотичний атрактор. Чіткі певні лінії граничних циклів розпливаються і заповнюють фазовий простір нескінченною безліччю траєкторій, що мають властивості фракталу.

Двоє з рівнянь системи Рьосслера лінійні. При $z=0$ вони набувають вигляду

$$\begin{cases} \frac{dx}{dt} = -y - x \\ \frac{dy}{dt} = x + ay \end{cases}$$

Тому стійкість руху у площині $z=0$ визначається власними значеннями матриці Якобі

$$\begin{pmatrix} 0 & -1 \\ 1 & a \end{pmatrix},$$

які дорівнюють

$$\lambda_{1,2} = \frac{a \pm \sqrt{a^2 - 4}}{2}.$$

Коли $0 < a < 2$ власні значення мають позитивну дійсну частину і комплексно спряжені. Тому фазові траєкторії розходяться від початку координат по спіралі.

При z , що не дорівнює 0, та, фіксуючи a в діапазоні $0 < a < 2$, отримуємо: поки x менше c , множник $(x - c)$ у рівнянні dz/dt буде утримувати траєкторію близькою до площини xu . Як тільки x стане більше c , z -координата почне рости. У свою чергу, великі значення z будуть гальмувати ріст x .

Утворення хаосу в цілому можна класифікувати через механізм подвоєння періоду. Це добре видно на біфуркаційних діаграмах (рис. 1.1).

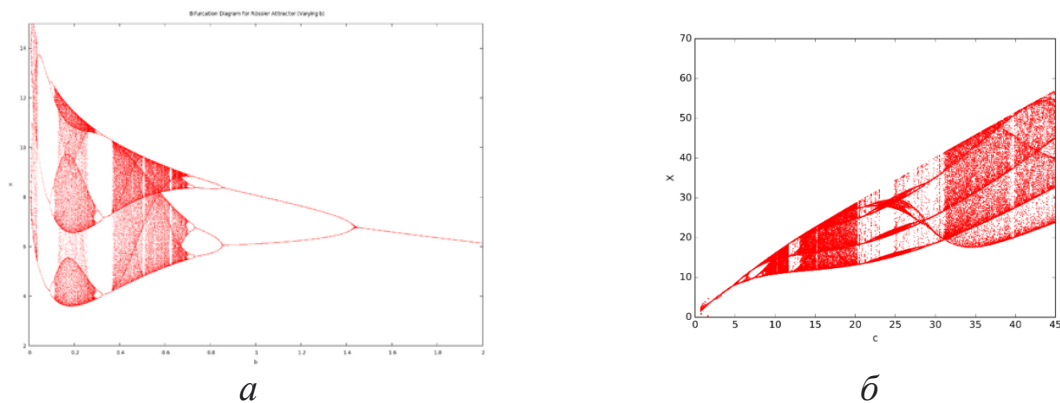


Рис. 1.1. Біфуркаційна діаграма системи Рюсслера. На осі абсцис – параметр, на осі ординат – положення особливої точки x : a – при сталих значеннях $a=0.2$, $c=5.7$ і змінному параметрі b від 0 до 2; b – при $a=b=0.1$ та змінному параметрі c від 0 до 45

Безпосередньо, типовий фазовий портрет системи Рьюсслера в хаотичному режимі має вигляд як на рис. 1.2.

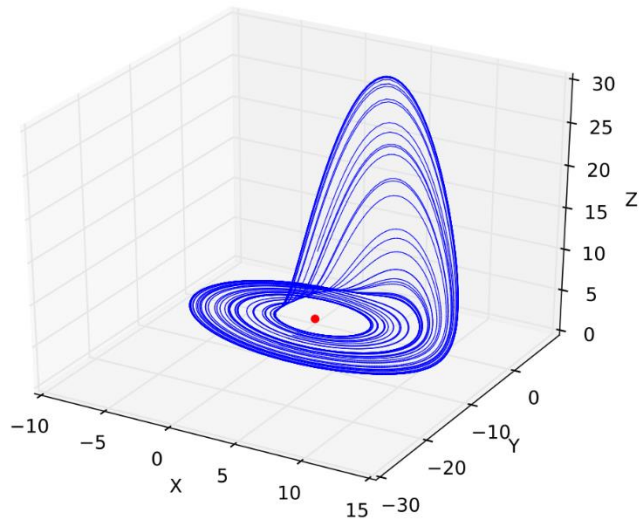


Рис. 1.2. Типовий фазовий портрет системи Рьюсслера в хаотичному режимі, зокрема, при $a=0.2$, $b=0.2$, $c=5.7$

2. Завдання

1. Побудувати Simulink-модель системи диференціальних рівнянь Лоренца та отримати її чисельний розв'язок:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(r - z) - y \\ \frac{dz}{dt} = xy - bz \end{cases} \quad (1)$$

Параметри системи вважати сталими: $\sigma=10$, $b=8/3$, $r=28+N$, де N – номер варіанта).

2. Побудувати Simulink-модель системи диференціальних рівнянь Рьосслера та отримати її чисельний розв'язок при різних значеннях параметрів:

$$\begin{cases} \frac{dx}{dt} = -y - x \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases} \quad (2)$$

Параметри системи a та b вважати сталими: $a = b = 0.1$, параметр c приймає значення $c = 4$; $c = 6$; $c = 8.5$; $c = 18$.

3. В хаотичному режимі для побудови трьохвимірного фазового портрету атрактора Рьосслера результати моделювання зберегти в робочій області Matlab.

4. Порівняти спектри Фур'є в хаотичному режимі системи Лоренца та Рьосслера.

3. Методичні вказівки

Приклад Simulink-моделі системи Лоренца може мати вигляд, наведений на рис. 3.1.

Контрольні запитання

1. Чим на якісному рівні відрізняється хаотична поведінка системи Лоренца та Рьосслера?
2. Які Фур'є- спектральні особливості для системи Лоренца й Рьосслера?
3. Як оцінити спектр Фур'є в системі Simulink?
4. Що характеризує показник Ляпунова?
5. Які фізичні явища моделює система Лоренца?

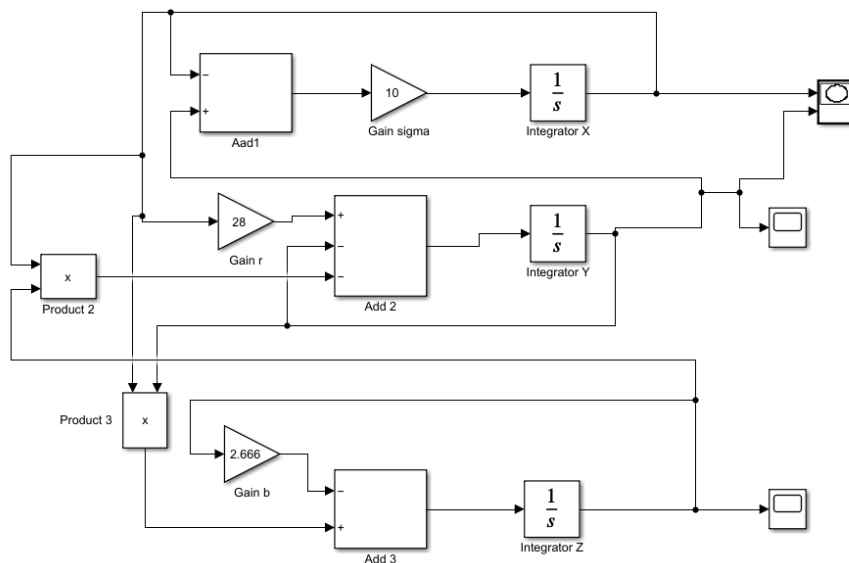


Рис. 3.1. Приклад Simulink-моделі системи Лоренца

Рекомендована література

1. James J. Scully, Alexander B. Neiman, Andrey L. Shilnikov; Measuring chaos in the Lorenz and Rössler models: Fidelity tests for reservoir computing. *Chaos* 1 September 2021; 31 (9): 093121. <https://doi.org/10.1063/5.0065044>
2. Лошицький П.П. Моделювання біофізичних процесів. Вступ до синергетики.[текст]: Навчальний посібник /П.П.Лошицький, М.О.Ніколов – Київ.: НТУУ»КПІ», 2014.-412с.
3. Wikipedia: Rössler attractor. – https://en.wikipedia.org/wiki/R%C3%B6ssler_attractor

ЛАБОРАТОРНА РОБОТА № 6. ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВНОГО ПРОГРАМУВАННЯ

Мета – отримання практичних навичок застосування об'єктно-орієнтованого програмування в програмному середовищі Matlab.

1. Теоретичні відомості

Визначення класу. Клас можна визначити за допомогою *classdef* у файлі *.m* з тим самим ім'ям, що й клас. Файл може містити безпосередні методи класу в блоці *classdef ... end* та локальні функції для використання у методах класу.

Найбільш загальне визначення класу MATLAB має таку структуру:

```
classdef (ClassAttribute = expression, ...) ClassName < ParentClass1 &
ParentClass2 & ...

    properties (PropertyAttributes)
        PropertyName
    end

    methods (MethodAttributes)
        function obj = methodName(obj, arg2, ...)
            ...
        end
    end

    events (EventAttributes)
        EventName
    end

    enumeration
        EnumName
    end

end
```

При створенні класу бажано встановити його параметри/властивості (properties) за умовчанням. Для цього використовують конструктор. Це звичайний метод, але при коректному програмуванні з дотриманням правил

програмування він визивається автоматично при створенні класу. *Конструктор* – це метод екземпляра, який зазвичай має те саме ім'я, що й клас.

```
classdef Car < handle % handle class so properties persist
    properties
        make
        model
        mileage = 0;
        cost
    end

    methods
        function obj = Car(make, model)
            obj.make = make;
            obj.model = model;
            obj.cost = [];
        end
        function drive(obj, milesDriven)
            obj.mileage = obj.mileage + milesDriven;
        end
    end
end
```

Екземпляр цього класу може бути створений за допомогою виклику конструктора:

```
>> myCar = Car('Ford', 'Mustang'); //creating an instance of car class
```

Виклик методу drive збільшить пробіг:

```
>> myCar.mileage

ans =
     0

>> myCar.drive(450);

>> myCar.mileage

ans =
    450
```

Виклик конструктора дочірнього класу. Класи можуть бути успадковані від батьківських класів. Ці батьківські класи можуть потребувати виклику власних конструкторів. Ця задача вирішується наступним чином.

Створимо батьківський клас Person

```
classdef Person
    properties
        name
        surname
        address
    end

    methods
        function obj = Person(name,surname,address)
            obj.name = name;
            obj.surname = surname;
            obj.address = address;
        end
    end
end
```

Створимо дочірній клас Member

```
classdef Member < Person
    properties
        payment
    end

    methods
        function obj = Member(name,surname,address,payment)
            obj = obj@Person(name,surname,address);
            obj.payment = payment;
        end
    end
end
```

Для виклику конструктора успадкованого класу Person код має вигляд

```
obj = obj@Person(name,surname,address);
```

Класи Value vs Handle. Класи в MATLAB поділяються на дві основні категорії: класи *значень* та класи *дескрипторів*. Основна відмінність полягає в тому, що при копіюванні екземпляра *класу значень* базові дані копіюються в новий екземпляр, тоді як для *класів дескрипторів* новий екземпляр вказує на вихідні дані, та зміна значень у новому екземплярі змінює їх у оригіналі. Клас може бути визначений як дескриптор шляхом успадкування класу handle.

```

classdef valueClass
    properties
        data
    end
end
classdef handleClass < handle
    properties
        data
    end
end
end

```

Порівняємо їх роботу.

```

>> v1 = valueClass;
>> v1.data = 5;
>> v2 = v1;
>> v2.data = 7;
>> v1.data
ans =
     5

```

Тут значення data екземпляра v1 не змінилося після створення та ідентифікації екземпляра v2. Тобто v1 та v2 є незалежними.

В класі handleClass зміна параметру в екземлярі h2 означає й зміну значення в h1.

```

>> h1 = handleClass;
>> h1.data = 5;
>> h2 = h1;
>> h2.data = 7;
>> h1.data
ans =
     7

```

Властивості і методи. Властивості – це параметри, що характеризують стан об’єкта. Методи – це функції, які визначають поведінку об’єкта.

Для захисту властивостей від їх зміни в основній програмі їх можна захистити:

```

classdef ScreenLogger
    properties (Access=protected)
        scrh;
    end

    methods
        function obj = ScreenLogger(screenhandler)
            obj.scrh = screenhandler;
        end
    end
end

```

```

        function LogMessage(obj, varargin)
            if ~isempty(varargin)
                varargin{1} = num2str(varargin{1});
                fprintf(obj.scrh, '%s\n', sprintf(varargin{:}));
            end
        end
    end
end
end

```

В даному прикладі властивість `scrh` захищена від зміни у зовнішній програмі; код

```
mylogger.scrh = 2;
```

викликає помилку. Змінити значення `scrh` можливо тільки за допомогою окремого методу класу:

```

classdef ScreenLogger
    properties (Access=protected)
        scrh;
    end
    properties
        a
    end
    methods
        function obj = ScreenLogger(screenhandler)
            obj.scrh = screenhandler;
            obj.a=0;
        end

        function LogMessage(obj, varargin)
            if ~isempty(varargin)
                varargin{1} = num2str(varargin{1});
                fprintf(obj.scrh, '%s\n', sprintf(varargin{:}));
            end
        end
        function obj=set_scrh(obj, value)
            obj.scrh = value;
        end
        function obj=add_a_scrh(obj)
            obj.a = obj.a+obj.scrh;
        end
    end
end
end

```

```

mylogger = ScreenLogger(1);
>> mylogger = ScreenLogger(1);

>> mylogger=mylogger.set_scrh(2);

>> mylogger=mylogger.add_a_scrh;

mylogger.a =2

```

Зверніть увагу, що в робочій області властивість `scrh` не візуалізується.

Більш детально про атрибути властивостей дивіться на веб сторінці центра допомоги Matlab: https://www.mathworks.com/help/matlab/matlab_oop/property-attributes.html

2. Завдання

1. Створити клас, що генерує модельні сигнали – *class_Model_Signals*.

Цей клас повинен мати можливість генерувати щонайменше три модельні сигнали F_1, F_2, F_3 :

- $F_1(t) = \sin(2\pi f_1 t + \varphi_1) + \sin(2\pi f_2 t + \varphi_2) + \sin(2\pi f_3 t + \varphi_3)$;
- $F_2(t) = \sin(2\pi f_1 t + \varphi_1) + 0.5 * randn$;
- фракталоподібний шумовий сигнал на основі функції Вейерштрасса:

$$F_3(t) = \sqrt{2}\sigma \frac{\left[1 - b^{(2D-4)}\right]^{0.5} \sum_{n=0}^N b^{(D-2)n} \sin(2\pi f s^n t + \psi_n)}{\left[1 - b^{(2D-4)(N+1)}\right]^{0.5}},$$

де σ – стандартне відхилення, s, f – параметри просторово-частотного масштабування, D – фрактальна розмірність, яка пов'язана з параметром Хьорста (Херста) для одновимірних сигналів співвідношенням $D' = 2 - D$, $N+1$ – кількість гармонік, ψ_n – фаза n -тої гармоніки, що розподілена випадковим чином, t – час.

Рекомендовані значення параметрів:

$s = f_1$, $b = 1.2$, $D = 1.3$, $N = 10$, ψ – розподілена за нормальним законом в межах $[0..2\pi]$.

Загальний час моделювання не менш 30 періодів; f_1, f_2 , та f_3 – різні, але співрозмірні, одного порядку, які відрізняються одна від одної не більш ніж в 4 рази.

При цьому, одиниці виміру часу – с, F_1 – мВ, F_2 – °C, F_3 – мА.

2. Створити клас (`class_Signal_Analysis`), основна задача якого проводити аналіз сигналів, зокрема спектральний (Фур'є спектр) та автокореляційний аналіз.

3. Створити клас (`class_Signal`), що зберігає та аналізує кожний сигнал окремо. При цьому `class_Signal` успадковує властивості `class_Spectr`.

`class_Signal` повинен забезпечувати наступні властивості:

- додавати поточне значення амплітуди та часу до сигналу;
- через фіксований час ($\Delta t \sim 10 \frac{1}{f_1}$) необхідно побудувати сигнал, спектр

та автокореляційну функцію (модель аналізу сигналу в реальному часі);

- містити інформацію про ім'я сигналу та одиниці виміру, розмір часового вікна аналізу Δt .

Контрольні запитання

1. Які переваги має ООП в порівнянні з лінійним програмуванням?
2. Які бувають атрибути властивостей об'єктів?
3. Що таке інкапсуляція в ООП?
4. Що таке поліморфізм?
5. При яких умовах виникає режим модуляції/биття при накладанні двох та більше синусоїдальних сигналів?
6. Які спектральні особливості фракталоподібного сигналу, що досліджувався?

Рекомендована література

1. Revinskaya O.G. Flexibility of using input and output parameters of standard and non-standard functions in MatLab // Software & Systems. - 2019, vol. 32, no. 1, pp. 042–054. DOI: 10.15827/0236-235X.125.042-054
2. DataSheet Matlab: Classes – Режим доступу: <https://www.mathworks.com/help/matlab/object-oriented-programming.html>
3. Об'єктно-орієнтоване програмування: конспект лекцій: навч. посіб. для студ. освітньої програми «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення» / Порєв В.М.; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 271 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/854443cb-3375-4e76-961a-1f9a74a15b07/content>
4. Об'єктно-орієнтоване програмування мовою Python: навчальний посібник для студентів напрямів підготовки 122-Комп'ютерні науки та 121-Інженерія програмного забезпечення / О.В. Мнушка, В.М. Савченко, О.Б. Маций – Х.: ХНАДУ, 2021. – 228 с.
5. Замуруєва О. В., Кримусь А. С., Ольхова Н. В. Об'єктно-орієнтоване програмування в Python : курс лекцій. Луцьк : Вежа-Друк, 2018. – 64 с. - <https://evnuir.vnu.edu.ua/bitstream/123456789/14344/1/Python.pdf>

ЛАБОРАТОРНА РОБОТА № 7. МОДЕЛЮВАННЯ ПОШИРЕННЯ АКУСТИЧНИХ ХВИЛЬ У СЕРЕДОВИЩАХ МЕТОДОМ СКІНЧЕННИХ РІЗНИЦЬ

Мета роботи – отримання навичок розв’язування хвильових рівнянь, зокрема поширення акустичних хвиль у середовищах, методом скінченних різниць.

1. Теоретичні відомості

1.1. Основні класичні типи рівнянь математичної фізики

Приведемо деякі найбільш важливі рівняння з частинними похідними. Для спрощення запису будемо використовувати наступні позначення:

$$u_t \equiv \frac{\partial u}{\partial t}, \quad u_x \equiv \frac{\partial u}{\partial x}, \quad u_{xx} \equiv \frac{\partial^2 u}{\partial x^2}, \dots$$

Деякі найбільш типові рівняння з частинними похідними:

$u_t = u_{xx}$	(одновимірне рівняння теплопровідності)
$u_t = u_{xx} + u_{yy}$	(двовимірне рівняння теплопровідності)
$u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} = 0$	(рівняння Лапласа у полярних координатах)
$u_{tt} = u_{xx} + u_{yy} + u_{zz}$	(тривимірне хвильове рівняння)
$u_{tt} = u_{xx} + \alpha u_t + \beta u$	(телеграфне рівняння)

У всіх наведених прикладах невідома функція u залежить більш ніж від однієї змінної. Змінна u (яку ми диференціюємо) називається *залежною змінною*. Змінні, за якими відбувається диференціювання, називаються *незалежними*

змінними. Наприклад, у рівнянні $u_t = u_{xx}$ залежна змінна $u(x, t)$ є функцією двох незалежних змінних x та t .

Для знаходження аналітичного розв'язку цих рівнянь (рівнянь з частинними похідними) найпоширенішими методами є:

1. **Метод розділення змінних.** Рівняння з частинними похідними з n незалежними змінними зводиться до n звичайних диференціальних рівнянь.
2. **Метод інтегральних перетворень.** Рівняння з частинними похідними з n незалежними змінними зводиться до рівняння з частинними похідними з $(n-1)$ незалежними змінними; отже, рівняння з частинними похідними з двома незалежними змінними можна звести до звичайного диференціального рівняння.
3. **Метод перетворення координат.** Вихідне рівняння з частинними похідними зводиться до звичайного диференціального рівняння або іншого, більш простого рівняння з частинними похідними за допомогою відповідного перетворення координат (наприклад, повороту координатних осей і т.п.).
4. **Перетворення залежної змінної.** Вихідне рівняння з частинними похідними перетворюється на таке рівняння з частинними похідними для іншої невідомої функції, яке розв'язується легше, ніж вихідне.
5. **Метод теорії збурень.** Вихідна нелінійна задача зводиться до послідовності лінійних задач, що апроксимують нелінійну задачу.
6. **Метод функцій Гріна.** Початкові та граничні умови замінюються системою найпростіших джерел, і завдання розв'язується для кожного найпростішого джерела. Повний розв'язок вихідної задачі виходить у результаті підсумовування розв'язків для елементарних джерел.

7. Метод інтегральних рівнянь. Рівняння з частинними похідними зводиться до інтегрального рівняння. Існує багато різних методів розв'язання інтегральних рівнянь.

8. Варіаційні методи. Замість рівняння з частинними похідними розв'язується деяке завдання мінімізації. Виявляється, що функція, яка відповідає умові мінімуму деякого виразу, є водночас розв'язком вихідного рівняння з частинними похідними.

9. Метод розкладання за власними функціями. Розв'язок рівняння з частинними похідними шукається у вигляді ряду за власними функціями. Ці власні функції знаходяться як розв'язки так званої задачі на власні значення, що відповідає вихідному завданню для рівняння з частинними похідними.

При комп'ютерному моделюванні, розв'язуючи задачі математичної фізики найчастіше використовують **метод скінченних різниць (МСР)** (рус. *Метод конечных разностей* (МКР), англ. *Finite difference method* (FDM)) та **метод скінченних елементів (МСЕ)** (рус. *Метод конечных элементов* (МКЭ), англ. *Finite element method* (FEM)).

1.2. Базові положення МСР

Нехай є функція $u(x,y)$, що має безперервні похідні по всім змінним $(k+1)$ -го порядку включно. Виберемо вузол (x_h, t_n) сітки D_h . Спочатку розглянемо різницю у напрямі x . Застосуємо формулу Тейлора розкладання функції $u(x, t^n)$ однієї змінної x в околиці обраного вузла (x, t_n) .

Пригадаємо ряд Тейлора (p – порядок похідної):

$$f(x_0 + h) = f(x_0) + \frac{f'(x_0)}{1!}h + \frac{f''(x_0)}{2!}h^2 + \dots + \frac{f^{(p)}(x_0)}{p!}h^p + R_p(x),$$

при $x-a=h \rightarrow 0$

$$f(x) = f(a+h) = f(a) + f'(a)h + R(h^2) \approx f(a) + f'(a)h = f(a) + f'(a)(x-a).$$

Тут $R(h^2)$ – залишковий член, вільний член – доданки, чиї значення за модулем суттєво нижче ніж перші елементу ряду. Часто його позначають літерою $O(h^2)$. Він визначає похибку наближення $f(x)$. Інша назва цієї похибки – **нев’язка** (англ. **residual**).

При $k=1$ маємо

$$u(x_i + h, t^n) = u(x_i, t^n) + u_x(x_i, t^n)h + u_{xx}(\xi_1, t^n)\frac{h^2}{2}, \quad \xi_1 \in (x_i, x_i + h), \quad (1)$$

чи

$$u(x_i - h, t^n) = u(x_i, t^n) - u_x(x_i, t^n)h + u_{xx}(\xi_2, t^n)\frac{h^2}{2}, \quad \xi_2 \in (x_i - h, x_i).$$

З (1) випливає

$$u_x(x_i, t^n) = \frac{u(x_i + h, t^n) - u(x_i, t^n)}{h} - u_{xx}(\xi_1, t^n)\frac{h}{2}.$$

Для того щоб чисельний розв’язок не прямував до нескінченності (був стійким), необхідно щоб

$$\left| u_{xx}(\xi_1, t^n)\frac{h}{2} \right| \leq \frac{h}{2} \max_{x \in [x_i, x_i + h]} \left| u_{xx}(\xi_1, t^n)\frac{h}{2} \right| = \frac{h}{2} M_2,$$

де

$$M_2 = \max_{x \in [x_i, x_i + h]} \left| u_{xx}(\xi_1, t^n)\frac{h}{2} \right|.$$

Звідси й отримують формулу апроксимації похідної:

$$\hat{u}_x(x_i, t^n) = \frac{u(x_i + h, t^n) - u(x_i, t^n)}{h}, \quad \left(\frac{h}{2} M_2 \right).$$

Це так звана формула/шаблон лівої різниці.

Поклавши $k=2$ отримуємо

$$u(x_i + h, t^n) = u(x_i, t^n) + u_x(x_i, t^n)h + u_{xx}(x_i, t^n)\frac{h^2}{2} + u_{xxx}(\xi_1, t^n)\frac{h^3}{6}, \quad \xi_1 \in (x_i, x_i + h),$$

$$u(x_i - h, t^n) = u(x_i, t^n) - u_x(x_i, t^n)h + u_{xx}(x_i, t^n)\frac{h^2}{2} - u_{xxx}(\xi_2, t^n)\frac{h^3}{6}, \quad \xi_2 \in (x_i - h, x_i).$$

Віднімаючи від першого рівняння друге, маємо

$$u(x_i + h, t^n) - u(x_i - h, t^n) = 2u_x(x_i, t^n)h + [u_{xxx}(\xi_1, t^n) + u_{xxx}(\xi_2, t^n)]\frac{h^3}{6}.$$

Однак $\xi \in (x_i - h, x_i + h)$, і $\frac{u_{xxx}(\xi_1, t^n) + u_{xxx}(\xi_2, t^n)}{2} = u_{xxx}(\xi, t^n)$. Тому

$$u_x(x_i, t^n) = \frac{u(x_i + h, t^n) - u(x_i - h, t^n)}{2h} - u_{xxx}(\xi, t^n)\frac{h^2}{6}, \quad \xi \in (x_i - h, x_i + h),$$

чи

$$\hat{u}_x(x_i, t^n) = \frac{u(x_i + h, t^n) - u(x_i - h, t^n)}{2h} - \left(\frac{h^2}{6} M_3\right),$$

де

$$M_3 = \max_{x \in [x_i - h, x_i + h]} |u_{xxx}(\xi, t^n)|.$$

Це так званий *шаблон центральної різниці*.

Таким чином наближене значення похідної можна отримати за формулами:

$$\hat{u}_x(x_i, t^n) = \frac{u_{i+1}^n - u_i^n}{h},$$

$$\hat{u}_x(x_i, t^n) = \frac{u_i^n - u_{i-1}^n}{h},$$

$$\hat{u}_x(x_i, t^n) = \frac{u_{i+1}^n - u_{i-1}^n}{2h}.$$

Подібно до отримання формули центральної різниці можна вивести апроксимацію другої похідної (тільки сумуючи вирази $u(x_i + h, t^n)$ й $u(x_i - h, t^n)$), яка має вигляд

$$\hat{u}_{xx}(x_i, t^n) = \frac{u(x_i + h, t^n) - 2u(x_i, t^n) + u(x_i - h, t^n)}{h^2} \left(\frac{h^2}{12} M^4 \right),$$

де

$$M_4 = \max_{x \in [x_i - h, x_i + h]} |u_{xxxx}(x, t^n)|.$$

Отримуємо апроксимацію змішаної похідної:

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{\partial u(x_i, t^n)}{\partial x} \right) &= \frac{1}{\tau} \left(\frac{u(x_i + h, t^n + \tau) - u(x_i, t^n + \tau)}{h} - \frac{u(x_i + h, t^n) - u(x_i, t^n)}{h} \right) = \\ &= \frac{u(x_i + h, t^n + \tau) - u(x_i, t^n + \tau) - u(x_i + h, t^n) + u(x_i, t^n)}{h \tau}. \end{aligned}$$

Інакше, більш короткий запис

$$\hat{u}_{xt}(x_i, t^n) = \frac{u_{i+1}^{n+1} - u_i^{n+1} - u_{i+1}^n + u_i^n}{h \tau}.$$

Ця формула відповідає першому порядку апроксимації.

Використовуючи шаблон центральної різниці апроксимації похідної, отримуємо:

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{\partial u(x_i, t^n)}{\partial x} \right) &= \frac{1}{2\tau} \left(\frac{u(x_i + h, t^n + \tau) - u(x_i - h, t^n + \tau)}{2h} - \frac{u(x_i + h, t^n - \tau) - u(x_i - h, t^n - \tau)}{2h} \right) = \\ &= \frac{u(x_i + h, t^n + \tau) - u(x_i - h, t^n + \tau) - u(x_i + h, t^n - \tau) + u(x_i - h, t^n - \tau)}{4h \tau} \end{aligned}$$

Інакше, більш короткий запис

$$\hat{u}_{xt}(x_i, t^n) = \frac{u_{i+1}^{n+1} - u_{i-1}^{n+1} - u_{i+1}^{n-1} + u_{i-1}^{n-1}}{4h \tau}.$$

Ця формула відповідає другому порядку апроксимації.

1.3. Хвильове рівняння д'Аламбера

Хвильове рівняння д'Аламбера для моделювання поширення акустичного тиску (p) має вигляд:

$$\frac{\partial^2 p}{\partial t^2} - V^2 \nabla^2 p = 0,$$

де V – швидкість поширення акустичної хвилі (швидкість звуку), ∇^2 – оператор Лапласа.

Іноді оператор Лапласа позначають як Δ :

$$\nabla^2 F = \Delta F = \frac{\partial^2 F}{\partial x_1^2} + \frac{\partial^2 F}{\partial x_2^2} + \dots + \frac{\partial^2 F}{\partial x_n^2}.$$

Тут F – деяка довільна функція, для якої існують другі частинні похідні в n -мірному просторі $\{x_1, x_2, \dots, x_n\}$.

Оператор Лапласа еквівалентний послідовному взяттю операцій градієнта та дивергенції:

$$\Delta = \operatorname{div} \operatorname{grad}.$$

Таким чином, значення оператора Лапласа в точці може бути розтлумачено як густина джерел (стоків) потенціального векторного поля $\operatorname{grad} F$ в цій точці.

У декартовій системі координат оператор Лапласа часто позначається так

$$\Delta = \nabla \cdot \nabla = \nabla^2.$$

Цей вираз – скалярний добуток оператора Набла на себе.

Якщо в системі є джерело механічних/акустичних коливань (F), то рівняння д'Аламбера записується так:

$$\frac{\partial^2 p}{\partial t^2} - V^2 \nabla^2 p = F(x, y, z, t).$$

Якщо при моделюванні необхідно врахувати ефекти поглинання енергії акустичних хвиль, то рівняння модифікується наступним чином:

$$\nabla^2 p - \frac{1}{V^2} \frac{\partial^2 p}{\partial t^2} + \tau \frac{\partial p}{\partial t} \nabla^2 p = F,$$

де τ – коефіцієнт, що має розмірність часу й характеризує швидкість затухання.

В більш загальному випадку похідна за часом оператора Лапласа представляє собою дрібну (нецілу) похідну (англ. **Fractional derivatives**):

$$\nabla^2 p - \frac{1}{V^2} \frac{\partial^2 p}{\partial t^2} + \tau^\phi \frac{\partial^\phi p}{\partial t^\phi} \nabla^2 p = F .$$

Детальніше про дрібну похідну дивись у навчальному посібнику: Дробное исчисление и аппроксимационные методы в моделировании динамических систем. Научное издание / В.В.Васильев, Л.А.Симак. — Киев, НАН Украины, 2008. — 256 с.

У газі та рідині акустичні хвилі повздовжні. В твердому тілі при деформації середовища в одному напрямку виникає також деформація в поперечному напрямку. Такі хвилі називають поперечними. Швидкість поширення цих хвиль може відрізнятись (зазвичай менше) від повздовжніх. Тому в твердих тілах необхідно в загальному випадку розв'язувати два рівняння – для повздовжніх (l -хвиль) та поперечних (t -хвиль):

$$\nabla^2 p - \frac{1}{V_l^2} \frac{\partial^2 p}{\partial t^2} + \tau^\phi \frac{\partial^\phi p}{\partial t^\phi} \nabla^2 p = F \text{ та } \nabla^2 p - \frac{1}{V_t^2} \frac{\partial^2 p}{\partial t^2} + \tau^\phi \frac{\partial^\phi p}{\partial t^\phi} \nabla^2 p = F .$$

Сумарний акустичний тиск буде визначатися:

$$p = p_l + p_t .$$

Для повного опису поширення акустичних хвиль в середовищах необхідно розглядати ще коливальну швидкість атомів ($\vec{v}$) та коливання густини середовища (ρ). Але рівняння будуть подібними.

При комп'ютерному моделюванні поширення акустичних хвиль на основі МСК для отримання розв'язку необхідне виконання умови збіжності за Курантом-Фрідріхсом-Леві (Courant-Friedrichs-Lewy Condition). Вона записується як

$$\frac{V \Delta t}{\Delta x} < C_{CFL} ,$$

де V – швидкість переносу (акустичної хвилі), Δt та Δx – крок інтегрування за часом та простором, відповідно, C_{CFL} – константа.

Для двухмірного випадку це співвідношення записується як:

$$\frac{V_x \Delta t}{\Delta x} + \frac{V_y \Delta t}{\Delta y} < C_{CFL}.$$

C_{CFL} повинна бути менша за одиницю – швидкість поширення хвилі не повинна перевищувати $\Delta x/\Delta t$. Якщо простір моделювання є неоднорідним (в різних точках простору різні значення V), то обирають найменші значення для Δt при фіксованих значеннях Δx , C_{CFL} . Наприклад так:

$$\Delta t = \frac{C_{CFL} \Delta x}{\max(V)}, \quad \Delta t = \frac{C_{CFL} \sqrt{\Delta x^2 + \Delta y^2}}{\max(V)}, \quad \Delta t = \frac{C_{CFL} \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\max(V)}.$$

1.4. Граничні умови

При розв'язуванні хвильових рівнянь необхідно задати граничні умови (boundary condition) – як поводить ся сигнал на границях: нульові – умови Perfect Electrical Conductor (PEC), тобто ідеальний провідник; умови симетрії; відбиваючі границі; поглинаючі границі – хвиля ідеально поглинається, що еквівалентно тому, що вона проходить крізь границю і прямує до нескінченності. Останні називають умовами ABC, від Absorbing Boundary Condition.

Нажаль, при комп'ютерному моделюванні ці умови, в загальному випадку, не завжди працюють ефективно, що може спотворювати результати. Необхідно проводити попередні дослідження наскільки коректно граничні умови відповідають задачам моделювання.

Для моделювання ідеального відбиття хвилі використовують нульові умови Діріхле (Dirichlet boundary conditions) (задача Діріхле). Для двовимірного простору $x \in [0..L_x]$, $y \in [0..L_y]$ вони записуються як

$$\begin{aligned} u(0, y, t) = 0 \text{ і } u(L_x, y, t) = 0, \\ u(x, 0, t) = 0 \text{ і } u(x, L_y, t) = 0, \end{aligned}$$

де u – шукана функція (для акустичного тиску замість u підставляють p). Ця умова відноситься до першого порядку.

Якщо границя поглинаюча (ABC), то досить поширеною є умова Мура (Mur's condition):

$$\left. \frac{\partial u}{\partial x} \right|_{x=0} = V \left. \frac{\partial u}{\partial t} \right|_{x=0} \quad \text{і} \quad \left. \frac{\partial u}{\partial x} \right|_{x=L_x} = -V \left. \frac{\partial u}{\partial t} \right|_{x=L_x},$$

$$\left. \frac{\partial u}{\partial y} \right|_{y=0} = V \left. \frac{\partial u}{\partial t} \right|_{y=0} \quad \text{і} \quad \left. \frac{\partial u}{\partial y} \right|_{y=L_y} = -V \left. \frac{\partial u}{\partial t} \right|_{y=L_y},$$

тут V , як і в попередніх підрозділах, швидкість акустичних хвиль.

Ця умова теж відноситься до першого порядку точності. В залежності від фази сигналу, що падає на границю, «якість поглинання» варіюється. Тому часто доводиться використовувати інші умови ABC, зокрема умови Мура другого порядку, Ліао (Liao) чи інші. Однак вони вимагають додаткових машинних ресурсів.

Універсальним підходом до мінімізації проблеми ABC є використання узгоджуваних додаткових шарів: просторову область моделювання штучно збільшують від 2 до 10 шарів, але цим шарам притаманне велике значення коефіцієнту згасання. Такі умови називають PML – Perfect Matched Layer. За рахунок додавання нових шарів простору теж підвищується час розрахунків (машинні ресурси), але цей підхід дозволяє досягти вимог до моделювання й поставленої мети.

2. Завдання

1. Отримати чисельний розв'язок хвильового рівняння д'Аламбера

$$\frac{\partial^2 p}{\partial t^2} - V^2 \nabla^2 p = F(x, t)$$

для одномірного однорідного простору $x \in [0..76]$ (мм). Форма збуджуючого сигналу – синусоїдальна, частота джерела сигналу $f=1$ МГц, амплітуда 1 Па, час генерування сигналу – 2 періоди. Область збудження («точка» збудження) довільна в межах $x \in [0..76]$. Граничні умови – ABC, Мура 1-го порядку.

Приблизний час інтегрування $50/f$ (с), швидкість поширення акустичних хвиль – 330 м/с.

Пошук розв’язку – прогонка по вузлам – ітераційна, за допомогою операторів циклу `for ...end`.

2. Отримати чисельний розв’язок хвильового рівняння д’Аламбера

$$\nabla^2 p - \frac{1}{V^2} \frac{\partial^2 p}{\partial t^2} + \tau \frac{\partial p}{\partial t} \nabla^2 p = F$$

для одномірного однорідного простору $x \in [0..500]$ (мм). Форма збуджуючого сигналу – синусоїдальна, частота джерела сигналу $f=1$ МГц, амплітуда 1 Па, час генерування сигналу – безперервно. Область збудження («точка» збудження) $x_0=0$. Граничні умови – АВС, Мура 1-го порядку. Приблизний час інтегрування $100/f$ (с), швидкість поширення акустичних хвиль – 330 м/с, τ – підібрати таким, щоб коефіцієнт згасання прямував до 750 дБ/м.

3. Отримати чисельний розв’язок хвильового рівняння д’Аламбера

$$\nabla^2 p - \frac{1}{V^2} \frac{\partial^2 p}{\partial t^2} + \tau \frac{\partial p}{\partial t} \nabla^2 p = F$$

для двовимірного простору $x \in [0..76]$ (мм), $y \in [0..76]$ (мм). Форма збуджуючого сигналу – синусоїдальна, частота джерела сигналу $f=1$ МГц, амплітуда 1 Па, час генерування сигналу – безперервно. Область збудження точкова $x_0=38$ мм, $y_0=38$ мм. Граничні умови – АВС, Мура 1-го порядку. Приблизний час інтегрування $50/f$ (с). В межах $x \in [0..5]$ мм та $y \in [0..5]$ мм швидкість поширення акустичних хвиль – 330 м/с, в межах $x \in [70..75]$ мм та $y \in [70..75]$ мм - 1500 м/с, τ – підібрати таким, щоб коефіцієнт згасання прямував до 750 дБ/м, в іншому просторі швидкість поширення акустичних хвиль – 1500 м/с, τ – підібрати таким, щоб коефіцієнт згасання прямував до 0,22 дБ/м.

Необхідно виміряти машинний час інтегрування.

4. Отримати чисельний розв’язок хвильового рівняння д’Аламбера, що зазначено в попередньому пункті 3, але при розв’язуванні рівняння застосувати векторні можливості Matlab – уникаючи застосування циклів `for ...end`.

Необхідно виміряти машинний час інтегрування. Порівняйте результати цього вимірювання зі значенням у попередньому пункті.

3. Методичні вказівки

1. Візуалізацію результатів моделювання не треба виконувати на кожному кроці Δt , а через задану кількість ітерацій чи через певний час. Інакше машинні ресурси будуть по більшій мірі зайняті процесами візуалізації й буде спостерігатись зависання роботи програми.

2. Для візуалізації одновимірних сигналів доцільно користуватися командами

```
plot(x,u);  
drawnow;
```

У двовимірному випадку:

```
imagesc(u);  
drawnow;
```

3. Щоб виміряти машинний час виконання коду використовують команди `tic` – початок відліку часу, `toc` – кінець вимірювання часу:

```
t0=tic;  
код програми  
t=toc(t0).
```

4. При розв’язуванні рівняння д’Аламбера в лабораторній роботі доцільно запам’ятовувати тільки три шари за часом акустичного тиску.

5. Приклад функції з класу, який відповідальний за інтегрування для трьохвимірною випадку з використанням векторизації, в наведеному нижче коді.

- `up` – шукана функція (акустичний тиск);

- Laplas – поточне значення лапласіану в вузлах. Його необхідно запам'ятати для знаходження похідної за часом в наступний момент часу інтегрування;
- S – амплітуда збуджуючого сигналу в даний момент часу;
- dx, dy, dz – просторовий крок інтегрування;
- obj.WaveParam.Time_impulse – час генерації сигналу (зовнішнього джерела);
- obj.nC – трьохзначний масив, що містить кількість вузлів вздовж осей x, y, z;
- t – поточний час інтегрування (не машинний час!!!);
- obj.Source – координати джерела сигналу.

```
function [un, Laplas]=wave_eq_DAlamberta_3D_vectorization(obj,t,dt,un,Laplas,S)

    dx=obj.dXYZ_metric(1);  dx_2=dx^2;
    dy=obj.dXYZ_metric(2);  dy_2=dy^2;
    dz=obj.dXYZ_metric(3);  dz_2=dz^2;
    % Source
    if t < obj.WaveParam.Time_impulse % менше встановленого часу
        try
            un(obj.Source.Y, obj.Source.X, obj.Source.Z, 3) = S;
        catch
            a=[S, t];
        end
    end

    d2un_dx=(un(2:obj.nC(1)-1,3:obj.nC(2),2:obj.nC(3)-1,2)-2*un(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1,2)+un(2:obj.nC(1)-1,1:obj.nC(2)-2,2:obj.nC(3)-1,2))./dx_2;
    d2un_dy=(un(3:obj.nC(1),2:obj.nC(2)-1,2:obj.nC(3)-1,2)-2*un(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1,2)+un(1:obj.nC(1)-2,2:obj.nC(2)-1,2:obj.nC(3)-1,2))./dy_2;
    d2un_dz=(un(2:obj.nC(1)-1,2:obj.nC(2)-1,3:obj.nC(3),2)-2*un(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1,2)+un(2:obj.nC(1)-1,2:obj.nC(2)-1,1:obj.nC(3)-2,2))./dz_2;

    L=d2un_dx+d2un_dy+d2un_dz;
    dt_Laplas=(L-Laplas(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1) );
    Laplas(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1)=L;

    Att=dt*obj.Attenuation_alfa(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1).*dt_Laplas;
    k=(dt*obj.C(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1) ).^2;

    un(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1,3) =2*un(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1,2)-un(2:obj.nC(1)-1,2:obj.nC(2)-1,2:obj.nC(3)-1,1)+k.*(L)+Att;

    un=obj.Absorbing_BC(un, dt, obj.dXYZ_metric); % розрахунок un на границях
end
```

Контрольні запитання

1. В чому полягає сенс PML граничних умов?
2. Яким чином можна зменшити машинне навантаження при моделюванні хвильових задач, що розглядаються в лабораторній роботі?
3. В чому полягає ідея методу скінченних елементів?
4. Що таке CFL-умова?
5. Від чого залежить коефіцієнт згасання при моделюванні поширення акустичних хвиль?
6. Яким чином можна оптимізувати візуалізацію розрахунків й зменшити машинне навантаження в динамічному режимі (під час моделювання)?
7. Які умови для проходження акустичної хвилі через границю розділу середовищ з мінімальними ефектами відбиття?

Рекомендована література

1. Теорія хвильових процесів: Акустичні хвилі: Навчальний посібник / І.П. Солов'янова, С.Н. Шабунін.- 2004. 142 с..
2. Конспект лекцій із курсу "Числові методи математичної фізики" / В.І.Кузьменко, Н.Л.Козакова. – Дніпропетровськ РВВ ДНУ, 2013. – 47 с.
3. Гой Т.П., Копач М.І., Федак І.В. Наближені методи розв'язування диференціальних рівнянь. Навчальний посібник для студентів напрямів підготовки “математика” та “прикладна математика”. – Івано-Франківськ: Видавничо-дизайнерський відділ Центру інформаційних технологій Прикарпатського національного університету імені Василя Стефаника, 2010. – 148 с.

4. Андруник В.А., Висоцька В.А., Пасічник В.В., Чирун Л.Б., Чирун Л.В.
Чисельні методи в комп'ютерних науках: навчальний посібник, Том 2 за ред.
В.В. Пасічника – Львів: Видавництво «Новий Світ – 2000», 2020. – 536 с
5. Implementation of Mur's absorbing boundaries with periodic structures to speed
up the design process using finite-difference time-domain method / G. Zheng, A.
A. Kishk, A. W. Glisson, and A. B. Yakovlev // Progress In Electromagnetics
Research, PIER. – 2006. – 58. – P. 101–114. – URL:
https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.jpier.org/ac_api/download.php%3Fid%3D0506213&ved=2ahUKEwjhm-Tr2JaGAxUyVPEDHV2fAl8QFnoECCKQAQ&usg=AOvVaw29d8bWxC550zwpsofHLUDs