



Міністерство освіти і науки України  
Запорізька державна інженерна академія

---

Волков В.О.

## **МОДЕЛЮВАННЯ ГІДРАВЛІЧНИХ ТА ЕЛЕКТРОМЕХАНІЧНИХ ПРОЦЕСІВ**

**Методичні вказівки до виконання лабораторних робіт**

**для студентів ЗДІА  
спеціальності 6.05060201 «Гідроенергетика»  
денної і заочної форм навчання**

Запоріжжя  
2013

Міністерство освіти і науки України  
Запорізька державна інженерна академія

## **МОДЕЛЮВАННЯ ГІДРАВЛІЧНИХ ТА ЕЛЕКТРОМЕХАНІЧНИХ ПРОЦЕСІВ**

**Методичні вказівки до виконання лабораторних робіт**

**для студентів ЗДІА  
спеціальності 6.05060201 «Гідроенергетика»  
денної і заочної форм навчання**

Рекомендовано до видання  
на засіданні кафедри гідроенергетики,  
протокол № 3 від 10.10.2013

**Моделювання гідравлічних та електромеханічних процесів.**  
Методичні вказівки до виконання лабораторних робіт для студентів ЗДІА спеціальності 6.05060201 «Гідроенергетика» денної і заочної форм навчання. / Укл. Волков В.О. – Запоріжжя, 2013. – 86 с.

Укладач: В.О. Волков, к.т.н., доцент

Відповідальний за випуск: зав. кафедри гідроенергетики  
д.т.н., професор О.В. Волков

## ЗМІСТ

	ВСТУП	4
	Лабораторна робота № 1	
1	Будова графіків з застосуванням Simulink та з використанням m-файлів	5
	Лабораторна робота № 2	
2	Розрахунок в пакеті програм MatLAB, розрахунок із застосування M-файлів та S-моделей	31
	Лабораторна робота № 3	
3	Обчислення диференціалів та інтегралів в пакеті програм MatLAB	53
	Лабораторна робота № 4	
4	Розрахунок гідравлічних процесів	66
	Лабораторна робота № 5	
5	Будова двумірних та тривимірних графіків в MatLAB.	
	Редагування графіків	71
	Список використаної літератури	86

## ВСТУП

У наші дні комп'ютерна математика отримала належну популярність і знаходиться на етапі швидкого розвитку [1]. Почавши свій шлях з застосування програмованих мікрокалькуляторів і розрахунків на персональних ЕОМ (комп'ютерах), вона породила цілий ряд своїх спеціальних програмних засобів - систем комп'ютерної математики (СКМ) і пакетів їх розширення. Серед СКМ, в першу чергу орієнтованих на чисельні розрахунки, особливо виділяється матрична математична система MATLAB – найефективніша серед систем для чисельних обчислень. Система фактично стала світовим стандартом в області сучасного математичного та науково-технічного програмного забезпечення. У розробці MATLAB і пакетів розширення цієї системи беруть участь великі наукові школи світу, багато провідних університетів та інші організації. Ефективність MATLAB обумовлена насамперед її орієнтацією на математичні обчислення з програмною емуляцією паралельних обчислень та спрощеними засобами завдання циклів. В MATLAB вдало реалізоване середовище роботи з багатовимірними масивами, великими і розрідженими матрицями і багатьма типами даних. Система пройшла багаторічний шлях розвитку від вузько спеціалізованого матричного програмного модуля, використовуваного тільки на великих ЕОМ, до універсальної інтегрованої СКМ, орієнтованої на масові персональні комп'ютери класу IBM PC, AT і Macintosh і робочі станції UNIX. MATLAB має потужні засоби діалогу, графіки та комплексної візуалізації обчислень, в тому числі із залученням коштів дескрипторної графіки та віртуальної реальності. Система MATLAB пропонується розробниками (корпорація The MathWorks, Inc.) як лідируюча обчислювальна система на ринку, в першу чергу на підприємствах військового промислового комплексу, в енергетиці, в аерокосмічній галузі та в автомобілі будівництві. В нових реалізаціях MATLAB число пакетів розширення фірми MathWorks складає понад 80. Найвідомішим з них стало розширення Simulink, що забезпечує блокове імітаційне моделювання різних систем і пристроїв із застосуванням сучасної технології візуально орієнтованого програмування.

Структура комплексу MATLAB + Simulink, крім основи - системи MATLAB і головного розширення Simulink, містить обширні групи пакетів розширення. Ці інструментальні ящики Toolboxes з великим числом пакетів розширення (до сотні) і група пакетів розширення Blocksets збільшують можливості системи візуально орієнтованого блокового імітаційного моделювання динамічних систем Simulink.

## Лабораторна робота № 1

### БУДОВА ГРАФІКІВ З ЗАСТОСУВАННЯМ SIMULINK ТА З ВИКОРИСТАННЯМ m-ФАЙЛІВ

**Мета роботи:** вивчення програмного інтерфейсу, налаштувань графічного редактора, команд креслення графічних примітивів і геометричних зображень на кресленнях

#### Теоретична частина

##### 1.1 Запуск SimuLink

Запуск SimuLink можна здійснити з командного вікна MatLAB або обравши команду New Model ("Нова модель") в меню File, або натиснувши відповідну піктограму в лінійці інструментів. При запуску SimuLink відкриваються два вікна (рис. 1.1а,б):

а) порожнє вікно untitled (вікно, куди буде виводитися схемне представлення модельованої системи, нової S-моделі, MDL-файлу);

б) вікно Simulink Library Browser, яке містить перелік основних бібліотек SimuLink (рис. 1.1).

Почнемо зі знайомства з бібліотеками SimuLink. У вікні Simulink Library Browser представлений перелік SimuLink-бібліотек, що входять до складу встановленої конфігурації системи MatLAB. З них головною є бібліотека SimuLink, розташована у першому рядку. Інші бібліотеки не є обов'язковими. Вони включаються в склад загальної бібліотеки в залежності від смаків користувача.

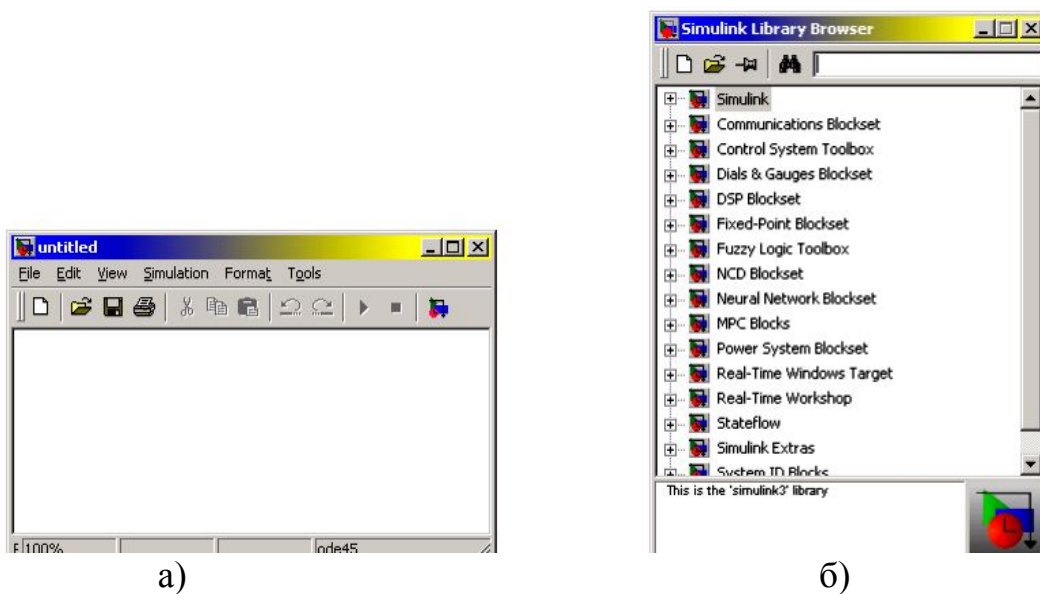


Рис. 1.1 Зображення стандартних вікон при запуску SimuLink

Вікно блок-схеми моделі (рис. 1.1a) містить рядок меню, панель інструментів (версії MatLAB 5.2 і 5.3) і робоче поле. Меню File (Файл) містить команди роботи з MDL-файлами, меню Edit (Редагування) – команди редагування блок-схеми, а меню View (Представлення) (починаючи з версії 5.2) – команди зміни зовнішнього вигляду вікна. У меню Simulation (Моделювання) містяться команди управління моделюванням, а в меню Format (Формат) – команди редагування формату (тобто зовнішнього зображення) блоків схеми і блок-схеми в цілому. Меню Tools (Інструменти) містить деякі додаткові сервісні засоби роботи з Simulink-моделлю.

### 1.1.1. Бібліотека модулів (блоків)

Бібліотека блоків SimuLink є набором візуальних об'єктів, використовуючи які, можна, з'єднуючи окремі модулі між собою лініями функціональних зв'язків, скласти функціональну блок-схему будь-якого пристрою. Бібліотека SimuLink блоків складається з 9 розділів. Вісім з них є головними і не можуть змінюватися користувачем:

- Sources (Джерела);
- Sinks (Приймачі);
- Continuous (Безперервні елементи);
- Discrete (Дискретні елементи);
- Math (Математичні блоки);
- Functions & Tables (Функції і таблиці)
- Nonlinear (Нелінійні елементи);
- Signals & Systems (Сигнали і системи).

Дев'ятий розділ – Blocksets & Toolboxes (Набори блоків та інструменти) – містить додаткові блоки, включені в робочу конфігурацію пакета. Блоки, які входять в розділ Sources (Джерела), призначені для формування сигналів, що забезпечують роботу S-моделі в цілому або окремих її частин при моделюванні. Всі блоки-джерела мають по одному виходу і не мають входів. Блоки, зібрані в розділі Sinks (Приймачі), мають тільки входи і не мають виходів. Умовно їх можна розділити на 3 види:

- блоки, які використовуються як оглядові вікна при моделюванні;
- блоки, що забезпечують збереження проміжних і вихідних результатів моделювання;
- блок управління моделюванням, який дозволяє переривати моделювання при виконанні тих чи інших умов.

Розділ Continuous (безперервні елементи) містить блоки, які можна умовно поділити на три групи:

- блоки загального призначення (інтегратори, диференціатор);
- блоки затримки сигналу;
- блоки лінійних стаціонарних ланок.

В розділ Discrete (дискретні елементи) входять блоки, за допомогою яких в моделі може бути описано поведінку дискретних систем. Розрізняють два основних типи таких систем: системи з дискретним часом і системи з

дискретними станами. Блоки, які входять в розділ Discrete забезпечують моделювання систем з дискретним часом.

Розділ Math (математичні елементи) – найбільший за складом. Він містить 20 блоків, які можна розділити на кілька груп:

- блоки, що реалізують елементарні математичні операції (Множення, підсумовування різних математичних об'єктів);
- блоки, що реалізують елементарні математичні функції;
- блоки, що забезпечують логічну обробку вхідних сигналів;
- блоки, які перетворюють комплекснозначних сигнал в два дійсних і навпаки тим чи іншим способом;
- блок, який реалізує відшукування нуля алгебраїчної функції.

У розділі Functions & Tables (функції і таблиці) зосереджені блоки двох видів:

- Блоки, що формують вихідний сигнал з вхідного відповідно до заданої таблицею відповіностей, здійснюючи лінійну інтерполяцію по цим значенням;
- Блоки, що дозволяють користувачеві створювати власні блоки з довільними функціями.

Розділ Nonlinear (нелінійні елементи) містить 10 елементів, з яких 7 блоків реалізують різного виду кусочно-лінійні залежності виходу від входу, а три здійснюють різного виду перемикання сигналу.

Більшість блоків розділу Signals & Systems (сигнали і системи) призначене для розробки складних S-моделей, що містять моделі більш низького рівня (підсистеми), і забезпечують встановлення необхідних зв'язків між декількома S-моделями.

Щоб перейти у вікно відповідного розділу бібліотеки, в якому розташовані графічні зображення блоків, досить двічі клацнути мишею на піктограмі цього розділу

Збірка блок-схеми S-моделі полягає в тому, що графічні зображення обраних блоків за допомогою миші перетягуються з вікна розділу бібліотеки у вікно блок-схеми, а потім виходи одних блоків у вікні блок-схеми з'єднуються з входами інших блоків також за допомогою миші.

Технологія перетягування зображення блоку така: курсор миші потрібно встановити на зображенні обраного блоку у вікні розділу бібліотеки, потім натиснути ліву клавішу мишки і, не відпускаючи її, пересунути курсор на поле блок-схеми, після чого відпустити клавішу. Аналогічно здійснюються з'єднання в блок-схемі лініями виходів одних блоків з входами інших блоків: курсор мишки підводять до потрібного виходу деякого блоку (при цьому курсор повинен набути форму хрестика), натискають ліву клавішу і, не відпускаючи її, курсор переміщують до потрібного входу іншого блоку, а потім відпускають клавішу. Якщо з'єднання здійснено вірно, на вході останнього блоку з'явиться зображення чорної затушувати стрілки.



## 1.1.2. Розділ Sinks (приймачі)

Після переходу до розділу Sinks на екрані з'являється вікно цього розділу, зображене на рис. 1.2.

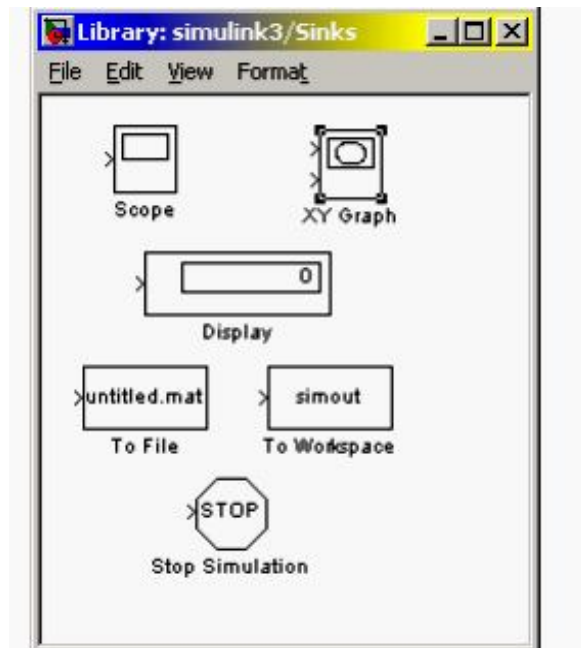


Рис. 1.2 Бібліотека Sinks

З його розгляду випливає, що в цьому розділі розміщені три групи блоків, які не мають виходів, а тільки входи:

1) блоки, які при моделюванні відіграють роль оглядових вікон; до них відносяться:

- блок Scope з одним входом, який виводить в графічне вікно графік залежності величини, що подається на його вхід, від модельного часу;
- блок XYGraph з двома входами, який забезпечує побудову графіка залежності однієї моделюється величини (другий зверху вхід) від іншої (перший вхід);
- блок Display з одним входом, призначений для відображення чисельних значень вхідної величини;

2) блоки для збереження результатів:

- блок To File, який забезпечує збереження результатів моделювання на диску в Маат-файлі (з розширенням. mat);
- блок To Workspace, який зберігає результати в робочому просторі;

3) блок керування моделюванням - Stop Simulation, що дозволяє переривати моделювання при виконанні тих чи інших умов; блок спрацьовує в тому випадку, коли на його вхід надходить ненульовий сигнал.

### Блок Scope

Цей блок дозволяє в процесі моделювання спостерігати за графіком процесу, які цікавлять дослідника.



Рис.1.3 Вид вікна Scope

Для настроювання параметрів цього блоку потрібно після установки зображення блоку у вікно блок-схеми двічі клацнути мишкою на цьому зображенні. В результаті на екрані з'явиться вікно Scope (рис. 1.3). Розмір і пропорції вікна можна змінювати довільно, користуючись мишею. За горизонтальної осі відкладаються значення модельного часу, а по вертикальній - значення вхідної величини, що відповідають цим моментам часу.

Якщо вхідна величина блоку Scope є вектором, у вікні будуються графіки зміни всіх елементів цього вектора, тобто стільки кривих, скільки елементів у вхідному векторі, причому кожна - свого кольору. Одночасно у вікні може відображатися до 30 кривих.

Для управління параметрами вікна в ньому передбачена панель інструментів, який містить 7 піктограм такого призначення (зліва направо):

- зміна масштабу одночасно по обом осям графіка;
- зміна масштабу по горизонтальній осі;
- зміна масштабу по вертикальній осі;
- автоматичне встановлення оптимального масштабу осей (повний огляд);
- збереження встановленого масштабу осей;
- виклик діалогового вікна настроювання параметрів блоку Scope;
- роздрукування вмісту вікна Scope на принтері.

Перші три піктограми є альтернативними, тобто в кожен момент часу може бути натиснута лише одна з них. Перші п'ять піктограм не активні до тих пір, поки немає графіка у вікні Scope. Активні з самого початку лише останні дві піктограми. Натискання шостий піктограми призводить до появи вікна настроювання параметрів (властивостей) блоку 'Scope' properties (рис.1.4)

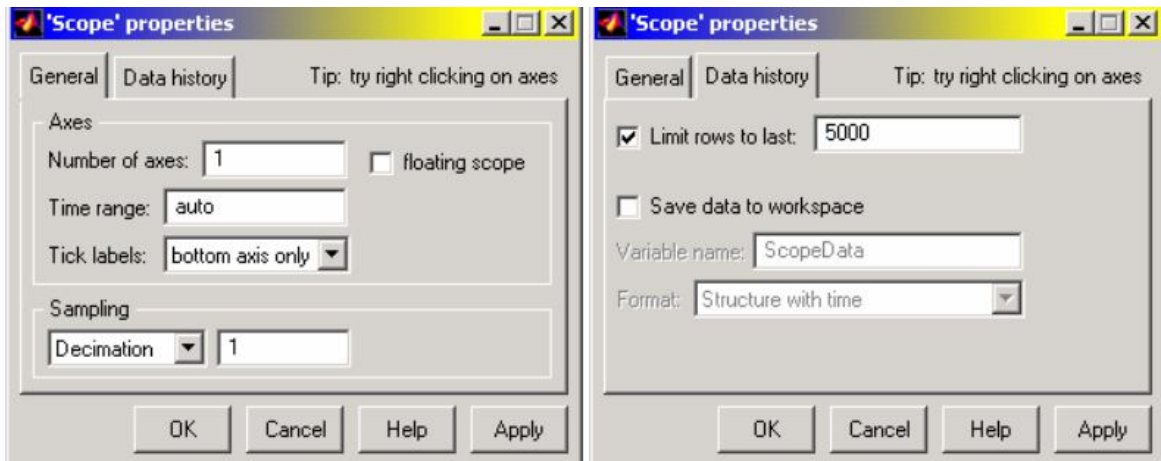


Рис.1.4 Вид вікна 'Scope' properties

Це вікно має дві вкладки:

- General (Загальні), вона дозволяє встановити параметри осей;
- Data history (Представлення даних), яка призначена для введення параметрів представлення даних блоку Scope. У нижній частині вікна розташовані кнопки: Apply (Застосувати), Help (Виклик довідки), Cancel (Вернутися назад) і OK (Підтвердити установку). На вкладці General маються поля Axes і Sampling. У полі Axes можна встановити:

- у віконці Number of axes (Кількість осей) – кількість графічних полів в графічному вікні Scope (одночасно змінюється кількість входів в блок Scope);

- верхню межу відображуваного модельного часу по осі абсцис (віконце Time range); при цьому слід брати до уваги наступне: якщо розмір заданого інтервалу моделювання ( $T_m$ ) не перевищує встановлене значення Time range (тобто весь процес вміщується у вікні Scope), то під графіком в рядку Time offset виводиться 0. В випадку ж, коли інтервал моделювання перевищує значення Time range, у вікні Scope відобразиться лише останній відрізок часу, менший за розміром, ніж Time range і рівний  $n * \text{Time range}$ , де  $n$  - ціле число; при цьому в рядку  $n * \text{Time offset}$  виводиться розмір "прихованого" інтервалу часу - Time range; наприклад, якщо значення Time range дорівнює 3, а тривалість інтервалу моделювання встановлена 17, то у вікні Scope буде виведений графік модельованого процесу за останні 2 одиниці часу, а рядок під графіком буде мати вигляд: Time offset: 15;

- у віконці Tick Labels - вид оформлення осей координат в графіках графічного вікна; якщо викликати спадний список у ньому, то в ньому побачимо три альтернативи: all (всі), none (немає), bottom axis only (тільки нижній осі). Обрання all призводить до того, що ділення по осях наносяться уздовж кожній з осей всіх графіків. Вибір bottom axis only викличе зникнення поділів по всьому горизонтальним осям графічних полів (якщо їх декілька), при цьому залишаться лише ділення по самій нижній з них. Якщо вибрати none, то зникнуть усі ділення по осях графіків і написи на них.

Віконце поруч з написом floating scope призначене для відключення входів в блок Scope. Для цього досить помістити в нього "галочку", клацнувши в ньому мишкою. Якщо "галочка" встановлена, то Scope відображається як блок

без входу, і якщо він був пов'язаний по входу з іншими блоками, то ці зв'язки "обриваються".

Поле Sampling містить лише одне віконце з написом Decimation. У ньому можна задати ціле позитивне число, яке дорівнює кількості кроків (дискретів) часу, в яких використовуються отримані дані для побудови графіків у вікні Scope.

Друга вкладка Data history дозволяє задати максимальну кількість елементів масивів даних, які використовуються для побудови графіків у вікні Scope (віконце поруч з написом Limit rows to last (Максимальна межа рядків)). Інші віконця цієї вкладки стають досяжними, якщо у віконці поруч з написом Save data to work space (Записати дані в робочий простір) поставити "галочку" (мишею). При цьому стає можливим записати дані, які виводяться на графіки вікна Scope, в робочий простір системи MatLAB, і стають активними віконця з написами Variable name (Ім'я змінної) і Format (Формат). У перше з них можна ввести ім'я змінної, під яким будуть зберігатися дані в робочому просторі системи (за умовчанням ці дані будуть записані під ім'ям ScopeData). Віконце Format дає можливість вибрати один з трьох форматів запису даних - Matrix (Матриця), Structure (Структура) і Structure with time (Структура із часом).

Будь-які зміни, зроблені у вікні Properties, змінюють вікно Scope лише у випадку, якщо після введення цих змін натиснута кнопка Apply в нижній частині вікна.

### Блок XYGraph

Цей блок також є оглядовим вікном. На відміну від Scope, він має два входи: на перший (верхній) подається сигнал, значення якого відкладаються по горизонтальній осі графіка, а на другий (нижній) - по вертикальній осі. Якщо перетягнути цей блок на полі блок-схеми, а потім на зображенні його клацнути двічі мишкою, то на екрані з'явиться вікно налаштування блоку (рис. 1.5), яке дозволяє встановити межі змін обох вхідних величин, між якими буде побудований графік залежності другої величини від першої, а також задати дискрет по часу.

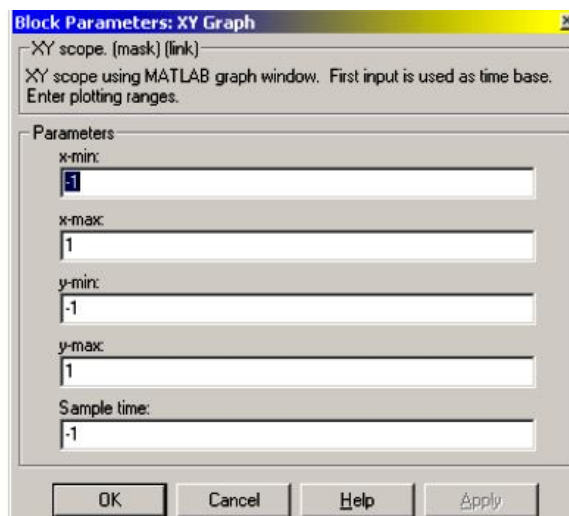


Рис.1.5 Вікно параметрів блоку XYGraph

Наведемо приклад використання блоку XYGraph. Для цього перетягнемо в вікно блок-схеми зображення цього блоку з вікна Library: simulink3/Sinks, а з вікна Library: simulink3/Sources - два блоки-джерела Clock і Sine Wave. З'єднаємо виходи блоків-джерел з входами блоку XYGraph. Одержимо блок-схему, наведену на рис. 1.6.

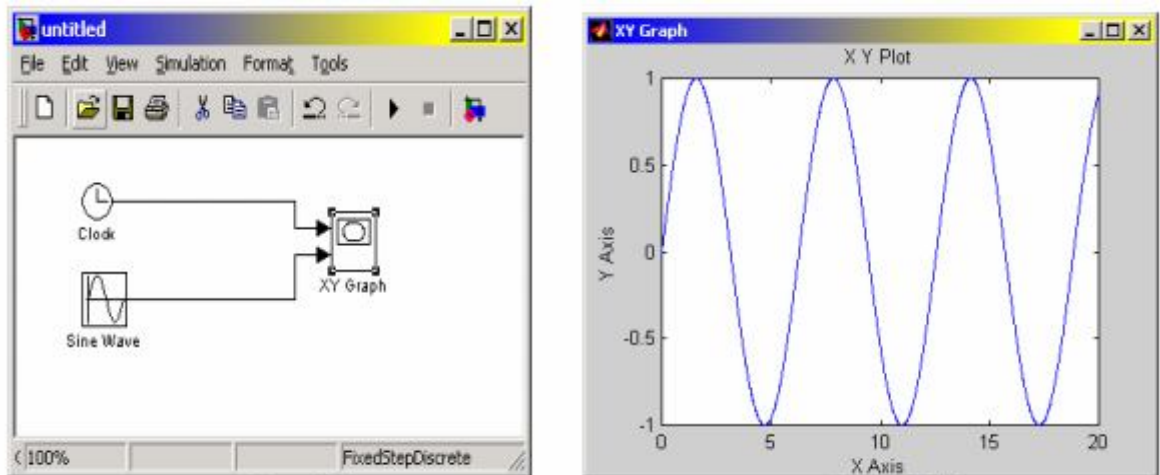


Рис.1.6 Блок схема і результат моделювання

Якщо тепер вибрати мишею меню Simulation в рядку меню вікна блок-схеми, а в ньому - команду Start, то по закінченні розрахунків на екрані виникне вікно XYGraph і в ньому буде побудовано зображення, представлене на рис.1.6.

### Блок Display

Цей блок призначений для виведення на екран чисельних значень величин, які фігурують у блок-схемі. Блок має 4 параметри настроювання (рис.1.7).

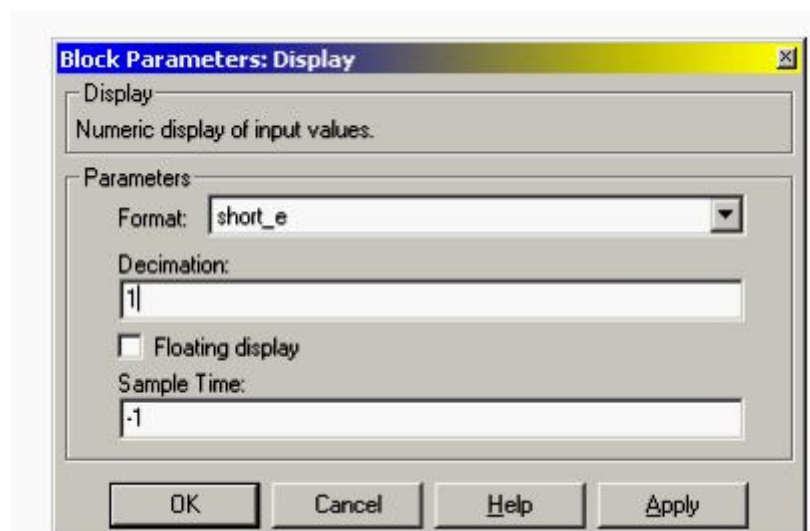


Рис.1.7 Вікно параметрів блоку Display

Список Format задає формат виводу чисел; вид формату обирається за допомогою спадного меню, містить 5 пунктів: short, long, short\_e, long\_e, bank.

Поле введення Decimation дозволяє задати періодичність (через скільки кроків часу) виведення значень у вікні Display.

Перемикач Floating display дозволяє визначати блок Display як блок без входу, обриваючи його зв'язку.

Блок Display може використовуватися для виводу як скалярних, так і векторних величин. Якщо відображувана величина є вектором, то уявлення блоку змінюється автоматично, про що свідчить поява маленького чорного трикутника в правому нижньому кутку блоку. Для кожного елемента вектора створюється своє міні-вікно, але щоб вони стали видимими, необхідно розтягнути зображення блоку. Для цього слід виділити блок, підвести курсор мишки до одного з його кутів, натиснути ліву клавішу миші і, не відпускаючи її, розтягнути зображення блоку, поки не зникне чорний трикутник.

### Блок To File

Цей блок забезпечує запис значень величини, поданої на його вхід, в MAT-файл даних для використання їх в подальшому в інших S-моделях. Блок має такі параметри налаштування (див. рис. 1.8):

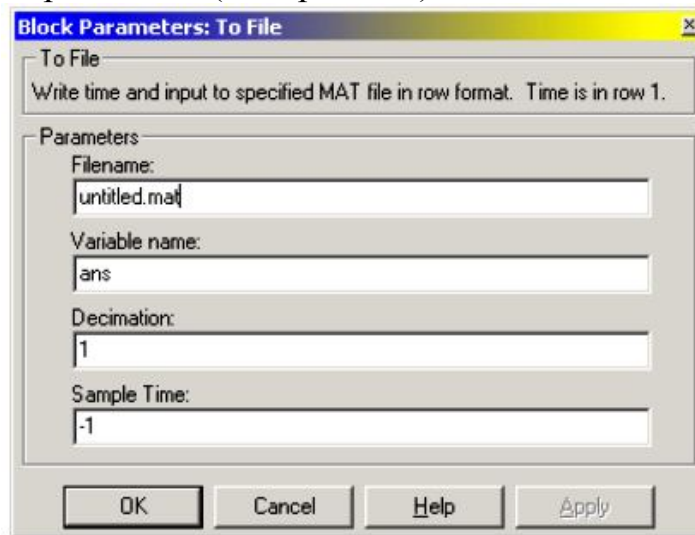


Рис. 1.8 Вікно параметрів блоку To File

File name - ім'я MAT-файлу, в який будуть записуватися значення вхідної величини; за замовчуванням - untitled. mat; ім'я файлу виводиться на зображенні блоку в блок-схемі;

Variable name - ім'я змінної, за яким можна буде звертатися до даними, записаним у файлі (для того, щоб переглянути або змінити їх в командному вікні MatLAB); за замовчуванням використовується системне ім'я ans;

Decimation - дискретність (в кількості дискретов часу) запису даних в файл;

Sample Time - розмір дискрет часу для даного блоку.

Слід зазначити, що значення даних, які подаються у вхід блоку записуються у вихідну змінну (наприклад, ans) так: перший рядок матриці утворюють значення відповідних моментів часу; другий рядок містить відповідні значення першого елемента вхідного вектора, третій рядок - значення другого елемента і т.д. В результаті записується матриця розміром

$(k + 1) * N$ , де  $k$  - кількість елементів вхідного вектора, а  $N$  – кількість точок вимірювання (або кількість моментів часу, в які здійснені вимір).

### Блок To Workspace

Цей блок призначений для збереження даних у робочому просторі системи MatLAB. Дані зберігаються у вигляді матриці розміром  $(N * k)$ , структура якої відрізняється від структури даних у MAT-файлі тим, що:

- значення величин, які зберігаються, розташовані по стовпцях, а не по рядках;
- не записуються значення модельного часу.

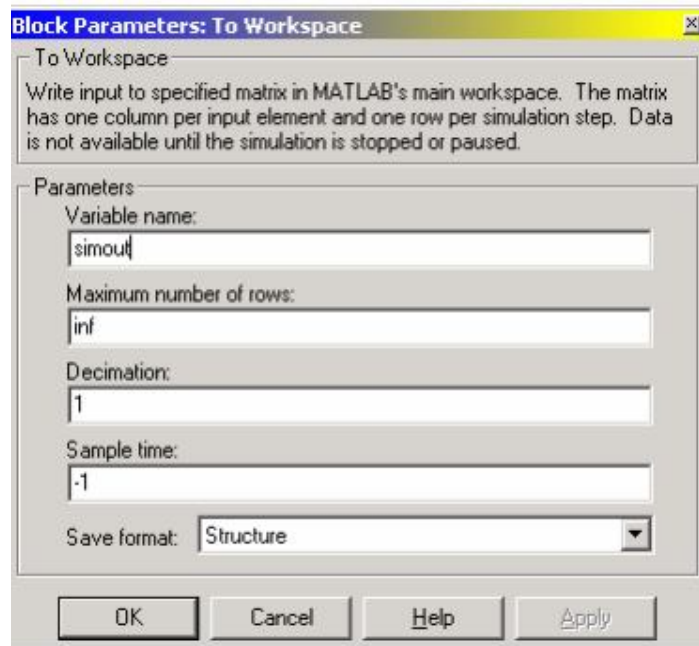


Рис.1.9 Вікно параметрів блоку To Workspace

Блок має 4 параметри настроювання (див. рис.1.9):

Variable name - ім'я, під яким дані зберігаються в робочому просторі (за замовчуванням - simout);

Maximum number of rows - максимально допустима кількість рядків, тобто значень даних, які записуються; за замовчуванням вона задається константою inf, тобто дані реєструються на всьому інтервалі моделювання;

Decimation і Sample Time мають те саме зміст, яке і раніше;

Віконце Save format (Формат запису) дозволяє вибрати один з трьох варіантів запису даних: Matrix (Матриця), Structure (Структура) і Structure with time (Структура із часом).

### **1.1.3. Розділ Sources (Джерела)**

Після вибору розділу Sources бібліотеки SimuLink на екрані з'явиться додаткове вікно, показане на рис. 1.10.

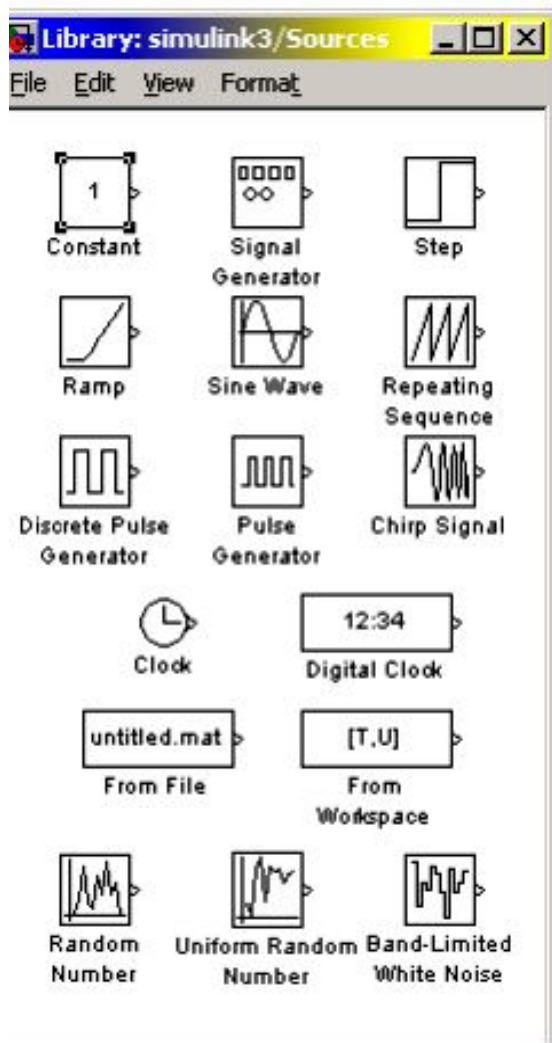


Рис. 1.10 Бібліотека Sources

лінійно змінюється з часом;

- Clock (Годинник) - джерело безперервного сигналу, пропорційного часу;

- Digital clock (Цифровий годинник) - формує дискретний сигнал, пропорційного часу;

- Random Number - джерело дискретного сигналу, значення якого є випадковою величиною, розподіленою за нормальним (Гаусовим) законом;

- Uniform Random Number - джерело дискретного сигналу, значення якого є випадковою рівномірно розподіленою величиною;

- Band-Limited White Noise - генератор білого шуму з обмеженою смугою частот.

Що залишилися два блоки з розділу Sources забезпечують використання в моделі даних, отриманих раніше. Перший з них - From File – призначений для введення в S-модель даних, які зберігаються на диску в MAT-файлі. Другий - From Workspace - забезпечує введення в модель даних безпосередньо з робочого простору MatLAB. Нагадаємо, що структура даних в MAT-файлі є багатовимірним масивом зі змінним кількістю рядків, яке визначається кількістю реєстрованих змінних. Елементи першого рядка містять послідовні значення модельного часу, елементи в інших рядках - значення змінних, відповідні окремим моментам часу.

Як бачимо, в цьому розділі бібліотеки в якості джерел сигналів передбачені такі блоки:

- Constant - формує постійну величину (скаляр, вектор або матрицю);

- Signal Generator - створює (генерує) безперервний коливальний сигнал однієї з хвильових форм на вибір - синусоїдальний, прямокутний, трикутний або випадковий;

- Step - генерує сигнал у вигляді одиночної сходинки (ступінчастий сигнал) із заданими параметрами (початку сходинки і її висоти);

- Ramp - створює лінійно висхідний (або спадний) сигнал;

- Sine Wave - генерує гармонійний сигнал;

- Repeating Sequence - генерує періодичну послідовність;

- Discrete Pulse Generator - генератор дискретних імпульсних сигналів;

- Pulse Generator - генератор безперервних прямокутних імпульсів;

- Chirp Signal - генератор гармонійних коливань з частотою, яка



Як і інші блоки бібліотеки SimuLink, блоки-джерела можуть наструюватися користувачем, за винятком блоку Clock, робота якого основана на використанні апаратного таймера комп'ютера.

### Блок Constant

Блок призначений для генерування процесів, який є незмінним в часі, тобто мають постійне значення. Він має один параметр наструювання - Constant value, який може бути введений і як вектор-рядок з кількох елементів за загальними правилами MatLAB.

### Блок Signal Generator

Вікно наструювання цього блоку виглядає так, як показано на рис.1.11.

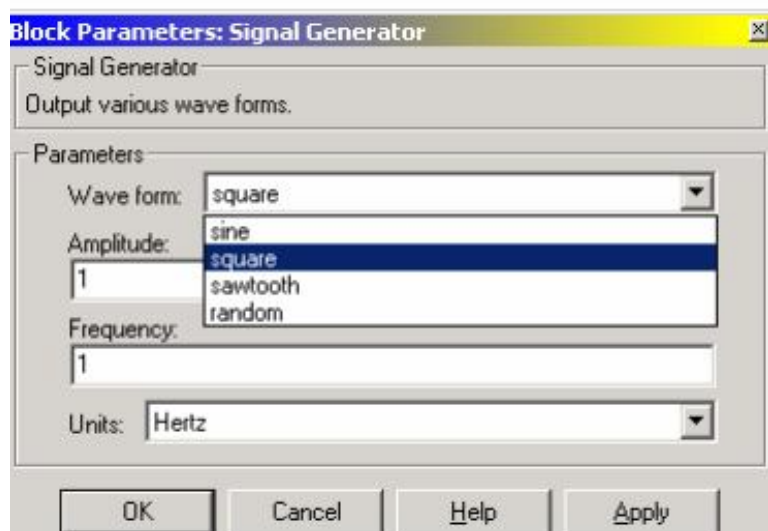


Рис. 1.11 Вікно параметрів блоку Signal Generator

Як видно, в параметри наструювання входять:

1 Wave form - форма хвилі - дозволяє вибрати одну з таких форм періодичного процесу:

- 1) Sine - синусоїдальні хвилі;
- 2) Square - прямокутні хвилі;
- 3) Sawtooth - трикутні хвилі;
- 4) Random - випадкові коливання;

- Amplitude - визначає значення амплітуди коливань, які генеруються;

- Frequency - задає частоту коливань;

- Units - дозволяє вибрати одну з одиниць вимірювання частоти за допомогою низхідного меню - Hertz (в Герцах) і Rad / Sec (в радіанах в секунду).

### Блок Step

Блок забезпечує формування сигналу у формі сходинок (або, як кажуть, - ступінчастого сигналу).

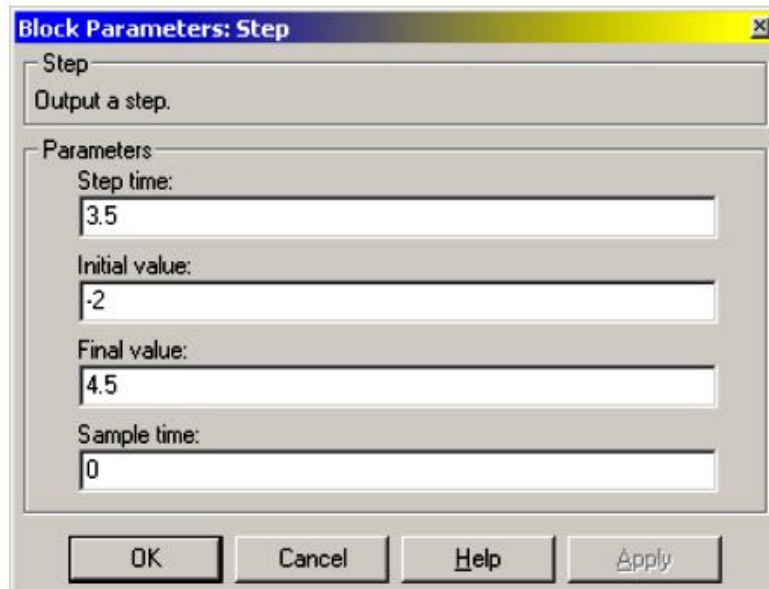


Рис.1.12 Вікно параметрів блоку Step

Блок має 3 параметри настроювання (рис. 1.12):

- Step time - час початку сходинки (момент стрибка сигналу) – визначає момент часу, в який відбувається стрибкоподібне зміна величини сигналу (за замовчуванням приймається рівним 1);
- Initial value - початкове значення - задає рівень сигналу до стрибка (значення за умовчанням – 0);
- Final value - кінцеве значення - задає рівень сигналу після стрибка (значення його за замовчуванням – 1).

### Блок Ramp

Блок формує безперервно наростаючий сигнал і має такі параметри настроювання (рис.1.13):

- Slope - значення швидкості наростання сигналу;
- Start time - час початку наростання сигналу;
- Initial output - значення сигналу до моменту початку його наростання.

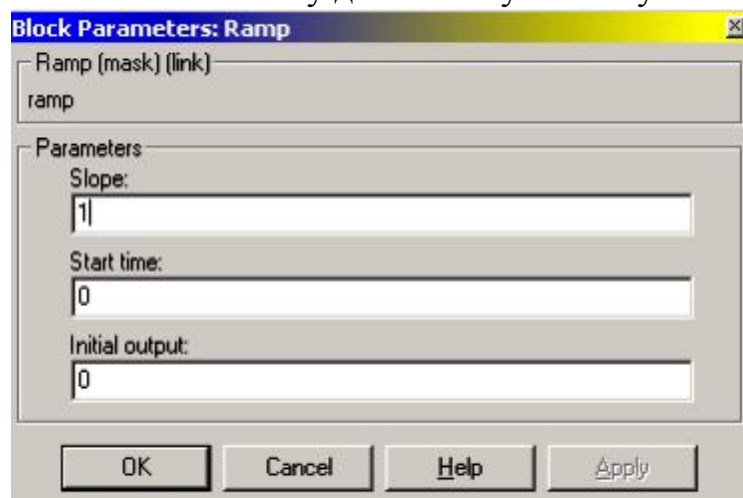


Рис.1.13 Вікно параметрів блоку Ramp

### Блок Sine Wave

Блок Sine Wave має такі настройки (рис. 1.14):

- Amplitude - визначає амплітуду синусоїдального сигналу;
- Frequency (rad / sec) - задає частоту коливань в радіанах в секунду;
- Phase (rad) - дозволяє встановити початкову фазу в радіанах;
- Sample time - визначає величину дискрет часу завдання значень синусоїдального сигналу.

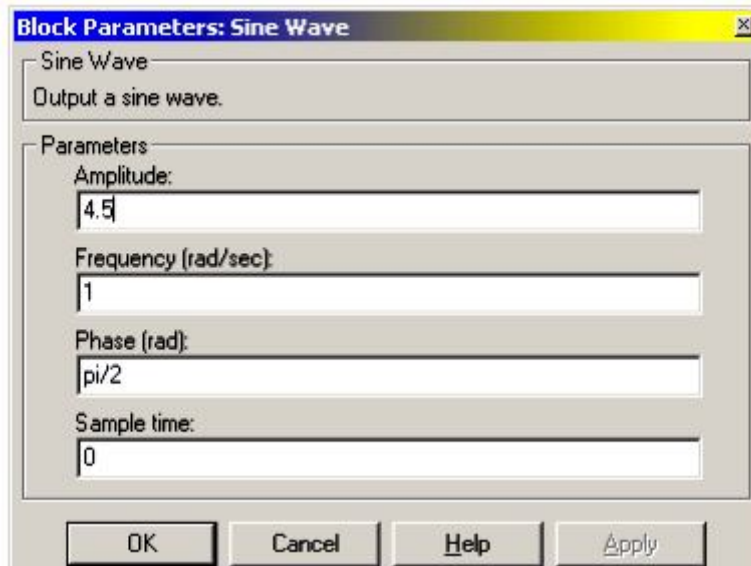


Рис.1.14 Вікно параметрів блоку Sine Wave

Відмінності цього блоку від генератора синусоїдальних коливань в блоці Signal Generator полягають у наступному:

- 1) в аналізованому блоці можна встановлювати довільну початкову фазу;
- 2) у ньому не можна задати частоту в Герцах.

### Блок Repeating Sequence

Цей блок містить два налаштування (рис.1.15):

- Time values - вектор значень часу, в які задані значення вихідної величини;
- Output values - вектор значень вихідної величини, які вона повинна прийняти в зазначені в першому векторі відповідні моменти часу.

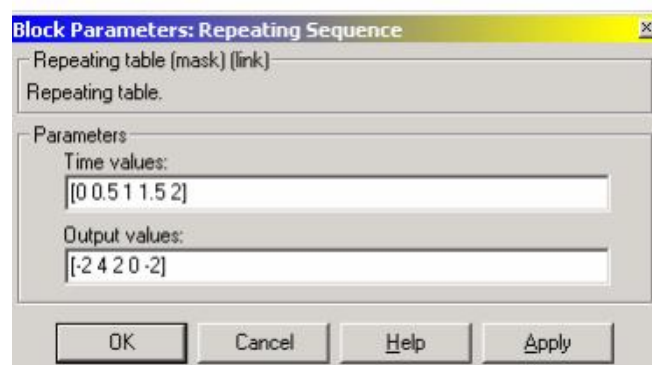


Рис.1.15 Вікно параметрів блоку Repeating Sequence

Блок забезпечує генерування коливань з періодом, рівним розходженню між останнім значенням вектора Time values і значенням першого його елемента. Форма хвилі усередині періоду є ламаною, що проходить через точки із зазначеними у векторах Time values і Output values координатами.

### Блок Diskret Pulse Generator

Блок генерує послідовності прямокутних імпульсів. У число параметрів, що настроюються цього блоку входять (рис.1.16):

- амплітуда сигналу (Amplitude), тобто висота прямокутного імпульсу;
- розмір періоду сигналу (Period), який відлічується в цілих числах кількості дискретів часу;
- ширина імпульсу (Pulse width), теж в дискретному часу;
- розмір затримки імпульсу щодо  $t = 0$  (Phase delay) - у дискретному часу;
- розмір дискретного часу (Sample time).

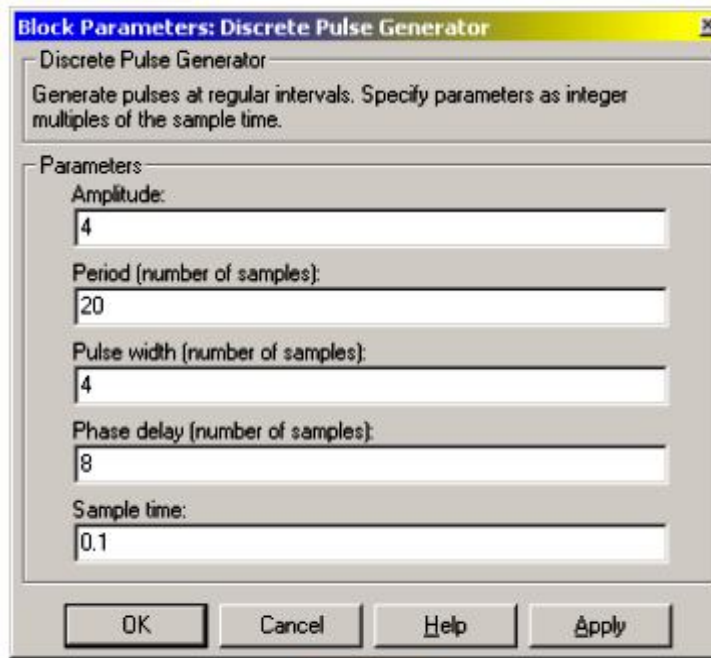


Рис.1.16 Вікно параметрів блоку Diskret Pulse Generator

Встановлення параметрів цього блоку доцільно починати з дискретного часу. Розмір кроку можна вказати як у формі константи, так і у формі обчислюваного виразу.

### Блок Pulse Generator

Блок Pulse Generator здійснює практично ту ж функцію, що і блок Discret Pulse Generator. Але параметри імпульсів задаються інакше (рис.1.17).

Тут вимагають завдання такі параметри:

- Period - період проходження імпульсів;
- Duty cycle (% of period) - тривалість прямокутного імпульсу (в відсотках від періоду);

- Amplitude - амплітуда (висота) імпульсу;
- Start time - початковий момент часу першого з імпульсів.

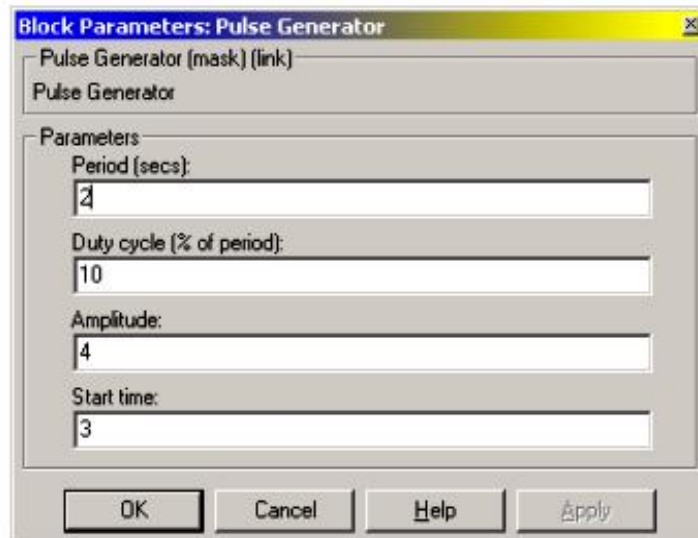


Рис.1.17 Вікно параметрів блоку Pulse Generator

### Блок Chirp Signal

Цей блок генерує синусоїдальний сигнал одиначної амплітуди змінної частоти, причому значення частоти коливань змінюється за впливом часу за лінійним законом. Відповідно цьому в ньому передбачені такі параметри налаштування (рис.1.18):

- Initial frequency (Hz) - початкове значення (при  $t = 0$ ) частоти в герцах;
- Target time (secs) - другий (позитивна величина) момент часу (в секундах);
- Frequency at target time (Hz) - значення частоти в цей другий момент часу.



Рис.1.18 Вікно параметрів блоку Chirp Signal

### Блок Random Number

Блок Random Number забезпечує формування сигналів, значення котрих в окремі моменти часу є випадковою величиною, розподіленої за нормальним (гауссових) законом із заданими параметрами.

Блок має чотири параметри настроювання (рис.1.19). Перші два – Mean і Variance - є середнім і середньоквадратичним відхиленням від цього середнього, третій - Initial seed - задає початкове значення бази для ініціалізації генератора послідовності випадкових чисел. При фіксованому значенні цього параметра генератор завжди виробляє одну і ту ж послідовність. Четвертий параметр (Sample time), як і раніше, задає величину дискрет часу.

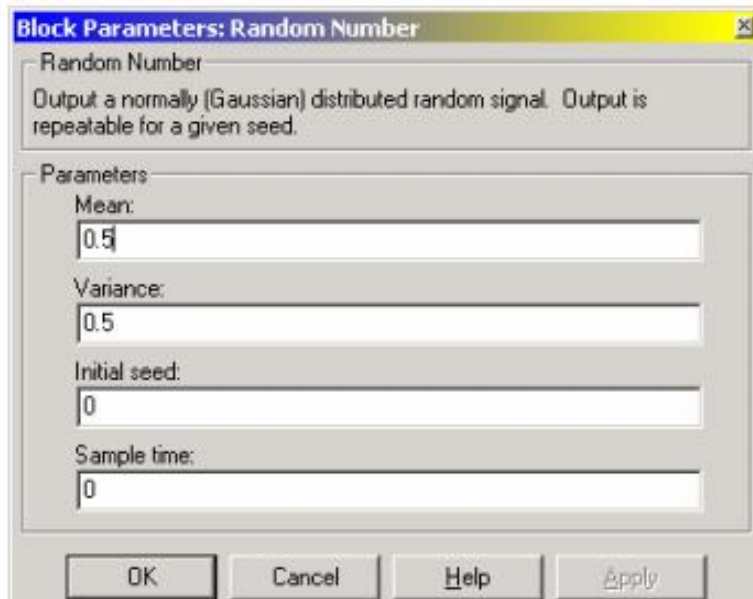


Рис. 1.19 Вікно параметрів блоку Random Number

### Блок Uniform Random Number

Цей блок формує сигнали, значення яких в окремі моменти часу є випадковою величиною, рівномірно розподіленої в заданому інтервалі. У число параметрів настроювання блоку входять (рис.1.20 ):

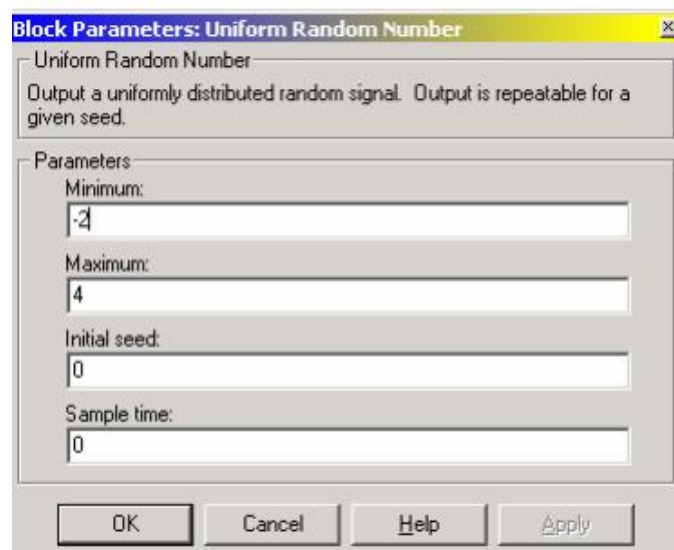


Рис.1.20 Вікно параметрів блоку Uniform Random Number

- Minimum - нижня межа випадкової величини;
- Maximum - верхня межа;
- Initial seed - початкове значення бази генератора випадкових чисел;
- Sample time - дискретного часу.

### Блок Band-Limited White Noise

Цей блок формує процес у вигляді частотно-обмеженого білого шуму. Параметри настроювання у нього такі (рис.1.21):

- Noise power - значення інтенсивності білого шуму;
- Sample time - значення дискрет часу (визначає верхнє значення частоти процесу);
- Seed - початкове значення бази генератора випадкової величини.

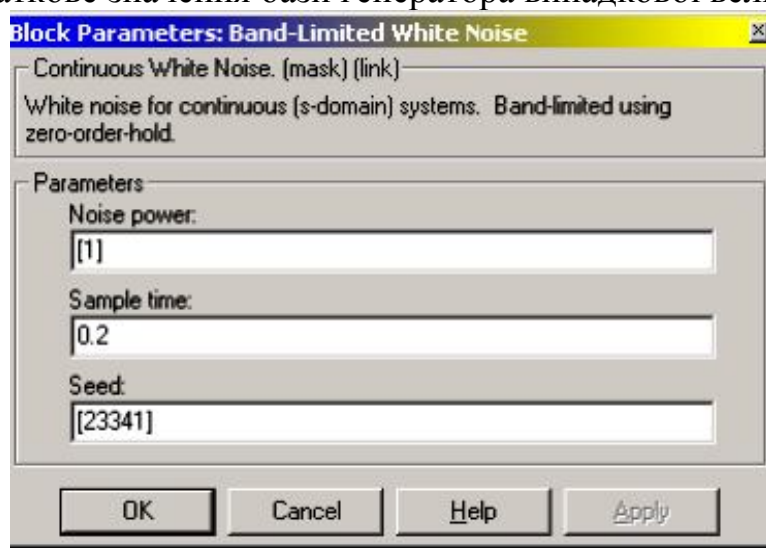


Рис.1.21 Вікно параметрів блоку Band-Limited White Noise

## **1.2. Розділ Math**

У цьому розділі містяться блоки, які реалізують деякі вбудовані математичні функції системи MatLAB (рис.1.22):

- Sum, який здійснює підсумовування сигналів, що надходять в нього;
- Product, що виконує множення або ділення вхідних сигналів;
- Dot Product - блок, який здійснює перемножування двох вхідних величин, якщо вони є скалярами, або визначення суми поелементно елементів двох вхідних векторів (однакової довжини);
- Gain - лінійне підсилювальний ланцюг;
- Slider Gain - ланка інтерактивного змінення коефіцієнта посилення;
- Matrix Gain - матричне підсилювальний ланцюг для багатовимірної системи;
- Шість блоків математичних стандартних операцій (MathFunction, TrigonometricFunction, MinMax, Abs, Sign і Rounding function);

- Три блоки логічних операцій (Combinatorial Logic, LogicalOperator і Relation Operator);
- Чотири блоки трансформування комплексних сигналів в дійсні і навпаки (Complex to Magnitude-Angle, Complex to Real-Imag, Magnitude-Angle to Complex і Real-Imag to Complex);
- AlgebraicConstraint - блок рішення алгебраїчних рівнянь.

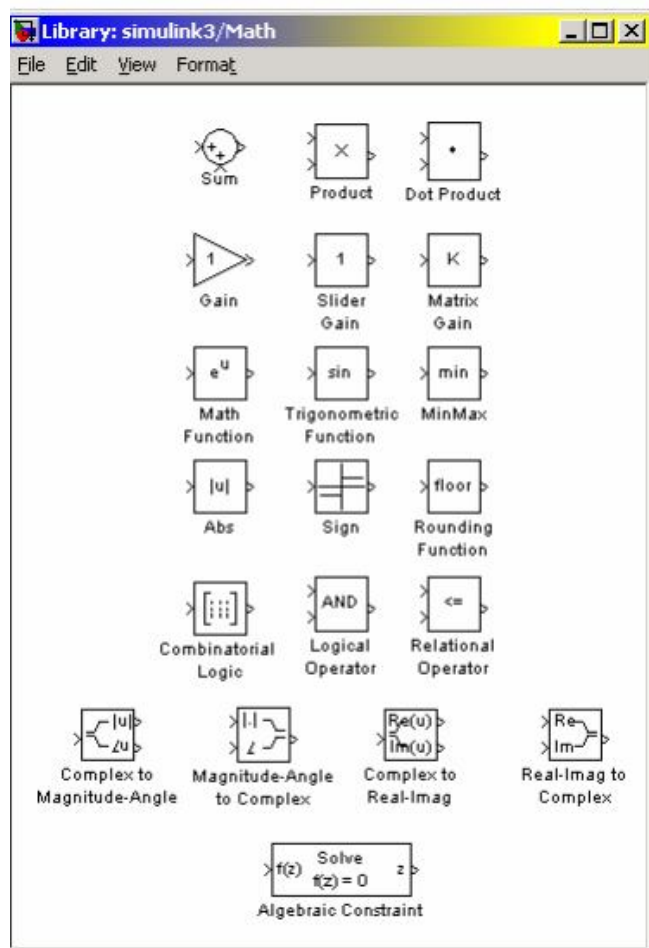


Рис.1.22 Бібліотека Math

Блок **Sum** може використовуватися в двох режимах:

- підсумовування вхідних сигналів (в тому числі з різними знаками);
- підсумовування елементів вектора, який надходить на вхід блоку.

Для керування режимами роботи блоку використовується два параметри настроювання - Icon Shape (Форма зображення) і List of signs (Список знаків). Перший може приймати два значення - round (круглий) і rectangular (прямокутний). Значення другого параметра можуть задаватися одним із трьох способів:

- у вигляді послідовності знаків "+" або "-"; при цьому кількість знаків визначає кількість входів блоку, а самий знак – полярність відповідного вхідного сигналу;

- у вигляді цілої позитивної і більше 1 константи; значення цієї константи визначає кількість входів блоку, а всі входи вважаються позитивними;

- у вигляді символу "1", який вказує, що блок використовується в другому режимі.

Блок **Product** виконує множення або ділення декількох вхідних сигналів. У параметри настроювання входять кількість входів блоку і вид виконуваної операції. Вікно настроювання блоку містить лише один параметр - Number of inputs (кількість входів). Якщо цей параметр (а, значить, кількість входів блоку) - позитивне число, більше 1, то всі вхідні величини перемножуються. Якщо в якості значення параметра настроювання блоку ввести "1", буде обчислюватися добуток елементів вхідного вектора.

При цьому на зображенні блоку виводиться символ P. В разі, коли результат виконання повинен містити ділення на деякі вхідні величини, в вікно Number of inputs слід вводити послідовність символів '\*' або '/' по числу входів блоку відповідно до того множиться або ділиться результат на



відповідну вхідну величину. Завдання значень цих параметрів аналогічно настроювання блоку Sum з бібліотеки Linear.

Блок **DotProduct** має лише два входи і не має параметрів настроювання. Його вхідні сигнали повинні бути векторами однакової довжини. Вихідна величина блоку в кожен момент часу дорівнює сумі творів відповідних елементів цих двох векторів.

Блок **Gain** здійснює множення вхідного сигналу на постійну величину, значення якої задається у вікні настроювання. У разі, коли вхідний сигнал є вектором довжиною N елементів, коефіцієнт посилення повинен бути вектором тієї ж довжини.

Блок **Slider Gain** є одним з елементів взаємодії користувача з моделлю. Він дозволяє в зручній діалоговій формі змінювати значення деякого параметра в процесі моделювання. Блок стає активним після того, як буде переміщений у вікно блок-схеми створюваної моделі. Щоб відчинити вікно з "повзункових" регулятором (рис. 7.51), необхідно двічі клацнути мишею на зображенні блоку.

Вікно Slider Gain має три поля введення інформації:

- для вказівки нижньої межі зміни параметра (Low);
- для вказівки верхньої межі зміни параметра (High);
- для вказівки поточного значення.

Поточне значення повинно розташовуватися всередині діапазону [Low, High]. Проте при виборі нового діапазону необхідно спочатку вказати нове значення параметра, а потім змінити межі діапазону.

Блок **Matrix Gain** здійснює перемноження вхідного вектора на постійну матрицю, зазначену у вікні настроювання (рис. 7.52). При цьому кількість рядків матриці посилення повинно збігатися з довжиною вхідного вектора. На виході одержується векторний сигнал довжиною, яка дорівнює кількості стовпців матриці посилення.

Нижче наводяться особливості тієї частини блоків, яка реалізує математичні функції.

Блок **Abs** формує абсолютне значення вхідного сигналу. Він не має параметрів настроювання.

Блок **Trigonometric Function** забезпечує перетворення вхідного сигналу за допомогою однієї з таких функцій MatLAB: sin, cos, tan, asin, acos, atan, atan2, sinh, cosh, tanh. Обрання необхідної функції здійснюється у вікні настроювання блоку за допомогою спадаючого меню.

Блок **Math Function** дозволяє вибрати для перетворення вхідного сигналу елементарно не тригонометричні і не гіперболічні функції, такі як exp, log,  $10^u$ , log10, magnitude<sup>2</sup>, square, sqrt, pow, reciprocal, hypot, rem, mod. Потрібна функція вибирається за допомогою меню, що випадає у вікні настроювання.

Блок **Rounding Function** містить різноманітні функції округлення, передбачені в MatLAB. Він здійснює округлення значень вхідного сигналу. Вибір конкретного методу округлення здійснюється також з допомогою випадаючого меню у вікні налаштування.

Блок **MinMax** здійснює пошук мінімального або максимального елемента вхідного вектора. Якщо входом є скалярна величина, то вихідна величина

збігається з вхідною. Якщо входів кілька, шукається мінімум або максимум серед входів. У число налаштувань входить вибір методу і кількість входів блоку.

Блок **Sign** реалізує нелінійність типу Сігнум-функції. У ньому немає параметрів налаштування. Блок формує вихідний сигнал, який приймає тільки три можливих значення: "+1" - у випадку, коли вхідний сигнал позитивний, "-1" – при негативному вхідному сигналі і "0" при вхідному сигналі, рівному нулю.

Для зазначених вище блоків ім'я вибраної функції виводиться на графічному зображенні блоку. Наступну групу утворюють блоки, які забезпечують логічну обробку вхідного сигналу. Загальним для всіх їх є те, що вихідна величину в них є булевої, тобто може досягати лише двох значень: "1" ("Істина") або "0" ("брехня"). У багатьох з них булевими повинні бути і все вхідні величини.

Блок **Relational Operator** реалізує операції відносини між двома вхідними сигналами  $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ,  $==$ ,  $\sim$  (відповідно: більше, менше, менше або дорівнює, більше або дорівнює, тотожно дорівнює, не дорівнює). Конкретна операція вибирається при налаштуванні параметрів блоку за допомогою випадального меню. Знак операції виводиться на зображенні блоку.

Блок **Logical Operator** містить набір основних логічних операцій. Вхідні величини повинні бути булевої. Вибір необхідної логічної операції здійснюється у вікні налаштування блоку за допомогою випадального меню. Другим параметром налаштування є кількість вхідних величин (Портів) блоку (Number of input ports), тобто кількість аргументів логічної операції.

Блок **Combinatorial Logic** забезпечує перетворення вхідних мулевих величин у вихідну в відповідності з заданою таблицею істинності. Блок має єдиний параметр налаштування - Truth table (таблиця істинності).

Чотири наступних блоку здійснюють перетворення комплексного вхідного сигналу в один або два дійсних вихідних сигналу, які є модулем, аргументом, дійсної чи уявної частиною вхідного сигналу (блоки Complex to Magnitude-Angle і Complex to Real-Imag), а також один або два вхідних дійсних сигналу в комплексний вихідний сигнал (блоки MagnitudeAngle to Complex і Real-Imag to Complex). Кількість входів або виходів визначається у вікні налаштування блоку.

### 1.3. Побудова графіків. Приклад

Завдання № 1. Треба побудувати графік наведений на рис.1.23 двома способами: в об'єктно-орієнтованому просторі Simulink та за допомогою m-файлів.

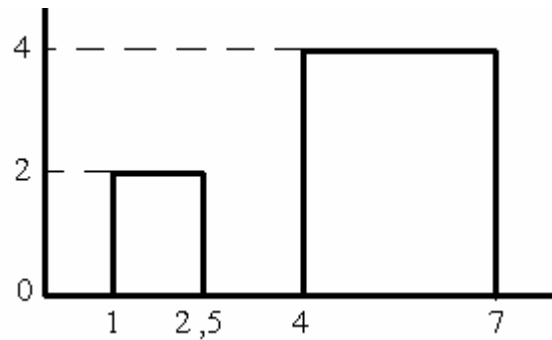


Рис.1.23 Побудова графіку № 1

Рішення.

1. Будова графіку за допомогою Simulink.

Для того щоб побудувати цей графік за допомогою Simulink треба використати блоки: Step, Add і Scope.

Складаємо з них схему:

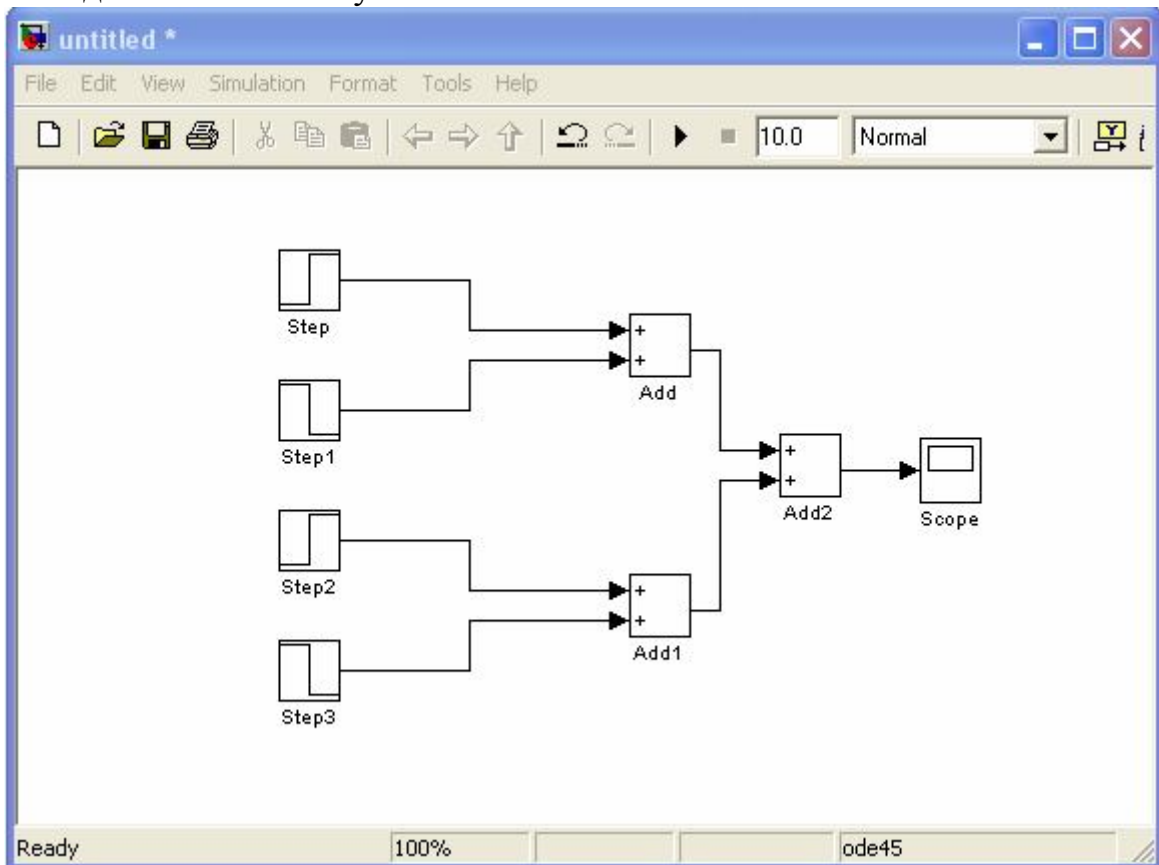


Рис.1.24 Блок-схема для першого прикладу

При цьому в блоках записуються наступні значення:

- блок Step:            Step time:            1
- Initial value:        0
- Final value:         2
- Sample time:        0

- блок Step1:      Step time:            2.5  
                           Initial value:        0  
                           Final value:         -2  
                           Sample time:        0
  
- блок Step2:      Step time:            4  
                           Initial value:        0  
                           Final value:         4  
                           Sample time:        0
  
- блок Step3:      Step time:            7  
                           Initial value:        0  
                           Final value:         -4  
                           Sample time:        0

Зображення отриманого графіку виводимо через блок Scope.

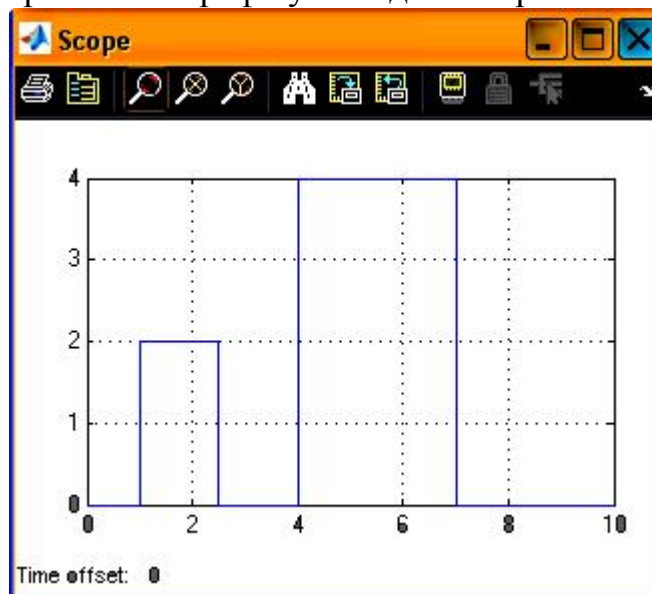


Рис.1.25 Отримане зображення в блоці Scope

## 2. Будова графіку за допомогою m-файлів

Для того щоб побудувати цей графік з використанням m-файлів треба у вікні MatLAB натиснути іконку «Створити новий M-файл», а потім у відкритому вікні набрати наступну програму:

```
x=[0 1 1 2.5 2.5 4 4 7 7 10];
y=[0 0 2 2 0 0 4 4 0 0];
```

```
plot(x,y)
```

де x, y – вектора, елементи яких є точками на побудованому нами графіку;  
 plot – команда відображення зображення, що використовує масиви даних.

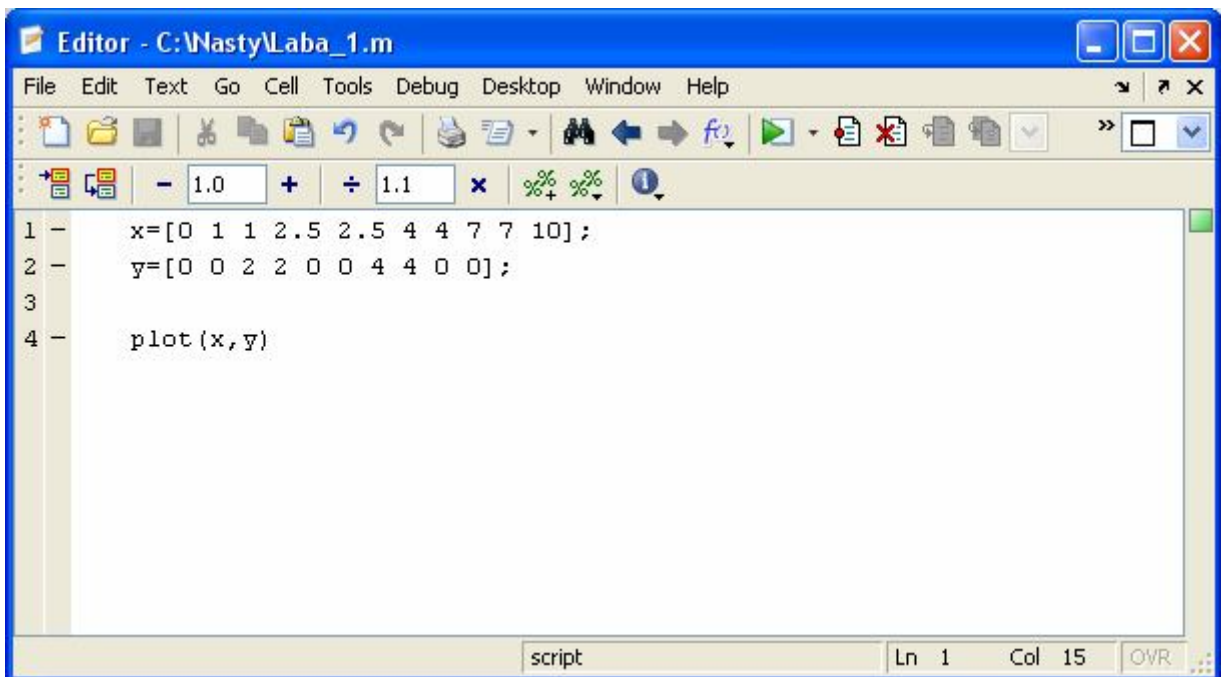


Рис.1.26 Вигляд вікна m-файлу з введеною програмою

Отримане зображення має вигляд:

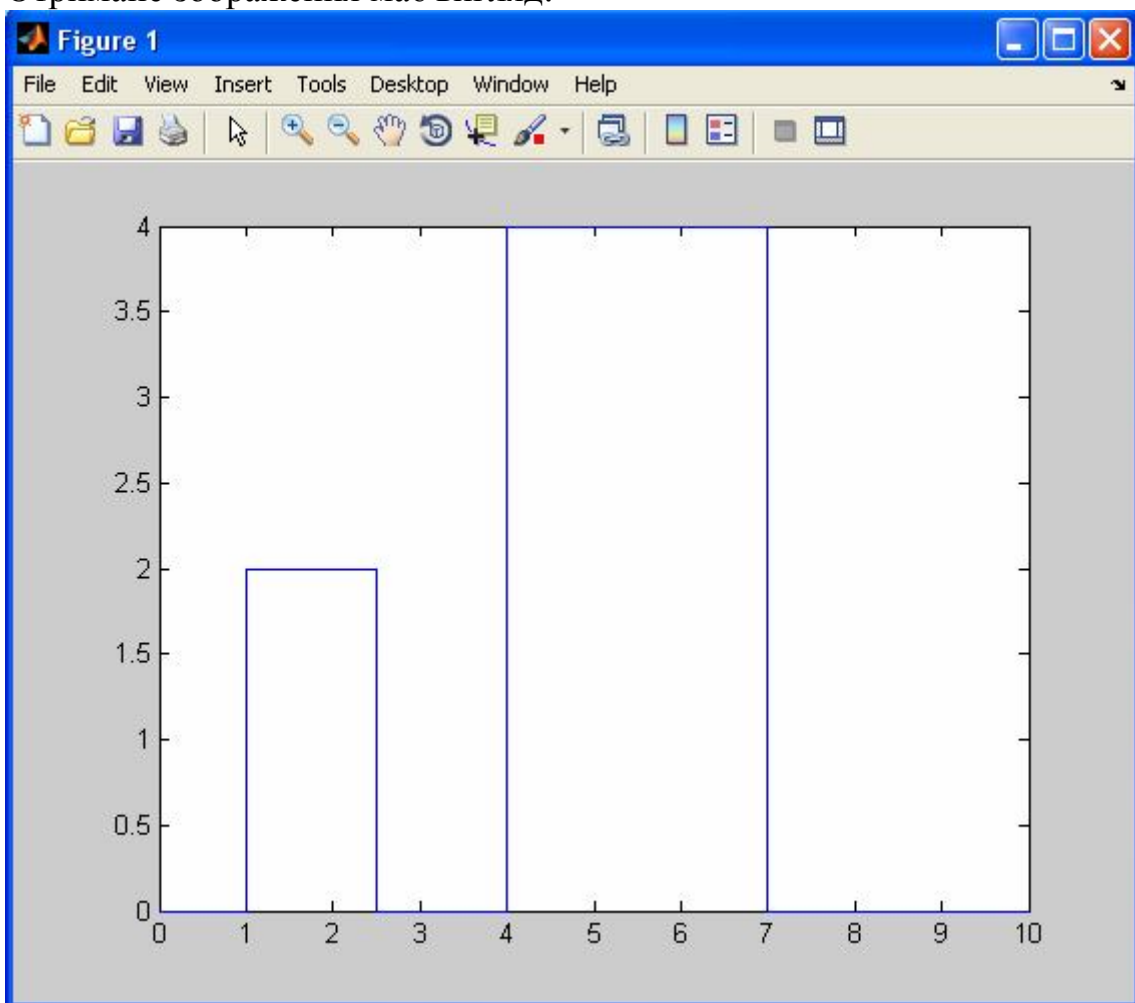


Рис. 1.27 Побудоване зображення із застосуванням команди plot

Завдання № 2. Треба побудувати графік наведений на рис.1.28 в об'єктно-орієнтованому просторі Simulink.

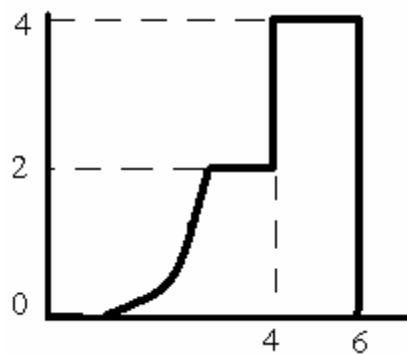


Рис.1.28 Графік завдання № 2

Для побудови цього графіку треба зібрати схему зображену на рис.1.29, що складається з таких елементів як: Ramp, Math Function, Saturation, Step, Add, Scope.

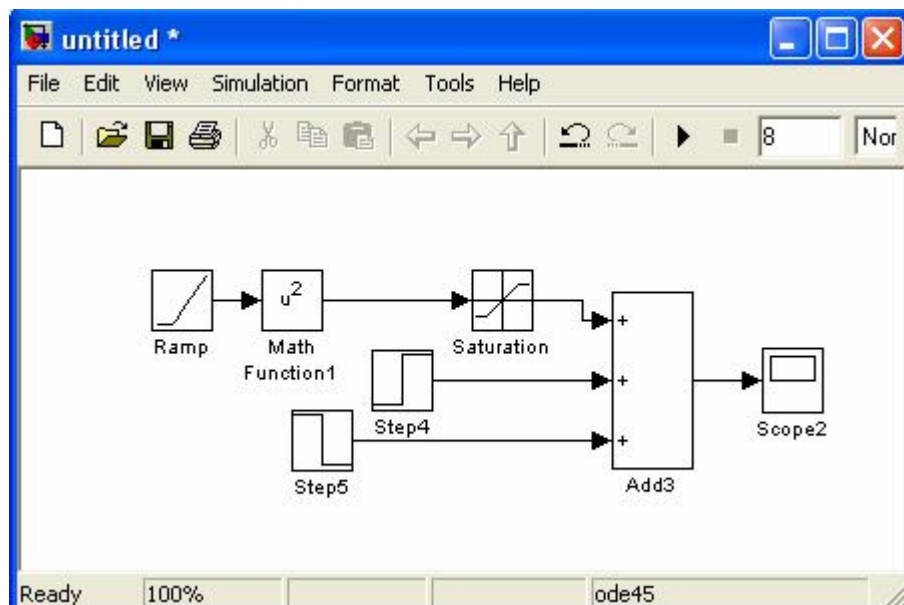


Рис.1.29 Блок схема завдання №2

При цьому в блоках записуються наступні значення:

- блок Step4:      Step time:            4  
                           Initial value:        0  
                           Final value:         2  
                           Sample time:        0
  
- блок Step5:      Step time:            6  
                           Initial value:        0  
                           Final value:         -4  
                           Sample time:        0

- Ramp:                    Slope:                    1  
                                  Start time:                1  
                                  Initial output:            0
  
- Math Function:    Square
  
- Saturation:            Upper limit:              2  
                                  Lower limit:              -0.5

Зображення отриманого графіку виводимо через блок Scope.

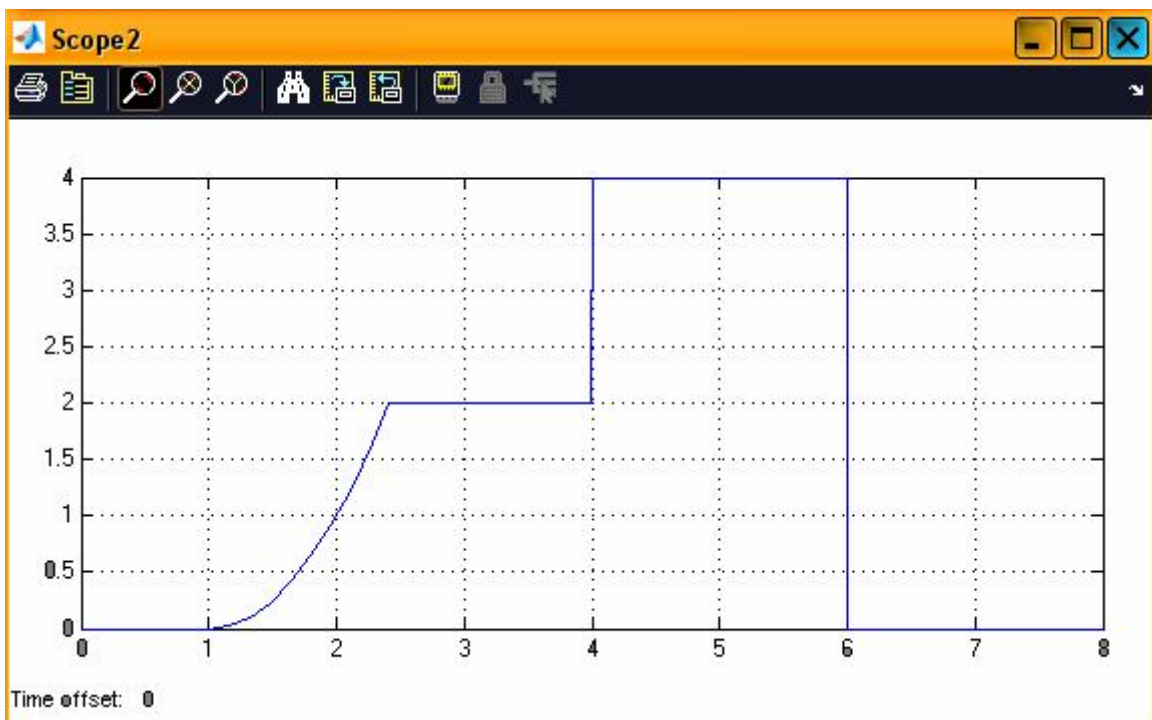


Рис.1.30 Зображення графіку завдання № 2 на Scope

#### 1.4 План виконання лабораторної роботи «Будова графіків з застосуванням Simulink та з використанням m-файлів»

1. Створюємо новий S-документ.
2. Надаємо ім'я S-документу і зберігаємо його.
3. За допомогою бібліотек Simulink виконуємо Завдання № 1 і Завдання № 2.
4. Виконуємо індивідуальні завдання.

## Лабораторна робота № 2

### РОЗРАХУНОК В ПАКЕТІ ПРОГРАМ МАТЛАВ. РОЗРАХУНОК ІЗ ЗАСТОСУВАННЯМ М-ФАЙЛІВ ТА S-МОДЕЛЕЙ

**Мета роботи:** вивчення основ виконання розрахунків із застосуванням m-файлів та s-моделей.

#### Теоретична частина

##### 2.1 Розрахунок в m-файлі.

Завдання 1. Потрібно виконати розрахунки:

$$Y1 = (X1 + 42,57 * X3 - \cos(X2))^{2X3 + \sqrt{X1 - X2}},$$

$$Y2 = \sin(X1 + X3 - \frac{X1 * X2^{3/2}}{X3}) + \sqrt[5]{X3},$$

де

$$X1 = 0,8056842$$

X2 являє собою графік вектор наступних значень: 5; 15; 25; 35; 45; 55.

X3 це результат виразу  $\pi * \sqrt{a}$ , де a змінюється від 1 до 6 з кроком 1.

Отримані Y1 та Y2 потрібно порівняти між собою, якщо  $Y1 > Y2$ , то потрібно логічну одиницю помістити в змінну `vidpovid`.

Рішення.

Відкриваємо новий m-файл. Зберігаємо під новою назвою, наприклад LAB2. Рішення програми ведеться зверху вниз построково. Спочатку в тексті програми потрібно визначити змінні або постійні X1, X2, X3, що використовуються в розрахунках.

$$X1 = 0.8056842$$

$$X2 = [5 \ 15 \ 25 \ 35 \ 45 \ 55]$$

$$X3 = 1 : 1 : 6$$

Останній вираз говорить, що потрібно створити вектор X3 у якому перший елемент буде дорівнювати одиниці, потім потрібно підвищити це



значення на «крок», в даному випадку на одиницю, записати отримане значення в вектор  $i$  продовжувати цей алгоритм поки значення  $X_3$  не стане дорівнювати шести. В загальному випадку цей вираз має вигляд:

$$X = X_0 : i : X_k, \quad (2.1)$$

де  $X$  – вектор, що створюється;  
 $X_0$  – значення першого елемента вектора;  
 $X_k$  – значення кінцевого елемента вектора;  
 $i$  – крок.

Результат запису перших трьох рядків буде:

```
>> LAB2
```

```
X1 =  
0
```

```
ans =  
8.056842000000000e-001
```

```
X2 =  
5 15 25 35 45 55
```

```
X3 =  
1 2 3 4 5 6
```

Тепер потрібно записати вирази які фігурують в нашому розрахунку.

```
Y1= (X1+X3.*42.57-cos(X2)) ^ ((X3.*2)+sqrt(X1-X2))
```

```
Y2= sin(X1+X3-((X1*(X2.^1.5))./X3))+(X3^(1/5))
```

Якщо тепер запустити наш m-файл от отримаємо результат:

```
Y1 =  
Column 1  
2.713514848154294e+002 +1.836989120160852e+003i  
Column 2  
-2.539821273736025e+007 -5.048935059346544e+007i  
Column 3  
1.212444147791127e+012 -4.126486973082565e+012i  
Column 4  
1.930861460788431e+017 -7.408717352104294e+017i  
Column 5
```

-8.982476392822596e+022 -1.712732138380843e+023i  
 Column 6  
 -7.998065515520705e+028 +1.042813447496832e+027i  
 Y2 =  
 Columns 1 through 3  
 2.050326905440538e-001 1.643175503938801e-001 2.242481343970636e+000  
 Columns 4 through 6  
 2.035462857378059e+000 2.290743552219665e+000 2.177147942050127e+000

Для того щоб порівняти отримані значення Y1 та Y2 потрібно застосувати оператор відношення «>»:

Vidpovid=Y1>Y2

Отримаємо такий вектор Vidpovid:

Vidpovid =  
 1 0 1 1 0 0

де «1» – означає істину, а «0» – неправду.

## 2.2 Розрахунок в S-моделі.

Завдання 2. Потрібно виконати розрахунки:

$$Y1 = 16 - 58X1\sqrt{X1-X2} - 4X3,$$

$$Y2 = X1 + X3 - \frac{X1 * X2^{3/2}}{X3} + \sqrt[5]{X3},$$

де

X1 – синусоїда з амплітудою 5, частотою 10 рад/с; кутом фази 0.

X2 – сигнал генератора імпульсів з амплітудою 2, періодом 0.2 с, шириною імпульсів 30 % від значення періоду.

X3 – сигнал генератора пилкоподібних імпульсів з амплітудою 10, періодом 0.2 с.

Отримані Y1 та Y2 потрібно вивести на один Score, таким чином щоб у вікні цього блоку вони розміщалися як по одинці, так і разом.

Рішення.

Для того щоб задати X1 потрібно вставити в вікно S-моделі блок Sine Wave із бібліотека Sources. І в його параметрах змінити амплітуду (Amplitude), частоту (Frequency) і кут фази (Phase) у відповідності до завдання.

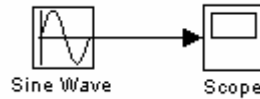


Рис.2.1 Піктограма блоку Sine Wave бібліотеки Sources

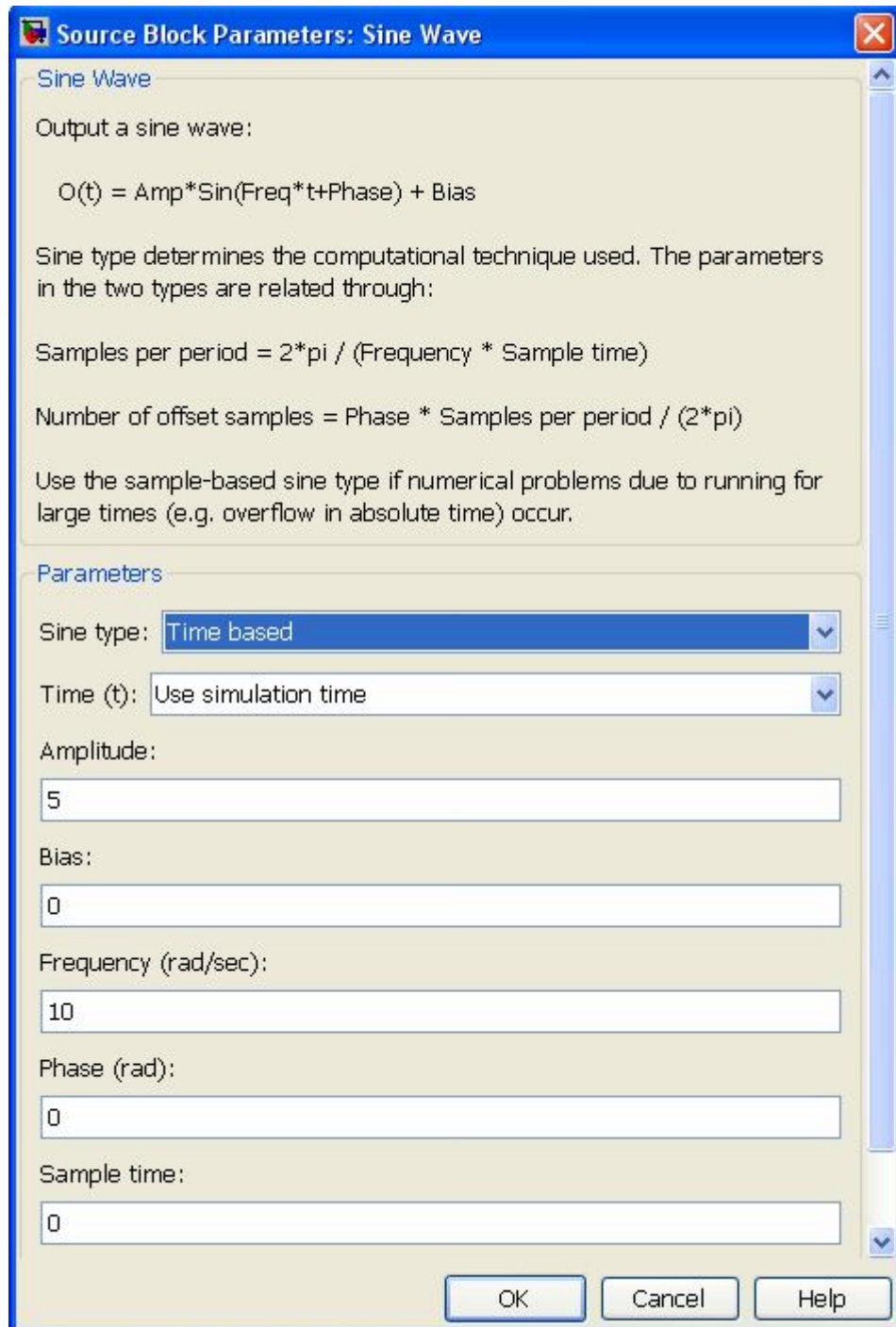


Рис.2.2 Параметри блоку Sine Wave

Для того щоб задати X2 потрібно вставити в вікно S-моделі блок Pulse Generator із бібліотека Sources. І в його параметрах змінити амплітуду (Amplitude), період (Period) і ширину імпульсів (Pulse Width) у відповідності до завдання.

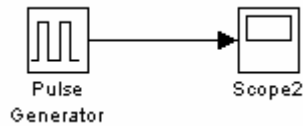


Рис.2.3 Піктограма блоку Pulse Generator бібліотеки Sources

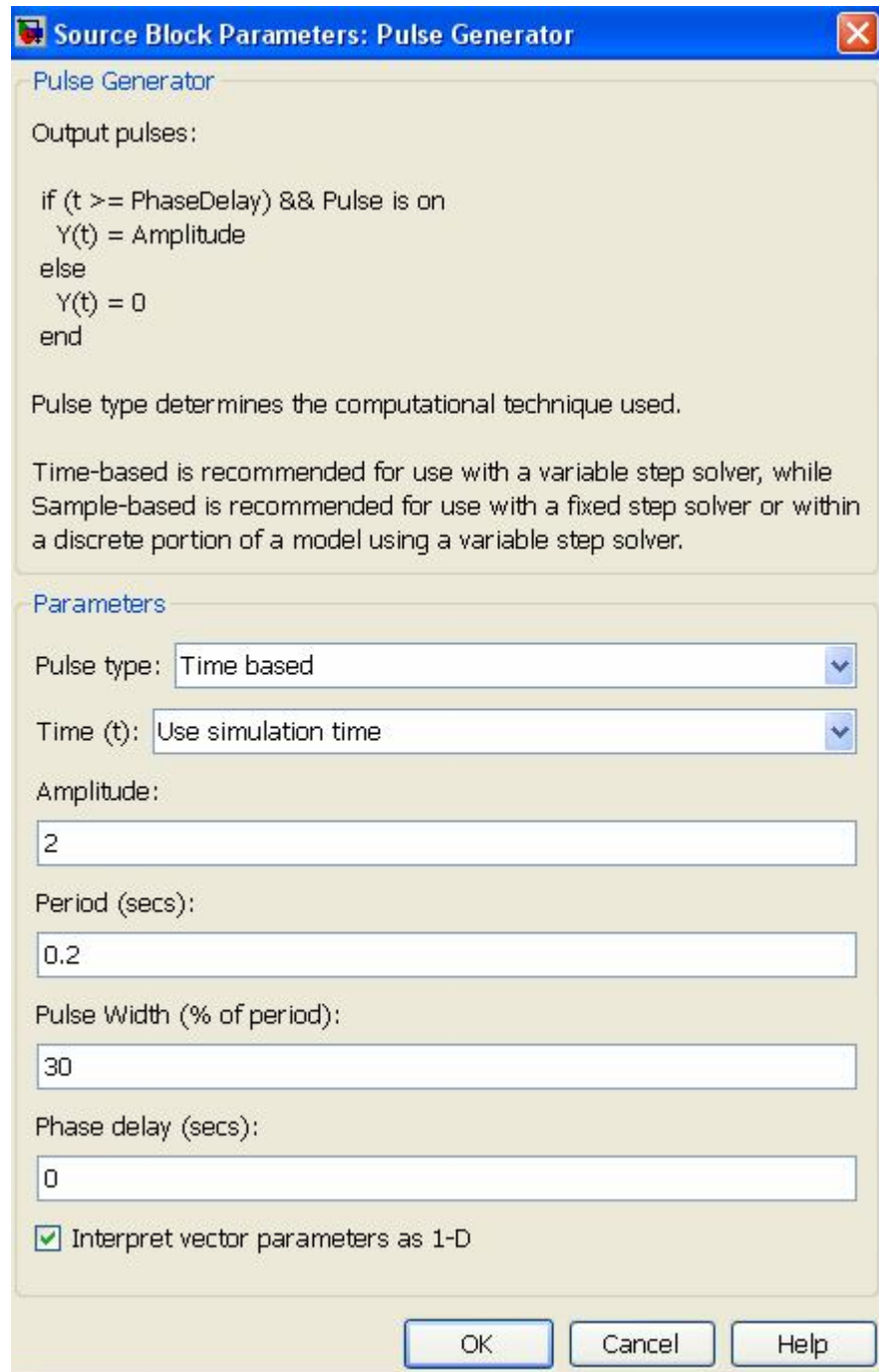


Рис.2.4 Параметри блоку Pulse Generator

Для того щоб задати X3 потрібно вставити в вікно S-моделі блок Repeating Sequence із бібліотека Sources. І в його параметрах змінити амплітуду (Output values) та період (Times values) у відповідності до завдання.

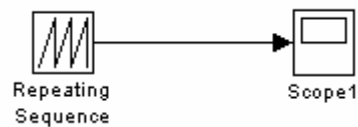


Рис.2.5 Піктограма блоку Repeating Sequence бібліотеки Sources

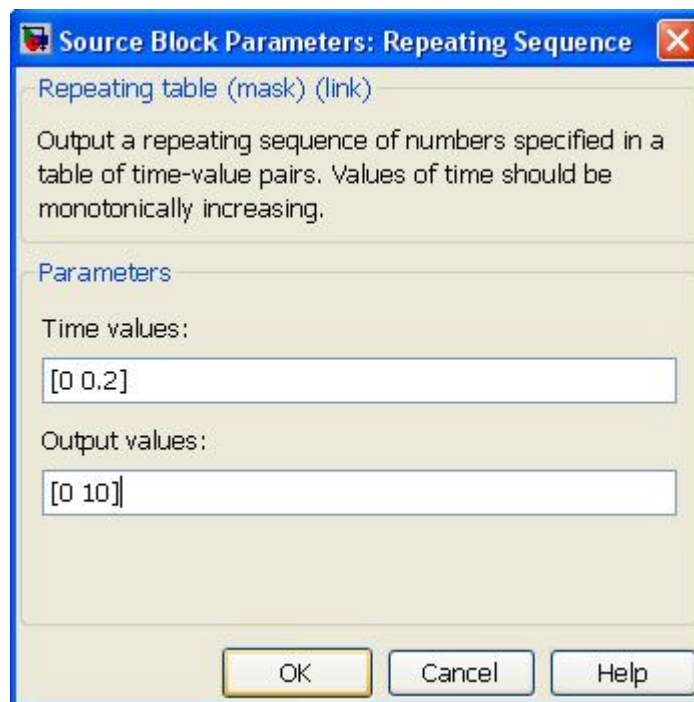
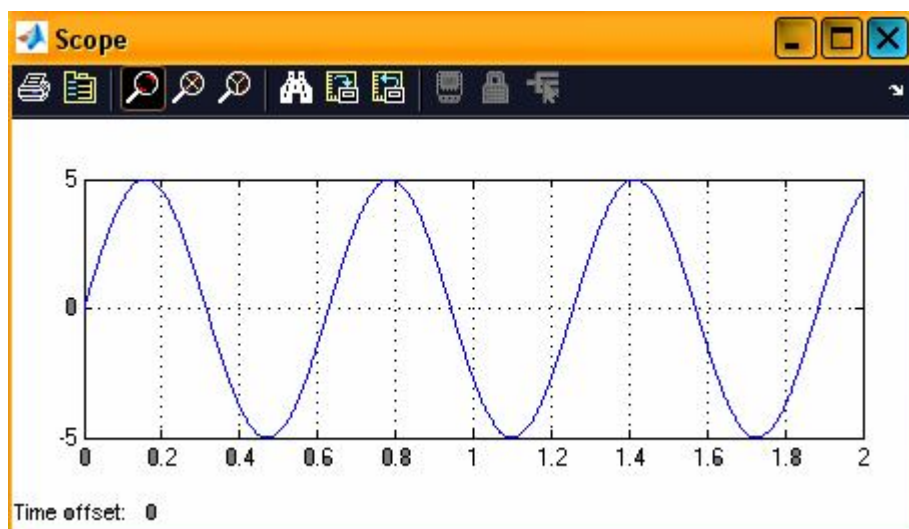
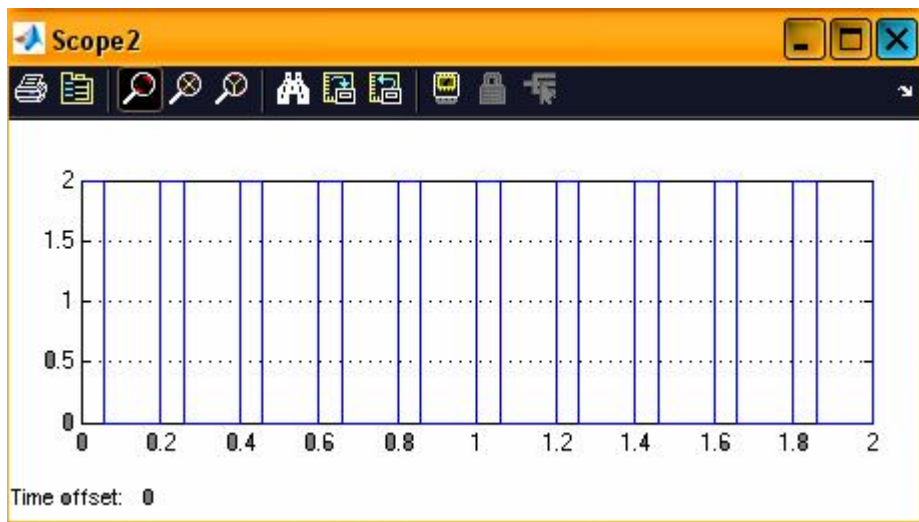


Рис.2.6 Параметри блоку Repeating Sequence

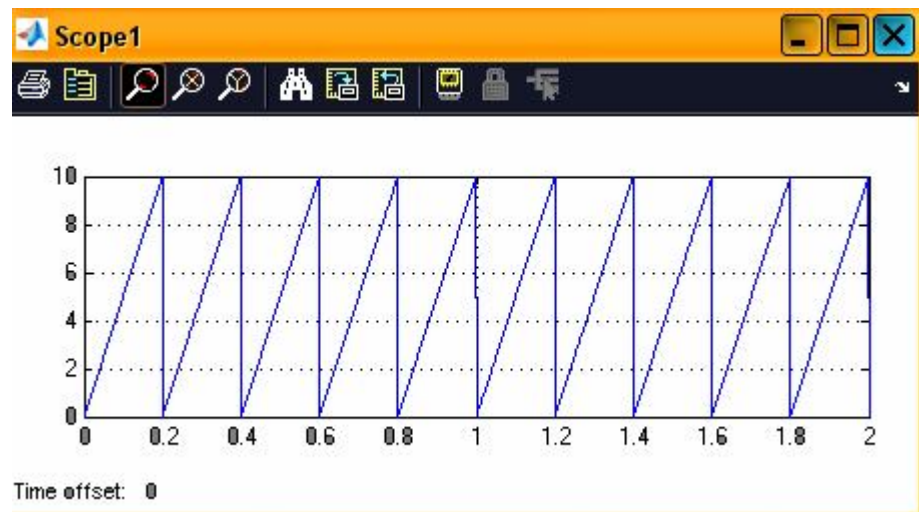
Отримаємо такі види сигналів:



a)



б)



в)

Рис.2.7 Форма сигналів:

- а) – сигнал X1 (Sine Wave); б) – сигнал X2 (Pulse Generator);  
в) – сигнал X3 (Repeating Sequence)

Тепер побудуємо співвідношення відповідні виразам

$$Y1 = 16 - 58X1\sqrt{X1-X2} - 4X3,$$

$$Y2 = X1 + X3 - \frac{X1 * X2^{3/2}}{X3} + \sqrt[5]{X3},$$

та знайдемо Y1 і Y2.

Блочна схема відповідна таким виразам буде мати вид:

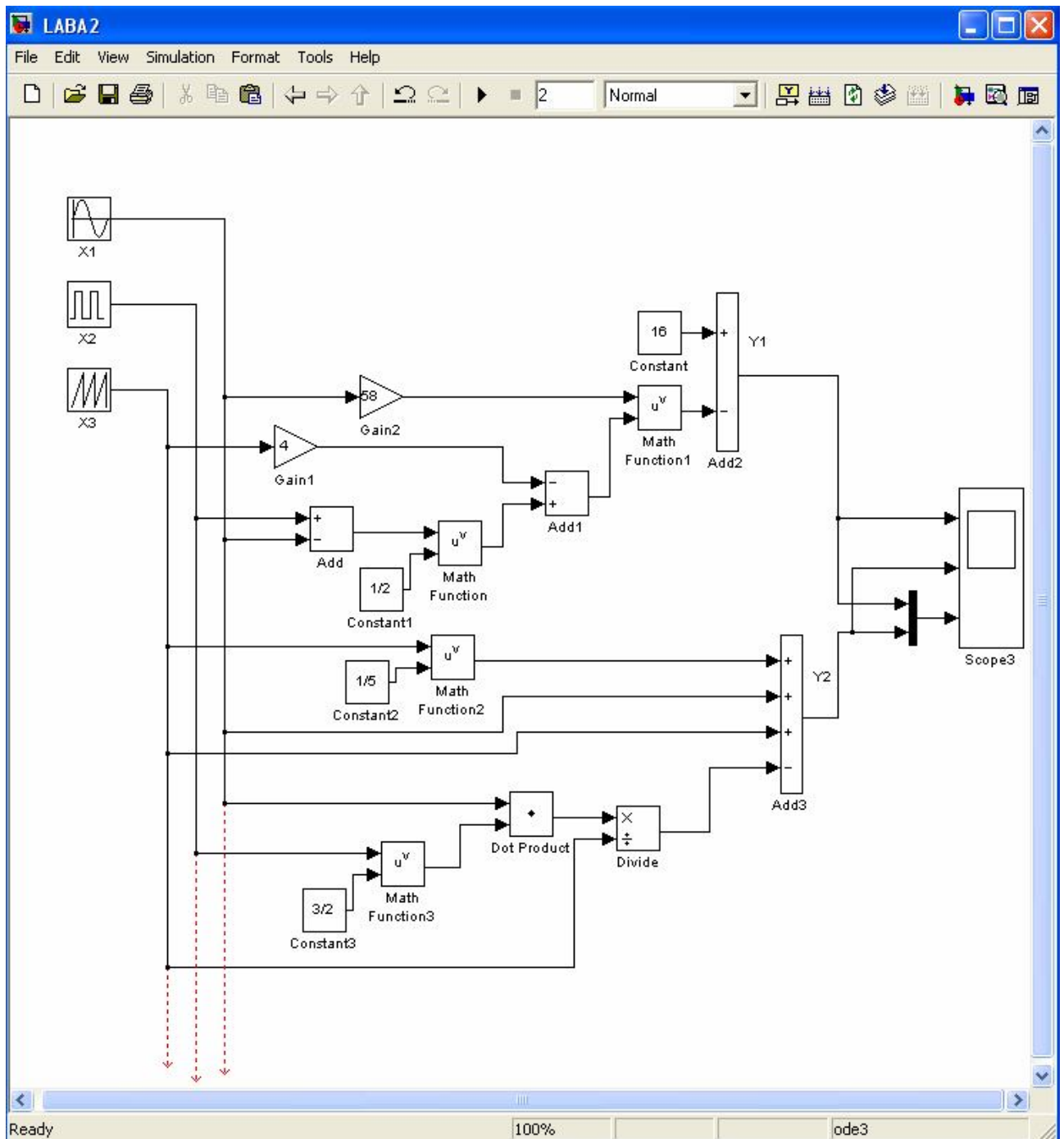


Рис.2.8 Отримана блок схема в S-моделі

Для того щоб зробити три або більше входів в блоці Scope потрібно два рази клацнути по блоці, в отриманому вікні

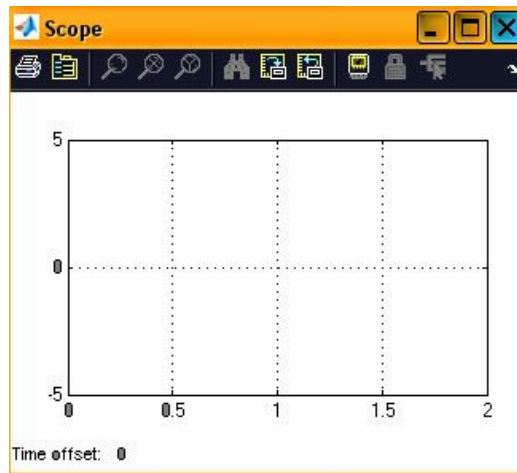



Рис.2.9 Вигляд робочої середовищі блоку Scope

натиснути піктограму  Parameters, з'явиться вікно параметрів блоку Scope.

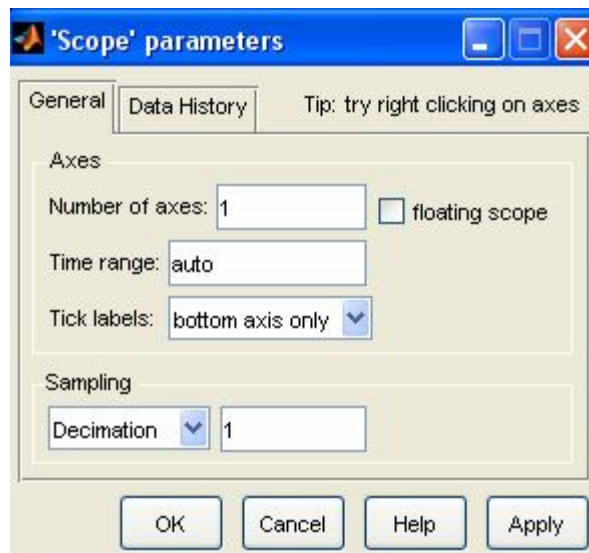


Рис.2.10 Вікно параметрів блоку Scope

В цьому вікні нас цікавить строчка «Number of axes», саме тут можна задати кількість входів і відповідно кількість окремих графіків, що будуть зображуватися в одному вікні Scope. Для нашого варіанту завдання нам потрібно поставити в це поле значення 3 та натиснути кнопку ОК.

Вид отриманих характеристик показано на рис.2.11., де в першому вікні зображене значення  $Y_1$ , у другому –  $Y_2$ , а в третьому вікні зображенні обидва ці графіки. Для того щоб це зробити ми скористалися блоком Mux із бібліотеки Signal Routing.



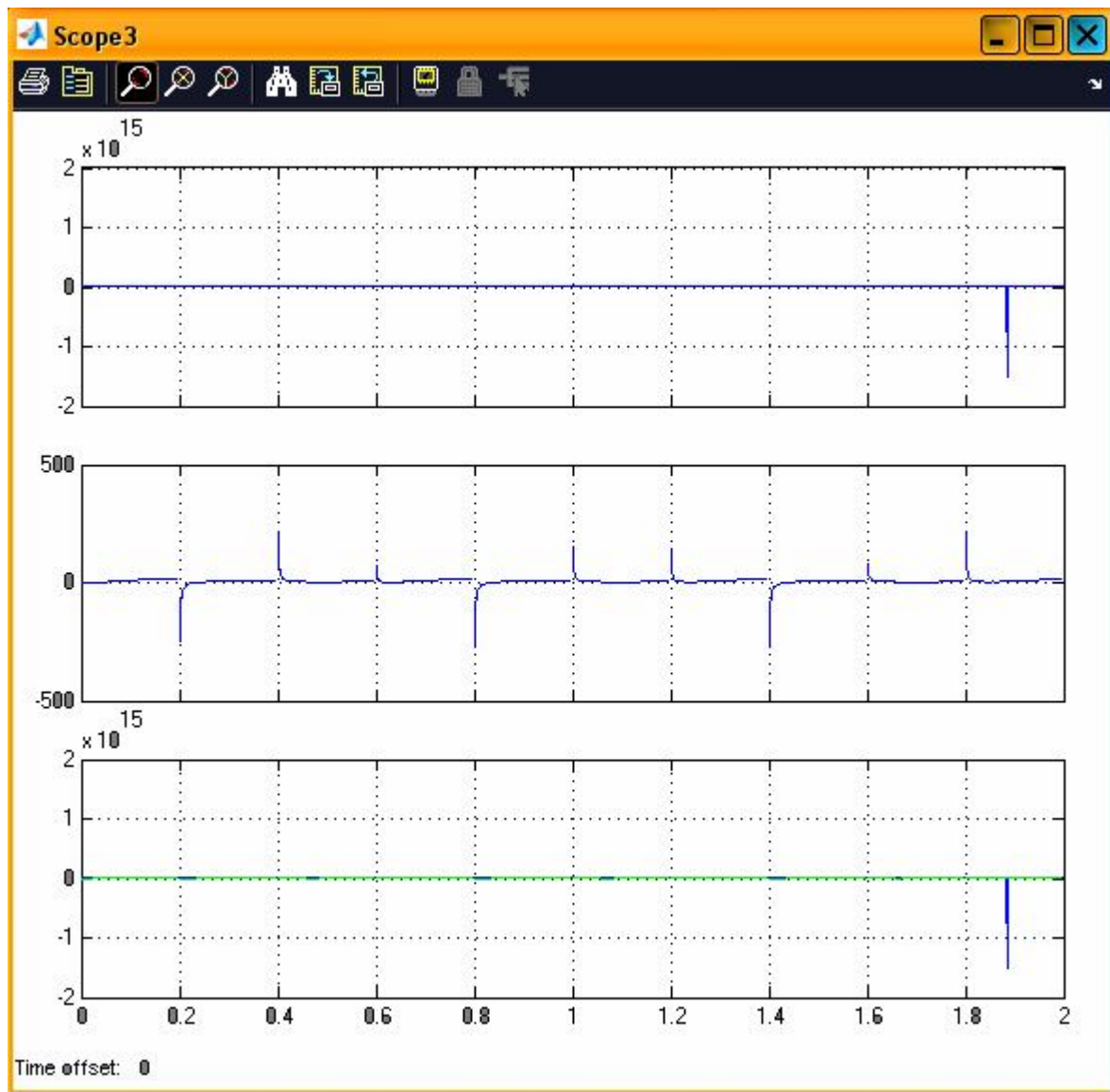


Рис.2.11 Результат рішення завдання 2

### 2.3 План виконання лабораторної роботи «Розрахунок в пакеті програм MATLAB. Розрахунок із застосуванням m-файлів та S-моделей»

1. Створюємо новий m-файл.
2. Надаємо ім'я m-файлу і зберігаємо його.
3. Виконуємо завдання № 1.
4. Створюємо новий S-документ.
5. Надаємо ім'я S-документу і зберігаємо його.
6. Виконуємо завдання № 2.
7. Виконуємо індивідуальні завдання.

## 2.4 Індивідуальні завдання

№ вар.	Завдання
1	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 1 по 15 з кроком 0,5;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 546 – 273+3<sup>4</sup>;  Знайти:</p> $Y1 = (256X1 + \frac{15}{X2 - 0.146} X3) \sqrt{13X1 - X2},$ $Y2 = \tan(X1 + X3 \frac{X1/X2^{3/2}}{Y1}) + 7\sqrt[7]{Y1},$
	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = (256X1 + \frac{15}{X2 - 0.146} X3) \sqrt{13X1 - X2},$ $Y2 = \tan(X1 + X3 \frac{X1/X2^{3/2}}{Y1}) + 7\sqrt[7]{Y1},$ <p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
2	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 15 по 25 з кроком 0,5;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 546,50146  Знайти:</p> $Y1 = (\sqrt[3]{256X2} + \frac{1}{2/X2 - 0.46} X1) \sqrt{X3 * X1 - X2},$ $Y2 = \cos(X1 + X3 \frac{X1/(X2^{1/2} - 1)}{Y1 * 2X2}) + 7\sqrt[7]{Y1},$
	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = (\sqrt[3]{256X2} + \frac{1}{2/X2 - 0.46} X1) \sqrt{X3 * X1 - X2},$ $Y2 = \cos(X1 + X3 \frac{X1/(X2^{1/2} - 1)}{Y1 * 2X2}) + 7\sqrt[7]{Y1},$

№ вар.	Завдання
	<p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
3	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 10 по 150 з кроком 10;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 12+2<sup>16</sup>;  Знайти:  <math display="block">Y1 = (256X1 - \frac{105}{X1 - 146} X2) \sqrt{X3 + X2 - X1},</math> <math display="block">Y2 = X1 + X3 \frac{X1/X2^{3/2}}{\sqrt{X1 - X2}},</math></p>
	Завдання № 2
<p>Розрахувати  <math display="block">Y1 = (256X1 - \frac{105}{X1 - 146} X2) \sqrt{X3 + X2 - X1},</math> <math display="block">Y2 = X1 + X3 \frac{X1/X2^{3/2}}{\sqrt{X1 - X2}},</math> де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>	
4	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 9 по 81 з кроком 3;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 4+3<sup>7</sup>;  Знайти:  <math display="block">Y1 = (1/(5X1) - \frac{8X3 - X2 * (X1 + 2X3)}{X1^{X2-3} - 1,46} X2) \sqrt{X1 - X2},</math> <math display="block">Y2 = X2 + X1 \frac{X3/X2^{5/2} - X3}{\sqrt{X1 - X2/X3}},</math></p>

№ вар.	Завдання
	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = (1/(5X1) - \frac{8X3 - X2 * (X1 + 2X3)}{X1^{X2-3} - 1,46} X2) \sqrt{X1-X2},$ $Y2 = X2 + X1 \frac{X3/X2^{5/2-X3}}{\sqrt{X1 - X2/X3}},$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3</p> <p>X1 – набір цифр з 24 по 56 з кроком 2;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 8+2<sup>6</sup>;  Знайти:</p> $Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 * (241/X2)))} - 505 \frac{1}{X3 - X1},$ $Y2 = X2 + X3 \frac{X1/X2^{15/2}}{\sqrt{X1 - X3}},$
5	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 * (241/X2)))} - 505 \frac{1}{X3 - X1},$ $Y2 = X2 + X3 \frac{X1/X2^{15/2}}{\sqrt{X1 - X3}},$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
6	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3</p> <p>X1 – набір цифр з 18 по 42 з кроком 2;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 8+3<sup>6</sup>;</p>

№ вар.	Завдання
	<p>Знайти:</p> $Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 - X2) * (241/X2))} - 55 \frac{1}{X3 + X1},$ $Y2 = X2 - X3 \frac{X1^2/X2^{15/2}}{\sqrt{X1 + X3}},$ <p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 - X2) * (241/X2))} - 55 \frac{1}{X3 + X1},$ $Y2 = X2 - X3 \frac{X1^2/X2^{15/2}}{\sqrt{X1 + X3}},$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 24 по 56 з кроком 2;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 8+2<sup>6</sup>;  Знайти:</p> $Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 * (241/X2))} - 505 \frac{1}{X3 - X1},$ $Y2 = X2 + X3 \frac{X1/X2^{15/2}}{\sqrt{X1 - X3}},$
7	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 * (241/X2))} - 505 \frac{1}{X3 - X1},$ $Y2 = X2 + X3 \frac{X1/X2^{15/2}}{\sqrt{X1 - X3}},$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>

№ вар.	Завдання
8	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 3 по 117 з кроком 3;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 8+2<sup>6</sup>;  Знайти:</p> $Y1 = \sqrt{X3 + (X2 - 51X1 * (X1 - 2) * (241/X2))} - 5 \frac{1}{X1 + X2},$ $Y2 = X3 - X2 \frac{X1/X3^{15/3.5}}{\sqrt{X1 - X3}},$
	Завдання № 2
	<p>Розрахувати</p> $Y1 = \sqrt{X3 + (X2 - 51X1 * (X1 - 2) * (241/X2))} - 5 \frac{1}{X1 + X2},$ $Y2 = X3 - X2 \frac{X1/X3^{15/3.5}}{\sqrt{X1 - X3}},$ <p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
9	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 3 по 225 з кроком 3;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 11+3<sup>2</sup>;  Знайти:</p> $Y1 = 5 \frac{X3}{X1 + X2} + \sqrt{X3 + (X2 + *(X1 - 2) * (241/X2))},$ $Y2 = X3^{3,46} - X2 \frac{X1/X3}{\sqrt{X1 - X3}},$
	Завдання № 2
	<p>Розрахувати</p> $Y1 = 5 \frac{X3}{X1 + X2} + \sqrt{X3 + (X2 + *(X1 - 2) * (241/X2))},$ $Y2 = X3^{3,46} - X2 \frac{X1/X3}{\sqrt{X1 - X3}},$

№ вар.	Завдання
	<p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
10	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 34 по 402 з кроком 2;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 7+4<sup>6</sup>;  Знайти:  <math display="block">Y1 = \sqrt{X3 + (X2 - 108X1 * (X1 - 39) * (41/X2))},</math> <math display="block">Y2 = X3 + X2 \frac{X1^{5/3.5}}{\sqrt{X1 + X3}},</math></p>
	Завдання № 2
	<p>Розрахувати  <math display="block">Y1 = \sqrt{X3 + (X2 - 108X1 * (X1 - 39) * (41/X2))},</math> <math display="block">Y2 = X3 + X2 \frac{X1^{5/3.5}}{\sqrt{X1 + X3}},</math> де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
11	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 4 по 42 з кроком 2;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 25<sup>2</sup>;  Знайти:  <math display="block">Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 * (241/X2)))}</math> <math display="block">Y2 = X2 + X3 \frac{X1/X2^{15/2}}{\sqrt{X1 - X3}},</math></p>
	Завдання № 2
	<p>Розрахувати  <math display="block">Y1 = \sqrt{X1 + (X2 - 54X1 * (X3 * (241/X2)))}</math> <math display="block">Y2 = X2 + X3 \frac{X1/X2^{15/2}}{\sqrt{X1 - X3}},</math></p>

№ вар.	Завдання
	<p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
12	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 12 по 291 з кроком 3;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 7+5<sup>4</sup>;  Знайти:  <math display="block">Y1 = \frac{\sqrt{X3 + (X2 - 108X1 * (X1 - 39) * (41/X2))}}{X3 + X2 - \arcsin(X1 + X2^2)},</math> <math display="block">Y2 = X3 + X2 \frac{\cos(X1^{5/3.5})}{\sqrt{X1 + \sin(X3)}},</math></p>
13	<p>Завдання № 2</p> <p>Розрахувати  <math display="block">Y1 = \frac{\sqrt{X3 + (X2 - 108X1 * (X1 - 39) * (41/X2))}}{X3 + X2 - \arcsin(X1 + X2^2)},</math> <math display="block">Y2 = X3 + X2 \frac{\cos(X1^{5/3.5})}{\sqrt{X1 + \sin(X3)}},</math> де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
13	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 5 по 45 з кроком 5;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 5<sup>4</sup>;  Знайти:  <math display="block">Y1 = 10 + 10 \sin(\sqrt{X3 + (X2 - 108X1 * (X1 - 39) * (41/X2))}),</math> <math display="block">Y2 = \cos(X3) + \arctan(X2 \frac{X1^{5/3.5}}{\sqrt{X1 + X3}}),</math></p>



№ вар.	Завдання
	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = 10 + 10 \sin(\sqrt{X3 + (X2 - 108X1 * (X1 - 39) * (41/X2))}),$ $Y2 = \cos(X3) + \arctan(X2 \frac{X1^{5/3.5}}{\sqrt{X1 + X3}}),$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
14	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3</p> <p>X1 – набір цифр з 11 по 19487171 з кроком 11;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 7+5<sup>4</sup>;</p> <p>Знайти:</p> $Y1 = \cos(\pi * \sqrt{\frac{X2}{X3} + (X2 + 108X1/(41/X2))}),$ $Y2 = X3 + \arctan(X2 \frac{(1/X1)^{3.5}}{\sqrt{X1 + \sin(\pi * X3)}}),$
	<p>Завдання № 2</p>
	<p>Розрахувати</p> $Y1 = \cos(\pi * \sqrt{\frac{X2}{X3} + (X2 + 108X1/(41/X2))}),$ $Y2 = X3 + \arctan(X2 \frac{(1/X1)^{3.5}}{\sqrt{X1 + \sin(\pi * X3)}}),$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>

№ вар.	Завдання
15	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 2 по 2<sup>32</sup> з кроком 2;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 2<sup>4</sup>+5<sup>2</sup>;  Знайти:</p> $Y1 = \sqrt{X3 + (X2 - 51X1 * (X1 - 2) * (21/X2))} - 2 \frac{1}{X1 + X2},$ $Y2 = X3 - X2 \frac{(X1^2 / X3)^{15/3.5}}{\sqrt{X1 - X3}},$
15	Завдання № 2
	<p>Розрахувати</p> $Y1 = \sqrt{X3 + (X2 - 51X1 * (X1 - 2) * (21/X2))} - 2 \frac{1}{X1 + X2},$ $Y2 = X3 - X2 \frac{(X1^2 / X3)^{15/3.5}}{\sqrt{X1 - X3}},$ <p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
16	Завдання № 1
	<p>Створити вектори X1, X2, X3  X1 – набір цифр з 5 по 5<sup>17</sup> з кроком 5;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 2<sup>4</sup>+5<sup>2</sup>;  Знайти:</p> $Y1 = \sqrt{X3 + (51X1 * (X3 - 2) * (21/X2))} - 7 \frac{1}{X1 - X2},$ $Y2 = X3 - 3X2 \frac{(X1^2 / X3)^{0.56}}{\sqrt{X1 - X3}},$
16	Завдання № 2
	<p>Розрахувати</p> $Y1 = \sqrt{X3 + (51X1 * (X3 - 2) * (21/X2))} - 7 \frac{1}{X1 - X2},$ $Y2 = X3 - 3X2 \frac{(X1^2 / X3)^{0.56}}{\sqrt{X1 - X3}},$ <p>де  X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.</p>

№ вар.	Завдання
	<p>X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
17	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 3 по 381 з кроком 3;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 3<sup>4</sup>+5<sup>4</sup>;  Знайти:</p> $Y1 = \sin(\sqrt{X3 + (X2 - 51X1 * (X1 - 2) * (21/X2))}) - 2 \frac{1}{X1 + X2},$ $Y2 = \cos(X3 - X2 \frac{(X1^2/X3)^{1/3.5}}{\sqrt{X1 + X3}}),$
17	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = \sin(\sqrt{X3 + (X2 - 51X1 * (X1 - 2) * (21/X2))}) - 2 \frac{1}{X1 + X2},$ $Y2 = \cos(X3 - X2 \frac{(X1^2/X3)^{1/3.5}}{\sqrt{X1 + X3}}),$ <p>де</p> <p>X1 – синусоїда з амплітудою N, частотою 20+N рад/с; кутом фази 0.  X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.  X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
18	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3  X1 – набір цифр з 3 по 540 з кроком 3;  X2 – будь-який вектор розміром співпадаючий із вектором X1;  X3 = 3<sup>4</sup>;  Знайти:</p> $Y1 = \cos(\sqrt{X3 + (X1 - 47X3 * (X1 - 2) * (21/X2))}) - 2 \frac{1}{X1 + X2},$ $Y2 = \sin(X3 - X2 \frac{(X1^2/X3)^{1/3.501}}{\sqrt{X1 + 540}}),$

№ вар.	Завдання
	<p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = \cos(\sqrt{X3 + (X1 - 47X3 * (X1 - 2) * (21/X2))}) - 2 \frac{1}{X1 + X2},$ $Y2 = \sin(X3 - X2 \frac{(X1^2/X3)^{1/3.501}}{\sqrt{X1 + 540}}),$ <p>де</p> <p>X1 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.</p> <p>X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.</p> <p>X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
19	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3</p> <p>X1 – набір цифр з 3 по 381 з кроком 3;</p> <p>X2 – будь-який вектор розміром співпадаючий із вектором X1;</p> <p>X3 = 5<sup>5</sup>;</p> <p>Знайти:</p> $Y1 = \sin(\sqrt{X3 + (X2 - \pi X1 * (X1 - 2) * (3\pi/X2))}) - \cos(\pi \frac{1}{X1 + X2}),$ $Y2 = \cos(0.109X3 - X2 \frac{(X1^2/X3)^{5\pi/6}}{\sqrt{X1 + X3}}),$ <p>Завдання № 2</p> <p>Розрахувати</p> $Y1 = \sin(\sqrt{X3 + (X2 - \pi X1 * (X1 - 2) * (3\pi/X2))}) - \cos(\pi \frac{1}{X1 + X2}),$ $Y2 = \cos(0.109X3 - X2 \frac{(X1^2/X3)^{5\pi/6}}{\sqrt{X1 + X3}}),$ <p>де</p> <p>X1 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.</p> <p>X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.</p> <p>X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>
20	<p>Завдання № 1</p> <p>Створити вектори X1, X2, X3</p> <p>X1 – набір цифр з 4 по 65536 з кроком 4;</p> <p>X2 – будь-який вектор розміром співпадаючий із вектором X1;</p> <p>X3 = 4<sup>2</sup>;</p>

№ вар.	Завдання
	Знайти: $Y1 = \sin(\sqrt{X3 + (-\pi X1 * (X1 - 2) * (5\pi/X2))}) * \cos(\pi \frac{1}{X1 + X2}),$ $Y2 = \cos(0.109X3 - X2 \frac{(X1^2 / X3)^{5\pi/6}}{(\sqrt{X1 + X3})^3}),$
	Завдання № 2
	Розрахувати $Y1 = \sin(\sqrt{X3 + (-\pi X1 * (X1 - 2) * (5\pi/X2))}) * \cos(\pi \frac{1}{X1 + X2}),$ $Y2 = \cos(0.109X3 - X2 \frac{(X1^2 / X3)^{5\pi/6}}{(\sqrt{X1 + X3})^3}),$ <p>де</p> <p>X1 – сигнал генератора імпульсів з амплітудою 5*N, періодом 0,3+0.1*N с, шириною імпульсів 15+N % від значення періоду.</p> <p>X2 – сигнал генератора пилкоподібних імпульсів з амплітудою 2*N, періодом 0.2+0.01*N с.</p> <p>X3 – вид характеристики із індивідуально завдання 2 Л.р.№1</p>

N – порядковий номер студента за журналом.

## Лабораторна робота № 3

### ОБЧИСЛЕННЯ ДИФЕРЕНЦІАЛІВ ТА ІНТЕГРАЛІВ В ПАКЕТІ ПРОГРАМ MATLAB

**Мета роботи:** вивчення способів обчислення диференціалів та інтегралів із застосуванням m-файлів та s-моделей.

#### Теоретична частина

##### 3.1 Розрахунок в m-файлі.

###### 1) Функція обчислення похідних - diff

Для обчислення в символьному вигляді похідних від виразу  $S$  служить функція **diff**, що записується в форматі **diff(S, x, n)**. Вона повертає символьне значення  $n$ -ї похідної від символьного виразу або масиву символьних виразів  $S$  по змінній  $x$ , тобто  $S_n(x) =$ .

У форматі **diff(S, x)** знаходиться перша похідна ( $n = 1$  за замовчуванням). Нехай потрібно знайти першу і третю похідні функції  $y = x^2 \sin x$ .

Рішення будуть мати вигляд:

```
>> syms x
```

```
>> y=(x^2)*sin(x);
```

```
>> diff(y,x)
```

```
ans =
```

```
2*x*sin(x)+x^2*cos(x)
```

```
>> diff(y,x,3)
```

```
ans =
```

```
6*cos(x)-6*x*sin(x)-x^2*cos(x)
```

Якщо вираз  $S$  залежить від декількох змінних, наприклад,  $S = S(x, y)$ , то її часткова похідна (або  $S'_x(x, y)$ ) по аргументу  $x$  є похідна цієї функції по  $x$  при постійному значенні  $y$ . У Декартовій системі координат на площині градієнт функції  $S(x, y)$  є вектор  $\text{grad } S =$ .

Частковими похідними другого порядку функції  $S = S(x, y)$  називаються часткові похідні від її перших похідних.

У форматі **diff(S)** знаходиться перша похідна по змінній функції  $S$  в порядку, зворотному алфавіту. Наприклад, якщо функція  $S$  містить змінні  $a, b$ , то диференціювання буде виконано по змінній  $b$ . Якщо при цьому у складі аргументів міститься змінна  $x$ , то вона має абсолютний пріоритет, незалежно від її положення в алфавіті змінних.

Наведемо приклади на всі перераховані випадки.

```
>> syms a b x z  
>> diff(7*a+b^3)
```

```
ans =  
    3*b^2
```

```
>> diff(a*z-2*b)
```

```
ans =  
    a
```

```
>> diff(x*a*z+b^2)
```

```
ans =  
    a*z
```

Функція  $S$  може бути масивом. У таких випадках відгуком буде масив, елементами якого будуть похідні від вихідних функцій, що утворюють масив.

```
>> syms a x  
>> y=[a*log(x);x^a];  
>> diff(y,x)
```

```
ans =  
    [ a/x  
      x^a*(a-1)]
```

Для функції двох змінних  $f(x, y) = \arcsin(x*y)$  обчислити  $\text{grad } f(x, y)$  в точці  $(0, 0)$  і знайти змішану похідну.

Рішення:

```
>> syms x y  
>> f=asin(x*y);  
>> x1=diff(f,x)
```

```
x1 =  
    y/(1-x^2*y^2)^(1/2)
```

```
>> x2=diff(f,x,2)  
>> y1=diff(f,y)
```

```
x2 =  
    (x*y^3)/(1 - x^2*y^2)^(3/2)
```

```
y1 =  
    x/(1 - x^2*y^2)^(1/2)
```

## 2) Функція обчислення інтегралів - int

У практичній роботі часто виникає необхідність обчислення невизначених і визначених інтегралів  $I = \int f(x)dx$  і  $I = \int_a^b f(x)dx$ . Де  $f(x)$  – підінтегральна функція незалежної змінної  $x$ ,  $a$  – нижній і  $b$  – верхній межі інтегрування для визначеного інтеграла.

Функція **int(f,x)** – повертає невизначений інтеграл від символічного виразу  $f$  по змінній  $x$ .

Функція **int(f,x,a,b)** – повертає певний інтеграл від символічного виразу  $f$  по змінній  $x$  з межами від  $a$  до  $b$ .

### Обчислимо невизначений інтеграл.

Рішення:

```
>> syms x
>> int(x/(1 - x ^ 2))
```

```
ans =
-log(x^2 - 1)/2
```

Перевіримо результат інтегрування, продиференціюємо результат інтегрування :

```
>> diff (-log(x^2 - 1)/2, x)
```

```
ans =
-x/(x^2 - 1)
```

Отримали підінтегральною функцією, тобто інтеграл обчислений правильно.

### Обчислимо невизначений інтеграл з символічними змінними.

Рішення:

```
>> syms x a b
>> int ( x / ( a + b * x ^ 2 ) , x )
```

```
ans =
log(b*x^2 + a)/(2*b)
```

У цьому випадку підінтегральна функція задана в аналітичному вигляді з символічними змінними.

### Обчислимо визначений інтеграл.

Рішення:

```
>> syms x a b c d
>> int ( x / ( a + b * x ^ 2 ) , x , c , d )
Warning: Explicit integral could not be found.
```



```
ans =
    piecewise([0 < a and 0 < b and c = 0 and d = 1, (log(a + b) - log(a))/(2*b)],
    [not 0 < a or not 0 < b or c <> 0 or d <> 1, int(x/(b*x^2 + a), x = c..d)])
```

У даному випадку підінтегральна функція задана в аналітичному вигляді з символьними змінними, а межі інтегрування – у вигляді символьних змінних. Це найбільш загальний випадок аналітичного обчислення інтеграла. Обчислити визначений інтеграл:  $\arcsin(x) dx$ .

```
рішення:
>> syms x
>> int(asin(x), x, 0, pi/4)
```

```
ans =
    (pi*asin(pi/4))/4 + (16 - pi^2)^(1/2)/4 - 1
```

Для отримання рішення в числовій формі треба виділити останній результат, вставити його в командну строку та натиснути клавішу <Enter>. Буде отримано наступну відповідь:

```
ans =
    3.284717709677651e-001
```

Обчислити інтеграл:  $x \cdot e^{-x} - x dx$ .

Рішення:

```
>> syms x
>> int(x*exp(-x), x, 0, inf)
```

```
ans =
    1
```

Обчислимо потрійний інтеграл:  $(x^2 + y^2)z dx dy dz$ .

Для обчислення такого інтегралу можна тричі використовувати функцію **int**:

```
>> syms a x y z
>> int( int( int((x^2 + y^2)*z, x, 0, a), y, 0, a), z, 0, a)
```

```
ans =
    a^6/3
```

## 3.2 Розрахунок в S-моделі.

### 1)Обчислення похідної

Для обчислення похідної використовується блок Derivative – він виконує чисельне диференціювання вхідного сигналу.

Для обчислення похідної використовується наближена формула Ейлера:

$$\frac{du}{dt} = \frac{\Delta u}{\Delta t}, \quad (3.1)$$

де  $u$  – величина зміни вхідного сигналу за час  $t$ ,  
 $t$  – поточне значення кроку модельного часу.

Значення вхідного сигналу блоку до початку розрахунку вважається рівним нулю. Початкове значення вихідного сигналу також покладається рівним нулю. Точність обчислення похідної істотно залежить від величини встановленого кроку розрахунку. Вибір меншого кроку розрахунку покращує точність обчислення **похідної**.

На рис. 3.1 показано приклад використання диференціюючого блоку для обчислення похідної прямокутного сигналу. У розглянутому прикладі, для підвищення наочності, крок розрахунку обраний досить великим.

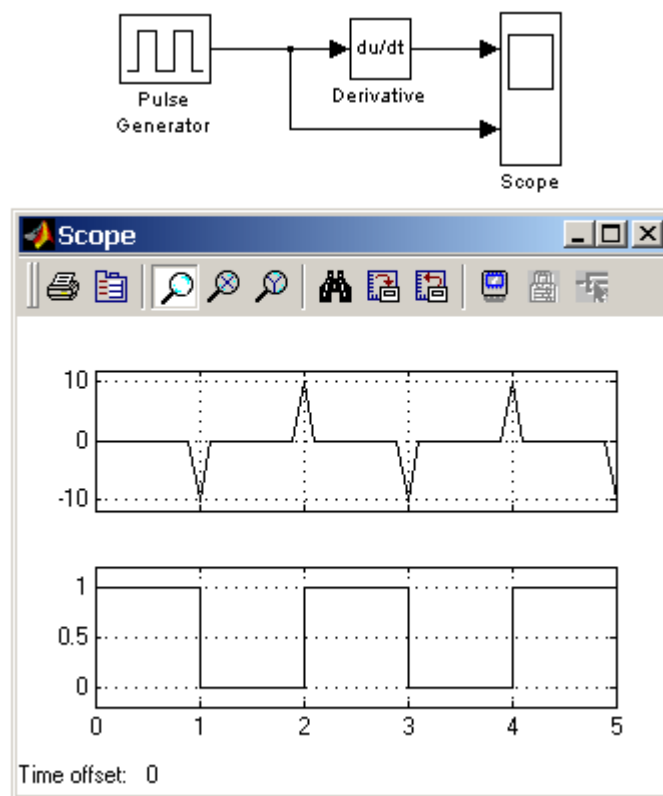


Рис. 3.1. Використання блоку Derivative для диференціювання сигналу.

Даний блок використовується для диференціювання аналогових сигналів. При диференціюванні дискретного сигналу з допомогою блоку Derivative його вихідний сигнал буде представляти собою послідовність імпульсів відповідних моментам часу стрибкоподібного зміни дискретного сигналу.

## 2) Обчислення інтегралу

Для обчислення інтегралу використовується блок **Integrator** – він виконує інтегрування вхідного сигналу.

Параметри:

- **External reset** – зовнішнє скидання. Тип зовнішнього керуючого сигналу, що забезпечує скидання інтегратора до початкового стану. Вибирається зі списку:

**none** – ні (скидання не виконується);

**rising** – наростаючий сигнал (передній фронт сигналу);

**falling** – спадаючий сигнал (задній фронт сигналу);

**either** – наростаючий або спадаючий сигнал;

**level** – не нульовий сигнал (скидання виконується якщо сигнал на керуючому вході стає рівним нулю).

У тому випадку, якщо обрано який не будь (але не none), тип керуючого сигналу, то на зображенні блоку з'являється додатковий керуючий вхід. Поруч з додатковим входом буде показано умовне позначення керуючого сигналу.

- **Initial condition source** – джерело початкового значення вихідного сигналу. Вибирається зі списку:

**internal** – внутрішній;

**external** – зовнішній.

У цьому випадку на зображенні блоку з'являється додатковий вхід, позначений  $x_0$ , на який необхідно подати сигнал задає початкове значення вихідного сигналу інтегратора.

- **Initial condition** – початкова умова. Установка початкового значення вихідного сигналу інтегратора. Параметр доступний, якщо обрано внутрішнє джерело початкового значення вихідного сигналу.

- **Limit output** ( прапорець ) – використання обмеження вихідного сигналу.

- **Upper saturation limit** – верхній рівень обмеження вихідного сигналу. Може бути заданий як числом, так і символічної послідовністю  $inf$ , тобто  $+\infty$ .

- **Lower saturation limit** – нижній рівень обмеження вихідного сигналу. Може бути заданий як числом, так і символічної послідовністю  $inf$ , тобто  $-\infty$ .

- **Show saturation port** – керує відображенням порту, котрий виводить сигнал, що свідчить про вихід інтегратора на обмеження. Вихідний сигнал даного порту може приймати такі значення:

**0**, якщо інтегратор не перебуває на обмеженні.

**+1**, якщо вихідний сигнал інтегратора досяг верхньої межі.

**-1**, якщо вихідний сигнал інтегратора досяг нижньої межі.

- **Show state port** (прапорець) – «відобразити/приховати» порт стану блоку. Даний порт використовується в тому випадку, якщо вихідний сигнал інтегратора потрібно подати в якості сигналу зворотного зв'язку цього ж інтегратора. Наприклад, при установці початкових умов через зовнішній порт або при скиданні інтегратора через порт скидання. Вихідний сигнал з цього порту може використовуватися також для організації взаємодії з керованою підсистемою.

- **Absolute tolerance** – абсолютна похибка.

На рис.3.2 показано приклад роботи інтегратора при подачі на його вхід ступеневої сигналу. Початкова умова прийнята рівним нулю.

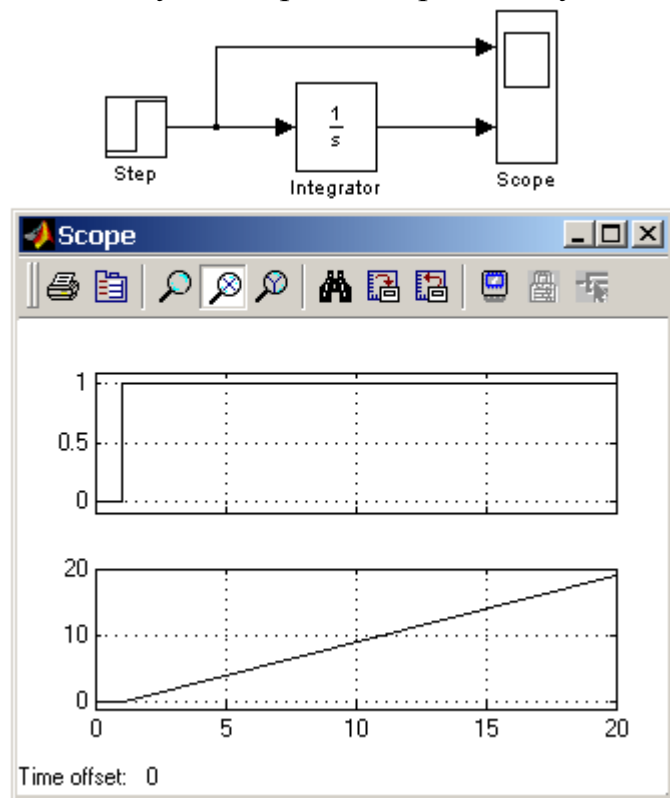


Рис. 3.2. Інтегрування ступеневого сигналу

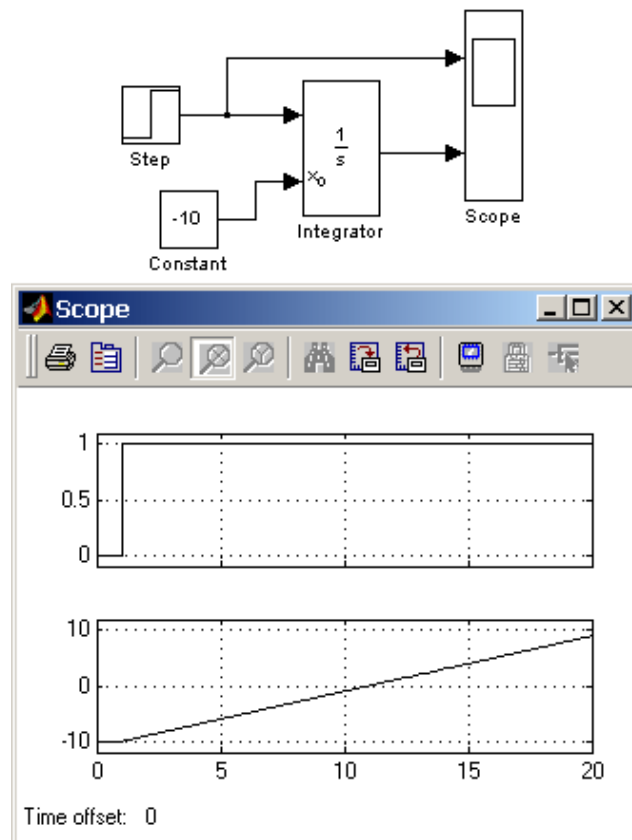


Рис.3.3. Інтегрування ступеневого сигналу з встановленням початкового значення вихідного сигналу

Приклад на рис.3.4 демонструє використання вхідного порту для скидання вихідного сигналу і порту стану інтегратора з метою організації зворотного зв'язку. Схема працює в такий спосіб: вхідний постійний сигнал перетвориться інтегратором в лінійно-змінний, по досягненні вихідним сигналом значення рівного **1** блок **Relational Operator** виробляє логічний сигнал, по передньому фронту якого відбувається скидання вихідного сигналу інтегратора до початкового значення рівного нулю. У результаті на виході інтегратора формується пилкоподібний сигнал, що змінюється від **0** до **+1**.

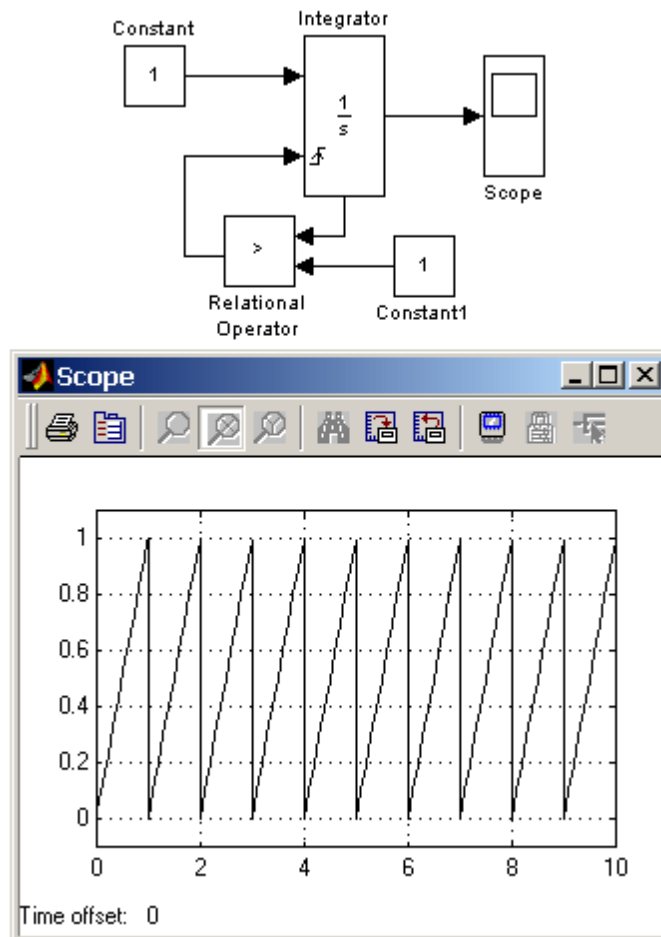


Рис.3.4. Генератор пилкоподібних сигналів на основі інтегратора

Наступна схема (рис.3.5) використовує установку початкового значення інтегратора за допомогою його вихідного сигналу. У перший момент часу початкове значення вихідного сигналу інтегратора за допомогою блоку **IC (Initial Condition)** встановлюється рівним нулю. По досягненні вихідним сигналом значення рівного **1** блок **Relational Operator** подає сигнал скидання вихідного сигналу інтегратора на початковий рівень, при цьому сигналом, що задає початковий рівень, виявляється інвертований вихідний сигнал інтегратора (тобто - **1**). Далі цикл роботи схеми повторюється. На відміну від попередньої схеми вихідним сигналом генератора є двуполлярний сигнал.

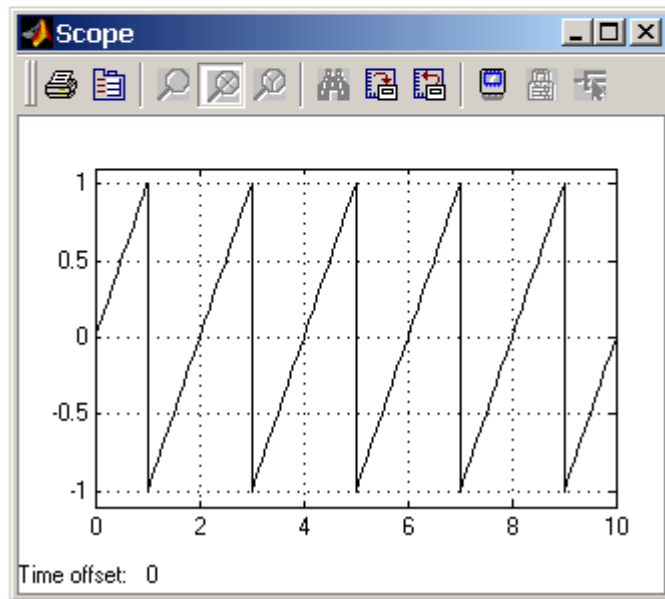
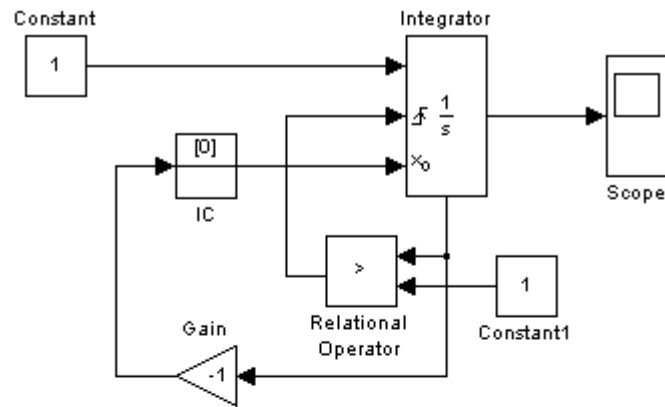


Рис.3.5. Генератор двуполярного пилоподібного сигналу на основі інтегратора

### 3.3 План виконання лабораторної роботи «Обчислення диференціалів та інтегралів в пакеті програм MATLAB»

1. Створюємо новий m-файл.
2. Надаємо ім'я m-файлу і зберігаємо його.
3. Виконуємо приклади, що наведено в п.3.1.
4. Створюємо новий S-документ.
5. Надаємо ім'я S-документу і зберігаємо його.
6. Виконуємо приклади, що наведено в п.3.2.
7. Виконуємо індивідуальні завдання.

### 3.4 Індивідуальні завдання

№ в-ту	Функція для диференціювання	Функція для інтегрування
1	$y(x) = \sin(x^2) - \cos\left(\frac{2\sqrt{x} - 5}{x}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_0^{10} y(x) - ?$
2	$y(x) = e^{x^5} + \sin\left(\frac{2\sqrt{x+4} - 5}{x+2}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_0^1 y(x) - ?$
3	$y(x) = x^5 - \sin^2\left(\frac{\sqrt{x+4} - 5x}{x-4}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_2^5 y(x) - ?$
4	$y(x) = x^5 - \operatorname{tg}^2\left(\frac{\sqrt{x+2} - a \cdot x}{x-4}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_2^{10} y(x) - ?$
5	$y(x) = x^5 \sin\left(\frac{\sqrt[3]{x+4} - 5}{x+2}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_5^{10} y(x) - ?$
6	$y(x) = 2x - \lg_2\left(\frac{\sqrt{x} - 5x^2}{x-4}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_{10}^{20} y(x) - ?$
7	$y(x) = \cos(x^5) \cdot \sin\left(\frac{\sqrt[3]{x+4} - 5}{x+2}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int_0^{20} y(x) - ?$

8	$y(x) = \sin(\sqrt{x}) - \ln\left(\frac{5\sqrt{x^3} + 2 + 3x^3}{0,2x}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 20 5
9	$y(x) = 0,5x - \ln\left(\frac{\sqrt{x} + 2 + 3x^3}{0,2x - 2}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 6 3
10	$y(x) = \frac{x^{0.5}}{\cos(x + 2 \cdot \pi/3)} \cdot \sin\left(\frac{\sqrt[3]{x+4} - 5}{x+2}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 6 4
11	$y(x) = \sqrt{x} - \ln\left(\frac{4\sqrt{x} + 2 + 3x^3}{0,2x}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 8 4
12	$y(x) = e^{x-2} - \cos\left(\frac{\sqrt{x} - 5x + 5}{3x}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 10 4
13	$y(x) = x^2 - \cos^2\left(\frac{\sqrt{x-4} - 5x + 10}{3x-1}\right)$	
	$y', y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 0 -8
14	$y(x) = \sin(a \cdot x^2) - \cos\left(\frac{2b\sqrt{x} - 5}{x+c}\right)$	
	$y', y''(b), y''' - ?$	$\int y(x), \int y(x) - ?$ 1 -10



15	$y(x) = e^{-x^5} + \sin\left(\frac{2\sqrt{x+4a}-5}{x+c}\right)$	
	$y'(a), y'', y''' - ?$	$\int y(x), \int y(x) - ?$ -1
16	$y(x) = x^5 - \sin^2\left(\frac{\sqrt{x+c}-5a}{x-4}\right)$	
	$y'(a), y'', y''' - ?$	$\int y(x), \int y(x) - ?$ 0 -2
17	$y(x) = x^a - \sin^b\left(\frac{\sqrt{x+4}-5x}{x-4}\right)$	
	$y', y''(b), y''' - ?$	$\int y(x), \int y(x) - ?$ 0 -3
18	$y(x) = x^b \sin\left(\frac{\sqrt[4]{x+4}-5}{x+2}\right)$	
	$y', y''(b), y''' - ?$	$\int y(x), \int y(x) - ?$ 2 -2
19	$y(x) = Kx - \lg_2\left(\frac{\sqrt{x}-bx^2}{x-4}\right)$	
	$y', y''(b), y''' - ?$	$\int y(x), \int y(x) - ?$ 3 -3
20	$y(x) = \cos(x^2 - 2\pi/3) \cdot \sin\left(\frac{\sqrt[3]{x+b}-5}{x+c}\right)$	
	$y', y''(c), y''' - ?$	$\int y(x), \int y(x) - ?$ 4 -4
21	$y(x) = \sin(\sqrt{x-a^2}) + 2\left(\frac{5\sqrt{x^3+b^2+3x^3}}{0,2x}\right)$	
	$y', y''(a), y''' - ?$	$\int y(x), \int y(x) - ?$ 6 -5

22	$y(x) = 0,5x - \ln\left(\frac{\sqrt{x} + c + 3x^3}{0,2x - 2b}\right)$	
	$y'(b), y'', y''' - ?$	$\int_0^6 y(x), \int_0^6 y(x) - ?$
23	$y(x) = \frac{x^{0.5+b}}{\cos(x + 2 \cdot \pi/3)} \cdot \sin\left(\frac{\sqrt[3]{x+a} - 5}{x + 2a}\right)$	
	$y', y'', y''' - ?$	$\int_2^6 y(x), \int_2^6 y(x) - ?$
24	$y(x) = \sqrt{xb} - \ln\left(\frac{4\sqrt{x+a} + b + 3x^3}{0,2x}\right)$	
	$y'(b), y'', y''' - ?$	$\int_3^9 y(x), \int_3^9 y(x) - ?$
25	$y(x) = e^{x-2b} - \cos\left(\frac{\sqrt{x} - 5x + 5}{3x + a}\right)$	
	$y'(b), y'', y''' - ?$	$\int_0^9 y(x), \int_0^9 y(x) - ?$

## Лабораторна робота № 4

### РОЗРАХУНОК ГІДРАВЛІЧНИХ ПРОЦЕСІВ

**Мета роботи:** отримання навиків в розрахунку гідравлічних процесів за допомогою пакету програм MatLab.

#### 4.1 Розрахунок витрати за допомогою витратоміру Вентурі

Витратомір Вентурі являє собою плавно звужену й розширену циліндричну вставку, що може бути встановлена в трубі.

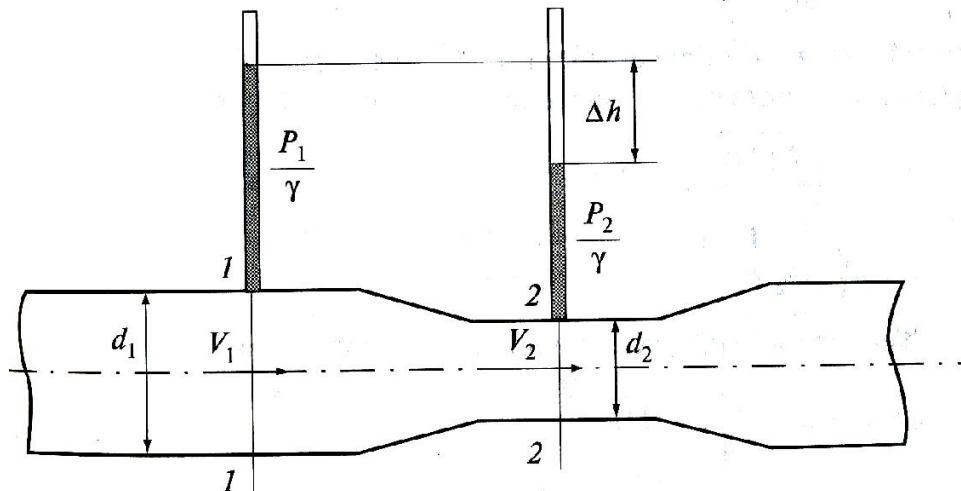


Рис.4.1 Витратомір Вентурі

Напишемо рівняння Бернуллі для перерізів 1-1 та 2-2 (приймаємо відсутність втрат напору,  $h_w = 0$ ):

$$z_1 + \frac{P_1}{\gamma} + \frac{V_1^2}{2g} = z_2 + \frac{P_2}{\gamma} + \frac{V_2^2}{2g} . \quad (4.1)$$

Оскільки  $V_1 < V_2$ , тому показання п'езометра в перерізі будуть більше, ніж у другому:

$$z_1 + \frac{P_1}{\gamma} > z_2 + \frac{P_2}{\gamma} . \quad (4.2)$$

Різниця показників п'езометрів складає

$$\Delta h = \left( z_1 + \frac{P_1}{\gamma} \right) - \left( z_2 + \frac{P_2}{\gamma} \right) . \quad (4.3)$$

Підставивши вираз (4.3) в рівняння (4.2), отримаємо:

$$\Delta h = \frac{V_1^2 - V_2^2}{2g} \quad (4.4)$$

Оскільки площини перерізів 1-1 та 2-2 відомі, тому, при використанні рівняння нерозривності для нестисливої рідини, маємо:

$$\omega_2 V_2 = \omega_1 V_1, \quad (4.5)$$

або

$$V_2 = V_1 \frac{\omega_1}{\omega_2}. \quad (4.6)$$

Підставивши отриманий вираз (4.6) в рівняння (4.4), отримаємо

$$V_1 = \frac{\sqrt{2g\Delta h}}{\sqrt{\left(\frac{\omega_1}{\omega_2}\right)^2 - 1}}. \quad (4.7)$$

Теоретична витрата рідини с трубопроводі складає:

$$Q_m = V_1 \omega_1 = \frac{\pi d_1^2}{4} \cdot \frac{\sqrt{2g\Delta h}}{\sqrt{\left(\frac{d_1}{d_2}\right)^4 - 1}} \quad (4.8)$$

або

$$Q_m = B_1 \cdot \sqrt{\Delta h}, \quad (4.9)$$

де  $B_1$  – постійна витратоміра.

$$B_1 = \frac{\pi d_1^2}{4} \cdot \frac{\sqrt{2g}}{\sqrt{\left(\frac{d_1}{d_2}\right)^4 - 1}} \quad (4.10)$$

Слід зазначити, що випадку руху ідеальної рідини приведений розрахунок повністю відповідає. При русі крізь витратомір в'язкої рідини можуть викликатися втрати напору, тому необхідно ввести в кінцеву формулу відповідну поправку у вигляді коефіцієнта витрати водоміра  $\mu$ ,  $Q = \mu \cdot Q_m$ .

Коефіцієнт витрати водоміра Вентурі, виготовленого у відповідності із стандартом по виміру витрати рідин, складає  $\mu = 0,97...0,99$ .

Кінцева формула витрати з урахуванням  $\mu$  буде:

$$Q = B \cdot \sqrt{\Delta h}, \quad (4.11)$$

де

$$B = \mu \cdot B_1 \quad (4.12)$$

## 4.2 Розрахунок гідравлічного удару

В трубопроводі діаметром  $d$  тече вода з витратою  $Q$ . Товщина стінки труби –  $\delta$ , початковий надлишковий тиск трубопроводу  $P_0$ . Визначити значення підвищення тиску біля затвора при раптовому його зачиненні.

Рішення. Середня швидкість в трубопроводі до зачинення затвору дорівнює:

$$V_0 = \frac{4Q}{\pi d^2} \quad (4.13)$$

Таблиця 4.1

Співвідношення модулів пружності води  $E_B$  й матеріалу стінок труб  $E_M$

Матеріал стінки труби	$\frac{E_B}{E_M}$
Сталь	0,01
Чавун	0,02
Азбоцемент	0,11
Вініпласт	0,68...0,73
Поліетилен	1...1,45

Модуль пружності води дорівнює  $E_B = 2 \cdot 10^9 \text{ Н/м}^2$  та щільність води дорівнює  $\rho_B = 1000 \text{ кг/м}^3$ .

Швидкість поширення ударної хвилі буде дорівнювати:

$$C = \sqrt{\frac{E_B}{\rho_B}} \cdot \frac{1}{\sqrt{1 + \frac{E_B}{E_M} \cdot \frac{d}{\delta}}} \quad (4.15)$$

Підвищення тиску визначаємо по формулі Н. Жуковського:

$$\Delta P = \rho \cdot V_0 \cdot C \quad (4.16)$$

Тиск біля затвору буде дорівнювати:

$$P = P_0 + \Delta P \quad (4.17)$$

### 4.3 План виконання лабораторної роботи «Розрахунок гідравлічних процесів»

1. Запускаємо MatLAB.
2. Ознайомлюємося з лабораторною роботою № 4.
3. Створюємо новий М-файл.
4. Виконуємо розрахунок витрати за допомогою витратоміру Вентурі та розрахунок гідравлічного удару згідно до відповідних варіантів з табл.4.2.
5. Надаємо ім'я М-файлу і зберігаємо його.
6. Захист лабораторної роботи.

#### 4.4 Індивідуальні завдання

1) Побудувати залежність (не менше ніж 50 точок)  $Q(\Delta h)$  при використанні витратоміру Вентурі

Таблиця 4.2

№ варіанту	$\Delta h$ , м		$d_1$ , м	$d_2$ , м	$\mu$
1	0,01	0,15	0,1	0,05	0,97
2	0,01	0,8	0,5	0,3	0,98
3	0,01	1	0,8	0,4	0,99
4	0,01	1	0,9	0,3	0,97
5	0,01	0,8	1,2	0,6	0,98
6	0,01	0,7	0,1	0,02	0,99
7	0,01	0,75	0,5	0,2	0,97
8	0,01	1	0,8	0,6	0,98
9	0,01	1,2	0,9	0,3	0,99
10	0,01	1,5	1,2	0,8	0,97
11	0,01	1	1	0,5	0,98
12	0,01	1,6	2	1	0,99
13	0,01	0,8	0,5	0,25	0,97
14	0,01	0,75	0,6	0,3	0,98
15	0,01	0,9	0,75	0,25	0,99
16	0,01	0,6	0,8	0,4	0,98
17	0,01	0,7	0,6	0,3	0,97
18	0,01	0,5	0,4	0,2	0,99
19	0,01	0,65	0,2	0,1	0,98
20	0,01	0,3	0,1	0,05	0,97
21	0,01	0,35	0,05	0,025	0,99
22	0,01	0,42	0,35	0,12	0,98
23	0,01	0,68	0,55	0,24	0,97
24	0,01	1,1	0,95	0,48	0,99
25	0,01	1,3	1,05	0,6	0,98

2) Визначити значення підвищення тиску біля затвора при раптовому його зачиненні (розрахунок виконується в системі СИ).

Таблиця 4.3

№ варіанту	d, мм	Q, л/с	$\delta$ , мм	P <sub>0</sub> , ат.	Матеріал труби
1	400	200	15	1,5	Сталь
2	100	250	20	2	Чавун
3	300	750	10	1	Азбоцемент
4	200	160	8	3	Вініпласт
5	500	130	5	1,2	Поліетилен
6	350	230	25	1,8	Сталь
7	220	245	18	2,5	Чавун
8	150	270	16	3,2	Азбоцемент
9	180	360	12	1,5	Вініпласт
10	60	345	14	2	Поліетилен
11	120	520	35	1	Сталь
12	380	600	15	3	Чавун
13	550	220	20	1,2	Азбоцемент
14	270	250	10	1,8	Вініпласт
15	250	400	8	2,5	Поліетилен
16	160	300	5	3,2	Сталь
17	130	420	25	1,5	Чавун
18	230	510	18	2	Азбоцемент
19	245	240	16	1	Вініпласт
20	270	280	12	3	Поліетилен
21	360	310	14	1,2	Сталь
22	345	380	35	1,8	Чавун
23	520	330	15	2,5	Азбоцемент
24	600	440	20	3,2	Вініпласт
25	220	290	10	1,5	Поліетилен

## Лабораторна робота № 5

### БУДОВА ДВУМІРНИХ ТА ТРИВИМІРНИХ ГРАФІКІВ В MATLAB. РЕДАГУВАННЯ ГРАФІКІВ.

**Мета роботи:** вивчення способів побудови графіків та відображення характеристик, а також вивчення способів їх редагування.

#### 5.1 Побудова таблиць значень функції однієї змінної в пакеті MatLab

Відображення функції у вигляді таблиці зручно, якщо є порівняно невелике число значень функції. Нехай потрібно вивести в командне вікно таблицю значень функції в точках 0.2, 0.3, 0.5, 0.8, 1.3, 1.7, 2.5.

Завдання вирішується в два етапи:

- 1) створюється вектор-рядок  $x$ , що містить координати заданих точок;
- 2) обчислюються значення функції  $y(x)$  від кожного елемента вектора  $x$  і записуються отримані значення в вектор-рядок  $y$ .

Значення функції необхідно знайти для кожного з елементів вектор-рядки  $x$ , тому операції у виразі для функції повинні виконуватися поелементно.

```
» x = [0.2 0.3 0.5 0.8 1.3 1.7 2.5]
x =
0.2000 0.3000 0.5000 0.8000 1.3000 1.7000 2.5000
» y = sin(x).^2./(1+cos(x))+exp(-x).*log(x)
y =
-1.2978 -0.8473 -0.2980 0.2030 0.8040 1.2258 1.8764
```

Якщо необхідно вивести значення функції в точках відрізка, віддалених один від одного на рівну відстань (крок). Припустимо, що необхідно вивести таблицю значень функції  $y(x)$  на відрізки  $[1, 2]$  з кроком 0.2. Можна, звичайно, ввести вектор-рядок значень аргументу  $x = [1, 1.2, 1.4, 1.6, 1.8, 2.0]$  з командного рядка і обчислити всі значення функції так, як описано вище. Однак, якщо крок буде не 0.2, а, наприклад 0.01, то робота по введенню вектора  $x$  зросте.

У MatLab передбачено просте створення векторів, кожний елемент яких відрізняється від попереднього на постійну величину, тобто на крок. Для введення таких векторів служить двокрапка. Наступні два оператори приводять до формування однакових вектор-рядків. Умовно можна записати

```
» x = [1, 1.2, 1.4, 1.6, 1.8, 2.0]
x =
1.0000 1.2000 1.4000 1.6000 1.8000 2.0000
» x = [1:0.2:2]
x =
1.0000 1.2000 1.4000 1.6000 1.8000 2.0000
```



Крок може бути і від'ємним:

```
» x = [1.9:-0.2:1]
x =
1.9000 1.7000 1.5000 1.3000 1.1000
```

Крок, рівний одиниці, допускається не вказувати при автоматичному заповненні:

```
» x = [1:5]
x =
1 2 3 4 5
```

Виведемо таблицю значень функції

$$y(x) = e^{-x} \sin(10x)$$

на відрізку  $[0, 1]$  з кроком 0.05.

Для виконання цього завдання необхідно зробити наступні дії:

- 1) сформувати вектор-рядок  $x$  за допомогою двокрапки;
- 2) обчислити значення  $y(x)$  від елементів  $x$ ;
- 3) записати результат у вектор-рядок  $y$ ;
- 4) вивести  $x$  та  $y$ .

```
» x = [0:0.05:1];
» y = exp(-x).*sin(10*x);
» x
x =
Columns 1 through 7
0 0.0500 0.1000 0.1500 0.2000 0.2500 0.3000
Columns 8 through 14
0.3500 0.4000 0.4500 0.5000 0.5500 0.6000 0.6500
Columns 15 through 21
0.7000 0.7500 0.8000 0.8500 0.9000 0.9500 1.0000
» y
y =
Columns 1 through 7
0 0.4560 0.7614 0.8586 0.7445 0.4661 0.1045
Columns 8 through 14
-0.2472 -0.5073 -0.6233 -0.5816 -0.4071 -0.1533 0.1123
Columns 15 through 21
0.3262 0.4431 0.4445 0.3413 0.1676 -0.0291 -0.2001
```

Вектор-рядки  $x$  і  $y$  складаються з двадцяти одного елемента, і не поміщаються на екрані в один рядок, тому виводяться по частинах. Так як  $x$  і  $y$  зберігаються в двовимірних масивах розмірністю один на двадцять один, то виводяться по стовпцях, кожен з яких складається з одного елемента. Спочатку виводяться стовпчики з першого по сьомий (columns 1 through 7), потім – з

восьмого по чотирнадцятий (columns 8 through 14), і, нарешті, – з п'ятнадцятого по двадцять перший (columns 15 through 21). Більш наочним і зручним є графічне представлення функції.

## 5.2 Побудова графіків функції однієї змінної

### 5.2.1 Графіки функцій в лінійному масштабі

MatLab володіє добре розвиненими графічними можливостями для візуалізації даних. Розглянемо побудову найпростішого графіка функції однієї змінної на прикладі функції

$$y(x) = e^{-x} \sin(10x) ,$$

визначеної на відрізку  $[0, 1]$ . Висновок функції у вигляді графіка складається з наступних етапів:

- 1) задання вектора значень аргументу  $x$ ;
- 2) обчислення вектора  $y$  значень функції  $y(x)$ ;
- 3) виклик команди **plot** для побудови графіка.

Команди для завдання вектора  $x$  і обчислення функції краще завершувати крапкою з комою для придушення виведення в командне вікно їх значень.

```
» x = 0:0.05:1;  
» y = exp(-x).*sin(10*x);  
» plot(x, y)
```

Після виконання команд на екрані з'являється вікно **Figure No. 1** з графіком функції. Вікно містить меню, панель інструментів і область графіка.

Для побудови графіка функції в робочому середовищі MatLab повинні бути визначені два вектори однакової розмірності, наприклад  $x$  і  $y$ . Відповідний масив  $x$  містить значення аргументів, а  $y$  – значення функції від цих аргументів. Команда **plot** з'єднує точки з координатами  $(x(i), y(i))$  прямими лініями, автоматично масштабується осі для оптимального розташування графіка у вікні. При побудові графіків зручно розташувати на екрані основне вікно MatLab і вікно з графіком поруч так, щоб вони не перекривалися.

Побудований графік функції має злами. Для більш точного побудови графіка функцію необхідно обчислити  $y(x)$  в більшому числі точок на відрізку  $[0, 1]$ , тобто задати менший крок при введенні вектора  $x$ :

```
» x = 0:0.01:1;  
» y = exp(-x).*sin(10*x);  
» plot(x, y)
```

У результаті виходить графік функції у вигляді більш плавною кривою.

Порівняння декількох функцій зручно робити, відобразивши їх графіки на одних осях. Наприклад, побудуємо на відрізку  $[-1, -0.3]$  графіки функцій

$$f(x) = \sin\left(\frac{1}{x^2}\right), \quad f(x) = \sin\left(\frac{1,2}{x^2}\right)$$

за допомогою наступної послідовності команд:

```
» x = -1:0.005:-0.3;
» f = sin(x.^-2);
» g = sin(1.2*x.^-2);
» plot(x, f, x, g)
```

Використання **plot** з одним аргументом-вектором – приводить до побудови "графіка вектора", тобто залежно від значень елементів вектора від їх номерів. Аргументом **plot** може бути і матриця, в цьому випадку на одні координатні осі виводяться графіки стовпців.

Іноді потрібно порівняти поведінку двох функцій, значення яких сильно відрізняються один від одного. Графік функції з невеликими значеннями практично зливається з віссю абсцис, і встановити його вид не вдається. У цій ситуації допомагає функція **plotyy**, яка виводить графіки у вікно з двома вертикальними осями, які мають відповідний масштаб.

Порівняємо, наприклад, дві функції:

$$f(x) = x^{-3} \quad \text{та} \quad F(x) = 1000 \cdot (x + 0,5)^{-4} .$$

```
» x = 0.5:0.01:3;
» f = x.^-3;
» F = 1000*(x+0.5).^-4;
» plot(x, f, x, F)
```

При виконанні цього прикладу зверніть увагу, що колір графіка збігається з кольором відповідної йому осі ординат.

## 5.2.2 Графіки функцій в логарифмічному масштабі

Для побудови графіків в логарифмічному і напівлогарифмічному масштабах служать наступні функції:

- Loglog (логарифмічний масштаб по обох осях);
- Semilogx (логарифмічний масштаб тільки по осі абсцис);
- semilogy (логарифмічний масштаб тільки по осі ординат).

Аргументи loglog, semilogx і semilogy задаються у вигляді пари векторів значень абсцис і ординат так само, як для функції plot, описаної в попередньому пункті. Побудуємо, наприклад, графіки функцій

$$f(x) = \ln(0,5x) \quad \text{і} \quad g(x) = \sin(\ln(0,5x))$$

на відрізьку  $[0.1, 5]$  в логарифмічному масштабі по осі  $x$ :

»  $x = 0.1:0.01:10$ ;  
 »  $f = \log(0.5*x)$ ;  
 »  $g = \sin(\log(x))$ ;  
 » `semilogx(x, f, x, g)`

### 5.3 Задання властивостей ліній на графіках функцій

Побудовані графіки функцій повинні бути максимально зручними для сприйняття. Часто потрібно нанести маркери, змінити колір ліній, а при підготовці до монохромного друку – задати тип лінії (суцільна, пунктирна, штрих-пунктирна і т.д.). MatLab надає можливість керувати виглядом графіків, побудованих за допомогою `plot`, `loglog`, `semilogx` і `semilogy`, для чого служить додатковий аргумент, що поміщається за кожною парою векторів. Цей аргумент записується в апострофи і складається з трьох символів, які визначають: колір, тип маркера і тип лінії. Використовується одна, дві або три позиції, залежно від необхідних змін. У таблиці наведені можливі значення даного аргументу із зазначенням результату.

Таблиця 5.1

Колір		Тип маркера		Тип лінії	
y	жовтий	.	точка	-	суцільна
m	фіолетовий	o	коло	:	пунктирна
c	блакитний	x	хрестик	-.	штрих-пунктирна
r	красний	+	плюс	--	штрихова
g	зелений	*	зірочка		
b	синій	s	квадрат		
w	білий	d	ромб		
k	чорний	v	трикутник вершиною до низу		
		^	трикутник вершиною до гори		
		<	трикутник вершиною вліво		
		>	трикутник вершиною вправо		
		p	п'яти кінцева зірка		
		h	шести кінцева зірка		

Якщо, наприклад, необхідно побудувати перший графік червоними точковими маркерами без лінії, а другого графік - чорною пунктирною лінією, то слід використовувати команду `plot(x, f, 'r.', X, g, 'k:')`.

## 5.4 Оформлення графіків функцій

Зручність використання графіків багато в чому залежить від додаткових елементів оформлення: координатної сітки, підписів до осей, заголовка і легенди. Сітка наноситься командою **grid on**, підписи до осей розміщуються за допомогою **xlabel**, **ylabel**, заголовок дається командою **title**. Наявність декількох графіків на одних осях вимагає приміщення легенди командою **legend** з інформацією про лінії. Всі перераховані команди застосовні до графіків як у лінійному, так і в логарифмічному і напівлогарифмічному масштабах. Наступні команди виводять графіки зміни добової температури, які забезпечені всією необхідною інформацією.

```
» time = [0 4 7 9 10 11 12 13 13.5 14 14.5 15 16 17 18 20 22];
» temp1 = [14 15 14 16 18 17 20 22 24 28 25 20 16 13 13 14 13];
» temp2 = [12 13 13 14 16 18 20 20 23 25 25 20 16 12 12 11 10];
» plot(time, temp1, 'ro-', time, temp2, 'go-')
» grid on
» title('Добова температура')
» xlabel('Час')
» ylabel('Температура (C)')
» legend('10 квітня, 11 квітня')
```

## 5.5 Побудова графіків функцій двох змінних

Побудова графіка функції двох змінних в MatLab на прямокутній області визначення змінних включає два попередніх етапи:

- 1) розбиття області визначення прямокутною сіткою;
- 2) обчислення значень функції в точках перетину ліній сітки та запис їх в матрицю.

Побудуємо графік функції  $z(x, y) = x^2 + y^2$  на області визначення у вигляді квадрата  $x \in [0, 1]$ ,  $y \in [0, 1]$ . Необхідно розбити квадрат рівномірною сіткою (наприклад, з кроком 0.2) і обчислити значення функцій у вузлах, позначених крапками.

Зручно використовувати два двовимірних масиви  $x$  і  $y$ , розмірністю шість на шість для зберігання інформації про координати вузлів. Масив  $x$  складається з однакових рядків, в яких записані координати  $x_1, x_2, \dots, x_6$ , а масив  $y$  складається з однакових стовпців  $y_1, y_2, \dots, y_6$ . Значення функції у вузлах сітки запишемо в масив  $z$  такої ж розмірності (6 x 6), причому для обчислення матриці  $Z$  використовуємо вираз для функції, але з по елементними матричними операціями. Тоді, наприклад  $z(3,4)$  як раз буде дорівнює значенню функції  $z(x, y)$  в точці  $(x_3, y_4)$ . Для генерації масивів сітки  $x$  і  $y$  за координатами вузлів в MatLab передбачена функція `meshgrid`, для побудови графіка у вигляді каркасної поверхні – функція `mesh`. Наступні оператори приводять до появи на екрані вікна з графіком функції (крапка з комою в кінці операторів не ставиться для того, щоб проконтролювати генерацію масивів):

```

» [X, Y] = meshgrid(0:0.2:1,0:0.2:1)
X =
0 0.2000 0.4000 0.6000 0.8000 1.0000
0 0.2000 0.4000 0.6000 0.8000 1.0000
0 0.2000 0.4000 0.6000 0.8000 1.0000
0 0.2000 0.4000 0.6000 0.8000 1.0000
0 0.2000 0.4000 0.6000 0.8000 1.0000
0 0.2000 0.4000 0.6000 0.8000 1.0000
Y =
0 0 0 0 0 0
0.2000 0.2000 0.2000 0.2000 0.2000 0.2000
0.4000 0.4000 0.4000 0.4000 0.4000 0.4000
0.6000 0.6000 0.6000 0.6000 0.6000 0.6000
0.8000 0.8000 0.8000 0.8000 0.8000 0.8000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
» Z = X.^2+Y.^2
Z =
0 0.0400 0.1600 0.3600 0.6400 1.0000
0.0400 0.0800 0.2000 0.4000 0.6800 1.0400
0.1600 0.2000 0.3200 0.5200 0.8000 1.1600
0.3600 0.4000 0.5200 0.7200 1.0000 1.3600
0.6400 0.6800 0.8000 1.0000 1.2800 1.6400
1.0000 1.0400 1.1600 1.3600 1.6400 2.0000
» mesh(X,Y,Z)

```

MatLab дозволяє наносити на графік додаткову інформацію, зокрема, відповідність кольорів значенням функції. Сітка генерується за допомогою команди **meshgrid**, спричиненої з двома аргументами. Аргументами є вектори, елементи яких відповідають сітці на прямокутній області побудови функції. Можна використовувати один аргумент, якщо область побудови функції – квадрат. Для обчислення функції слід використовувати поелементні операції.

Розглянемо основні можливості, надані MatLab для візуалізації функцій двох змінних, на прикладі побудови графіка функції

$$z(x, y) = 4 \sin(2\pi x) \cdot \cos(1,5\pi y) \cdot (1 - x^2) \cdot y \cdot (1 - y),$$

де  $x \in [-1, 1]$ ,  $y \in [0, 1]$ .

Підготуємо матриці з координатами вузлів сітки і значеннями функції:

```

» [X, Y] = meshgrid(-1:0.05:1, 0:0.05:1);
» Z = 4*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y);

```

Для побудови каркасної поверхні використовується функція **mesh**, що викликається з трьома аргументами:

```

» mesh(X,Y,Z)

```

Колір ліній поверхні відповідає значенням функції. MatLab малює тільки видиму частину поверхні.

За допомогою команди **hidden off** можна зробити каркасну поверхню "прозорою", додавши приховану частину. Команда **hidden on** прибирає невидиму частину поверхні, повертаючи графіком колишній вигляд.

Функція **surf** будує каркасну поверхню графіка функції і заливає кожну клітину поверхні певним кольором, залежних від значень функції в точках, відповідних кутах клітини. У межах кожної клітини колір постійний. Подивіться результати виконання команди

```
» surf(X,Y,Z)
```

Команда **shading flat** дозволяє прибирати каркасні лінії. Для отримання поверхні, плавно залитої кольором, залежних від значень функції, призначена команда **shading interp**.

За допомогою **shading faceted** можна повернутися до поверхні з каркасними лініями.

Тривимірні графіки, одержані за допомогою описаних вище команд, зручні для отримання уявлення про форму поверхні, однак по них важко судити про значення функції. У MatLab визначена команда **colorbar**, яка виводить поруч з графіком стовпчик, встановлює відповідність між кольором і значенням функції.

Команду **colorbar** можна застосовувати в поєднанні з усіма функціями, що будують тривимірні об'єкти.

Користуючись кольоровою поверхнею, важко зробити висновок про значення функції в тій чи іншій точці площини  $xy$ . Команди **meshc** або **surfz** дозволяють отримати більш точне уявлення про поведінку функції. Ці команди будують каркасну поверхню або залиту кольором каркасну поверхню і розміщують на площині  $xy$  лінії рівня функції (лінії сталості значень функції):

```
» surfz(X,Y,Z)
» colorbar
```

MatLab дозволяє побудувати поверхню, що складається з ліній рівня, за допомогою функції **contour3**. Цю функцію можна використовувати так само, як і описані вище **mesh**, **surf**, **meshc** і **surfz** з трьома аргументами. При цьому число ліній рівня вибирається автоматично. Є можливість задати четвертим аргументом в **contour3** або число ліній рівня, або вектор, елементи якого дорівнюють значенням функції, що відображаються у вигляді ліній рівня. Завдання вектора (четвертого аргументу **levels**) зручно, коли потрібно досліджувати поведінку функції в деякій області її значень (зріз функції). Побудуємо поверхню, що складається з ліній рівня, що відповідають значенням функції від 0 до 0.5 з кроком 0.01:

```
» levels = [0:0.01:0.5];
» contour3(X, Y, Z, levels)
» colorbar
```

## 5.6 Побудова контурних графіків функцій двох змінних

MatLab надає можливість отримувати різні типи контурних графіків за допомогою функцій **contour** і **contourf**. Розглянемо їх можливості на прикладі функції

$$z(x, y) = 4 \sin(2\pi x) \cdot \cos(1,5\pi y) \cdot (1 - x^2) \cdot y \cdot (1 - y).$$

Використання **contour** з трьома аргументами

```
» contour(X,Y,Z)
```

призводить до графіку, на якому показані лінії рівня на площині  $xy$ , але без вказівки числових значень на них. Такий графік є малоінформативним, він не дозволяє дізнатися значення функції на кожній з ліній рівня. Використання команди **colorbar** також не дозволить точно визначити значення функції. Кожну лінію рівня можна постачити значенням, яке приймає на ній досліджувана функція, за допомогою визначеної в MatLab функції **clabel**. Функція **clabel** викликається з двома аргументами: матрицею, що містить інформацію про лінії рівня та покажчиком на графік, на якому слід нанести розмітку. Користувачеві не потрібно самому створювати аргументи **clabel**. Функція **contour**, викликана з двома вихідними параметрами, не тільки буде лінії рівня, але і знаходить необхідні для **clabel** параметри. Використовуйте **contour** з вихідними аргументами **CMatr** і **h** (в масиві **CMatr** міститься інформація про лінії рівня, а в масиві **h** – покажчики). Завершіть виклик **contour** крапкою з комою для придушення виведення на екран значень вихідних параметрів і нанесіть на графік сітку:

```
» [CMatr, h] = contour(X, Y, Z);  
» clabel(CMatr, h)  
» grid on
```

Додатковим аргументом функції **contour** (так само, як і **contour3**, описаної вище) може бути або число ліній рівня, або вектор, що містить значення функції, для яких потрібно побудувати лінії рівня.

Наочну інформацію про зміну функції дає заливка прямокутника на площині  $xy$  кольором, залежних від значення функції в точках площини. Для побудови таких графіків призначена функція **contourf**, використання якої не відрізняється від застосування **contour**. Побудуємо графік, який складається з двадцяти ліній рівня, а проміжки між ними заповнені квітами, відповідними значеннями досліджуваної функції:

```
» contourf(X, Y, Z, 20)  
» colorbar
```



## 5.7 Побудова декількох графіків на одній осі

Для відображення декількох графіків функцій однієї змінної на одних осях використовувалися можливості функцій **plot**, **plotyy**, **semilogx**, **semilogy**, **loglog**. Вони дозволяють виводити графіки кількох функцій, задаючи відповідні векторні аргументи парами, наприклад **plot(x,f,x,g)**. Однак для об'єднання тривимірних графіків їх використовувати не можна. Для об'єднання таких графіків призначена команда **hold on**, яку потрібно задати перед побудовою графіка. У наступному прикладі об'єднання двох графіків (площини і конуса) призводить до їх перетину. Конус задається параметрично наступними залежностями:

$$x(u, v) = 0,3 \cdot u \cdot \cos v, \quad y(u, v) = 0,3 \cdot u \cdot \sin v, \quad z(u, v) = 0,6 \cdot u, \quad u, v \in [-2\pi, 2\pi].$$

Для графічного відображення конуса спочатку необхідно згенерувати за допомогою двокрапки вектор-стовпець і вектор-рядок, що містять значення параметрів на заданому інтервалі (важливо, що **u** - вектор-стовпець, а **v**-вектор-рядок):

```
» u = [-2*pi:0.1*pi:2*pi]';  
» v = [-2*pi:0.1*pi:2*pi];
```

Далі формуються матриці **X**, **Y**, містять значення функцій  $x(u, v)$  і  $y(u, v)$  в точках, що відповідають значенням параметрів. Формування матриць виконується за допомогою зовнішнього добутку векторів.

Сформуємо матриці **X**, **Y**, необхідні для графічного відображення конуса:

```
» X = 0.3*u*cos(v);  
» Y = 0.3*u*sin(v);
```

Матриця **Z** повинна бути того ж розміру, що й матриці **X** і **Y**. Крім того, вона повинна містити значення, що відповідають значенням параметрів. Якби у функцію  $z(u, v)$  входило **u** і **v**, то матрицю **Z** можна було заповнити аналогічно матрицям **X** і **Y** за допомогою зовнішнього твору. З іншого боку, функцію  $z(u, v)$  можна представити у вигляді  $z(u, v) = 0,6 \cdot u \cdot g(v)$ , де  $g(v) = 1$ . Тому для обчислення **Z** можна застосувати зовнішній добуток векторів **u** і  $g(v)$ , де вектор-рядок має ту ж розмірність, що **v**, але складається з одиниць:

```
» Z = 0.6*u*ones(size(v));
```

Всі необхідні матриці для відображення конуса створені. Завдання площини виконується наступним чином:

```
» [X,Y] = meshgrid(-2:0.1:2);  
» Z = 0.5*X+0.4*Y;
```

Тепер запишемо повну послідовність команд для побудови пересічних конуса і площини:

```

» u = [-2*pi:0.1*pi:2*pi]';
» v = [-2*pi:0.1*pi:2*pi];
» X = 0.3*u*cos(v);
» Y = 0.3*u*sin(v);
» Z = 0.6*u*ones(size(v));
» surf(X, Y, Z)
» [X,Y] = meshgrid(-2:0.1:2);
» Z = 0.5*X+0.4*Y;
» hold on
» mesh(X, Y, Z)
» hidden off

```

Результат послідовності:

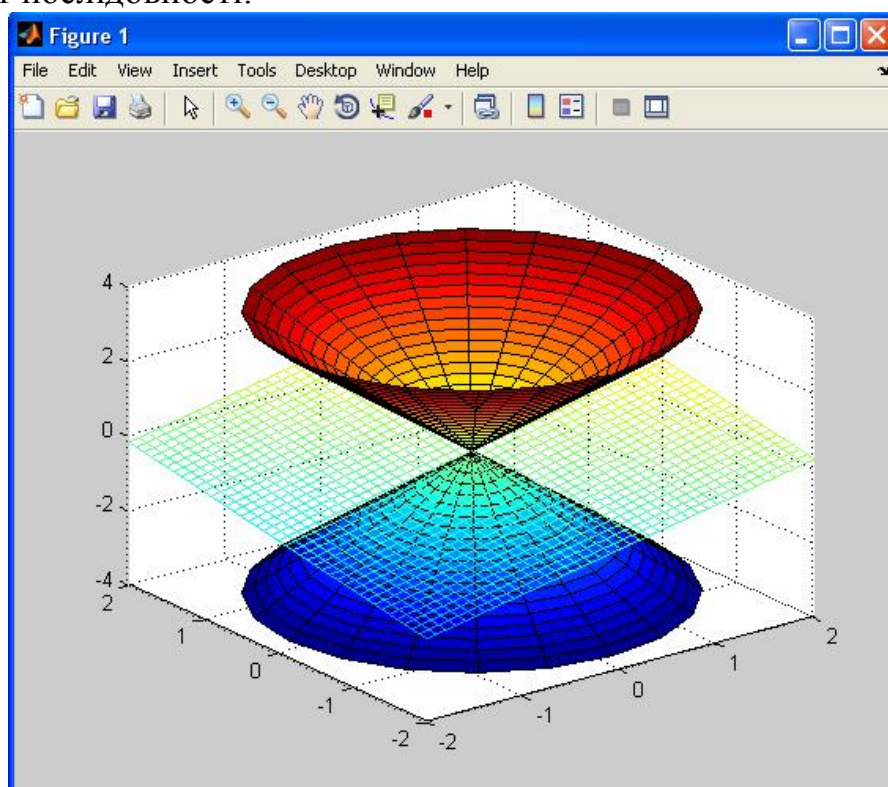


Рис.5.1 Побудова двох конусів

Команда **hidden off** застосована для того, щоб показати частину конуса, що знаходиться під площиною.

Зверніть увагу, що команда **hold on** поширюється на всі наступні висновки графіків в поточне вікно. Для розміщення графіків у нових вікнах слід виконати команду **hold off**. Команда **hold on** може застосовуватися і для розташування декількох графіків функцій однієї змінної, наприклад,

```
» plot(x,f,x,g)
```

еквівалентно послідовності

```

» plot(x,f)
» hold on
» plot(x,g)

```

## 5.8 Побудова декількох вікон графіків в одному вікні фігури

Скористаємося прикладом наведеним в Л.р. №2:

$$Y1 = 16 - 58X1\sqrt{X1-X2} - 4X3,$$
$$Y2 = X1 + X3 - \frac{X1 * X2^{3/2}}{X3} + \sqrt[5]{X3},$$

структура S-моделі має вигляд приведенний на рис.2.8.

Виведемо графіки  $Y1$  та  $Y2$  в два різні вікна, в одному вікні фігури один під одним. Для цього треба у вільному місці S-моделі вставити зв'язку двох блоків показаних на рис.3.8.1.

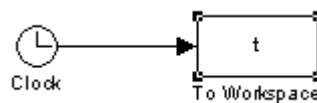


Рис.5.2 Задання вектора часу  $t$

Де в блок **To Workspace** відповідає за пересилку значень в вектор  $t$ , що відобразиться  $\square t$  в робочій області MatLAB.

При цьому в блоці **To Workspace** потрібно прописати нове значення змінної (в даному випадку це буде  $t$ ) та змінити **Save format** на значення **Array**. Вікно параметрів блоку **To Workspace** відображене на рис.3.8.2.

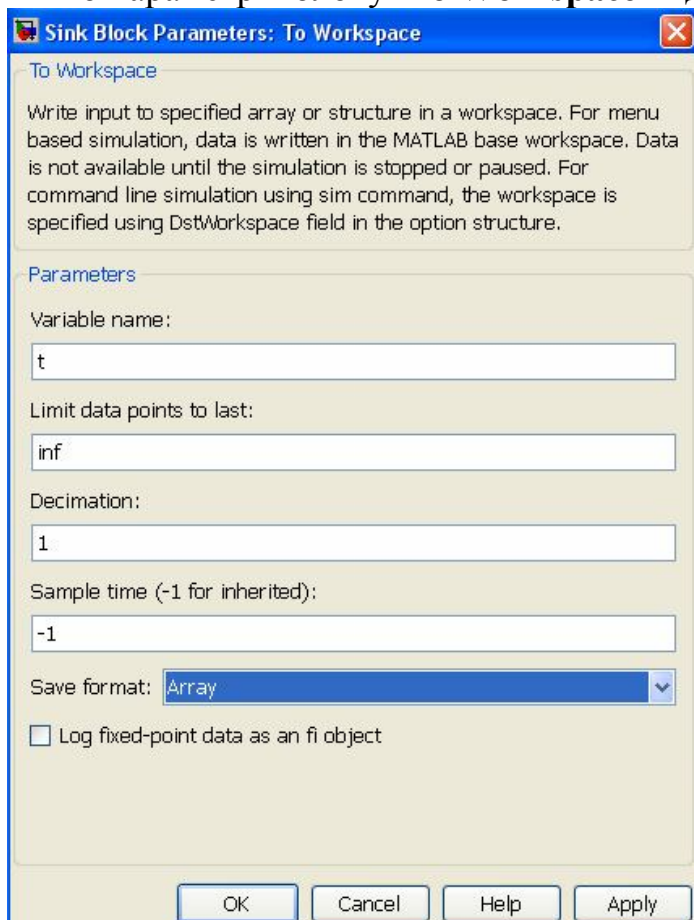


Рис.5.3 Вікно параметрів блоку **To Workspace**

Далі потрібно встановити ще два блоки **To Workspace** для змінних **Y1** та **Y2**, отримаємо схему приведену на Рис.3.8.3.

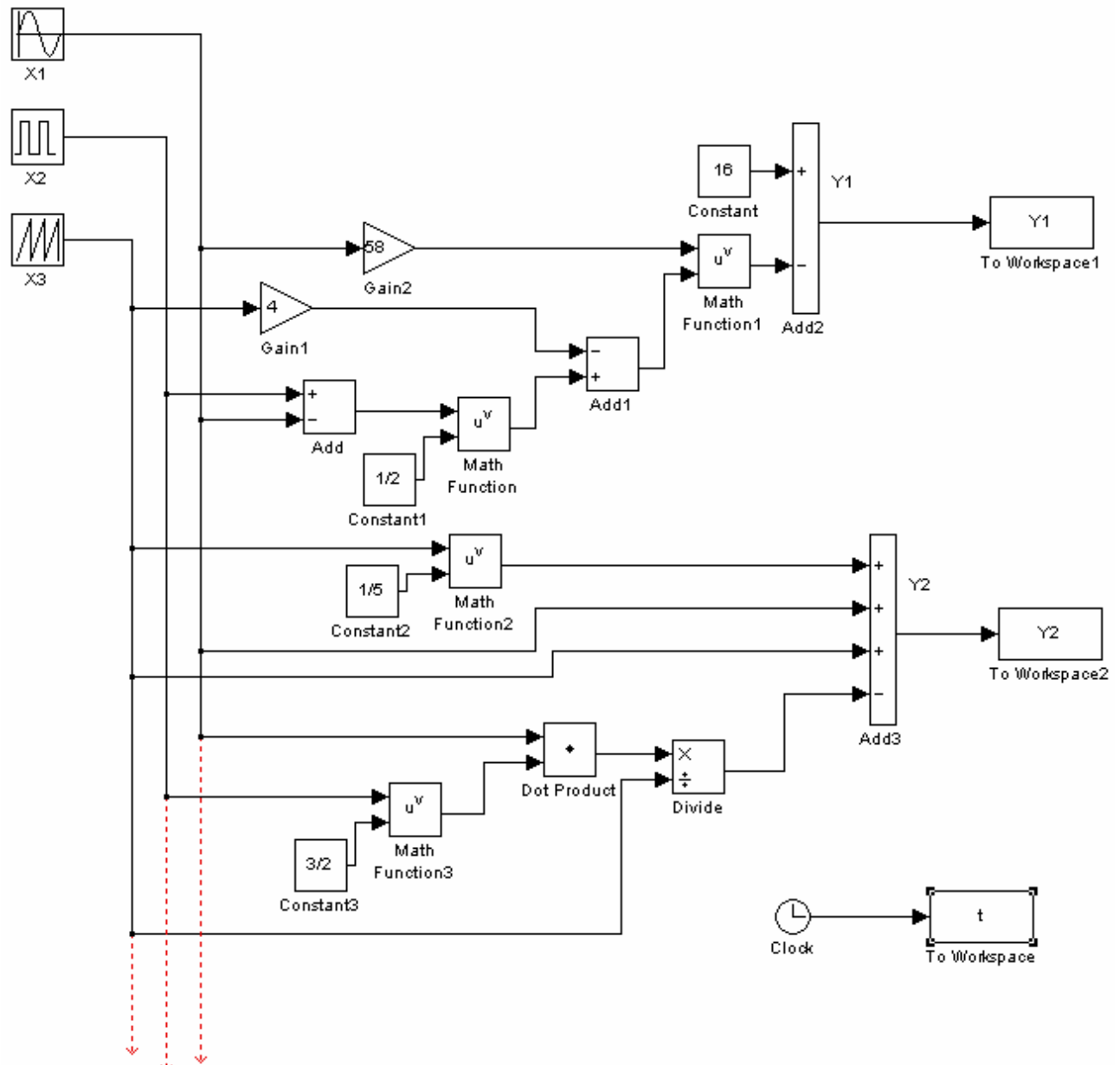


Рис.5.4 Схема S-моделі приклада з використанням блоків **To Workspace**

Запуск моделі призводить до того, що в робочій області MatLAB створюються три вектора  $t$ ,  $Y1$ ,  $Y2$ . Тепер створимо  $m$ -файл з наступною програмою:

```
subplot(2,1,1)
plot(t,Y1)
grid
subplot(2,1,2)
plot(t,Y2)
grid
```

Збережемо  $m$ -файл та запусимо його в командному вікні MatLAB. Отримаємо вікно фігури, що зображене на рис.3.8.4.

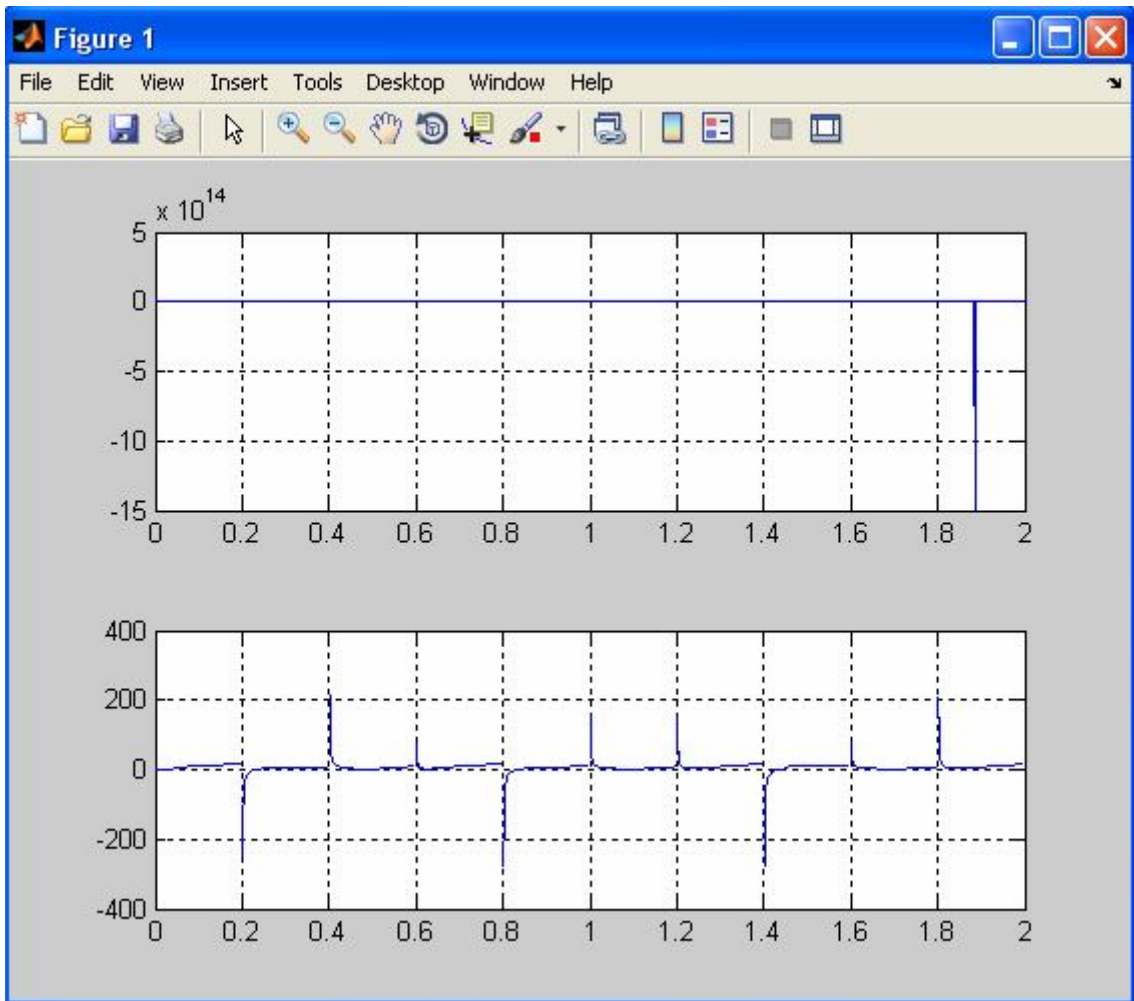






Рис.5.5 Зображення вікна фігури, що відповідає розглянутому прикладу

В цьому вікні можна продовжити редагування зображення. Ярлик **Rotate 3D**  змінює кут зору на вибраній графік. Ярлики   дозволяють приблизити або віддалити вибраний фрагмент графіку відповідно.

Натиснемо на ярлик курсору в панелі інструментів , потім клацнемо правою клавішею миші на сірому просторі у вікні фігури і з запропонованого списку виберемо строчку **Show Property Editor** та натиснемо ліву клавішу миші. Після цих операцій під вікном з'явиться додаткове вікно для редагування вікна і зображення. Тепер натиснемо лівою клавішею (ЛК) миші на верхньому вікні і поглянемо як зміниться додаткове вікно (рис.3.8.5).

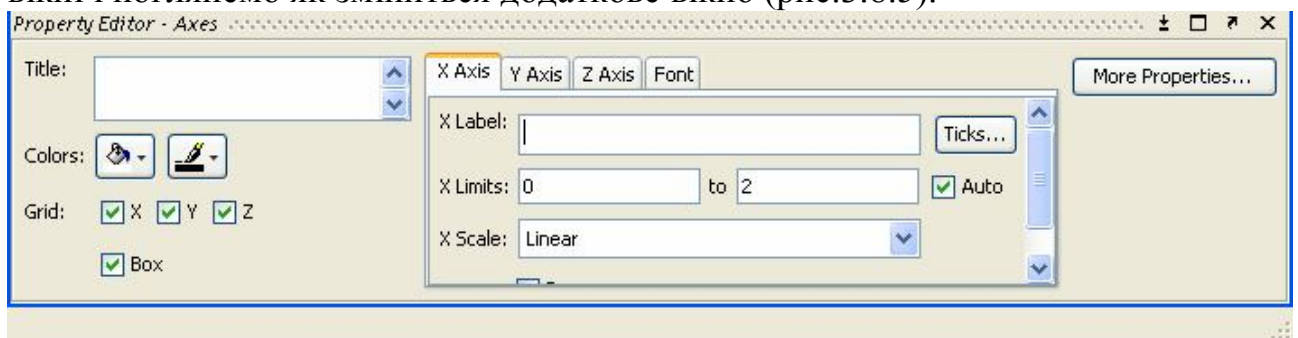


Рис.5.6 Виділення вікна в Show Property Editor

В цьому вікні можна змінити або задати напис Title і Label, можна змінити масштаб вибраних осей, можна змінити значення точок (це можна зробити відкривши вкладене вікно натиснувши кнопку Ticks). Тепер клацнемо ЛК на лінії, що зображена на першому графіку, – додаткове меню одразу зміниться і буде мати вигляд показаний на рис.3.8.6.

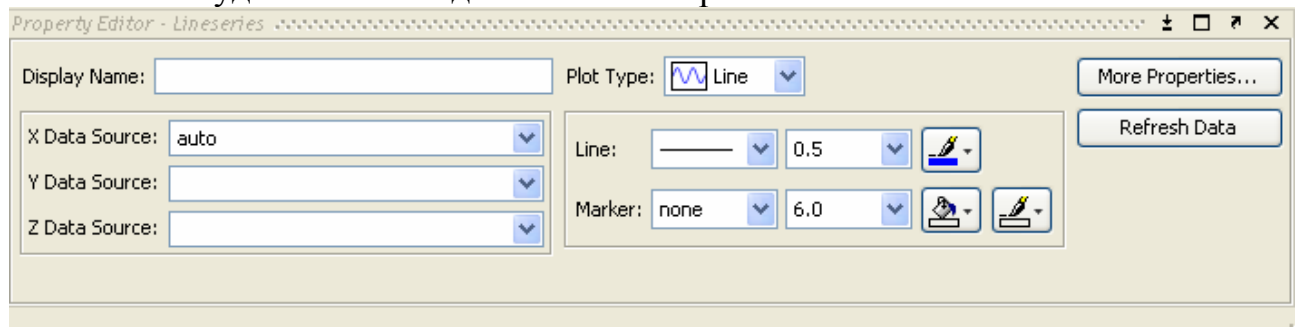


Рис.5.7 Виділення лінії в Show Property Editor

Це вікно призначене для редагування характеристик лінії, можна змінювати тип лінії, її розмір та колір. Для зміни додаткових параметрів натиснемо на кнопку More Properties. Вікно More Properties містить велику кількість параметрів.

## 5.9 План виконання лабораторної роботи «Будова графіків в MATLAB. Редагування графіків»

1. Запускаємо MatLAB.
2. Ознайомлюємося з лабораторною роботою № 5.
3. Виконуємо побудову графіків різними способами.
4. Створюємо новий S-документ.
5. Надаємо ім'я S-документу і зберігаємо його.
6. Для завдання № 2 Л.р. № 4 виконуємо будову трьох графіків:
  - в першому вікні X1, X2, X3 один під одним;
  - в другому вікні Y1 ліворуч, а Y2 праворуч, та під ними графік на якому будуть зображені Y1 і Y2 разом.
7. Виконуємо індивідуальні завдання.

## Список використаної літератури

1. Дьяконов В.П. MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simu-link 5/6. Инструменты искусственного интеллекта и биоинформатики // В.П. Дьяконов, В.В. Круглов. – М.: СОЛОН-ПРЕСС, 2006. – 456 с.
2. Ухин Б.В. Гидравлика: Учебник // Б.В. Ухин, А.А. Гусев. – М.: ИНФРА-М, 2010. – 432 с.
3. Гультияев А.К. Визуальное моделирование в среде MATLAB // А.К. Гультияев. – СПб.: Питер, 2000. – 430 с.
4. Дьяконов В.П. MATLAB5. Система символьной математики // В.П. Дьяконов, И.В. Абраменкова. – М.: Нолидж, 1999. – 633с.
5. Дьяконов В.П. MATLAB. Обработка сигналов и изображений. Специальный справочник // В.П. Дьяконов, И.В. Абраменкова. – СПб.: Питер, 2002. – 608с.