

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

**МЕТОДИЧНІ ВКАЗІВКИ  
до лабораторних робіт з курсу  
«Прикладне програмне забезпечення»**

**Модуль 3.**

**Основи роботи в Simulink**

**для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані  
технології та робототехніка»  
денної та заочної форми навчання**

**Харків 2023**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

**МЕТОДИЧНІ ВКАЗІВКИ**

**до лабораторних робіт з курсу «Прикладне програмне забезпечення»**

Модуль 3. Основи роботи в Simulink

для студентів спеціальності 174 «Автоматизація, комп’ютерно-інтегровані  
технології та робототехніка» денної та заочної форми навчання

Затвержено  
редакційно-видавничою  
радою університету,  
протокол № 1 від 16.02.2023р.

Харків 2023

Методичні вказівки до лабораторних робіт з курсу «Прикладне програмне забезпечення» Модуль 3. Основи роботи в Simulink для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» денної та заочної форми навчання /уклад. Е.Є. Герман, С.Д. Деменкова – Харків: НТУ «ХПІ», 2023. – 36 с .

Укладачі: Е.Є. Герман

С.Д. Деменкова

Рецензент Бабіченко А.К.

Кафедра автоматизації технологічних систем та екологічного моніторингу

# ЛАБОРАТОРНА РОБОТА № 1

## *Початок роботи з мовою візуального програмування Simulink*

### **Мета роботи:**

- ознайомиться інтерфейсом пакета Simulink;
- ознайомиться із бібліотекою блоків пакету Simulink;
- засвоїти принципи побудови моделей мовою візуального програмування Simulink.

### **Короткі теоретичні відомості.**

#### **1. Опис пакету Simulink**

**Simulink** – це середовище, яке є додатком MATLAB, для моделювання, імітації та аналізу динамічних систем, що включають *дискретні, безперервні, гібридні, нелінійні* системи.

Simulink дає можливість будувати графічні блок-діаграми, імітувати динамічні системи, досліджувати працездатність систем. Simulink повністю інтегрований з MATLAB, що забезпечує швидкий доступ до широкого спектру інструментів аналізу і проектування. Ця перевага робить Simulink найпопулярнішим інструментом для проектування систем управління і комутації, цифрової обробки та інших додатків моделювання.

**Simulink** реалізує *принцип візуального програмування*, відповідно до якого, користувач з бібліотеки стандартних блоків на екрані створює модель пристрою, системи або процесу і здійснює відповідні розрахунки та обчислення. На відміну від класичних способів моделювання, користувачеві не потрібно досконально вивчати мову програмування і чисельні методи математики, а досить загальних знань потрібних при роботі на комп'ютері і знань в тій області, в якій він працює.

Simulink є самостійним інструментом. При роботі з Simulink не потрібно знати сам MATLAB та інші його додатки. З іншого боку доступ до функцій MATLAB та його інструментів відкритий і їх можна використовувати в

Simulink. Частина пакетів, що входять до складу MATLAB, має інструменти, вбудовані в Simulink (наприклад, Control System Toolbox – пакет для розробки систем керування).

При роботі в Simulink користувач має можливість :

- *модернізувати* бібліотечні блоки,
- *створювати* свої власні блоки,
- *складати* нові бібліотеки блоків.

При моделюванні користувач може вибирати метод розв’язання диференціальних рівнянь, а також спосіб зміни модельного часу.

Під час моделювання є можливість стежити за процесами, що відбуваються в системі. Для цього використовуються спеціальні пристрої спостереження, що входять до складу бібліотеки Simulink. Результати моделювання можуть бути представлені у вигляді графіків або таблиць.

## 2. Запуск Simulink

Для запуску програми необхідно попередньо запустити середовище MATLAB. Після відкриття основного вікна MATLAB запуск додатку Simulink можна зробити одним із таких способів:

– натиснути відповідну іконку (  Simulink Library) на вкладці HOME панелі інструментів командного вікна MATLAB :

– в командному рядку системи MATLAB виконати команду **simulink**:

```
>> simulink
```

– виконати команду New... на вкладці HOME панелі інструментів і вибрати у групі Simulink “**Simulink Model**”.

– виконати команду Open ... на вкладці HOME та відкрити файл моделі з розширенням .mdl або .slx.

Використання першого та другого способів призводить до відкриття вікна *оглядача розділів бібліотеки Simulink*. Список розділів бібліотеки Simulink представлений у вигляді дерева, і правила роботи з ним є загальними для

списків такого виду. При виборі відповідного розділу бібліотеки в правій частині вікна відображається його вміст.

### 3. Вікно моделі

Вікно моделі призначене для безпосередньої роботи з пакетом Simulink. Моделі, що створені у цьому “вікні” зберігаються у файлах з розширенням .mdl або .slx. Вимоги до імені файлу моделі такі ж, як і до m-файлу. Саме ім'я моделі повинно бути унікальним і не збігатися з іменами файлів з іншими розширеннями.

Вікно моделі містить наступні елементи (див. рис. 1.1):

1. Заголовок з назвою вікна. Новоствореному вікна присвоюється ім'я Untitled з відповідним номером.

2. Меню команд.

3. Панель інструментів.

4. Поле для створення схеми моделі.

5. Рядок стану, що містить інформацію про поточний стан моделі.

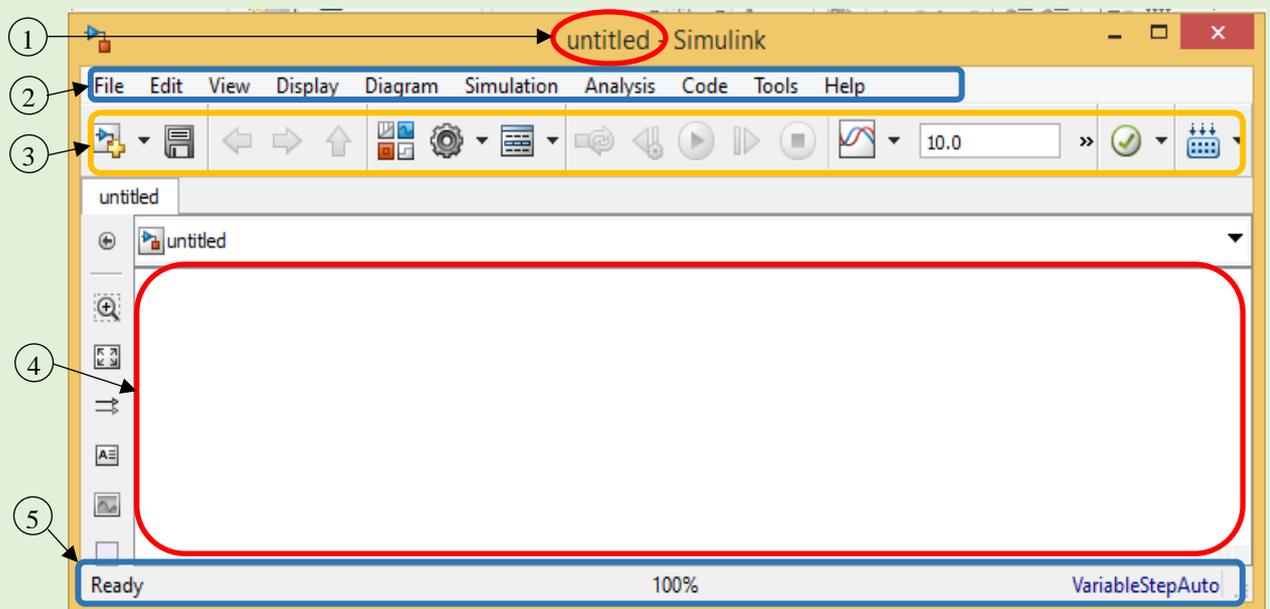


Рисунок 1.1. Вікно моделі Simulink

#### 4. Створення моделі

Для створення моделі в середовищі SIMULINK необхідно послідовно виконати ряд дій.

- Крок 1:** *Створення нового файлу моделі.* Відкрити нове вікно моделі та зберегти його.
- Крок 2:** *Розташувати блоки у вікні моделі.* Переміщення необхідних блоків з відповідних розділів бібліотек в створене вікно файлу моделі.
- Крок 3:** *Зміна параметрів блоку, що встановлені “за замовчуванням”.* Якщо це необхідно, двічі клацнувши лівою клавішею “миші” відкривають вікно редагування параметрів блоку, в якому вносять необхідні зміни.
- Крок 4:** *З’єднання елементів схеми.* Для з’єднання блоків необхідно вказати курсором на “вихід” блоку, а потім, натиснути і, не відпускаючи ліву клавішу “миші”, провести лінію до входу іншого блоку. Після чого відпустити кнопку. Для створення точки розгалуження в з’єднувальній лінії потрібно підвести курсор до передбачуваного вузла і, натиснувши праву клавішу “миші”, протягнути лінію. Для видалення лінії потрібно вибрати лінію, а потім натиснути клавішу Delete.
- Крок 5:** *Встановлення параметрів розрахунку.* Перед виконанням розрахунків необхідно попередньо задати параметри розрахунку, яке виконується в панелі управління меню Simulation/Parameters.

#### 5. Встановлення параметрів розрахунку в Simulink

Перед виконанням розрахунків необхідно попередньо задати параметри розрахунку. У Simulink параметри розрахунку задаються у вікні **Configuration Parameters**, яке відкривається з панелі управління меню за посиланням:

Simulation → Model Configuration Parameters... ,

або комбінацією клавіш Ctrl+E. Також на панелі інструментів є відповідна іконка, через яку можна отримати доступ до вікна редагування параметрів.

При відкритті Вікно редагування параметрів моделювання має вигляд, як вказано на рисунку 1. Це вікно має кілька вкладок перерахованих зліва у вигляді дерева. За замовчуванням відкривається на вкладці **Solver (Розв'язувач)**.

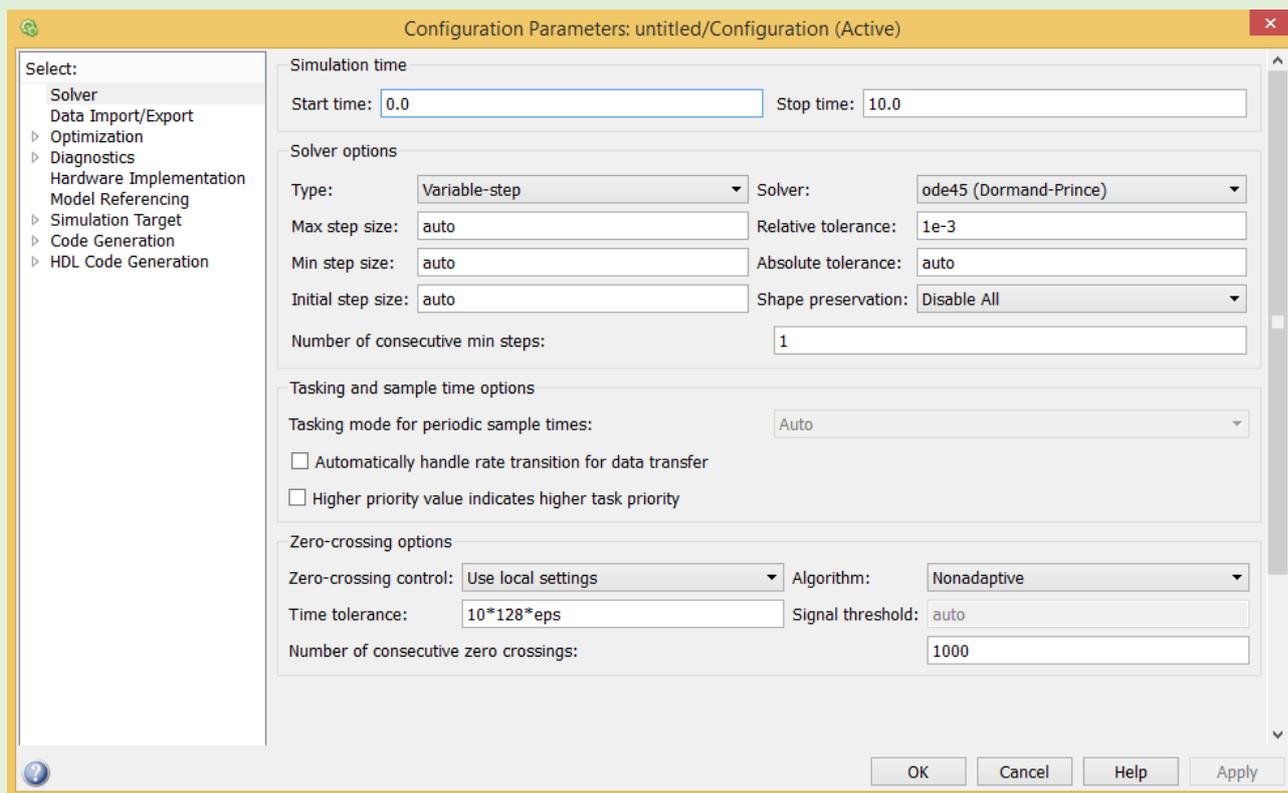


Рисунок 1.1. Вікно редагування параметрів моделювання

### 5.1. Вкладка Solver

На цій вкладці встановлюються параметри розв'язувача – програми, що реалізує чисельне рішення диференціальних рівнянь для системи моделювання Simulink.

Вкладка **Solver** розбита на кілька розділів, першим з яких є **Simulation time**, в якій задаються два параметра:

- **Start time** – початок часу моделювання;
- **Stop time** – кінцевий час моделювання.

У другому розділі вкладки Solver – **Solver options** – задаються тип розв’язувача, метод вирішення і його параметри. Можливі два типи розв’язувача (**Type**):

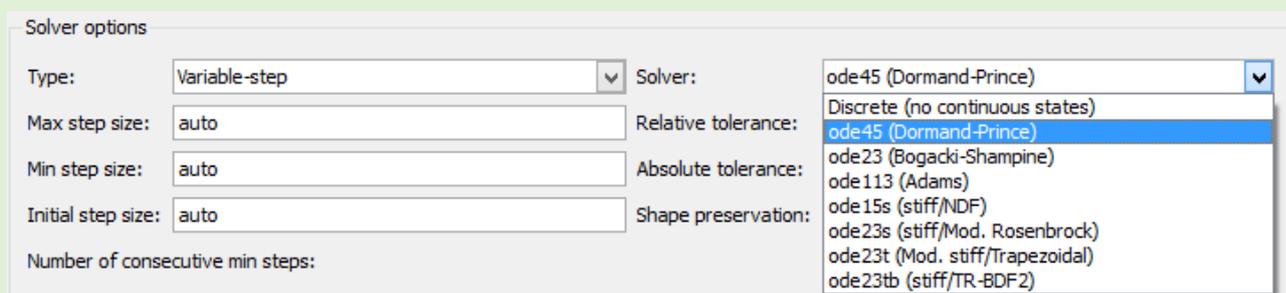
- **Variable-step** – метод рішення зі змінним кроком;
- **Fixed-step** – метод рішення з фіксованим кроком.

За замовчуванням використовується тип розв’язувача зі змінним кроком.

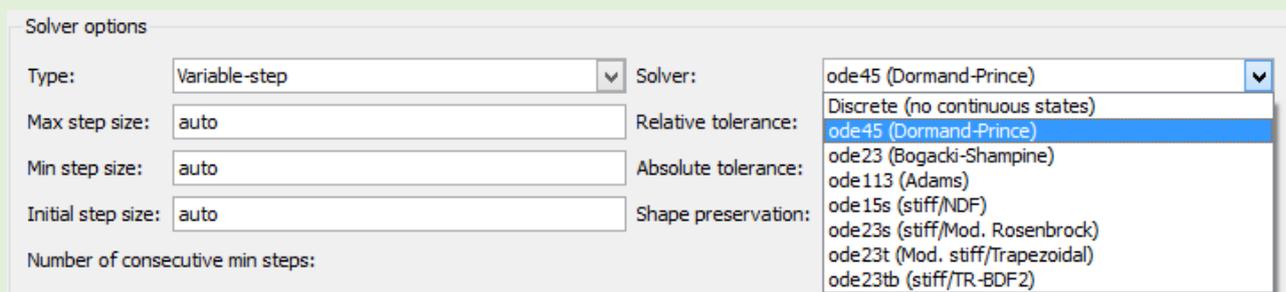
Як правило, кращі результати (зазвичай за часом, але не завжди) дає розв’язувач зі **змінним кроком**. У цьому випадку крок автоматично зменшується, якщо швидкість зміни результатів в процесі рішення зростає. І навпаки, якщо результати змінюються слабо, крок рішення автоматично збільшується.

Моделювання з **фіксованим кроком** варто застосовувати, коли це обумовлено специфікою виконання завдання. Цей метод дає непогані результати, якщо поведінка системи описується майже монотонними функціями. В іншому випадку крок часу доведеться сильно зменшувати для опису найбільш швидких ділянок зміни результатів моделювання, що веде за собою значне зростання часу моделювання.

Залежно від **типу розв’язувача** (зі змінним або постійним кроком) пропонуються різні набори методів для вирішення диференціальних рівнянь (рис. 1.2).



a.



б.

Рисунок 1.2. Розділ Solver options вкладки Solver:

а) розв'язувач зі змінним кроком; б) розв'язувач з постійним кроком

Для розв'язувача *зі змінним кроком* також задаються наступні параметри:

- **Max step size** – максимальний крок інтегрування СОДР;
- **Min step size** – мінімальний крок інтегрування;
- **Initial step size** – початковий крок інтегрування.

За замовченням ці параметри задаються автоматично (**auto**). Для параметрів **Max step size** и **Fixed-step size** за замовчуванням встановлюється значення рівне 1/50 часу моделювання (**Stop time**).

Також важливими параметрами моделювання є *точність інтегрування*:

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

За замовченням вони задані, відповідно, рівними  $10^{-3}$  і  $10^{-6}$ .

Якщо, графіки результатів моделювання виглядають складеними явно з окремих фрагментів, це вказує на необхідність зменшення зазначених значень похибки. Однак занадто малі похибки можуть викликати значне збільшення часу обчислень. Неоптимально обрані значення похибки (як дуже малі, так і дуже великі) можуть викликати нестійкість і навіть «зациклення» процесу моделювання.

## 5.2. Вкладка Data Import/Export

Вкладка **Data Import/Export** має в собі параметри для конфігурації вхідних даних для моделювання (наприклад блок Inport) та вихідних даних (наприклад з блоку Outport). Ці параметри дозволяють імпортувати вхідний сигнал та початковий стан системи з робочого простору (**workspace**) та експортувати під час моделювання вихідний сигнал та дані стану в робочій простір MATLAB. Ця можливість дозволяє використовувати стандартні або настроювані функції MATLAB для генерації змодельованих вхідних сигналів системи та для побудови графіків, аналізу або іншої постобробки вихідних даних системи.

## 6. Блоки бібліотеки Simulink

Опис деяких блоків бібліотеки Simulink надано у Додатку 2

### Порядок виконання роботи:

1. Створити файл моделі і зберегти його.
2. Обчислити вираз, використовуючи пакет Simulink:

Непарний варіант	$\lg \frac{15^2 - (\sqrt{5-1} + 3) \cdot 5}{(10^3 + 2^3 \cdot 3)^{\frac{1}{4 + \frac{9 \cdot 2}{8-5}}}}$
Парний варіант	$\sqrt[3]{\frac{\lg^6 \left( 6^2 + \frac{4^4}{4} \right) - 4 \operatorname{tg} \frac{\pi}{4}}{4}} + \frac{5^4}{(5 + 4 - \cos \pi)^{\sqrt{5+4}}}$

Відповідь відобразити у блоку Display

3. У новому файлі моделі вивести у вікні блоку осцилографа на 4-х осях графіки наступних генераторів сигналів, встановивши вказані параметри:

	Генератор сигналу	Параметри
Непарний варіант	<b>Ramp</b>	коефіцієнт нахилу (slope) – 0.5, час початку (start time) – 2 с;
	<b>Pulse Generator</b>	амплітуда – 3, період – 4 с; ширина імпульсу у відсотках від періоду (Pulse Width) – 10;
	<b>Uniform Random Number</b>	мінімальне значення – 0, максимальне значення – 2, час дискретизації – 0.2 с;
	<b>Sine Wave</b>	–
Парний варіант	<b>Ramp</b>	коефіцієнт нахилу (slope) – 1.5, час початку (start time) – 4 с;
	<b>Repeating Sequence</b>	значення часу (Time values) – [0: 0.5: 2]; значення виходу (Output values) – [-2 -1 2 3 -1];

<b>Random Number</b>	середня (Mean) – 1, дисперсія (Variance) – 2;
<b>Repeating Sequence Stair</b>	вектор вихідних значень – [3 1 7 4 1]; час дискретизації – 0.4 с.

4. У новому файлі побудувати модель, яка генерує синусоїдальний сигнал зі вказаним періодом та амплітудою, використовуючи джерело сигналу **Ramp** і блок тригонометричних функцій. Визначити абсолютне значення сигналу і його знак. Вивести отримані сигнали на осцилограф. Зменшити максимальний крок.

	період	амплітуда
Непарний варіант	$\pi$	2.5
Парний варіант	2	$0.5 \pi$

5. Об'єкт описаний рівняннями в просторі станів:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

де

Непарний варіант	$A = \begin{vmatrix} 0 & 1 \\ -5 & -2 \end{vmatrix}, B = \begin{vmatrix} 0 \\ 3 \end{vmatrix}, C = \begin{vmatrix} 0 & 1 \end{vmatrix}, D = 0;$
Парний варіант	$A = \begin{vmatrix} -9 & -8 \\ -12.5 & 0 \end{vmatrix}, B = \begin{vmatrix} 2 \\ 0 \end{vmatrix}, C = \begin{vmatrix} 1 & 2 \end{vmatrix}, D = 0.$

У новому файлі визначити реакцію на одиничний стрибок:

- а. в розімкнутому контурі;
- б. в замкнутому контурі.

На осцилографі на одних осях координат відобразити скачок і реакцію об'єкта в обох випадках.

6. У новому файлі побудувати фігуру Лісажу: на вхід блоку XY Graph з блоку Sine Wave подаються 2 синусоїдальних сигнали (максимальний розмір кроку 0.1):

Непарний варіант	Частота – 2 і 3 рад/с, час моделювання – 2π.
Парний варіант	Частота – 1.5 і 4 рад/с, час моделювання – 4π.

7. Визначити реакцію замкненої системи з негативним зворотним зв'язком на одиничний скачок:

а) Система містить об'єкт керування, що описаний передатною функцією.

б) Система містить об'єкт керування та ПІД регулятор в прямому контурі (перед об'єктом управління). Коефіцієнти ПІД регулятора:

Вивести на осцилограф три сигнали: одиничний скачок, реакцію системи без контролера і з контролером.

	Передатна функція моделі об'єкту	Коефіцієнти ПІД регулятора
Непарний варіант	$W(s) = \frac{s + 0.5}{3s^3 + 12s^2 + 0.2}$	$K_p=9, K_i=1.7, K_d=15.$
Парний варіант	$W(s) = \frac{3s^2 + 2s + 0.5}{40s^3 + 20s + 1}$	$K_p=1, K_i=0.2, K_d=3.$

## ЛАБОРАТОРНА РОБОТА № 2

### Метод Зіглер-Нікольса

#### Мета роботи:

- вивчення коливального методу налаштування параметрів ПІД регулятора (метод Зіглера-Нікольса);
- реалізація методу Зіглера-Нікольса в пакеті Simulink.

#### Короткі теоретичні відомості

##### 1. Контур керування з ПІД регулятором

Структурна схема контуру керування з регулятором ПІД типу має вигляд, представлений на рисунку 2.1. На схемі об'єкт керування знаходиться на прямій ділянці ланцюга, а регулятор – на ділянці зворотного зв'язку. Таке положення регулятора – на ділянці зворотного зв'язку – зумовлено тим, що система керування є системою стабілізації (керування по збуренню). Такі системи та процеси превалюють в хімічній промисловості.

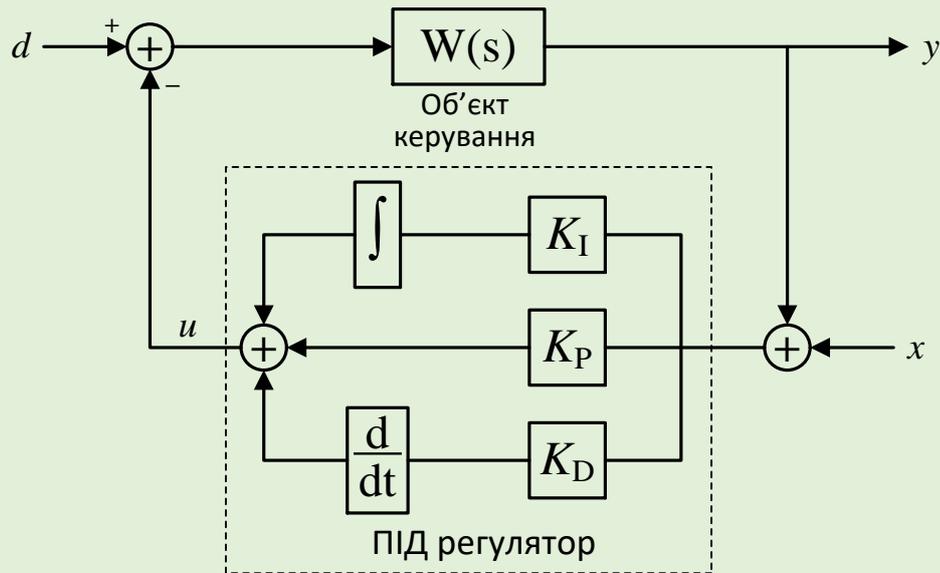


Рисунок 2.1. Структурна схема контуру керування з ПІД регулятором

В таких процесах регулятори ПІД типу використовуються для підтримки заданого значення  $x$  через зміну значення  $u$ , де  $x$  – завдання (або уставка), а різниця  $e = x - u$  – помилка регулювання (або нев'язка), тобто відхилення величини від заданого значення.

Вихідний сигнал ПІД регулятора визначається трьома доданками згідно з формулою:

$$u(t) = P + I + D = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de}{dt},$$

а його передатна функція має вигляд:

$$W_{PID}(s) = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s},$$

де  $K_p$ ,  $K_I$ ,  $K_D$  – коефіцієнти посилення пропорційної, інтегральної і диференціальної складових регулятора, відповідно. Таку форму опису ПД регулятора називають *ідеальною* або *паралельною*.

Деякі методи налаштування ПД-регуляторів використовують іншу схему – так звану *стандартну* (рисунок 2.2), яка описується виразом:

$$u(t) = K_p \left( e(t) + K_{ip} \int_0^t e(\tau) d\tau + K_{dp} \frac{de}{dt} \right), \quad (3)$$

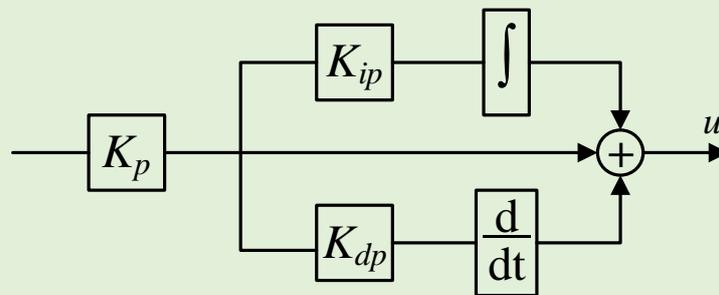


Рисунок 2.2. Структурна “стандартна” схема ПД регулятора.

Часто в якості параметрів ПД-регулятора використовують наступні:

– відносний діапазон:

$$P_b = \frac{1}{K_p},$$

– постійні інтегрування і диференціювання, що мають розмірність часу:

$$T_i = \frac{1}{K_{ip}} K_p T, \quad T_d = K_{dp}.$$

Слід враховувати, що терміни використовуються по-різному в різних джерелах і різними виробниками регуляторів.

## **2. Метод Зіглера-Нікольса**

### **2.1. Опис методу Зіглера-Нікольса**

При дослідженнях перехідних процесів для різних об'єктів регулювання, присутня закономірність, яку помітили J.G. Ziegler і N.V. Nichols, під час проведення випробувань пневматичних регуляторів, в 1941 р. в компанії “Taylor Instruments” (м. Ротчестер, штат Нью-Йорк).

Вона полягала в тому, що оптимальна зона пропорційності П-регулятора, як правило, в два рази більше величини зони пропорційності, при якій в системі керування починається автоколивальний процес.

Ziegler і Nichols також визначили залежність між періодом автоколивань виникають та постійними часу інтегрування і диференціювання. Завдяки знайденим співвідношенням, з'явилася можливість швидко і просто налаштувати П-, ПІ- і ПІД-регулятори, не вдаючись до складних математичних розрахунків.

На даний момент цей метод використовується для попереднього (грубого) налаштування. Однак, найчастіше, його використання задовольняє поставленим завданням.

### **2.2. Алгоритм налаштування регулятора за методом Зіглера-Нікольса**

Алгоритм налаштування регулятора ПІД типу методом Зіглера-Нікольса складається з наступних кроків:

- Крок 1** Коефіцієнти посилення регулятора встановлюються рівними 0.
- Крок 2** Поступово змінюємо пропорційний коефіцієнт посилення  $K_P$  до тих пір, поки в контурі не почнуться автоколивання.
- Крок 3** Фіксується автоколивальне посилення  $K_{ак}$  регулятора та

період коливань  $T_{ак}$ .

**Крок 4** Розраховуються параметри регулятора ПІД типу згідно з таблицями 2.1 та 2.2.

Таблиця 2.1

Розрахунок коефіцієнтів ПІД регулятора через  $K_P$

	$K_P$	$K_I$	$K_D$
<b>П</b>	$0.5K_{ак}$	–	–
<b>ПИ</b>	$0.45K_{ак}$	$\frac{1.2K_P}{T_{ак}}$	–
<b>ПІД</b>	$0.6K_{ак}$	$\frac{2K_P}{T_{ак}}$	$\frac{K_P T_{ак}}{8}$

Таблиця 2.2

Розрахунок коефіцієнтів ПІД регулятора через  $K_{ак}$

	$K_P$	$K_I$	$K_D$
<b>П</b>	$0.5K_{ак}$	–	–
<b>ПИ</b>	$0.45K_{ак}$	$\frac{0.54K_{ак}}{T_{ак}}$	–
<b>ПІД</b>	$0.6K_{ак}$	$\frac{1.2K_{ак}}{T_{ак}}$	$\frac{3K_{ак} T_{ак}}{40}$

### Порядок виконання роботи.

1. Створити файл моделі Simulink та зберегти його.
2. Побудувати на мові візуального програмування Simulink контур регулювання відповідно до рисунку 2.1. Модель об'єкту задана передатною функцією (див. у додатку Таблиця Д2.1) і співпадає з моделлю об'єкта з завдання до 2-ї лабораторної роботи II модулю.
3. Встановити коефіцієнти регулятора рівними 0. Подати на вхід східчастий сигнал. Результат відобразити на осцилографі.
4. Визначити пропорційний коефіцієнт регулятора при автоколиваннях.
4. Визначити період автоколивань.
5. Розрахувати по таблиці параметри налаштування ПІД регулятора за методом Зіглера-Нікольса.

6. Встановити визначені параметри регулятора та відобразити на осцилографі перехідні процеси контуру управління без регулятора, контуру управління з регулятором та вхідний сигнал.

## Лабораторна робота № 3

### Комунікація MATLAB і Simulink

#### Мета роботи:

- освоїти взаємозв'язок MATLAB і Simulink;
- створення файл-функцій в MATLAB.

#### Короткі теоретичні відомості

##### 1. Комунікація MATLAB і Simulink

Запуск програми на мові візуального програмування з системи MATLAB здійснюється за допомогою функції `sim()`, аргументом якої є ім'я файлу, що виконується:

```
sim('ModelName')
```

У разі, якщо необхідно виконати моделювання з певними параметрами, не встановленими заздалегідь, до функції `sim()` додають аргументи:

```
sim('ModelName', T, opt)
```

де **T** – параметр, що визначає час моделювання, який може бути заданий в таких варіантах:

- значенням *часу моделювання* у вигляді скаляра,
- вектором з двох елементів, які відповідають *початку та кінцю часу моделювання*,

– вектором, що визначає *часовий масив*.

Якщо час моделювання визначати не потрібно (воно задано в самому файлі моделі), то ставляться порожні квадратні дужки `[]`.

`opt` – змінна, яка визначає параметри моделювання за допомогою функції `simset`:

```
opt = simset(...)
```

У нашому випадку пропонується використовувати наступні параметри моделювання:

```
opt = simset('solver', 'ode5', 'SrcWorkspace', 'Current');
```

Більш детально ознайомитись з функцією `simset()` та її можливими аргументами можна в документації набравши команду

```
>> doc simset
```

Функція `sim()` може повертати певні значення, які описуються в файлі моделі. Для цього у файлі моделі додається блок `Out`, на вхід якого подається потрібний для подальшого аналізу сигнал.

За посиланням **Simulation**→**Configuration Parameters** в діалоговому вікні на вкладці **Data Import/Export** можна переглянути і редагувати вхідні і вихідні значення.

Функція `sim()`, яка повертає значення моделювання має такий вигляд :

```
[tout,xout,yout] = sim(...);
```

де `tout` – вектор часу перехідної характеристики;

`xout` – матриця станів системи керування;

`yout` – вектор значень перехідної характеристики по осі ординат.

Більш детально ознайомитись з функцією **sim** можна в документації набравши команду:

```
>> doc sim
```

## 2. Функції в MATLAB

Функції в MATLAB реалізуються через файл-функції.

### 2.1. Файл-функції

*Файл-функції* (або *m-файл-функції*) є типовим повноцінним об'єктом мови програмування MATLAB. Водночас вони є повноцінним модулем з точки зору структурного програмування, оскільки містить вхідні та вихідні параметри і використовує апарат локальних змінних.

#### 2.1.1. Властивості файл-функції:

- функція повертає своє значення і може використовуватися у виразах;
- всі змінні, які присутні середині файлах-функції, є локальними, тобто діють тільки в межах тіла функції;
- файл-функція є самостійним програмним модулем, який спілкується з іншими модулями через свої вхідні та вихідні параметри;
- файл-функція є засобом розширення системи MATLAB;
- при виявленні файлу-функції він компілюється, а потім виконується, а створені машинні коди зберігаються в робочій області MATLAB.

#### 2.1.2 Структура файл-функції

Файл-функція починається з об'явлення **function**. Якщо файл-функція має один вихідний параметр, то після **function** вказується ім'я змінної – вихідного параметра, знак рівняння, ім'я самої функції та в круглих дужках список аргументів функції. Далі, за потребою, йде основний та додатковий коментарі. Потім йде тіло функції. Функція завершується виразом, де змінній, що оголошена після **function**, надається значення (або результат виразу),

який функція повертає. Загалом структура файл-функції з одним вихідним параметром виглядає наступним чином:

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

Якщо вихідних параметрів більше, то вони вказуються в квадратних дужках після **function**. При цьому структура модуля має наступний вигляд:

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

У випадку, коли не має потреби повертати хоча б який результат, а сама функція повинна робити якісь визначені дії (наприклад будувати графіки або друкувати результат обчислень), то після **function** одразу оголошують функцію:

```
function f_name(Список_параметрів)
%Основний коментар
%Додатковий коментар
Тіло функції з будь-якими виразами
```

Функція завершується, коли всі команди у тілі функції будуть виконані.

## 2.2. Підфункції

В загальному випадку кожна функція може бути представлена окремим файлом-функцією, до якої звертаються за іменем файлу, шлях до якого є у

відповідних списках. Однак така організація програмного коду може виявитися не зручною. В такому випадку використовують підфункції.

Підфункції в MATLAB – це функції які оголошені в файл-функції після основної (головної) функції файлу і можуть бути викликані тільки в межах цього файлу-функції. З інших файлів ці функції не видно. При цьому головна функція не приймає і не повертає аргументів.

### **Порядок виконання роботи.**

1. Створити файл-функцію.
2. На основі Лабораторної роботи №2 II-го модулю у створеному файлі написати підфункцію, яка за заданими параметрами передатної ( $T_1$ ,  $T_2$ ,  $k$ ,  $\tau$ ,  $\psi$ ) розраховує коефіцієнти III регулятора методом Зіглера-Нікольса та повертає передатну функцію регулятора.
3. Побудувати перехідний процес контуру регулювання.
4. На основі Лабораторної роботи №4 II-го модулю у створеному файлі написати підфункцію, яка розраховує параметри перехідного процесу та друкує їх у Command Window.
5. Зі створеного файлу запусити Simulink модель, що була отримана за результатами Лабораторної роботи №2 III-го модулю, з встановленими параметрами III та ПІД регулятора.
6. Побудувати перехідні процес контуру регулювання з III та ПІД регуляторами, розрахованими за методом Зіглера-Нікольса, та визначити параметри перехідних процесів.
7. Побудувати порівняльний графік перехідних процесів контуру регулювання з регуляторами розрахованими за методи РАФЧХ та Зіглера-Нікольса.



## Додатки

### Додаток 1. Завдання до лабораторної роботи №2

Таблиця Д1

1.	$W(s) = \frac{e^{-3s}}{15s^2 + 10s + 1};$	2.	$W(s) = \frac{1.1e^{-7\tau}}{20s + 1};$
3.	$W(s) = \frac{0.8e^{-60s}}{100s^2 + 250s + 1};$	4.	$W(s) = \frac{0.92e^{-0.09\tau}}{1.2s + 1};$
5.	$W(s) = \frac{1.3e^{-1.5s}}{5s^2 + 7s + 1};$	6.	$W(s) = \frac{1.2e^{-70\tau}}{220s + 1};$
7.	$W(s) = \frac{0.65e^{-30s}}{125s^2 + 75s + 1};$	8.	$W(s) = \frac{0.7e^{-9\tau}}{49s + 1};$
9.	$W(s) = \frac{0.87e^{-6s}}{18s^2 + 30s + 1};$	10.	$W(s) = \frac{e^{-110\tau}}{290s + 1}.$
11.	$W(s) = \frac{1.2e^{-8s}}{120s^2 + 21s + 1};$	12.	$W(s) = \frac{1.25e^{-27\tau}}{64s + 1};$
13.	$W(s) = \frac{0.96e^{-75s}}{111s^2 + 265s + 1};$	14.	$W(s) = \frac{0.62e^{-0.18\tau}}{0.75s + 1};$
15.	$W(s) = \frac{1.14e^{-1.6s}}{3s^2 + 8s + 1};$	16.	$W(s) = \frac{0.4e^{-320\tau}}{787s + 1};$
17.	$W(s) = \frac{0.5e^{-25s}}{175s^2 + 25s + 1};$	18.	$W(s) = \frac{0.8e^{-3.5\tau}}{16s + 1};$
19.	$W(s) = \frac{0.5e^{-12s}}{118s^2 + 28s + 1};$	20.	$W(s) = \frac{1.17e^{-725\tau}}{2943s + 1}.$
21.	$W(s) = \frac{e^{-0.3s}}{2.3s^2 + 1.1s + 1};$	22.	$W(s) = \frac{e^{-17\tau}}{37s + 1};$
23.	$W(s) = \frac{1.5e^{-42s}}{98s^2 + 181s + 1};$	24.	$W(s) = \frac{2.1e^{-4\tau}}{22s + 1};$

## Додаток 2. Короткий опис деяких блоків бібліотеки Simulink

### Sources – джерела сигналів

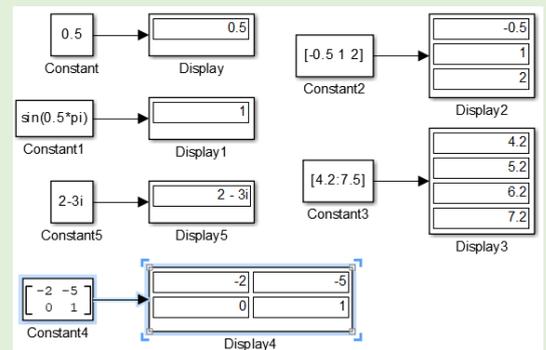
**Constant** – джерело постійного сигналу.

**Призначення:** Задає постійний за рівнем сигнал.

**Параметри:**

**Constant value** – константне (постійне) значення сигналу.

Значення константи може бути дійсним або комплексним числом, виразом що обчислюється, вектором або матрицею.



**Step** – генератор східчастого сигналу.

**Призначення:** формує східчастий сигнал.

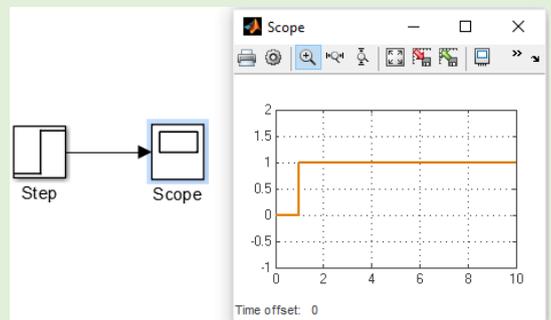
**Параметри:**

**Step time** – час настання перепаду сигналу;

**Initial value** – початкове значення сигналу;

**Final value** – кінцеве значення сигналу.

Кінцеве значення може бути більше ніж початкове, так і менше. Значення початкового і кінцевого рівнів можуть бути не тільки позитивними, але й негативними (наприклад, зміна сигналу з рівня -5 до рівня -3). На рисунку показано використання генератора східчастого сигналу.



**Ramp** – джерело впливу, що змінюється лінійно.

**Призначення:** формує лінійний сигнал виду:

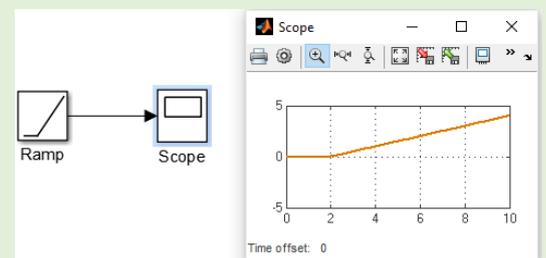
$$y = \text{Slope} * \text{time} + \text{Initial value}.$$

**Параметри:**

**Slope** – коефіцієнт нахилу;

**Start time** – час початку дії сигналу;

**Initial output** – початкове значення сигналу.

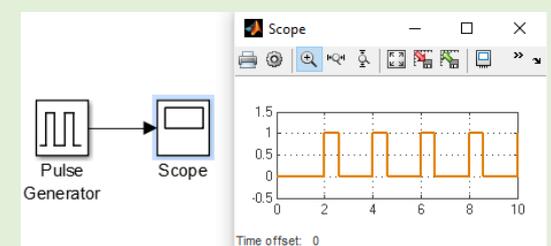


**Pulse Generator** – джерело імпульсного сигналу.

**Призначення:** формує прямокутні імпульси.

**Параметри:**

**Amplitude** – амплітуда імпульсного



сигналу;

**Period** – період імпульсного сигналу;

**Pulse Width** – ширина імпульсу у відсотках від періоду;

**Phase Delay** – затримка по фазі.

**Sine Wave** – джерело синусоїдального сигналу.

**Призначення:** формує синусоїдальний сигнал із заданою частотою, амплітудою, фазою і зміщенням.

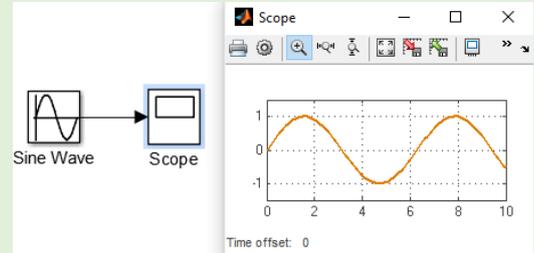
**Amplitude** – амплітуда сигналу;

**Bias** – зсув по амплітуді;

**Frequency** – частота сигналу;

**Phase** – фаза сигналу;

**Sample time** – час дискретизації.



**Random Number** – джерело випадкового сигналу з нормальним розподілом.

**Призначення:** формування випадкового сигналу з нормальним розподілом.

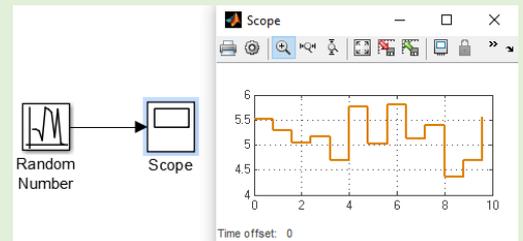
**Параметри:**

**Mean** – середнє значення;

**Variance** – дисперсія;

**Seed** – початкове число;

**Sample time** – час дискретизації.



**Uniform Random Number** – джерело випадкового сигналу з рівномірним розподілом.

**Призначення:** формування випадкового сигналу з рівномірним розподілом.

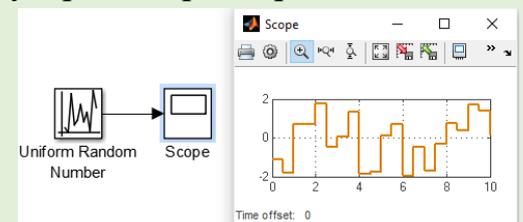
**Параметри:**

**Minimum** – мінімальне значення сигналу;

**Maximum** – максимальне значення сигналу;

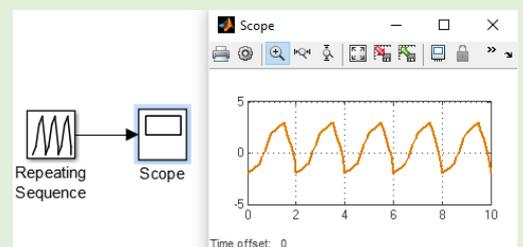
**Seed** – початкове число;

**Sample time** – час дискретизації.



**Repeating Sequence** – джерело послідовного сигналу, що повторюється.

**Призначення:** формує періодичний скалярний сигнал із формою, яка задана двома парними векторами.



**Time values** – вектор вихідних значень по часу;

**Output values** – вектор амплітуд сигналу у відповідний час.

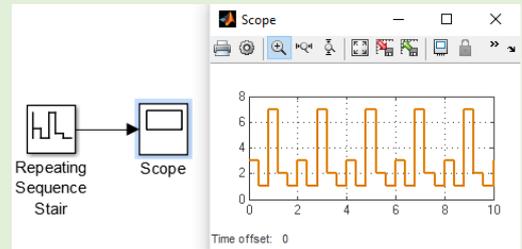
**Repeating Sequence Stair** – джерело повторюваної східчастої послідовності.

**Призначення:** формування дискретної часової послідовності, що повторюється.

**Параметри:**

**Vector of output values** – середнє значення;

**Sample time** – час дискретизації.



## Sinks – приймачі сигналів

**Scope** – осцилограф.

**Призначення:** Будує графіки сигналів як функції часу. Дозволяє спостерігати за змінами сигналів в процесі моделювання.

Для перегляду сигналів необхідно відкрити вікно блоку. Це можна зробити на будь-якому етапі розрахунку (як до, так і під час розрахунку). Якщо на вхід блоку надходить векторний сигнал, то крива для кожного елемента вектору будується окремим кольором.

**Налаштування** вікна осцилографа виконується за допомогою **панелі інструментів**. Вона містить наступні елементи:

**Print** – друк вмісту вікна осцилографа;



**Parameters** – доступ до вікна настройки параметрів;

**Zoom** – збільшення масштабу по обох осях;

**Zoom X-axis** – збільшення масштабу по горизонтальній осі;

**Zoom Y-axis** – збільшення масштабу по вертикальній осі;

**Autoscale** – автоматичне масштабування по обох осях;

**Save current axes settings** – збереження поточних налаштувань вікна;

**Restore saved axes settings** – установка раніше збережених налаштувань вікна;

**Floating scope** – переведення осцилографу у “вільний” режим;

**Axes are currently locked** – закріпити/розірвати зв'язок між поточною координатною системою вікна і сигналом, що відображається (доступний у режимі **Floating scope**);

**Signal selection** – вибір сигналів для відображення. (доступний у режимі **Floating scope**).

**Параметри** блоку встановлюються у вікні **Parameters**, яке відкривається через елемент **Parameters** панелі інструментів. Вікно параметрів має три вкладки:

**General** – загальні параметри;

**History** – параметри збереження сигналів в робочій області MATLAB;

**Style** – параметри графічного відображення.

### General:

**Number of axes** – число входів (осей координат) осцилографа. При зміні цього параметра на зображенні блоку з'являються додаткові входи.

**Time range** – величина часового інтервалу для якого відображаються графіки (auto).

**Tick labels** – друк/приховування осей і міток осей (має значення зі списку):

**all** – підписи для всіх осей,

**none** – відсутність усіх осей і підписів до них,

**bottom axis only** – підписи горизонтальній осі тільки для нижнього графіка.

**Sampling** – установка параметрів виведення графіків. Приймає значення зі списку:

**Decimation** – проріджування/кратність виведення. Встановлюється числом, що задає крок друку розрахункових координат.

**Sample time** – час дискретизації. Визначає інтервал квантування при відображенні сигналу.

**Floating scope** – змінює блок Scope на блок Floating Scope.

**legends** – відображає легенду назв сигналів із кольором та стилем.

### History:

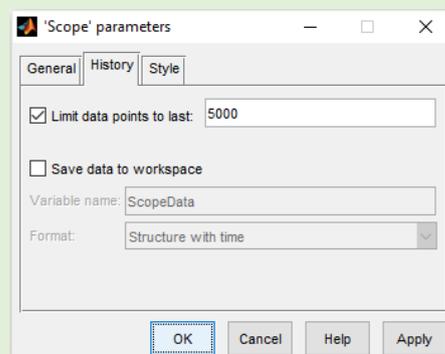
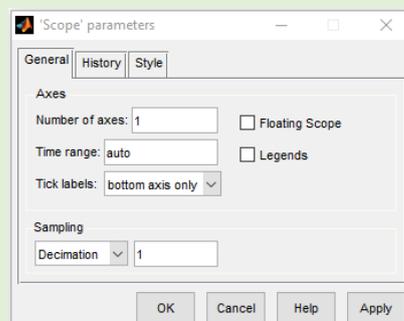
**Limit data points to last** – максимальна кількість точок графіка що відображаються. При перевищенні цього значення початкова частина графіка обрізається. Якщо прапорець не встановлений, значення параметру ігноруються і відображається повний графік.

**Save data to workspace** – збереження значень сигналів в робочій області MATLAB.

**Variable name** – ім'я змінної для збереження сигналів в робочій області;

**Format** – формат даних при збереженні в робочій області (зі списку): **Array**, **Structure**, **Structure with time**.

### Style:



**Figure color** – колір фону для вікна Score.

**Axes colors** – вибір кольору фону осей координат (перша кольорова палітра) та кольору сітки (друга кольорова палітра).

**Properties for line** – вибір номеру графіка для якого встановлюються властивості.

**Line** – вибір стилю, ширини та кольору лінії.

Блок Score відображає один вхід на вісь. При відображенні вектора або матриці сигналів на одній осі, блок Score призначає кольори кожному сигналу в такому порядку: жовтий, пурпурний, блакитний, червоний, зелений і темно-синій. Якщо є більше шести сигналів, блок Score переробляє кольори вдруге.

**Marker** – вибір стиль маркера.

**XY Graph** – графопобудовник.

**Призначення:** будує параметричний графік залежності одного сигналу від іншого:  $Y(t)(X(t))$ .

**Параметри:**

Граничні значення області визначення по осях:

**x-min** – мінімальне значення по осі X.

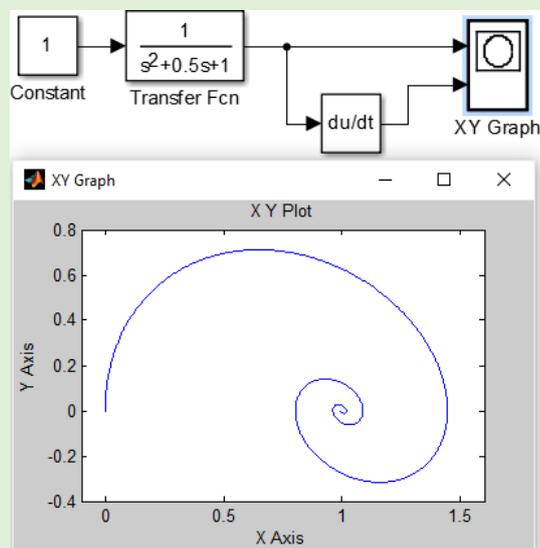
**x-max** – максимальне значення по осі X

**y-min** – мінімальне значення по осі Y.

**y-max** – максимальне значення по осі Y

**Sample time** – час дискретизації.

Блок має два входи. Верхній вхід призначений для подачі сигналу, який є аргументом (X), нижній – для подачі значень функції (Y).



**Display** – цифровий дисплей.

**Призначення:** відображає значення сигналу у вигляді числа.

**Параметри:**

**Format** – формат відображення даних.

Приймає значення зі списку:

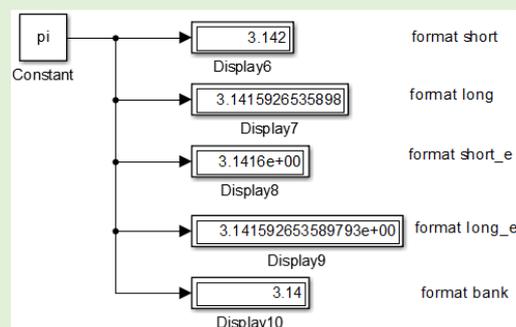
**short** – 5 значущих десяткових цифр.

**long** – 15 значущих десяткових цифр.

**short\_e** – 5 значущих десяткових цифр і 3 символу ступеня десяти.

**long\_e** – 15 значущих десяткових цифр і 3 символу ступеня десяти.

**bank** – “грошовий” формат (з фіксованою точкою і двома десятковими цифрами в дробовій частині).



**Decimation** – кратність відображення вхідного сигналу.

**Sample time** – час дискретизації (-1 за замовчуванням).

**Floating display** (прапорець) – переведення блоку в “вільний” режим.

**To Workspace** – блок збереження даних в робочій області.

**Призначення:** Блок записує дані, що надходять на його вхід, в робочу область MATLAB.

**Параметри:**

**Variable name** – ім'я змінної, куди записуються дані.

**Limit data points to last** – максимальна кількість значень вектору даних по часу (від моменту завершення моделювання). Якщо значення параметра Limit data points to last задано як inf, то в робочій області будуть збережені всі дані.

**Decimation** – кратність відображення вхідного сигналу.

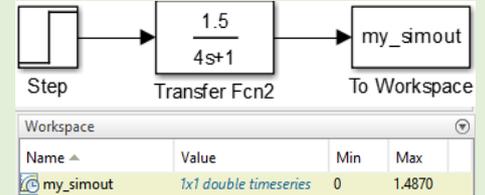
**Sample time** – час дискретизації.

**Save format** – формат збереження даних (значення зі списку):

**Array** – матриця. Дані зберігаються як масив, в якому число рядків визначається числом розрахункових точок по часу, а число стовпців – розмірністю вектору, що подається на вхід блоку. Якщо на вхід подається скалярний сигнал, то матриця буде містити лише один стовпець.

**Structure** – дані зберігаються у вигляді структури.

**Structure with Time** – структура з додатковим полем (час).



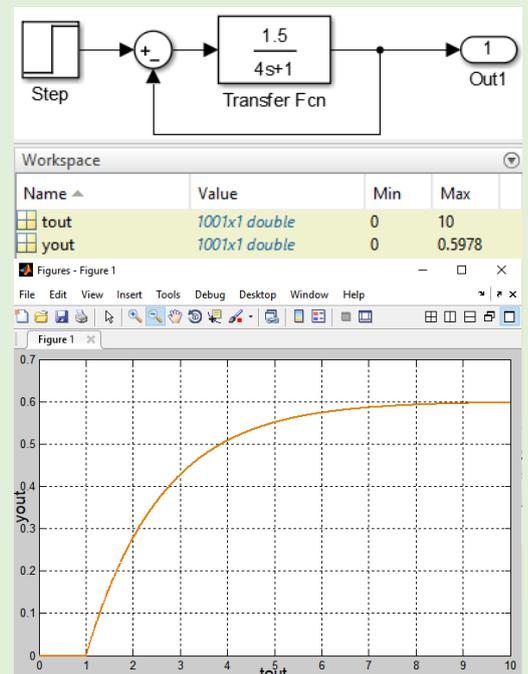
**Outport** або **Out** – вихідний порт.

**Призначення:** Порт виводу даних сигналів з підсистем або до зовнішнього ресурсу.

Блок пов'язує сигнали від системи до місця призначення за межами системи. Він може з'єднувати сигнали, що надходять від підсистеми до інших частин моделі, а також може забезпечувати зовнішні вихід даних на верхні рівні ієрархії моделі.

**Параметри:**

**Port number** – номер порту блоку. Цей параметр контролює порядок, у якому порт, що відповідає блоку, з'являється в батьківській підсистемі або блоці моделі.



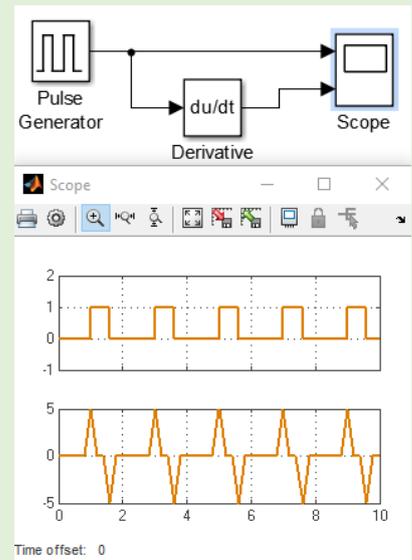
**Icon display** – тип інформації, що відобразиться на значку вхідного порту.

### Continuous – аналогові блоки.

**Derivative** – блок обчислення похідної.

**Призначення:** Виконує чисельне диференціювання вхідного сигналу.

Для обчислення похідної використовується наближена формула Ейлера:  $\frac{du}{dt} = \frac{\Delta u}{\Delta t}$ , де  $\Delta u$  – величина зміни вхідного сигналу за час  $\Delta t$ , де  $\Delta t$  – поточне значення кроку модельного часу. Значення вхідного сигналу блоку до початку розрахунку вважається рівним нулю. Початкове значення вихідного сигналу також вважається рівним нулю. Точність обчислення похідної істотно залежить від величини встановленого кроку розрахунку. Вибір меншого кроку розрахунку покращує точність обчислення похідної.



**Параметри:**

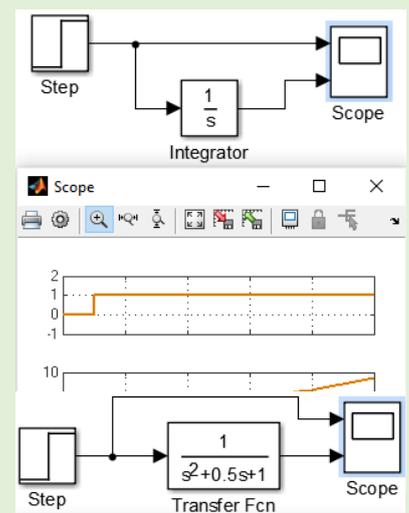
**Coefficient for linearization** – коефіцієнт апроксимаційної передатної

функції  $\frac{s}{cs+1}$ .

**Integrator** – блок інтегрування.

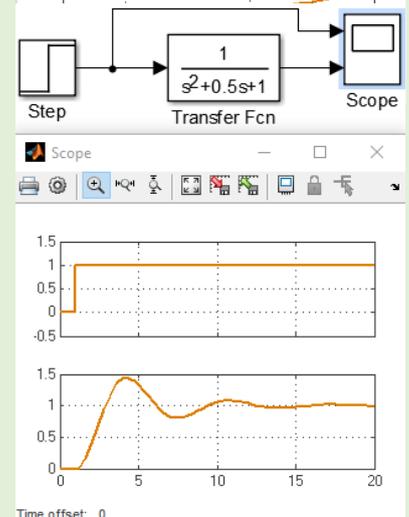
**Призначення:** Виконує інтегрування вхідного сигналу.

**Параметри:** Має велику кількість параметрів, проте всі вони будуть використовуватися за замовчуванням, якщо не вказано інше.



**Transfer Fcn** – блок моделі у вигляді передатної функції.

**Призначення:** задає модель у вигляді передатної функції, як відношення поліномів:



$H(s) = \frac{y(s)}{u(s)} = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}{b_d s^d + b_{d-1} s^{d-1} + \dots + b_1 s + b_0}$ , де  $n$  і  $d$  – порядок чисельника і знаменника передатної функції,  $n \leq d$ .

**Параметри:**

**Numerator** – вектор або матриця коефіцієнтів полінома чисельника;

**Denominator** – вектор коефіцієнтів полінома знаменника;

**Absolute tolerance** – абсолютна похибка.

Вхідний сигнал блоку повинен бути скалярним. У тому випадку, якщо коефіцієнти чисельника задані вектором, то вихідний сигнал блоку буде також скалярним (як і вхідний сигнал).

**Zero-Pole** – блок моделі у вигляді полюсів та нулів.

**Призначення:** визначає модель із заданими полюсами

і нулями:  $H(s) = \frac{y(s)}{u(s)} = k \frac{(s - z_1)(s - z_2) \dots (s - z_n)}{(s - p_1)(s - p_2) \dots (s - p_n)}$ , де  $z_i$

– нулі моделі (корені полінома чисельника),  $p_i$  – полюси моделі (корені полінома знаменника),  $n \leq d$ ;  $k$  – коефіцієнт передачі,

**Параметри:**

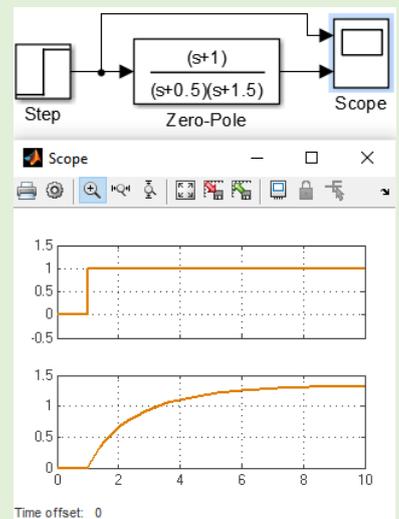
**Zeros** – вектор або матриця нулів.

**Poles** – вектор полюсів.

**Gain** – коефіцієнт передачі. Якщо нулі моделі визначені вектором, то він є скаляром. Якщо нулі задані матрицею – це вектор, при цьому розмір вектору  $k$  визначається числом рядків матриці нулів.

**Absolute tolerance** – абсолютна похибка.

Якщо нулі моделі задані матрицею, то блок Zero-Pole моделює векторну передавальну функцію. Нулі або полюса можуть бути задані комплексними числами. В цьому випадку нулі або полюса повинні бути задані комплексно-сполученими парами полюсів або нулів, відповідно. Початкові умови при використанні блоку Zero-Pole покладаються нульовими.



**State-Space** – блок моделі динамічного об'єкту.

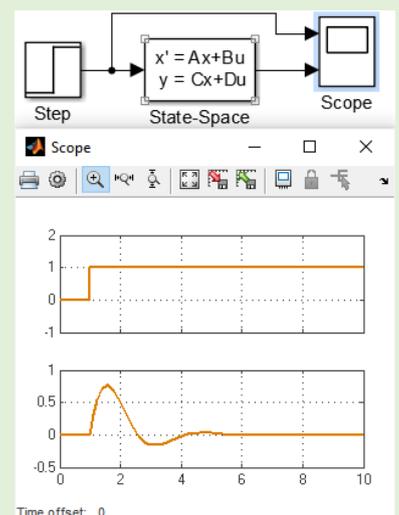
**Призначення:** створює динамічний об'єкт,

описуваний рівняннями в просторі станів:  $\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$ ,

де  $x$  – вектор стану,  $u$  – вектор вхідних впливів,  $y$  – вектор вихідних сигналів,  $A, B, C, D$  – матриці.

**Параметри:**

**A** – матриця системи;



**B** – матриця входу;

**C** – матриця виходу;

**D** – матриця обходу.

**Initial condition** – вектор початкових умов.

**Absolute tolerance** – абсолютна похибка.

**Transport Delay** – блок затримки сигналу.

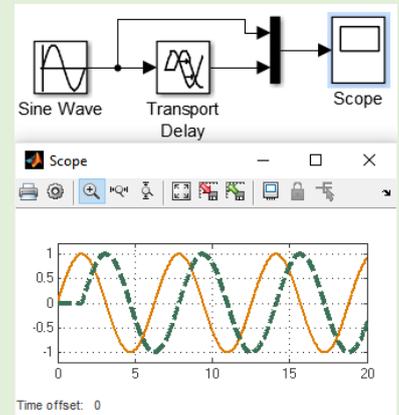
**Призначення:** затримує введення сигналу на певний проміжок часу.

**Параметри:**

**Time delay** – час затримки вхідного сигналу;

**Initial value** – значення виходу, який генерує блок, доки час моделювання не перевищить Time delay;

**Initial buffer size** – визначає початкове виділення пам'яті для кількості точок збереження.



## Math – блоки математических операций

**Gain** – посилювач.

**Призначення:** Виконує множення вхідного сигналу на постійний коефіцієнт.

**Параметри:**

**Gain** – коефіцієнт посилення.

**Multiplication** – спосіб виконання операції.

Може приймати значення (зі списку):

**Element-wise  $K*u$**  – поелементний.

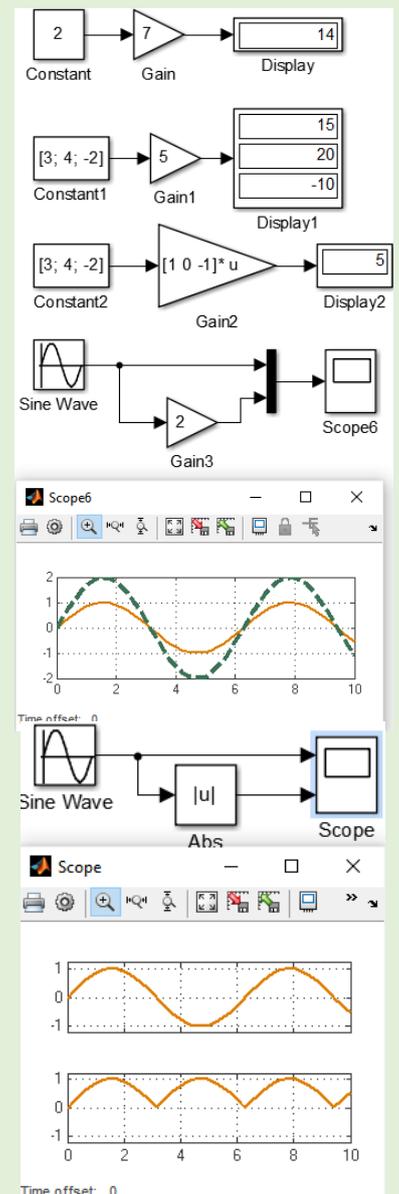
**Matrix  $K*u$**  – матричний коефіцієнт посилення є лівостороннім операндом.

**Matrix  $u*K$**  – матричний коефіцієнт посилення є правостороннім операндом.

**Saturate on integer overflow** (прапорець) – насичення при цілочисловому переповненні.

*Зазначемо*, що блоки посилювачів Gain і Matrix Gain є один і тим самим блоком, але з різними початковими значеннями параметра Multiplication.

**Abs** – блок обчислення модуля.



**Призначення:** Виконує обчислення абсолютного значення величини сигналу.

**Параметри:** зазвичай використовується за замовченням

Тип вхідного сигналу блок Abs може бути комплексним.

**Sign** – блок визначення знаку сигналу.

**Призначення:** Визначає знак вхідного сигналу.

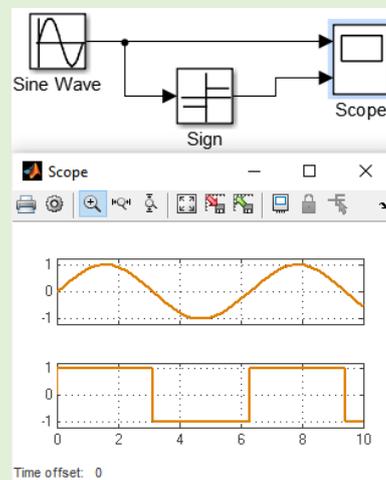
**Параметри:** Немає.

Блок працює у відповідності з наступним алгоритмом:

Якщо вхідний сигнал блоку додатний, то вихідний сигнал дорівнює 1.

Якщо вхідний сигнал блоку від'ємний, то вихідний сигнал дорівнює -1.

Якщо вхідний сигнал блоку дорівнює 0, то вихідний сигнал також дорівнює 0.



**Sum** – блок обчислення суми.

**Призначення:** Виконує обчислення суми поточних значень сигналів.

**Параметри:**

**Icon shape** – форма блоку. Вибирається зі списку.

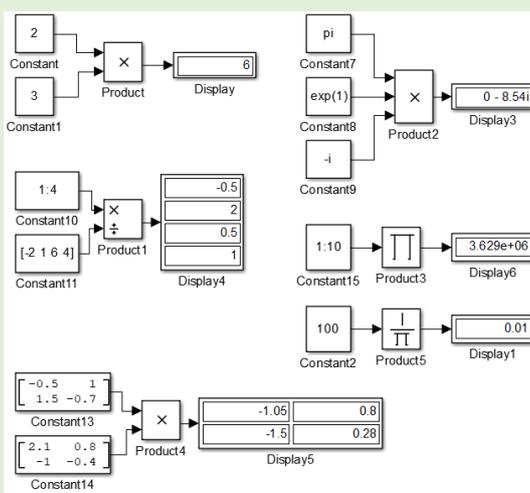
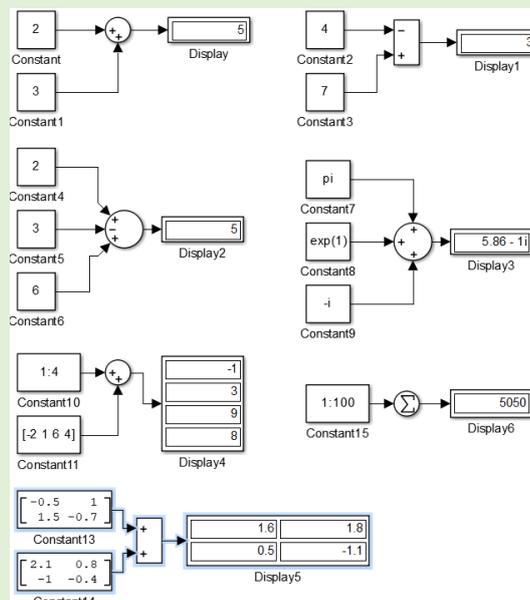
**round** – округність,

**rectangular** – прямокутник.

**List of sign** – список знаків.

У списку використовуються наступні знаки: “+” (плюс), “-” (мінус) та “|” (роздільник знаків). Тут же визначається кількість входів і операція (+ або -) списком знаків параметра **List of sign**.

В якості параметра **List of sign** можна також вказати число входів блоку. В цьому випадку всі входи будуть доданками суми. Якщо кількість входів блоку перевищує 3, то зручніше використовувати блок **Sum** прямокутної форми. Блок може використовуватися для підсумовування скалярних, векторних або матричних сигналів. Типи сигналів, що підсумовуються, повинні збігатися.



**Product** – блок множення.

**Призначення:** Виконує обчислення добутку поточних значень сигналів.

**Параметри:**

**Number of inputs** – кількість входів. Може визначатися як число або як список операторів. Список операторів може містити тільки “\*” (множення) і “/” (ділення).

**Multiplication** – тип операції (вибирається зі списку):

**Element-wise** – поелементне;

**Matrix** – матричне.

Якщо параметр **Number of inputs** заданий списком, що включає крім знаків множення також знаки поділу, то мітки входів будуть позначені символами відповідних операцій. Блок може використовуватися для операцій множення або ділення скалярних векторних або матричних сигналів. Типи вхідних сигналів блоку повинні збігатися. Якщо в якості кількості входів вказати цифру 1 (один вхід), то блок можна використовувати для визначення добутку елементів вектору.

**Math Function** – блок обчислення математичних функцій.

**Призначення:** Виконує обчислення математичної функції.

**Параметри:**

**Function** – вид функції (вибирається зі списку):

**exp** – обчислення експоненти,

**log** – обчислення натурального логарифму,

**log10** – обчислення десяткового логарифму,

**10<sup>u</sup>** – обчислення ступеня 10,

**magnitude<sup>2</sup>** – обчислення квадрата модуля вхідного сигналу,

**square** – обчислення квадрата вхідного сигналу,

**sqrt** – обчислення кореня,

**pow** – обчислення ступеня,

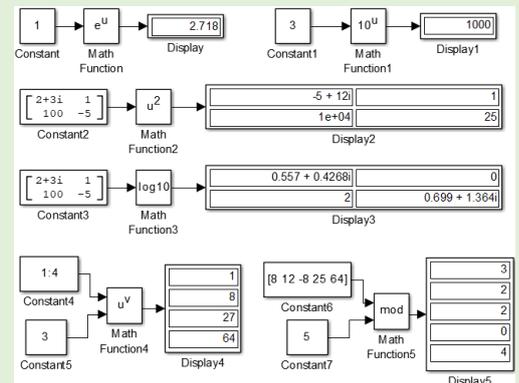
**conj** – обчислення комплексно-сполученого сигналу,

**reciprocal** – обчислення частки від розподілу вхідного сигналу на 1,

**hypot** – обчислення квадратного кореня із суми квадратів вхідних сигналів (гіпотенузи прямокутного трикутника за значеннями катетів),

**rem** – обчислює залишок від ділення першого вхідного сигналу на другий,

**mod** – обчислює залишок від ділення з урахуванням знака,



**transpose** – транспонування матриці,

**hermitian** – обчислення ермітової матриці.

**Output signal type** – тип вихідного сигналу (вибирається зі списку):

**auto** – автоматичне визначення типу

**real** – дійсний сигнал

**complex** – комплексний сигнал.

**Trigonometric Function** – блок обчислення тригонометричних функцій.

**Призначення:** Виконує обчислення тригонометричної та гіперболічних функцій.

**Параметри:**

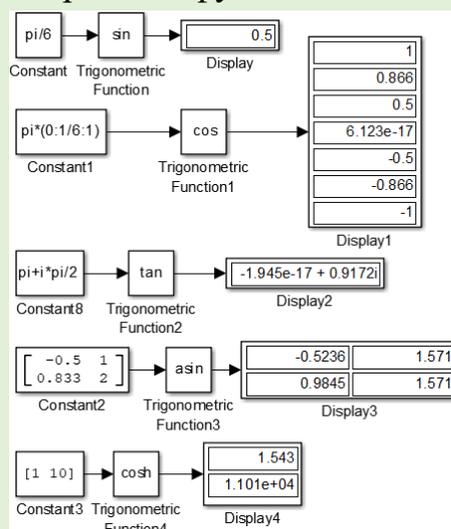
**Function** – вид функції (вибирається зі списку): **sin**, **cos**, **tan**, **asin**, **acos**, **atan**, **atan2**, **sinh**, **cosh** і **tanh**.

**Output signal type** – тип вихідного сигналу (зі списку):

**auto** – автоматичне визначення типу

**real** – дійсний сигнал

**complex** – комплексний сигнал.



При векторному або матричному вхідному сигналі блок виконує поелементне обчислення заданої функції.

**Signal Routing** – блоки перетворення сигналів і допоміжні блоки

**Mux** – мультиплексор (змішувач).

**Призначення:** Об'єднує вхідні сигнали в вектор.

**Параметри:**

**Number of Inputs** – кількість входів.

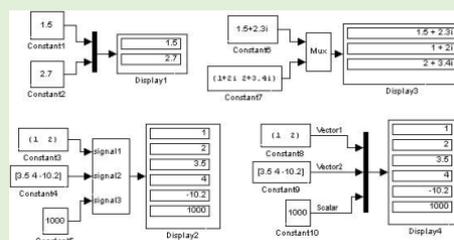
**Display option** – вигляд відображення (зі списку):

**bar** – вертикальний вузький прямокутник.

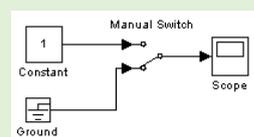
**signals** – прямокутник з білим фоном і відображенням міток вхідних сигналів.

**none** – прямокутник з білим фоном без відображення міток вхідних сигналів.

Вхідні сигнали блоку можуть бути скалярними і (або) векторними. Сигнали, що подаються на входи блоку повинні бути одного типу (дійсного або комплексного).



**Manual Switch** – блок ручного перемикача.



**Призначення:** Виконує перемикання між вхідними сигналами по команді користувача.

## Навчальне видання

### МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з курсу «Прикладне програмне забезпечення» Модуль 3.  
Основи роботи в Simulink для студентів спеціальності 174 «Автоматизація,  
комп'ютерно-інтегровані технології та робототехніка» денної та заочної форми  
навчання

Укладачі:     ГЕРМАН Едуард Євгенович  
                  ДЕМЕНКОВА Світлана Дмитрівна

Відповідальний за випуск проф. Подустов М.О.

Роботу до видання рекомендував: доц. Дудник О. В.

Редактор Н.В. Верстюк

План 2023 р., поз.145 . Підп. до друку 17.02.2023 Формат 60x84 1/16. Папір  
офсетний.ГарнітураTimes.

Ум. друк. арк. 5,9. Наклад 40 прим. Зам. №\_\_ . Ціна договірна.

Видавець НТУ «ХП», 61002, Харків вул. Кирпичова, 2

Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 г.

---

Надруковано з готового оригінал-макета у друкарні ФОП В.В. Петров  
Єдиний державний реєстр юридичних осіб та фізичних осіб – підприємців.

ДК №2480000000106167 від 08.01.2009 р.

61144, м. Харків, вул. Гв. Широнінців, 79в, к. 137, тел. (057)78-17-137.

e-mail: bookfabrik@mail.ua