МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут»

Математичне моделювання на ЕОМ (частина 2)

Методичні вказівки
до виконання комп'ютерних практикумів
для студентів напряму підготовки
6.051003 «Приладобудування»

Рекомендовано Вченою радою
Приладобудівного факультету НТУУ «КПІ»

Київ
НТУУ «КПІ»
2015

Математичне моделювання на ЕОМ (частина 2) [Текст] : метод. вказівки до викон. комп. практикумів для студентів напряму підгот. 6.051003 «Приладобудування» / Уклад.: Ю.Ф. Лазарєв, Д.О. Півторак, С.Л. Лакоза. – К.: НТУУ «КПІ», 2015. – 86 с.

Рекомендовано Вченою радою ПБФ НТУУ «КПІ»
(Протокол № 8/15 від 28.09.2015 р.)

Н а в ч а л ь н о – м е т о д и ч н е в и д а н н я

Математичне моделювання на ЕОМ
(частина 2)
до виконання комп'ютерних практикумів
для студентів напряму підготовки
6.051003 «Приладобудування»

Укладачі: *Лазарєв Юрій Федорович*, канд. техн. наук, доц.
Півторак Діана Олександрівна, канд. техн. наук
Лакоза Сергій Леонідович

Відповідальний редактор *Павловський О.М.*, канд. техн. наук.

Рецензенти: *Вислоух С.П.*, канд. техн. наук, доц.

ЗМІСТ

ВСТУП

Комп'ютерний практикум з дисципліни «Математичне моделювання на ЕОМ» для студентів напряму підготовки 6.051003 «Приладобудування» сприятимуть закріпленню, поглибленню та узагальненню отриманих знань та умінь, а також сприятимуть розвитку навичок самостійної творчої роботи студентів у процесі їх навчання, при дипломному проектуванні, а також у професійній діяльності.

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 2.1

Основи графічної візуалізації розрахунків

Мета роботи: навчитися побудові графіків за допомогою вбудованих процедур у середовищі MATLAB.

Теоретичні відомості

*Побудова графіків з використанням процедури **plot***

Виведення графіків у системі MATLAB є настільки простою і зручною процедурою, що її можна використовувати навіть при обчисленнях у режимі калькулятора.

Головною функцією, що забезпечує побудову графіків на екрані дисплея, є функція ***plot***. Загальний формат звернення до цієї функції такий:

plot(x1,y1,x2,y2,s2,…),

де x1, y1 – відомі вектори, елементами яких є масиви значень аргументу x1 та функції y1, що відповідають першій кривій графіка; x2, y2 – масиви значень аргументу і функції другої кривої і т. ін. при цьому вважається, що значення аргументу відкладаються вдовж горизонтальної осі графіка, а значення функції – вздовж вертикальної осі. Змінні s1, s2,… є символьними (вказання їх не є обов'язковим). Кожна з них може містити до трьох спеціальних символів, які визначають відповідно:

- тип лінії, що з'єднує окремі точки графіка;
- тип точки графіка;
- колір лінії.

Якщо змінні s не вказані, то тип лінії за замовчуванням – відрізок прямої, тип точки – пік сель, а колір встановлюється за таким чергуванням: - жовтий, зелений, фіолетовий, блакитний, червоний, зелений, синій, білий та чорний – залежно від того, яка по черзі лінія виводиться на графік виводиться на графік. Наприклад, звернення вигляду ***plot(***x1,y1,x2,y2, …) приведе до побудови графіка, у якому перша крива буде лінією з відрізків прямих жовтого кольору, друга – такого ж типу фіолетовою лінією і т.ін.

Графіки у MATLAB завжди виводяться в окреме (графічне) вікно, яке називається фігурою.

Приклад. Нехай потрібно вивести графік функції

$$y = 3\sin\left(x + \pi/3\right)$$

на відрізку від -3π до $+3\pi$ з кроком $\pi/100$.

Спочатку потрібно сформувати масив значень аргументу x:

$$x = -3 * pi : pi/100 : 3 * pi,$$

потім обчислити масив відповідний значень функції:

$$y = 3 * \sin\left(x + pi/3\right)$$

і, нарешті, побудувати графік залежності $y(x)$.

У цілому в командному вікні ця послідовність операцій буде мати такий вигляд:

```
>> x=-3*pi:pi/100:3*pi;
>> y=3*sin(x+pi/3);
>> plot(x,y)
```

У результаті на екрані з'явиться додаткове вікно з графіком (рис.1.1,а).

Якщо вектор аргументу при звернені до функції ***plot*** не вказано явно, то система обирає як аргумент номер елемента вектора функції.

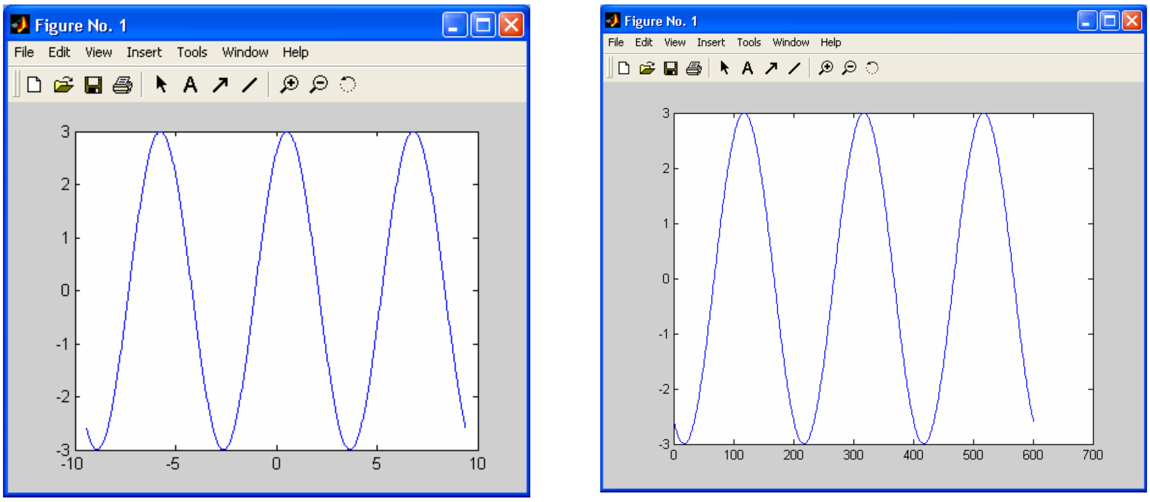
Наприклад, якщо ввести команду

```
>> plot(y)
```
,

то результатом буде поява графіка у вигляді, наведеному на рис.1.1,б.

Графіки, на рис. 1.1,а та 1.1,б мають кілька недоліків:

- на них не нанесено сітку з координатних ліній, що ускладнює «читання» графіків;
- немає загальної інформації про криві графіка (заголовки);
- невідомо, які величини відкладено по осях графіка.



а) б)

Рис.1.1 Вікна з прикладом побудови графіка функції $y = 3\sin(x + \pi/3)$

Перший недолік усувається за допомогою функції ***grid***. Якщо цю функцію записати одразу після звернення до функції plot:

```
>> x=-3*pi:pi/100:3*pi;
>> y=3*sin(x+pi/3);
>> plot(x,y), grid,
```

то графік буде споряджений координатною сіткою (рис.1.2, а).

Цінною особливістю графіків, побудованих у системі MATLAB, є те, що сітка координат завжди відповідає «цілим» кроком змінювати, що робить графіки «читабельними», тобто за графіком можна відлічувати значення функції за будь-яким заданим значенням аргументу і навпаки. Такої властивості не має жоден з графічних пакетів-додатків до мов програмування високого рівня.

Заголовок графіка виводиться за допомогою процедури ***title.*** Якщо після звернення до процедури ***plot*** звернутися до ***title*** таким чином:

title(‘<текст>’),

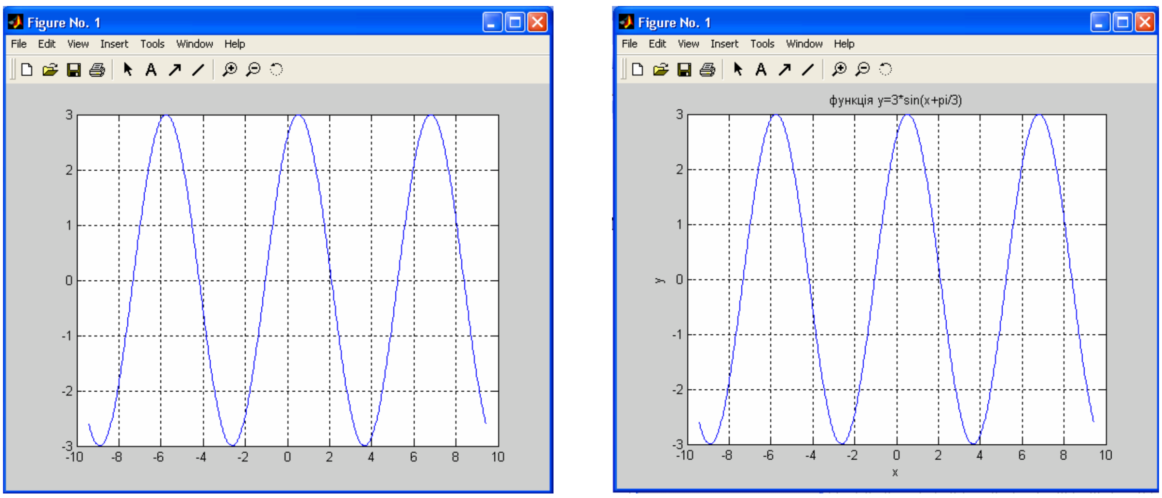
то зверху у полі фігури з’явиться текст, який записано між дужками. При цьому слід пам’ятати, що текст завжди повинен записуватися між апострофами.

Аналогічно можна вивести пояснення до графіка, що виводиться вдовж горизонтальної осі (функція ***xlabel***) і вдовж вертикальної осі (функція ***ylabel***).

Наприклад:

```
>> x=-3*pi:pi/100:3*pi;
>> y=3*sin(x+pi/3);
>> plot(x,y), grid
>> title ('функція y=3*sin(x+pi/3)');
>> xlabel('x'); ylabel('y');
```

Приведе до оформлення поля фігури у вигляді, наведеному на рис. 1.2,б. очевидно, така форма вже повністю задовольняє вимоги, які ставляться до інженерних графіків.



а) б)

Рис.1.2 Вікна з прикладом побудови графіка функції $y = 3\sin\left(x + \pi/3\right)$

Завдання до виконання комп'ютерного практикуму

Побудуйте в графічному вікні MATLAB графік функції відповідно до варіанту згідно табл.1.1. Зробіть текстове оформлення графіка.

Табл. 1.1 Варіанти індивідуальних завдань

Варіант	$f(x)$	a	b	h
1	2	3	4	5
1	$\dfrac{x^2}{1+0,25\sqrt{x}}$	1,1	3,1	0,2
2	$\dfrac{x^3-0,3x}{\sqrt{1+2x}}$	2,05	3,05	0,1
3	$\dfrac{2e^{-x}}{2\pi+x^3}$	0	1,6	0,16
4	$\dfrac{\cos\pi x^2}{\sqrt{1-3x}}$	-1	0	0,1
5	$\sqrt{1+4x}\sin\pi x$	0,1	0,8	0,07
6	$\dfrac{e^{x/3}}{1+x^2}$	1,4	2,4	0,1
7	$e^{-2x}+x^2-1$	0,25	2,25	0,2
8	$(e+x)\sin(\pi\sqrt{x-1})$	1,8	2,8	0,1
9	$\sqrt{3+2x}\cdot tg\dfrac{\pi x^3}{2}$	0,1	0,9	0,08
10	$\sqrt{2+3x}\cdot\ln(1+3x^2)$	-0,1	0,9	0,1
11	$\sqrt[3]{x^2+3}\cdot\cos\dfrac{\pi x}{2}$	1	2,5	0,15

1	2	3	4	5
12	$(4+7x)\sin(\pi^3\sqrt{1+x})$	0	7	0,7
13	$e^{-x^2}(1+3x-x^2)$	0	2	0,2
14	$x^3-3x+\dfrac{8}{\sqrt{1+x^2}}$	0	1,7	0,17
15	$\sqrt{sh\sqrt{2\pi x}},\ \left(sh\,x=\dfrac{e^x-e^{-x}}{2}\right)$	0	1,2	0,12
16	$\sqrt{ch\dfrac{x}{\sqrt{2x}}},\ \left(ch\,x=\dfrac{e^x+e^{-x}}{2}\right)$	0,5	1,5	0,1
17	$\dfrac{x^3+2x}{\sqrt{1+e^x}}$	-0,2	0,8	0,1
18	$\sqrt{1+2x^2}\cdot\sin\dfrac{3x}{2}$	2	4	0,2
19	$\sqrt{3x^2+5}\cdot\cos\dfrac{\pi x}{2}$	0,5	1,5	0,1
20	$\arccos\ e^{-\sqrt[3]{3x}}$	0,2	0,5	0,03
21	$\arcsin\ e^{-x^2/5}$	8	13	0,5
22	$x+\ln(x+\sqrt{1+x^2})$	-0,5	0,5	0,1
23	$\dfrac{1+e^{-x/2}}{\sqrt{3x^2+1}}$	3	5	0,2
24	$3x^3+\dfrac{1}{x}+e^{-2x^2}$	1,2	2,2	0,1
25	$x^{2x+1}+x^3-2x$	1	5	0,4

Контрольні запитання

1. Які функції MATLAB здійснюють виведення графіків на екран?
2. Яким чином можна змінити тип лінії при використанні функції ***plot***?
3. Якими функціями забезпечується супровід графіка координатними лініями і надписами?
4. Яка функція здійснює виведення заголовку графіка?
5. Яка функція здійснює виведення пояснення до графіка вздовж горизонтальної та вертикальної осей?
6. Де можна ознайомитися з розширеним описом функції ***plot?***

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 2.2

Побудова графіка у логарифмічному масштабі.
Побудова амплітудно-частотної та фазочастотної характеристик

Мета роботи: навчитися будувати графіки в логарифмічному масштабі за допомогою вбудованих процедур у середовищі MATLAB.

Теоретичні відомості

MATLAB має кілька функцій, що дозволяють будувати графіки у логарифмічному масштабі.

Функція logspace зі зверненням

x = ***logspace***(d1, d2, n)

формує вектор-рядок «x», що містить «n» рівновіддалених у логарифмічному масштабі одна від одної точок, які покривають діапазон від 10^{d1} до 10^{d2}.

Функція ***loglog*** є повністю аналогічною функції ***plot***, але графіки по обох осях будуються у логарифмічному масштабі.

Для побудови графіків, які використовують процедури ***semilogx*** та ***semilogy***. Перша процедура будує графіки з логарифмічним масштабом вдовж горизонтальної осі, друга – вдовж вертикальної осі.

Звернення до останніх трьох процедур повністю аналогічне зверненню до функції ***plot***.

Як приклад розглянемо побудову графіків амплітудно-частотної і фазо-частотної характеристик ланки, що описується передатною функцією:

$$W(p) = \frac{p+4}{p^2 + 4 \cdot p + 100}.$$

Для цього слід, по-перше, створити вектори-поліноми чисельника

$$Pc = [1 \quad 4]$$

і знаменника передатної функції

$$Pz = [1 \quad 4 \quad 100].$$

По-друге, визначити корені цих двох поліномів:

```
>> P1=[1 4]; P2=[1 4 100];
>> roots(P1)
ans =
    -4
>> roots(P2)
ans =
                         -2 +      9.79795897113271i
                         -2 -      9.79795897113271i
```

По-третє, задати діапазон змінювання частоти таким чином, щоб він охоплював усі знайдені корені:

```
om=1e-2;   omk=1e2
```

Тепер треба задатися кількістю точок на майбутньому графіку:

n=41,

і сформулювати масив точок за частотою

OM=logspace(-2,2,41),

де значення -2 і +2 відповідають десятковим порядкам початкового om0 і кінцевого omk значень частоти.

Користуючись функцією ***polyvar,*** можна обчислити спочатку вектор комплексних значень «ch» чисельника частотної передатної функції, яка відповідає заданій передатній функції, якщо як аргумент функції ***polyvar*** використати сформований вектор частот OM, елементи якого домножені на явну одиницю. Аналогічно обчислюється комплексно значний вектор «zn» знаменника ЧПФ (частотна передатна функція).

Вектор АЧХ (амплітудно-частотна характеристика) можна знайти, обчислюючи модулі векторів чисельника і знаменника ЧПФ і ділячи поелементно отримані вектори. Щоб знайти вектор значень ФЧХ (фазо-частотної характеристики) треба поділити поелементно комплексно значні вектори чисельника і знаменника ЧПФ і визначити вектор аргументів елементів отриманого вектора. Для того, щоб фази подати у градусах, отримані результати слід домножити на 180 і поділити на π.

Нарешті, для побудови графіка АЧХ у логарифмічному масштабі, достатньо застосувати функцію ***loglog***, а для побудови ФЧХ зручніше користуватися функцією ***semilogx***.

У цілому послідовність дій може бути такою:

```
>> loglog(OM,ACH); grid; title('Амплітудно-частотна характеристика');
>> xlabel('Частота (рад/с)'); ylabel('Відношення амплітуд');
>> loglog(OM,ACH); grid; title('Амплітудно-частотна характеристика')...
, xlabel('Частота (рад/с)'); ylabel('Відношення амплітуд')
>> OM=logspace(-2,2,40);
>> ch=polyval(P1,i*OM);
>> zn=polyval(P2,i*OM);
>> ACH=abs(ch)./abs(zn);
>> loglog(OM,ACH); grid; title('Амплітудно-частотна характеристика')...
, xlabel('Частота (рад/с)'); ylabel('Відношення амплітуд')
>> FCH=angle(ch./zn)*180/pi;
>> semilogx(OM,FCH); grid; title('Фазо-частотна характеристика')...
, xlabel('Частота (рад/с)'); ylabel('Фаза (градуси)')
```

В результаті отримуються графіки, наведені на рис.2.1 та 2.2.

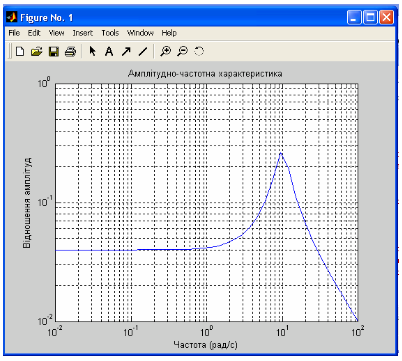


Рис. 2.1 Вікно з графіком амплітудно-частотної характеристики

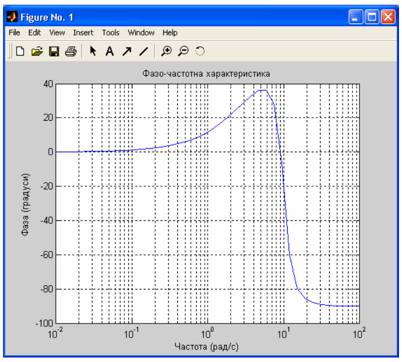


Рис. 2.2 Вікно з графіком фазочастотної характеристики

Звичайно графіки, які отримані за допомогою функції ***plot***, ***loglog***, ***semilogx*** і ***semilogy***, автоматично будуються по осях у таких масштабах, щоб у полі графіка вміщувалися усі обчислювальні точки графіка, включаючи максимальні і мінімальні значення аргументу і функції. Але MATLAB має можливості і встановлення режимів масштабування. Це досягається за рахунок використання процедури ***axis***.

Команда ***axis***([xmin xmax ymin ymax]) встановлює жорсткі межі поля графіка у одиницях величин, що відкладаються по осях.

Команда ***axis***(‘auto’) повертає масштаби по осях до штатних значень (прийнятих за замовчуванням).

Команда ***axis***(‘ij’) переміщує початок відліку у лівий верхній кут і реалізує відлік з верхнього лівого кута (матрична система координат).

Команда ***axis***(‘xy’) повертає декартову систему координат з початком відліку у лівому нижньому куті.

Команда ***axis***(‘square’) встановлює однаковий діапазон змінювання змінних по осях графіка.

Команда ***axis***(‘equal’) забезпечує однаковий масштаб по осях графіка.

В основному графічному вікні можна побудувати кілька графіків з власними полями за допомогою процедури ***subplot***. Звернення до цієї процедури повинно передувати зверненню до процедури ***plot***, ***loglog***, ***semilogx*** і ***semilogy*** і мати такий вигляд:

subplot(m,n,p),

де m – вказує, на скільки частин поділяється графічне вікно по вертикалі; n – по горизонталі, а p – номер підвікна, в якому будуватиметься графік. При цьому підвікна нумеруються зліва направо порядково зверху вниз.

Наприклад, два попердніх графіки можна вмістити у одному графічному вікні таким чином:

```
>> subplot(2,1,1);
>> loglog(OM,ACH); grid; title('Амплітудно-частотна характеристика')...
, xlabel('Частота (рад/с)'); ylabel('Відношення амплітуд')
>> subplot(2,1,2);
>> semilogx(OM,FCH); grid; title('Фазо-частотна характеристика')...
, xlabel('Частота (рад/с)'); ylabel('Фаза (градуси)')
```

Результат подано на рис. 2.3.

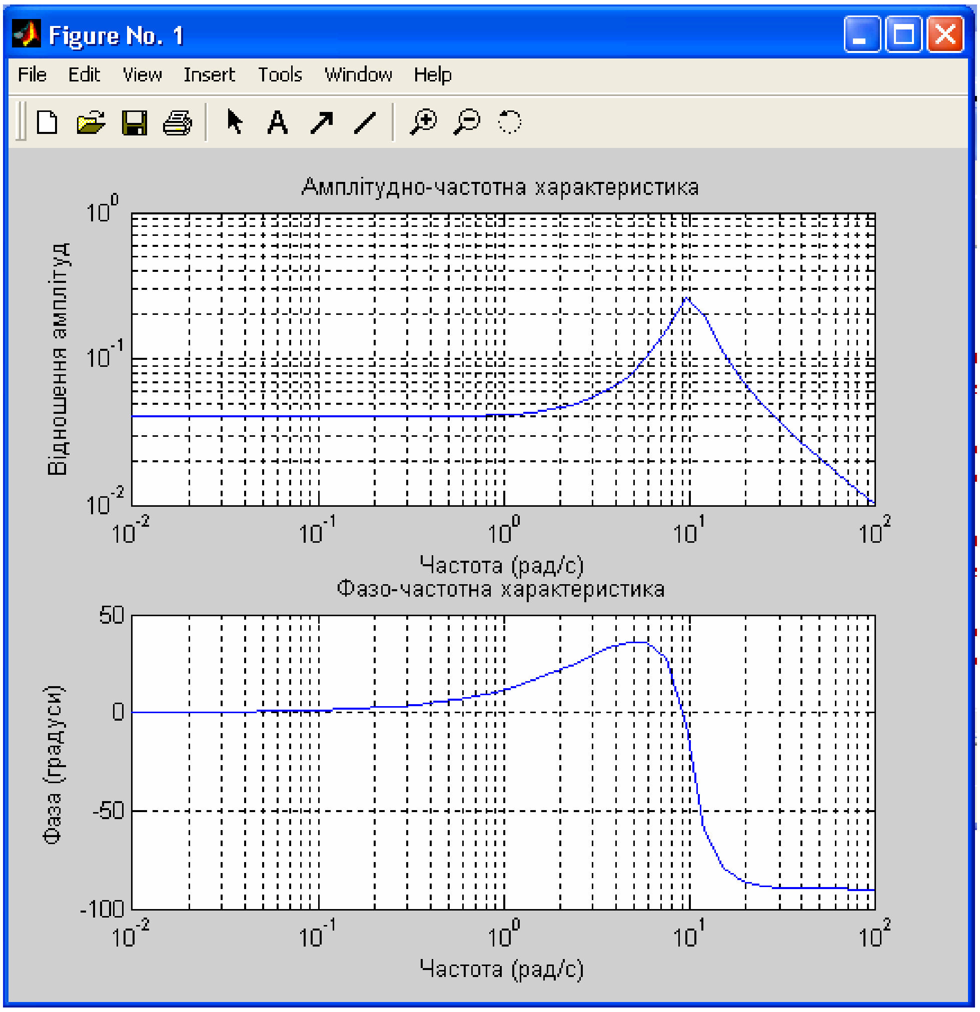


Рис. 2.3 Вікно з графіками амплітудно-частотної та фазочастотної характеристик

Команда ***text***(x,y,’<текст>’) дозволяє розташувати вказаний текст га полі графіка, при цьому початок текста розміщується у точці з координатами x і y. При цьому значення вказаних координат повинні знаходиться усередені діапазону змінювання величин, що відкладаються по осях графіка.

Більш зручним для розміщення всередині поля графіка є використання команди ***gtext***(‘<текст>’), яка висвічує в активному графічному вікні перехрестя, переміщування якого за допомогою «мишки» дозволяє вказати місце початку виведення вказаного тексту. Після цього натисканням кнопки «мишки» або будь-якої клавіші вводимо текст у вказане місце:

```
>> gtext(‘ФЧХ’)
>>subplot(2,1,1);
>>gtext(‘АЧХ’) .
```

Щоб утворити кілька графічних вікон, у кожному з яких розміщуються відповідні графіки, можна використати команду ***figure***, яка утворить наступне графічне вікно, залишаючи попередні.

Завдання до виконання комп’ютерного практикуму

Побудуйте в графічному вікні MATLAB графіки амплітудно-частотної і фазочастотної характеристик функції. Зробіть текстове оформлення графіка.

Табл. 2.1 Варіанти індивідуальних завдань

Варіант	Чисельник	Знаменник
1	2	3
1	$1.82p + 67.56$	$p^4 + 2.65p^3 + 3.09p^2 + 7.04p + 34.05$
2	$4.61p^2 + 1.82p + 67.56$	$p^4 + 3.65p^3 + 45p^2 + 7.04p + 125$
3	$p^2 + 4p + 23$	$p^4 + 2p^3 + 39p^2 + 2p + 45$
4	$3p^2 + 1.82p + 67.56$	$p^2 + 7.04p + 34.05$

1	2	3
5	$p+6$	$p^2+0.7p+48$
6	$p^3+4.61p^2+1.82p$	$2.65p^3+3p^2+4p+87$
7	$p^3+4.61p^2+1.82p+67.56$	$p^4+2.65p^3+68p^2+5p+34$
8	$4.61p^2+68$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
9	7.56	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
10	$p^3+1.8p+7$	$p^4+6.5p^3+39p^2+7p+45$
11	$p^3+4.61p^2+1.82p+67.56$	$p^3+3.09p^2+70p+34$
12	$p^2+1.8p+78$	$2.65p^3+3.09p^2+7.04p+34.05$
13	$p^3+1.82p+67.56$	$p^4+2.6p^3+3p^2+4p+34$
14	$p^3+4.61p^2+1.82p+67.56$	$7p^2+7p+34$
15	$4.61p^2+1.82p+67.56$	$p^2+7.04p+560$
16	$1.82p+67.56$	$3.09p^2+7.04p+34.05$
17	p^3	$3.09p^2+7.8p+125$
18	$1.82p$	$p^3+3.09p^2+7.04p+34.05$
19	$4.61p^2$	$p^2+7.04p+34.05$
20	$p^3+67.56$	$p^4+2.65p^3+3.09p^2+7.04p+34.05$
21	p^3	$p^4+2p^3+3p^2+12p+100$
22	$p^3+4.61p^2+1.82p+67.56$	$p^4+5p^3+30p^2+7p+305$
23	$p^2+1.82p+67.56$	$p^4+2p^3+9p^2+4p+35$
24	$p^3+61p^2+182p+67$	$p^4+3p^3+9p^2+0.04p+39$
25	$p^2+1.82p+67.56$	$p^4+5p^3+20p^2+7p+34$

Контрольні запитання

1. Які функції здійснюють виведення графіків на екран?
2. Якими функціями забезпечується супровід графіка координатними лініями та надписами?
3. Чи можна побудувати кілька графіків в одній системі координат і водному графічному вікні?
4. Як вивести кілька окремих графіків в одному графічному вікні?
5. Які дії виконуються за допомогою функції text, axis, gtext? Як ними користуватися?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 2.3

Апроксимація та інтерполяція

Мета роботи: навчитися використовувати функції апроксимації та інтерполяції, які вбудовані у MATLAB.

Теоретичні відомості

Система MATLAB надає зручні процедури для апроксимування та інтерполювання даних вимірювань.

Поліноміальне апроксимування даних вимірювань, які сформовані як деякий вектор Y, за деяких значень аргументу, котрі утворюють вектор X тієї ж довжини, що й вектор Y, здійснюється процедурою ***polyfit***(X,Y,n), де n – порядок апроксимуючого полінома. Результатом дії цієї процедури є вектор довжиною (n+1) коефіцієнтів апроксимуючого полінома.

Нехай масив значень аргументу є таким:

x=[1 2 3 4 5 6 7 8],

а масив відповідних значень вимірювальної величини – наступним:

y=[-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1].

Тоді, застосовуючи згадану функцію за різних значень порядку апроксимуючого полінома, отримуємо:

```
>> x=[1 2 3 4 5 6 7 8];
>> y=[-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1];
>> polyfit(x,y,1)
ans =
        0.114285714285714       -0.239285714285714
>> polyfit(x,y,2)
ans =
       -0.102380952380952         1.03571428571429                   -1.775
>> polyfit(x,y,3)
ans =
  Columns 1 through 3
       0.0176767676767678       -0.341017316017318         1.94606782106783
  Column 4
        -2.65000000000002

>> polyfit(x,y,4)
ans =
  Columns 1 through 3
     -0.00435606060606057       0.0960858585858584       -0.814583333333337
  Columns 4 through 5
         3.03259379509382        -3.38928571428575
```

Це означає, що дану залежність можна апроксимувати або прямою

$$y(x) = 0.1143x - 0.2393,$$

або квадратною параболою

$$y(x) = -0.1024x^2 + 1.0357x - 1.7750,$$

або кубічною параболою

$$y(x) = 0.0177x^3 - 0.3410x^2 + 1.9461x - 2.6500,$$

або параболою четвертого ступеня

$$y(x) = -0.0044x^4 + 0.0961x^3 - 0.8146x^2 + 3.0326x - 3.3893.$$

Кубічна інтерполяція функції однієї змінної здійснюється функцією ***icubic***(Y,Xi). Ця функція інтерполює значення вектора Y у точках Xi всередині області визначення функції, використовуючи кубічні поліноми. При цьому обчислюється вектор значень інтерполюючої залежності за значень аргументу, що вказані вектором Xi. При такому зверненні за замовчуванням вважається, що аргументом функції є номер відповідного елемента вектора Y, тобто областю визначення функції є проміжок між 1 і максимальним номером елемента заданого вектора Y.

Якщо ж значення аргументу не відповідають цим умовам, то вони повинні бути задані як вектор значень аргументу X і звернення до функції має бути дещо іншим:

$$\textbf{\textit{icubic}}(X,Y,Xi)$$

Наведемо приклад,

```
>> x=[1 2 3 4 5 6 7 8];
>> y=[-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1];
>> x=-0.5:0.1:0.2;
>> y=[-1.1 0.2 0.5 0.8 0.7 0.6 0.4 0.1];
>> y1=icubic(x,y,x1);
>> plot(x,y,x1,y1), grid
```

На рис. 3.1 зображено дві криві: перша – з'єднані відрізками прямих точки заданих векторів Y(x), друга – результат інтерполяції за більш щільних значень аргументу.

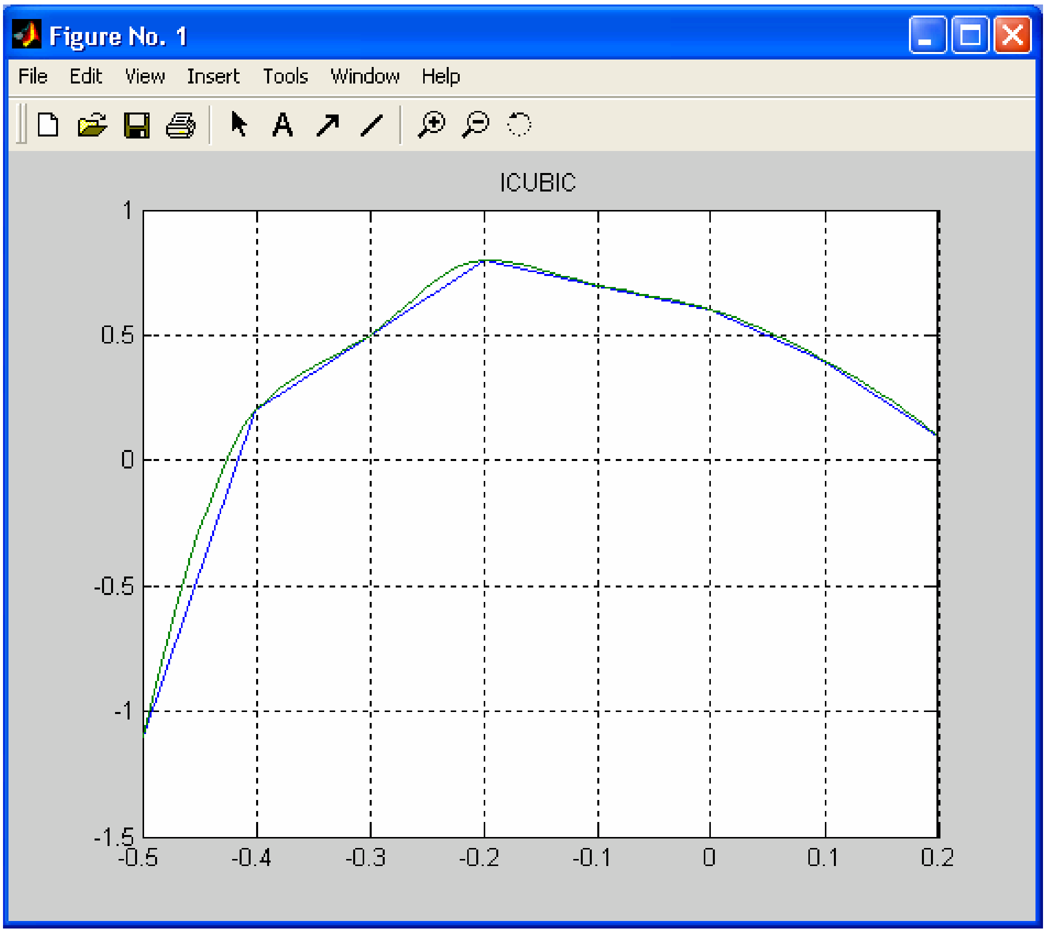


Рис. 3.1 Вікно з прикладом застосування інтерполяції

Функція ***spline***(X,Y,Xi) здійснює інтерполювання кубічними сплайнами. Як приклад розглянемо інтерполяцію попереднього вектора:

```
>> y2=spline(x,y,x1);
>> plot(x,y,x1,y2); grid; title('SPLINE')
```

Результат наведено на рис. 3.2.

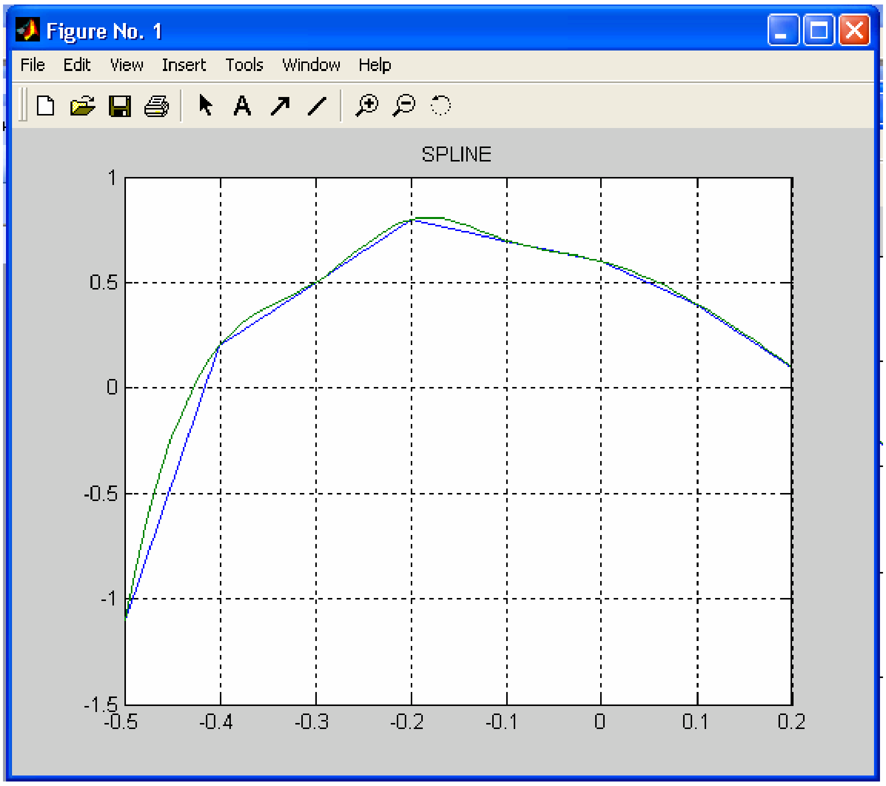


Рис. 3.2 Вікно з прикладом застосування інтерполювання кубічними сплайнами

Усі попередні процедури охоплюються єдиною процедурою ***interp***1(X,Y,Xi), яка здійснює одновимірну табличну інтерполяцію. Звернення до неї у загальному випадку має вигляд:

interp1(X,Y,Xi,'<метод>')

і дозволяє додатково вказати метод інтерполяції:

'linear' – лінійна,

'cubic' – кубічна,

'spline' – кубічні сплайни.

Якщо метод не вказано, здійснюється за замовчуванням лінійна інтерполяція. Наприклад, (для того ж вектора):

```
>> y1=interp1(x,y,x1);
>> y2=interp1(x,y,x1,'cubic');
>> y3=interp1(x,y,x1,'spline');
>> plot(x1,y1,x1,y2,x1,y3); grid; title('INTERP1 (linear, cubic, spline)')
```

Результат наведено на рис. 3.3.

Рис.3.3 Вікно з прикладом застосування процедури ***interp1***

Завдання на виконання комп'ютерного практикуму

Завдання 1. За допомогою функції ***polyfit*** провести поліноміальне апроксимування даних, значення векторів X та Y обираються з табл. 3.1 відповідно до варіанту.

Табл. 3.1 Варіанти індивідуальних завдань

Варіант	Вектор X	Вектор Y
1	2	3
1	$[1\ \ 3\ \ 6\ \ 8\ \ 9\ \ 12\ \ 15\ \ 17]$	$[-0.5\ \ -0.01\ \ 0\ \ 1.2\ \ 1.6\ \ 1.8\ \ 2.2\ \ 2.8]$
2	$[1\ \ 2\ \ 3\ \ 6\ \ 9\ \ 10\ \ 11\ \ 12]$	$[-1\ \ -0.8\ \ -0.4\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1.2]$
3	$[1\ \ 3\ \ 4\ \ 6\ \ 8\ \ 9\ \ 11\ \ 15]$	$[-2\ \ -1.5\ \ -1.2\ \ -0.8\ \ -0.2\ \ 0.2\ \ 0.3\ \ 1]$
4	$[1\ \ 3\ \ 5\ \ 8\ \ 11\ \ 12\ \ 15\ \ 18]$	$[0\ \ 0.2\ \ 0.4\ \ 0.6\ \ 0.8\ \ 0.9\ \ 1\ \ 1.2]$
5	$[1\ \ 2\ \ 3\ \ 5\ \ 7\ \ 8\ \ 9\ \ 10]$	$[-0.3\ \ -0.2\ \ -0.1\ \ 0\ \ 0.2\ \ 0.4\ \ 0.6\ \ 0.8]$
6	$[1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 8\ \ 9\ \ 12]$	$[0\ \ 0.2\ \ 0.4\ \ 0.6\ \ 0.9\ \ 1\ \ 1.2\ \ 1.5]$
7	$[1\ \ 2\ \ 3\ \ 5\ \ 7\ \ 8\ \ 9\ \ 10]$	$[-2\ \ -1.5\ \ -1.2\ \ -0.8\ \ -0.2\ \ 0.2\ \ 0.3\ \ 1]$
8	$[1\ \ 3\ \ 6\ \ 8\ \ 9\ \ 12\ \ 15\ \ 17]$	$[-1\ \ -0.8\ \ -0.4\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1.2]$
9	$[1\ \ 2\ \ 6\ \ 7\ \ 9\ \ 13\ \ 15\ \ 17]$	$[-0.5\ \ -0.04\ \ 0\ \ 1.2\ \ 1.6\ \ 1.9\ \ 2.2\ \ 2.8]$
10	$[1\ \ 2\ \ 5\ \ 6\ \ 7\ \ 9\ \ 10\ \ 12]$	$[0\ \ 0.1\ \ 0.3\ \ 0.5\ \ 0.7\ \ 0.9\ \ 1\ \ 1.2]$
11	$[1\ \ 2\ \ 3\ \ 5\ \ 7\ \ 8\ \ 9\ \ 12]$	$[-2\ \ -1.5\ \ -1.2\ \ -0.8\ \ -0.2\ \ 0.2\ \ 0.3\ \ 1]$
12	$[1\ \ 2\ \ 3\ \ 5\ \ 7\ \ 8\ \ 9\ \ 10]$	$[-1\ \ -0.8\ \ -0.4\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1.2]$
13	$[1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8]$	$[-0.1\ \ -0.01\ \ 0\ \ 1.2\ \ 1.6\ \ 1.7\ \ 2.2\ \ 3]$

1	2	3
14	$[1\ \ 3\ \ 5\ \ 8\ \ 9\ \ 13\ \ 15\ \ 17]$	$[-1\ \ -0.8\ \ -0.4\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1.2]$
15	$[1\ \ 2\ \ 3\ \ 6\ \ 7\ \ 9\ \ 13\ \ 15]$	$[-2\ \ -1.8\ \ -1.2\ \ -0.8\ \ -0.2\ \ 0.3\ \ 0.5\ \ 1]$
16	$[1\ \ 2\ \ 5\ \ 7\ \ 9\ \ 13\ \ 15\ \ 17]$	$[0\ \ 0.3\ \ 0.4\ \ 0.7\ \ 0.8\ \ 0.9\ \ 1\ \ 1.4]$
17	$[1\ \ 2\ \ 6\ \ 7\ \ 9\ \ 11\ \ 12\ \ 13]$	$[-2\ \ -1.5\ \ -1\ \ -0.8\ \ -0.2\ \ 0.3\ \ 0.5\ \ 1]$
18	$[2\ \ 3\ \ 4\ \ 5\ \ 7\ \ 8\ \ 9\ \ 12]$	$[-1\ \ -0.8\ \ -0.4\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1.2]$
19	$[1\ \ 3\ \ 5\ \ 6\ \ 7\ \ 9\ \ 10\ \ 12]$	$[-0.2\ \ -0.1\ \ 0\ \ 1.2\ \ 1.6\ \ 1.8\ \ 2.5\ \ 2.8]$
20	$[1\ \ 2\ \ 6\ \ 8\ \ 10\ \ 14\ \ 15\ \ 18]$	$[-1\ \ -0.5\ \ -0.3\ \ -0.2\ \ 0\ \ 0.2\ \ 0.5\ \ 1]$
21	$[1\ \ 2\ \ 3\ \ 5\ \ 6\ \ 8\ \ 9\ \ 12]$	$[-3\ \ -2\ \ -1.5\ \ -1\ \ -0.8\ \ 0\ \ 0.8\ \ 1]$
22	$[1\ \ 2\ \ 4\ \ 6\ \ 7\ \ 11\ \ 14\ \ 16]$	$[-3\ \ -2\ \ -1.2\ \ -0.8\ \ 0\ \ 0.2\ \ 0.3\ \ 1]$
23	$[1\ \ 2\ \ 3\ \ 4\ \ 7\ \ 8\ \ 9\ \ 10]$	$[-0.9\ \ -0.5\ \ 0\ \ 0.1\ \ 0.2\ \ 1\ \ 2\ \ 3]$
24	$[1\ \ 2\ \ 3\ \ 4\ \ 7\ \ 8\ \ 9\ \ 12]$	$[0\ \ 0.1\ \ 0.8\ \ 1\ \ 1.8\ \ 1.9\ \ 2\ \ 2.2]$
25	$[1\ \ 3\ \ 4\ \ 7\ \ 8\ \ 12\ \ 15\ \ 17]$	$[-0.5\ \ -0.2\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1\ \ 1.8]$

2. За допомогою функції ***icubic*** провести кубічну інтерполяцію даних, значення векторів X та Y обираються з табл. 3.2 відповідно до варіанту.

Табл. 3.2 Варіанти індивідуальних завдань

Варіант	Вектор X та X1	Вектор Y
1	2	3
1	$-0.5:0.01:0.8$	$[-0.5\ \ -0.01\ \ 0\ \ 1.2\ \ 1.6\ \ 1.8\ \ 2.2\ \ 2.8]$
2	$0.1:0.01:1.8$	$[-1\ \ -0.8\ \ -0.4\ \ 0\ \ 0.2\ \ 0.6\ \ 0.8\ \ 1.2]$
3	$-0.5:0.02:0.3$	$[-2\ \ -1.5\ \ -1.2\ \ -0.8\ \ -0.2\ \ 0.2\ \ 0.3\ \ 1]$
4	$-0.8:0.01:0.5$	$[0\ \ 0.2\ \ 0.4\ \ 0.6\ \ 0.8\ \ 0.9\ \ 1\ \ 1.2]$

11	2	3
5	$1.5:0.02:2.8$	$[-0.3 \quad -0.2 \quad -0.1 \quad 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8]$
6	$-1:0.05:0.4$	$[0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.9 \quad 1 \quad 1.2 \quad 1.5]$
7	$-0.5:0.01:0.3$	$[-2 \quad -1.5 \quad -1.2 \quad -0.8 \quad -0.2 \quad 0.2 \quad 0.3 \quad 1]$
8	$-0.6:0.02:0.6$	$[-1 \quad -0.8 \quad -0.4 \quad 0 \quad 0.2 \quad 0.6 \quad 0.8 \quad 1.2]$
9	$0:0.01:1.9$	$[-0.5 \quad -0.04 \quad 0 \quad 1.2 \quad 1.6 \quad 1.9 \quad 2.2 \quad 2.8]$
10	$0.5:0.05:2.5$	$[0 \quad 0.1 \quad 0.3 \quad 0.5 \quad 0.7 \quad 0.9 \quad 1 \quad 1.2]$
11	$-0.5:0.05:0.8$	$[-2 \quad -1.5 \quad -1.2 \quad -0.8 \quad -0.2 \quad 0.2 \quad 0.3 \quad 1]$
12	$0.1:0.01:2.8$	$[-1 \quad -0.8 \quad -0.4 \quad 0 \quad 0.2 \quad 0.6 \quad 0.8 \quad 1.2]$
13	$-0.5:0.01:0.8$	$[-0.1 \quad -0.01 \quad 0 \quad 1.2 \quad 1.6 \quad 1.7 \quad 2.2 \quad 3]$
14	$-0.8:0.02:1.5$	$[-1 \quad -0.8 \quad -0.4 \quad 0 \quad 0.2 \quad 0.6 \quad 0.8 \quad 1.2]$
15	$1.1:0.02:2.3$	$[-2 \quad -1.8 \quad -1.2 \quad -0.8 \quad -0.2 \quad 0.3 \quad 0.5 \quad 1]$
16	$-1:0.02:0.3$	$[0 \quad 0.3 \quad 0.4 \quad 0.7 \quad 0.8 \quad 0.9 \quad 1 \quad 1.4]$
17	$-1:0.02:1.0$	$[-2 \quad -1.5 \quad -1 \quad -0.8 \quad -0.2 \quad 0.3 \quad 0.5 \quad 1]$
18	$0.5:0.02:2.8$	$[-1 \quad -0.8 \quad -0.4 \quad 0 \quad 0.2 \quad 0.6 \quad 0.8 \quad 1.2]$
19	$-0.5:0.01:1.8$	$[-0.2 \quad -0.1 \quad 0 \quad 1.2 \quad 1.6 \quad 1.8 \quad 2.5 \quad 2.8]$
20	$-0.6:0.02:1.3$	$[-1 \quad -0.5 \quad -0.3 \quad -0.2 \quad 0 \quad 0.2 \quad 0.5 \quad 1]$
21	$-0.9:0.02:0.3$	$[-3 \quad -2 \quad -1.5 \quad -1 \quad -0.8 \quad 0 \quad 0.8 \quad 1]$
22	$0.8:0.02:3.2$	$[-3 \quad -2 \quad -1.2 \quad -0.8 \quad 0 \quad 0.2 \quad 0.3 \quad 1]$
23	$-0.7:0.02:1.4$	$[-0.9 \quad -0.5 \quad 0 \quad 0.1 \quad 0.2 \quad 1 \quad 2 \quad 3]$
24	$-0.5:0.01:0.4$	$[0 \quad 0.1 \quad 0.8 \quad 1 \quad 1.8 \quad 1.9 \quad 2 \quad 2.2]$
25	$0.5:0.01:2.5$	$[-0.5 \quad -0.2 \quad 0 \quad 0.2 \quad 0.6 \quad 0.8 \quad 1 \quad 1.8]$

Контрольні запитання

1. Що таке апроксимація?
2. Що таке інтерполяція?
3. Якими функціями здійснюється апроксимація?
4. Якими функціями здійснюється інтерполяція?
5. Яку інтерполяцію здійснює функції ***icubic***?

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 2.4

Чисельне інтегрування функцій. Знаходження мінімумів функцій

Мета: вивчити можливості системи MATLAB для чисельного інтегрування функцій, знаходження мінімумів функцій. Набути практичних навичок у використанні відповідних вбудованих в MATLAB функцій.

Теоретичні відомості

При розв'язуванні багатьох інженерних задач доводиться обчислювати визначені інтеграли від функцій, у тому числі і від заданих при окремих заданих значеннях аргументу (таблично).

Більшість чисельних методів інтегрування функцій спираються на ідею заміни підінтегральної функції $x(t)$ деякою наближаючою її функцією $X(t)$, інтеграл від якої обчислюється досить просто. Наближення зазвичай здійснюється інтерполюванням у межах заданого діапазону змінювання аргументу. Зазвичай функція задана масивом (вектором) своїх значень у рівновіддалених точках діапазону змінювання аргументу t від a до b так, що $h=(b-a)/(n-1)$ є кроком задання функції (одночасно це є крок інтегрування). На рис. 4.1. наведено графічне подання такої функції.

Рис. 4.1 Графічне подання табличної функції

Існує декілька методів чисельного інтегрування, найвідомішими з яких є:

1. Метод правих прямокутників. Цей найпростіший метод чисельного інтегрування полягає у тому, що на кожному кроці $t_i \le t \le t_{i+1}$ інтегрування функція $x(t)$ замінюється на сталу величину, рівну значенню x_i на лівому кінці відповідного інтервалу.

Отже, для *визначення повного інтегралу за методом правих прямокутників достатньо просумувати усі задані значення функції, окрім останнього, і результат домножити на крок інтегрування*. при цьому похибка методу визначається середнім значенням похідної на інтервалі визначення інтегралу і пропорційна другому степеню кроку. Зменшуючі величину кроку інтегрування, можна суттєво зменшити похибку визначення інтегралу.

2. Метод лівих прямокутників. Сутність цього методу аналогічна, за винятком того, що апроксимуючі прямокутники тепер примикають зліва (див. рис. 4.3 б) до заданих точок.

Властивості цього методу майже не відрізняються від попереднього. Слід звернути увагу лише на те, що похибка його має зворотний знак. З цього одразу можна зробити висновок, що комбінуючи ці два методи, можна суттєво збільшити точність чисельного інтегрування. У такий спосіб приходимо до методу трапецій.

3. Метод трапецій. Формули цього методу є такими:

$$\int_a^b x(t)dt = h\left(\frac{x_1 + x_n}{2} + \sum_{i=1}^{n-1} x_i\right) - \frac{h^2 \cdot (b-a)}{12} X''(\xi).$$

Графічна інтерпретація: при інтегруванні цим методом обчислюються площі (рис. 4.2.в) трапецій, які утворюються, якщо з'єднати окремі задані точки відрізками прямих, тобто за лінійної інтерполяції.

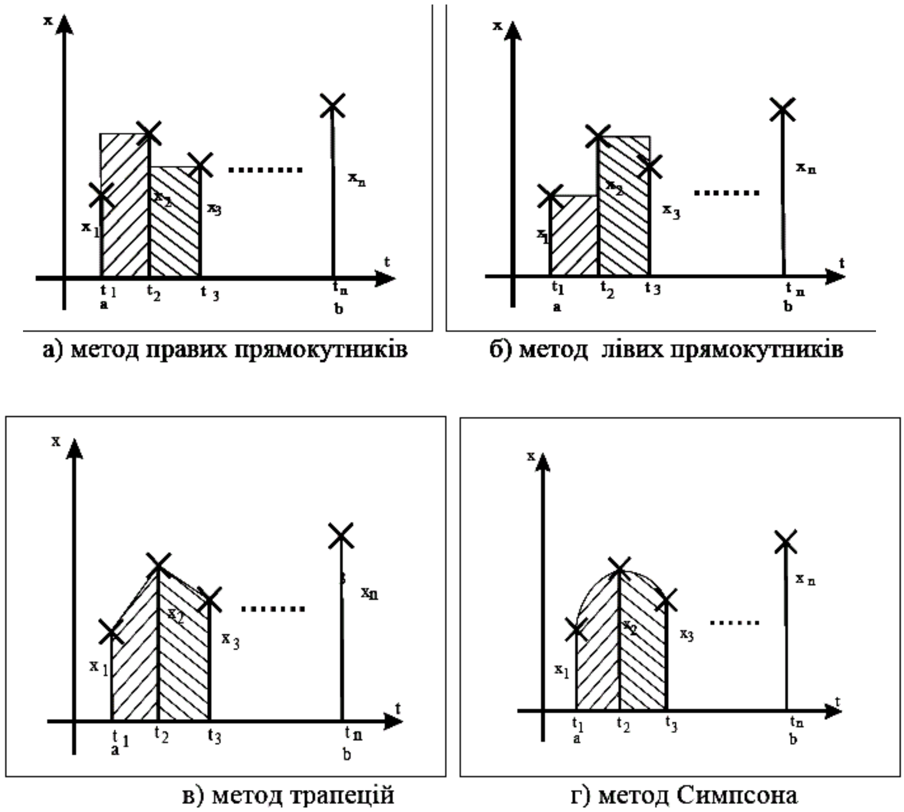


Рис. 4.2 Графічне подання методів інтегрування

4. Метод Сімпсона. У даному методі інтервал інтегрування *[a,b]* необхідно поділити на парну кількість рівних частин так, що загальна кількість точок буде непарною, а крок інтегрування дорівнюватиме

$$h=(b-a)/(n-1)=(b-a)/2(k-1).$$

На кожному з інтервалів виконується інтерполяція заданих точок квадратною параболою. Інтегруючи цю функцію в інтервалі $t_i \le t \le t_{i+2}$, можна одержати формулу Сімпсона для окремої ділянки . Підсумовуючи цей результат по усіх частинних відрізках $(i=1,3,5,...,n-2)$, отримують *квадратурну формулу Сімпсона*

$$\int_a^b x(t)dt = \frac{h}{3}\left(x_1+4x_2+2x_3+4x_4+2x_5+...+2x_{n-2}+4x_{n-1}+x_n\right)-\frac{h^4(b-a)}{180}X^{IV}(\xi)$$

На відміну від чисельного диференціювання, *чисельне інтегрування є обчислювально стійкою операцією*. Похибка будь-якого методу зменшується зі збільшенням кількості заданих точок початкової функції у заданому інтервалі, причому степень зменшення залежить від методу.

Похибку метода оцінюють з розгляду залишкових членів. Їх формула є степеневою функцією від кроку інтегрування. *Показник степеня у цій залежності похибки від кроку прийнято називати порядком метода інтегрування*. Таким чином, методи прямокутників є методами першого порядка, метод трапеції – другого порядка, а метод Сімпсона є методом четвертого порядка.

Засоби чисельного диференціювання і інтегрування на ЕОМ

Практично в усіх мовах програмування високого порядку відсутні вбудовані функції, які б дозволяли здійснювати чисельне диференціювання і інтегрування функцій. Деякі вбудовані функції (процедури) передбачені лише у новітніх комп‘ютерних математичних системах.

У системі MATLAB існують три вбудовані функції, які здійснюють чисельне інтегрування функцій – ***trapz***, ***quad*** і ***quad8***.

Процедура ***trapz*** здійснює обчислення площі під графіком функції $y(x)$ у випадку, коли функція подана у вигляді двох масивів (векторів) x – значень аргумента у зростаючому порядку і y – відповідних значень функції при цих значеннях аргумента. При цьому під графіком функції $y(x)$ розуміється ламана, що з’єднує сусідні точки функції. Звернення виду

I=***trapz***(x,y)

приводить до обчислення вказаної площі і присвоєння обчисленого значення ідентифікатору I.

Приклад. Обчислимо інтеграл від функції $y=\sin(x)$ у діапазоні від 0 до π. Його точне значення дорівнює 2. Візьмемо рівномірну сітку значень аргументу з 100 елементів. Тоді обчислення зведеться до сукупності операцій:

```
>> x = 0 : pi/100 : pi;
>> y = sin(x);
>> disp(trapz(x,y))
         1.99983550388744
```

Обчислення інтегралів методом квадратур у випадку, коли обчислення функції здійснюється у m-файлі з деяким ім‘ям, наприклад, FUN.m, здійснюється процедурою:

[I, cnt] = ***quad***('FUN', a, b),

де a і b – нижня і верхня межі змінювання аргументу функції (межі інтегрування; I – отримане значення інтеграла; cnt – кількість звернень до обчислення підінтегральної функції (яку репрезентовано m-файлом FUN.m) при обчисленні інтегралу.

Функція ***quad*** використовує квадратурні формули Ньютона – Котеса. Аналогічна функція ***quad8*** використовує більш точні формули 8-го порядку. Звернення до неї аналогічне.

Функції ***quad*** і ***quad8*** здійснюють розбиття заданого діапазону змінювання аргументу автоматично, забезпечуючи відносну похибку обчислення інтегралу, меншу за задану, яка за замовчуванням приймається рівною $1 \cdot 10^{-3}$. Якщо потрібно змінити точність обчислення інтегралу, слід задати граничну допустиму відносну похибку, додаючи у перелік вхідних параметрів під ім‘ям 'RelTol'. Наприклад, щоб встановити відносну похибку обчислення інтегралу меншу $1 \cdot 10^{-8}$, можна це зробити, звертаючись до цих функцій у такий спосіб:

[I, cnt] = ***quad***('FUN', a, b, 'RelTol', 1e-8).

Як приклад обчислимо той самий інтеграл, використовуючи як файл, що обчислює підінтегральну функцію, вбудовану функцію сінуса 'sin':

[I, cnt] = quad('sin', 0, pi)

Результат буде наступним:

I = 2.0000

cnt = 17

Завдання на виконання комп'ютерного практикуму

Завдання 1. Відповідно до таблиці 4.1 виконати обчислення точних (по стандартних функціях MATLAB) значень відповідної функції у діапазоні змінювання аргументу від x1 до x2 в m рівновіддалених точках цього діапазону, включаючи його межі. Побудуйте графік цієї функції.

Табл. 4.1 Варіанти індивідуальних завдань

Варіант	x_1	x_2	m	$f(x)$
1	2	3	4	5
1	-2	5	80	$\sin(x)$
2	0	10	50	$\cos(x)$
3	0.3	3	40	$\exp(x)$
4	0.4	4	50	$\ln(1+x)$
5	0.5	5	30	$\ln(x)$
6	0.6	6	40	$\ln(x)$
7	0.7	7	50	$\ln(x)$
8	0.8	8	45	$\ln(x+a)$
9	1.1	11	40	$ctg(x)$

1	2	3	4	5
10	1.2	12	50	$\cos ec(x)$
11	1.3	13	50	$\cos ec(x)$
12	1.4	14	60	$arctg(x)$
13	1.5	15	45	$arctg(x)$
14	1.6	16	40	$\ln(x)$
15	0.9	9	50	$\sin(x)$
16	-2	10	50	$\cos(x)$
17	0.6	5	50	$\ln(x)$
18	-0.9	0.9	45	$arcctg(x)$
19	1	20	50	$sh(x)$
20	1	20	50	$ch(x)$
21	-0.9	0.9	50	$arcth(x)$
22	1	20	50	$arcth(x)$
23	-0.8	0.8	50	$\arcsin(x)$
24	-0.8	0.8	50	$\arccos(x)$
25	-5	5	50	$\exp(x)$

Завдання 2. Створіть М-файл, що обчислює функцію з завдання 1. Побудуйте графік цієї функції за допомогою процедури ***plot*** у межах, визначених у завданні 1. Обчисліть інтеграл від функції у тих самих межах, використовуючи процедури ***quad*** і ***quad8***. Знайдіть значення абсциси та ординати точки локального мінімуму функції, та найближчий корінь функції (нуль: $f\left(x\right)=0$).

Завдання 3. Знайдіть точку локального мінімуму і локальний мінімум функції двох змінних, прийнявши за початкову точку із заданими координатами (табл. 4.2). Попередньо створіть відповідну файл-функцію.

Табл. 4.2 Варіанти індивідуальних завдань

Варіант	x_0	y_0	$f(x,y)$
1	0	1	$e^{x+y}+(x-y)^2-2x-2y$
2	0.7	-1.2	$(x-y)^2-\cos(x-y-1)$
3	1.5	-0.5	$e^{x+y}-2x-2y-\cos(x-y-1)$
4	0.5	1.5	$e^{x+y}+4x^2-3x-3y$
5	0	1	$4x^2+\ln(x+y)+\dfrac{1}{x+y}$
6	1.2	0.7	$2^{x+y}-2x-2y+2(x-y)^2$
7	0	-0.9	$e^{x-y}+2x+2y+(x+y)^2$
8	0.8	1.3	$(x-y)^2-\cos(x+y-1)$
9	1.5	0.5	$e^{x-y}-2x+2y-\cos(x+y-1)$
10	0.5	-1.5	$e^{x-y}-3x+3y+4x^2$
11	0	-1	$4x^2+\ln(x-y)+\dfrac{1}{x-y}$
12	1.2	-0.8	$2^{x-y}-2x+2y+2(x+y)^2$

Контрольні запитання

1. У чому полягає задача чисельного інтегрування функції?

2.Які існують найпростіші методи чисельного інтегрування функції

3. Які вбудовані функції, що здійснюють чисельне диференціювання і чисельне інтегрування функції, існують у мовах високого рівня, математичних комп‘ютернх системах, у MATLAB?

4. Чим відрізняється обчислення інтегралу у функціях ***trapz***, ***quad*** і ***quad8*** системи MATLAB?

5. Які формули використовуються для інтегрування у функціях ***quad*** і ***quad8*** системи MATLAB?

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 2.5

Чисельне інтегрування диференціальних рівнянь

Мета: навчитися створювати власні програмні засоби у системі MATLAB для виконання чисельного інтегрування звичайних диференціальних рівнянь заданим методом.

Теоретичні відомості

Деякі алгоритми є загальними для усіх видів функцій певного вигляду. Тому їх програмна реалізація є єдиною, але потребує знання алгоритму обчислення функції, який може суттєво змінюватися при переході від однієї функції до іншої. Останній алгоритм може бути зафіксований у вигляді деякої файл-функції. Щоб перший, більш загальний алгоритм був пристосований для будь-якої функції, що використовується в ньому, потрібно, щоб ім’я цієї функції набувало будь-якого вигляду, тобто було деякою змінною, що набуває певного значення (текстового імені файл-функції) лише при зверненні до основного алгоритму.

До таких функцій належать процедури:

- обчислення інтегралів від функції, які потребують вказання імені М-файлу, що містить обчислення значення підінтегральної функції;
- чисельного інтегрування диференційних рівнянь, які потребують вказання імені М-файлу, в якому обчислюються праві частини рівнянь у формі Коші;
- алгоритмів чисельного обчислення коренів нелінійних алгебричних рівнянь, які потребують вказання файл-функції, корень якої відшукується;

- алгоритми пошуку мінімуму функції, яку потрібно задавати відповідним М-файлом тощо.

На практиці досить часто виникає потреба створити такі власні процедури. MATLAB надає такі можливості.

Процедура feval

У MATLAB будь-яка функція (процедура), наприклад за ім'ям FUN1, може бути виконана не лише за допомогою звичайного звернення вигляду:

[y1,y2….,yk] = FUN1(x1,x2….,xn),

а і за допомогою спеціальної процедури *feval* таким чином:

[yl ,y2,...,yk] = ***feval***('FUN1',xl,x2,...,xn),

де ім'я функції FUN1 є вже однією з вхідних змінних - текстовою змінною, і тому вміщено у апострофи.

Перевагою виклику функції у другій формі є те, що він має той самий вигляд при змінюванні імені функції, наприклад, на FUN2. Це дозволяє уніфікувати звернення до усіх функцій певного вигляду, тобто таких, що мають однакову кількість вхідних і вихідних параметрів відповідного типу. При цьому ім'я функції, яка використовується, може бути довільним і змінюватися протягом повторних обчислень.

Через те, що при виклику функції за допомогою процедури ***feval*** ім'я функції розглядається як один з вхідних параметрів процедури, останнє (ім'я функції) можна використовувати як змінну і оформляти у М-файлі звернення до неї, не маючи ще конкретного імені функції.

Приклади створення процедур від функцій

Процедура метода чисельного інтегрування Рунге-Кутта 4-го порядку

Нехай маємо систему звичайних диференційних рівнянь (ЗДР) у вигляді Коші:

$$\frac{dy}{dt} = Z(y,t),$$

де y – вектор змінних стану системи; t - аргумент (час); Z – вектор заданих нелінійних функцій, який визначає систему ЗДР.

Якщо значення вектора y у момент часу t є відомим, то загальна формула, за якою знаходиться вектор ***yout*** значень змінних стану системи у момент часу $tout = t + h$, де h – крок інтегрування за вектором y значень цих змінних на попередньому кроці, має вигляд:

$$\boldsymbol{yout} = y + h * \boldsymbol{F}(y,t).$$

Функція $F(y_y t)$, пов'язана з вектором Z, може набувати різного вигляду залежно від обраного методу чисельного інтегрування. Для метода Рунге-Кутта 4-го порядку оберемо таку її форму:

$$F = \left(k_1 + 3 \cdot k_2 + 3 \cdot k_3 + k_4\right)/8,$$

де

$$\begin{aligned} k_1 &= Z(y,t); \\ k_2 &= Z(y + h \cdot k_1/3, t + h/3); \\ k_3 &= Z(y + h \cdot k_2 - h \cdot k_1/3, t + 2 \cdot h/3); \\ k_4 &= Z(y + h \cdot k_3 - h \cdot k_2 - h \cdot k_1, t + h). \end{aligned} \quad k1 = Z(y,t);$$

Утворимо М-файл процедури, що здійснює ці обчислення, назвавши його ***rko43****:*

```
function [tout,yout] = rko43(Zpfun,h,t,y)
%RK041 Інтегрування ЗДР методом Рунге-Кутта 4-го порядку.
%   праві частини яких задані процедурою Zpfun.
%   Вхідні змінні:
% Zpfun - рядок, що містить ім'я процедури обчислення правих частин ЗДР
%   звернення: z = fun(t,y), де Zpfun = 'fun'
%   t   - поточний момент часу
%   y   - вектор поточних значень змінних стану
%   z   - обчислені значення похідних z(i) = dy(i)/dt.
%   h - крок інтегрування
%   t - попередній момент часу
%   y - попереднє значення вектора змінних стану.
%   Вихідні змінні:
% tout - новий момент часу
% yout - обчислене значення вектора змінних стану через крок інтегрування
%   Ю.Ф. Лазарев, 274-67-31.
% Copyright (c) 1998 by The PSON, NTU of Ukraine "KPI".
%
%   Розрахунок проміжних значень похідних
k1 = feval(Zpfun, t, y);
k2 = fevaI(Zpfun, t+h/3, y+h/3*k1);
k3 = fevaI(Zpfun, t+2*h/3, y+h*k2-h/3*k1); k4 = feval(Zpfun, t+h, y+h*(k3+k1-k2));
%   Розрахунок нових значень вектора змінних стану
tout = t + h;
yout = y + h*(k1 + 3*k2 + 3*k3 + k4)/8;
% Кінець процедури RK043
```

Зверніть увагу на такі обставини:

- звернення до процедури обчислення правих частин не конкретизовано; ім'я цієї процедури входить до вхідних змінних процедури інтегрування і конкретизується при зверненні до останньої;
- проміжні змінні *k* є векторами-рядками (те саме стосується і змінних *y*, а також змінних z, що обчислюються у процедурі правих частин).

Процедура правих частин ЗДР маятника

Розглянемо процес утворення процедури обчислення правих частин ЗДР на прикладі рівняння» маятника, точка підвісу якого поступально переміщується з часом за гармонійним законом:

$$J \cdot \ddot{\varphi} + R \cdot \dot{\varphi} + mgl \cdot \left(1 + n_{my} \cdot \sin\left(\omega \cdot t + \varepsilon_y\right)\right) \cdot \sin\varphi =$$
$$= -mgl \cdot n_{mx} \cdot \sin\left(\omega \cdot t + \varepsilon_x\right) \cdot \cos\varphi,$$

де J – момент інерції маятника; R – коефіцієнт демпфірування; mgl – опорний маятниковий момент маятника; n_{my} – амплітуда віброперевантаження точки підвісу маятника у вертикальному напрямку; n_{mx} – амплітуда віброперевантаження точки підвісу маятника у горизонтальному напрямку; φ – кут відхилення маятника від вертикалі; ω – частота коливань точки підвісу; ε_x , ε_y – початкові фази коливань (з прискорень) точки підвісу у горизонтальному і вертикальному напрямках.

Щоб скласти М-файл процедури обчислення правих частин заданої системи ЗДР, перш за все слід привести початкову систему ЗДР до форми Коші. Для цього введемо позначення:

Тоді початкове рівняння маятника можна подати у вигляді двох диференційних рівнянь 1-го порядку:

$$\frac{dy1}{dt} = y2;$$

$$\frac{dy2}{dt} = \{-m \cdot g \cdot l \cdot n_{mx} \cdot \sin(\omega \cdot t + \varepsilon_x) \cdot \cos(y1) - R \cdot y2 -$$
$$- m \cdot g \cdot l \cdot [1 + n_{my} \cdot \sin(\omega \cdot t + \varepsilon_y)] \cdot \sin(y1)\} / J .$$

Порівнюючи отриману систему з загальною формою рівнянь Коші, можна зробити висновок, що

$$z1 = y2;$$

$$z2 = \{-m \cdot g \cdot l \cdot n_{mx} \cdot \sin(\omega \cdot t + \varepsilon_x) \cdot \cos(y1) - R \cdot y2 -$$
$$- m \cdot g \cdot l \cdot [1 + n_{my} \cdot \sin(\omega \cdot t + \varepsilon_y)] \cdot \sin(y1)\} / J .$$

Саме обчислення цих двох функцій і повинно відбуватися у процедурі правих частин. Назвемо майбутню процедуру *fmO*. Вихідною змінною у ній буде вектор

$$z = [zl\ z2],$$

а вхідними – момент часу *t* та вектор

$$y = [y1y2].$$

Деякою складністю є те, що сталі коефіцієнти, які є у правих частинах, не можна передавати у процедуру через її заголовок. Тому об'єднаємо їх у вектор коефіцієнтів k = [J,R,mgl, nmy,nmx, om,ey,ex] і віднесемо цей вектор до категорії глобальних ***global*** K.

Тоді М-файл матиме такий вигляд:

```
function z = FMO(t,y);
% Процедура правих частин рівняння Фізичного Маятника.
%   Здійснює розрахунок вектора "z" похідних    від вектора "y" змінних стану
%   за формулами:
%   z(1)=y(2);
%   z(2)=(-mgl*nmx*sin(om*t+ex) * cos(y( 1 ))-R*y(2)-...
%   mgl*(1 +nmy*sin(om*t+ey))*sin(y(1)))/J,
%  Коефіцієнти передаються у процедуру через глобальний %   вектор
%   K=|J,R,mgl,nmy,nmx,om,ey,ex|
%     Ю.Ф.Лазарев   5-10-1998р.
global K
z(1) = y(2);
z(2) = (-K(3)*K(5)*sin(K(6)*t+K(8))*cos(y(1)) - K(2)*y(2) -...
K(3)*(1+K(4)*sin(K(6)*t+K(7)))*sin(y(1)))/K(1);
% Кінець процедури FMO
```

При використанні цієї процедури слід пам'ятати, що у тексті програми попередньо має бути визначений вектор K з 8 елементів і потім він має бути перетворений на глобальний за допомогою службового слова ***global.***

Цю ж процедуру можна дещо ускладнити, групуючи разом обчислення усіх зовнішніх моментів сил, окрім моменту сил тяжіння, і оформлюючи їх як окрему процедуру. Для цього спочатку дещо перетворимо початкове рівняння, записуючи його у вигляді:

$$\varphi'' + \sin\varphi = S(\tau, \varphi, \varphi'),$$

де штрих – позначення похідної за безрозмірним часом

$$\omega_0 = \sqrt{\frac{m \cdot g \cdot l}{J}},$$

а через $S(\tau, \varphi, \varphi')$ позначено деяку задану функцію безрозмірного часу, куту повороту маятника та його безрозмірної швидкості

$$\varphi' = \frac{d\varphi}{d\tau}.$$

У розглядуваному випадку ця функція набуває вигляду:

$$S(\tau, \varphi, \varphi') = -2 \cdot \varsigma \cdot \varphi' -$$
$$-\left[n_{mx} \cdot \sin(\nu \cdot \tau + \varepsilon_x) \cdot \cos\varphi + n_{my} \cdot \sin(\nu \cdot \tau + \varepsilon_y) \cdot \sin\varphi\right],$$

причому безрозмірні величини ς і ν визначаються виразами:

$$\varsigma = \frac{R}{2 \cdot \sqrt{m \cdot g \cdot l \cdot J}}; \qquad \nu = \frac{\omega}{\omega_0}.$$

Така безрозмірна форма подання рівнянь є зручнішою, бо дозволяє скоротити кількість параметрів (у нашому випадку замість трьох розмірних параметрів J, R та $m \cdot g \cdot l$ залишається один – ς*),* а також подавати розв’язок рівняння у більш узагальненій формі.

Винесення обчислення моментів зовнішніх дій у окрему обчислювальну процедуру також дозволяє зробити процедуру правих частин рівняння маятника більш загальною, якщо звернення до процедури обчислення моментів здійснювати також через функцію ***feval.***

Утворимо процедуру ***MomFm1***, яка обчислюватиме моменти сил, що діють на маятник:

```
function m = MomFM1(t,y);
% Обчислення Моментів сил, що діють на Фізичний Маятник.
%   Здійснює розрахунок моменту "m□ сил
%   за формулою:
%   m=-2*dz*y(2) - (nmx*sin(nu*t+ex)*cos(y(1)) +...
%   + nmy*sin(nu*t+ey)*sin(y(1)),
% Коефіцієнти передаються у процедуру через глобальний
% вектор □
%   KM1=(dz,nmy,nmx,nu,ey,ex)
% Ю.Ф Лазарев    5-10-1998р.
global KM1
m = -2*KM1(1)*y(2)- (KM1(3)*sin(KM1(4)*t+KM1(6))*cos(y(1)) +...
KM1(2)*sin(KM1(4)*t+KM1(5))*sin(y(1)));
% Кінець процедури MomFM1
```

Відтепер слід перебудувати процедуру правих частин. Назвемо цей варіант FM1:

```
function z = FM1(mpfun,t,y);
% Процедура правих частин рівняння Фізичного Маятника.
% Здійснює розрахунок вектора "z" похідних вектора"y" змінних стану
%   за формулами:
%   z(1)=y(2);
%   z(2)= - sin(y(1)) +S(t,y),
%   де
%   вхідні параметри:   >
%   mpfun - ім'я процедури S(t,y)
%   mpfun = 'S';
%   t - поточний момент часу;
%   y - поточне значення вектору змінних стану;
%   вихідні параметри:
%   z - вектор значень похідних від змінних стану.
% Ю.ФЛазарев    5-10-1998р.
%
z(1) = y(2);
z(2) = - sin(y(1)) + feval(mpfun,t,y);
% Кінець процедури FM1
```

Через те, що вид звернення до процедури правих частин змінився, необхідно також перебудувати і процедуру чисельного методу. Назвемо її RK043m:

```
function [tout,yout] = rko43m(Zpfun,Mpfun,h,t,y)
% RK043m Інтегрування ЗДР методом Рунге-Кутта 4-го порядку.
%  праві частини яких задані процедурою Zpfun і Mpfun.
%  Вхідні змінні:
% Zpfun - рядок, що містить і'мя процедури обчислення правих частіш ЗДР.
%   звернення: z = fun(Mpfun,t,y), де Zpfun = □fun□,
%   а Mpfun - рядок, що містить ім□я процедури, до якої
%   звертається процедура fun;
%   t - поточний момент часу
%   y вектор поточних значень змінних стану
%   z - обчислені значення похідних z(i) = dy(i)/dt.
%   h - крок інтегрування
%   t - попередній момент часу
%         y - попереднє значення вектора змінних стану.
% Вихідні змінні:
%             tout - новий момент часу
%             yout - обчислене значення вектора змінних стану через крок інтегрування
% Ю.Ф.Лазарєв, 274-67-31.
% Copyright (c) 1998 by The PSON, NTU of Ukraine "KPI" .

%
%   Розрахунок проміжних значень похідних
k1 = feval(Zpfun, Mpfun,t, y);
k2 = feval(Zpfun, Mpfun, t+h/3, y+h/3*k1);
k3 = feval(Zpfun, Mpfun, t+2*h/3, y+h*k2-h/3*k1);
k4 = feval(Zpfun, Mpfun, t+h, y+h*(k3+k1-k2));
%   Розрахунок нових значень вектора змінних стану
tout = t + h;
yout = y + h*(k1 + 3*k2 + 3*k3 + k4)/8;
% Кінець процедури RK043m
```

Ця форма подання процедури обчислення правих частин диференційних рівнянь не є зручною, бо, по-перше, процедуру вигляду FM1 неможливо використовувати у зверненні до процедур MATLAB ***ode23*** та ***ode45*** (останні потребують, щоб у процедурі правих частин було лише два вхідних параметри, а у процедури FM1 їх три), а по-друге, така форма потребує утворення нових М-файлів методів чисельного інтегрування, як це створено зараз.

Можна цього уникнути, перетворюючи ім’я додаткової функції Mpfun на глобальну змінну. Тоді процедура правих частин може бути записана у вигляді:

```
function z = FM2(t,y);
% Процедура правих частин рівняння Фізичного Маятника.
% Здійснює розрахунок вектора □z□ похідних вектора"у" змінних стану
%   за формулами:
%   z(1)=y(2);
%   z(2)= - sin(y(1)) +S(t,y),
%   де
%   вхідні параметри:
%   mpfun - ім'я процедури S(t,y)
%   mpfun = 'S□;
%   t - поточний момент часу;
%   y - поточне значення вектора змінних стану;
%   вихідні параметри:
%   z - вектор значень похідних від змінних стану.
%   Ю.Ф.Лазарєв 7-10-1998р.
%
global MPFUN z(1) = y(2);
z(2) = - sin(y(1)) + feval(MPFUN,t,y);
% Кінець процедури FM2
```

Тепер процедура FM2 має лише два вхідних параметри, що передаються через заголовок, і може бути використаною будь-якою процедурою чисельного методу інтегрування, у тому числі, процедурами ***ode23*** та ***ode45***. Необхідно лише пам’ятати, що у головній програмі, де здійснюється звернення до процедури, змінній MPFUN спочатку слід присвоїти деяке текстове значення (ім'я функції, яка буде використовуватися у процедурі правих частин), а потім вона має бути оголошеною як глобальна. Наприклад, якщо буде використана раніше утворена процедура MomFunl, у головному Script-файлі повинні існувати рядки

```
MPFUN = 'MomFml';
global MPFUN
```

Завдання на виконання комп’ютерного практикуму

Завдання 1. Створіть М-файл метода чисельного інтегрування диференціальних рівнянь відповідно до формул, наведених у табл. 5.1.

Табл.5.1 Метод Рунге-Кутта

$$y_{m+1} = y_m + hF(t_m; y_m)$$

Варіант	Формула метода	Допоміжні величини	Назва методу
1	$F = k_1$	$k_1 = Z(t_m, y_m)$	Ейлера
2	$F = (k_1 + k_2)/2$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h, y_m + hk_1)$	Модифікований Ейлера-1
3	$F = Z(t_m + h/2, y_m + hk_1/2)$	$k_1 = Z(t_m, y_m)$	Модифікований Ейлера-2
4	$F = (k_1 + 4k_2 + k_3)/6$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h/2, y_m + hk_1/2)$ $k_3 = Z(t_m + h, y_m + h(2k_2 - k_1))$	Хойне-1
5	$F = (k_1 + 3k_3)/4$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h/3, y_m + hk_1/3)$ $k_3 = Z(t_m + 2h/3, y_m + 2hk_2/3)$	Хойне-2
6	$F = (k_1 + 2k_2 + 2k_3 + k_4)/6$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h/2, y_m + hk_1/2)$ $k_3 = Z(t_m + h/2, y_m + hk_2/2)$ $k_4 = Z(t_m + h, y_m + hk_3)$	Рунге-Кутта-1
7	$F = (k_1 + 3k_2 + 3k_3 + k_4)/8$	$k_1 = Z(t_m, y_m)$ $k_2 = Z(t_m + h/3, y_m + hk_1/3)$ $k_3 = Z(t_m + 2h/3, y_m + h(k_2 - k_1/3))$ $k_4 = Z(t_m + h, y_m + h(k_{31} - k_2 + k_3))$	Рунге-Кутта-2

Завдання 2. Створіть М-файл процедури правих частин диференціальних рівнянь динаміки:

1) двоступеневого гіроскопічного компаса:

$$J_1\ddot{\beta} + H\left[\omega_3 \cos\varphi \sin\beta + u_E(t)\cos\beta + u_N(t)\sin\beta\right] = 0,$$

де J_1 – момент інерції гірокомпаса; H – його власний кінетичний момент; ω_3 – кутова швидкість власного обертання Землі; β – кут відхилення головної осі гірокомпаса від площини географічного меридіана місця ; φ – географічна широта місця об'єкта, на якому встановлений гірокомпас; $u_N(t) = u_{Nm}\sin(\omega t + \varepsilon_N),\ u_E(t) = u_{Em}\sin(\omega t + \varepsilon_E)$ – відповідно північна і східна складові кутової швидкості повороту основи, на якій встановлений гірокомпас;

2) обсягів популяцій $x_1(t)$ хижаків і $x_2(t)$ жертв (модель Вольтерра):

$$\dot{x}_1 = -a_{11}x_1 + a_{12}x_1x_2$$
$$\dot{x}_2 = a_{22}x_1 - a_{21}x_1x_2$$

3) торпеди, яка керується нелінійним виконуючим елементом:

$$J\frac{d^2\psi}{dt^2} + R\frac{d\psi}{dt} + kF(\psi) = 0,$$

процедура має передбачати можливість використання кількох істотно нелінійних законів керування, зокрема, релейного із зонами нечутливості і гістерезисом:

$$\text{якщо } \dot{x} > 0, \qquad \text{то} \qquad F(x) = \begin{cases} -c \text{ при } x < b_1 \\ 0 \text{ при } -b_1 < x < b_2 \\ +c \text{ при } x > b_2 \end{cases}$$

$$\text{якщо ж } \dot{x} < 0, \qquad \text{то} \qquad F(x) = \begin{cases} +c \text{ при } x > b_1 \\ 0 \text{ при } -b_2 < x < b_1 \\ -c \text{ при } x < b_2 \end{cases}$$

4) штучного супутника Землі, керованого за логічним законом:

$$J\frac{d\omega}{dt} = k\Phi(\omega,\varphi)$$
$$\frac{d\varphi}{dt} = \omega$$

де ω – кутова швидкість руху супутника; J – його момент інерції; $\Phi(\omega,\varphi)$ – логічна нелінійна функція задана своїми значеннями згідно таблиці:

Табл. 5.2 Значення функції $\Phi(\omega,\varphi)$

Знак ω	Знак φ		
	-	0	+
-	+1	0	0
0	+1	0	-1
+	0	0	-1

5) гіроскопа в кардановому підвісі (ГКП), установленого на нерухомій основі:

$$\begin{cases} (J_1 + J_2\cos^2\beta)\ddot{\alpha} - 2J_2\dot{\alpha}\dot{\beta}\sin\beta\cos\beta + H\dot{\beta}\cos\beta = -f_2\dot{\alpha} + N_0 + N_m\sin(\omega t + \varepsilon_N) \\ J_3\ddot{\beta} + J_2\dot{\alpha}\sin\beta\cos\beta - H\dot{\alpha}\cos\beta = -f_1\dot{\beta} + L_0 + L_m\sin(\omega t + \varepsilon_L) \end{cases}$$

де J_1, J_2, J_3 – моменти інерції гіроскопа; β α, – кути повороту головної осі ГКП навколо зовнішньої і внутрішньої осей підвісу; H – власний кінетичний момент ГКП; f_1, f_2 – коефіцієнти в'язкого тертя вдовж внутрішньої і зовнішньої осей підвісу; N_0, L_0 – сталі складові моментів зовнішніх сил, спрямованих по зовнішній і внутрішній осях підвісу; N_m, L_m – амплітуди гармонійних складових моментів сил, що діють по відповідних осях; ω – частота змінювання гармонічних складових моментів сил; $\varepsilon_N, \varepsilon_L$ – початкові фази гармонічних складових моментів сил;

6) гіроскопічного тахометра (ГТ), встановленого на обертовій основі:

$$J_1\ddot{\beta} + f\dot{\beta} + c\beta = Hu_{z1} - J_1\dot{u}_Y,$$

де J_1 – момент інерції чутливого елемента ГТ; H – власний кінетичний момент гіроскопа; c – кутова жорсткість пружного зв'язку ГТ з основою; f – коефіцієнт кутового демпфірування; u_{X1}, u_{Y1}, u_{Z1} – проекції кутової швидкості основи на осі, пов'язані з ГТ. Останні пов'язані із проекціями на осі, пов'язані з основою, співвідношеннями:

$$\begin{aligned} u_{X1} &= u_X \cos\beta - u_Z \sin\beta \\ u_{Z1} &= u_Z \cos\beta + u_X \sin\beta \\ u_{Y1} &= u_Y \end{aligned}$$

проекції кутової швидкості основи на осі тієї ж основи змінюються у часі за законами:

$$\begin{aligned} u_X &= u_{X0} + u_{Xm} \sin(\omega t + \varepsilon_X) \\ u_Y &= u_{Y0} + u_{Ym} \sin(\omega t + \varepsilon_Y) \\ u_Z &= u_{Z0} + u_{Zm} \sin(\omega t + \varepsilon_Z) \end{aligned}$$

7) лампового генератора (рівняння Ван-дер-Поля):

$$\ddot{q} + \omega^2 q = (\chi - \lambda q^2)\dot{q}$$

8) маятника на рухомій основі з урахуванням моменту сил уздовж осі маятника:

$$J\ddot{\varphi} + R\dot{\psi} + mgl\sin\varphi = M_{tr} + M_0 + M_m \sin(\omega t),$$

де $\psi = \varphi - \vartheta(t)$, $\vartheta(t)$ – поточний кут повороту основи навколо осі маятника; M_0 – постійна складова моменту зовнішніх сил, що діють на маятник; M_m – амплітуда гармонічної складової моменту зовнішніх сил; ω – частота змінювання моменту сил; M_{tr} – момент сил сухого тертя по осі маятника: $M_{tr} = -Msign(\dot{\psi})$, де

$$sign(x) = \begin{cases} +1 & якщо \quad x > 0 \\ 0 & якщо \quad x = 0 \\ -1 & якщо \quad x < 0 \end{cases}$$

а також $\vartheta(t) = \vartheta_0 + \vartheta_0 t + \vartheta_m \sin(\omega t + \varepsilon)$.

Контрольні запитання

1.Для чого прислуговується функція ***feval***? Як її застосовують?

2. Що таке нормальна форма Коши диференціальних рівнянь? Як до неї дійти?

3. Що таке процедура правих частин диференціальних рівнянь? Які дії мають передувати складанню цієї процедури? Що обчислюється внаслідок звернення до неї?

4. Як забезпечити передавання у процедуру правих частин диференціальних рівнянь потрібних сталих коефіцієнтів?

5. Що називають змінними стану системи? Чи збігаються вони з шуканими змінними заданих диференціальних рівнянь?

6. Чим визначається кількість змінних стану?

7. У чому полягає задача чисельного інтегрування системи диференційних рівнянь?

8. Що робить процедура, яка реалізує той чи інший метод чисельного інтегрування диференціальних рівнянь? Які аргументи у такої процедури? Що обчислюється в результаті застосування цієї процедури?

9. До якої процедури обов’язково звертається процедура методу інтегрування? Як забезпечується звернення до потрібної процедури, якщо заздалегідь невідомо, які саме рівняння інтегруватимуться?

10. Чим вирізняються однокрокові методи різних порядків?

11. Що визначає величина порядку методу чисельного інтегрування?

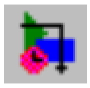
12. Які функції MATLAB здійснюють чисельне інтегрування диференціальних рівнянь і у який спосіб? Як визначається крок інтегрування? припустима похибка інтегрування? Як забезпечується інтегрування саме тих рівнянь, розв’язок яких потрібно знайти?

КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 2.6

Моделювання динамічних систем з використанням пакета Simulink. Створення S-моделей, використовуючи розділ Sources

Мета: ознайомлення з бібліотекою Simulink та набуття навичок створення S-моделей для генерації вхідних сигналів, використовуючи розділ Sources.

Теоретичні відомості

Для запуску Simulink необхідно на панелі інстментів командного вікна системи MATLAB натиснути на кнопку  (New Simulink Model) або у командному вікні системи, за допомогою клавіатури ввести наступний текст:

>> simulink

та натиснути на <Enter>.

Після запуску Simulink на дисплеї комп’ютера з’явиться вікно перегляду бібліотек – Simulink Library Browser (рис. 6.1).

Бібліотека Simulink системи MATLAB складається з наступних розділів:

1. Continuous − лінійні блоки.
2. Discrete − дискретні блоки.
3. Functions & Tables − функції і таблиці.
4. Math − блоки математичних операцій.
5. Nonlinear − нелінійні блоки.
6. Signals & Systems − сигнали і системи.
7. Sinks − реєструючі пристрої.

8. Sources – джерела сигналів і впливів.

9. Subsystems – блоки підсистем.

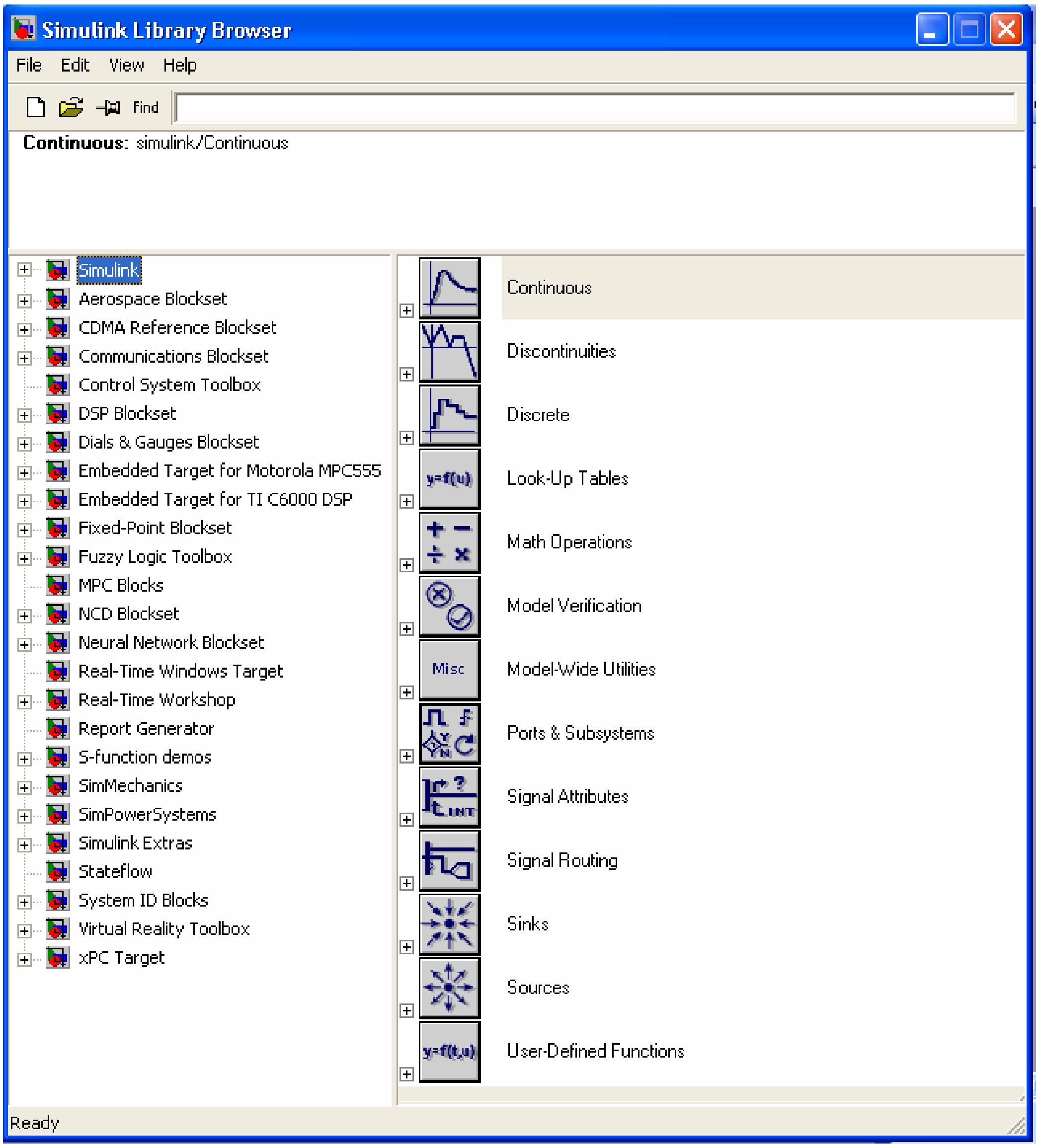


Рис.6.1 Вікно перегляду бібліотеки Simulink

Розділ Sources

Блоки, що входять в розділ Sources (джерела сигналів та впливів), призначені для формування сигналів, які при моделюванні забезпечують роботу S-моделі в цілому або окремих її частин. Всі блоки-джерела мають по одному виходу і не мають входів. На рис. 6.2 представлено вікно розділу бібліотеки з джерелами Sources.

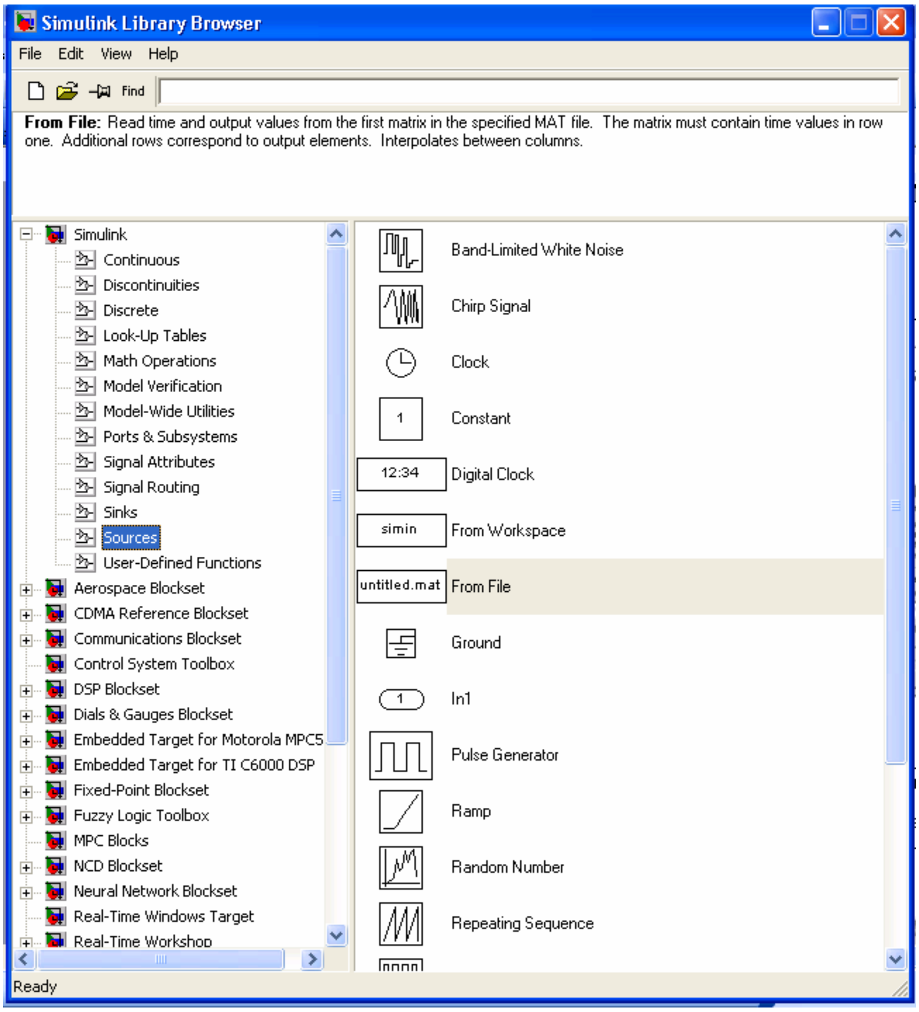
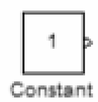
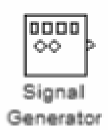


Рис.6.2. Вікно з переліком джерел сигналів і впливів

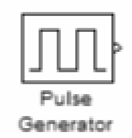
Зображення блоків, що входять до розділу Sources бібліотеки Simulink:



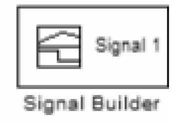
- формує постійну величину. На рис. 6.3 представлено застосування цього джерела і контроль рівня його впливу за допомогою осцилографа і цифрового індикатора дисплея.



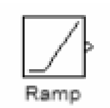
- створює безперервний коливальний сигнал однієї з хвильових форм (синусоїдальної, прямокутної, трикутної) або випадковий сигнал.



- генератор безупинних прямокутних імпульсів (рис.6.4).

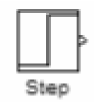


- створює один або декілька процесів, що апроксимується відрізками прямих (до п'яти відрізків у кожному).



- створює лінійний висхідний (або спадний) сигнал. На рис. 6.5 представлено джерело лінійно наростаючого впливу виду $F(t) = k * t + i$ і вікно його параметрів. До параметрів джерела відносяться: *Slope* − кутовий коефіцієнт часової залежності k; *Start time* − час, починаючи з якого вплив наростає; *Initial value* − початковий рівень впливу.

- генерує гармонійний сигнал. Він характеризується амплітудою *Amplitude*, зміщенням по вертикалі *Bias*, частотою *Frequence*, фази *Phase* і еталонним часом *Sample time*. Останнє використовується для узгодження роботи джерела. На рис. 6.6 представлено застосування цього джерела і контроль рівня його впливу за допомогою осцилографа.



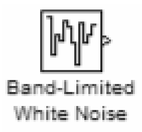
генерує сигнал у вигляді сходинки (ступінчастий сигнал) із заданими параметрами (час початку сходинки і її висота). На рис.6.7. представлено джерело впливу у вигляді одиночного перепаду. До параметрів джерела відносяться: *Step time* − час появи перепаду (стрибка); *Initial value* − початкове значення впливу (до перепаду); *Final value* − кінцеве значення впливу (після перепаду); *Sample time* − еталонне час.

- генерує періодичну послідовність.

- генератор гармонійних коливань, частота яких лінійно змінюється у часі.

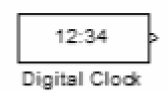
- джерело дискретного сигналу, значення якого є випадковою величиною, розподіленою за нормальним законом.

- джерело дискретного сигналу, значення якого є випадковою рівномірно розподіленою величиною.

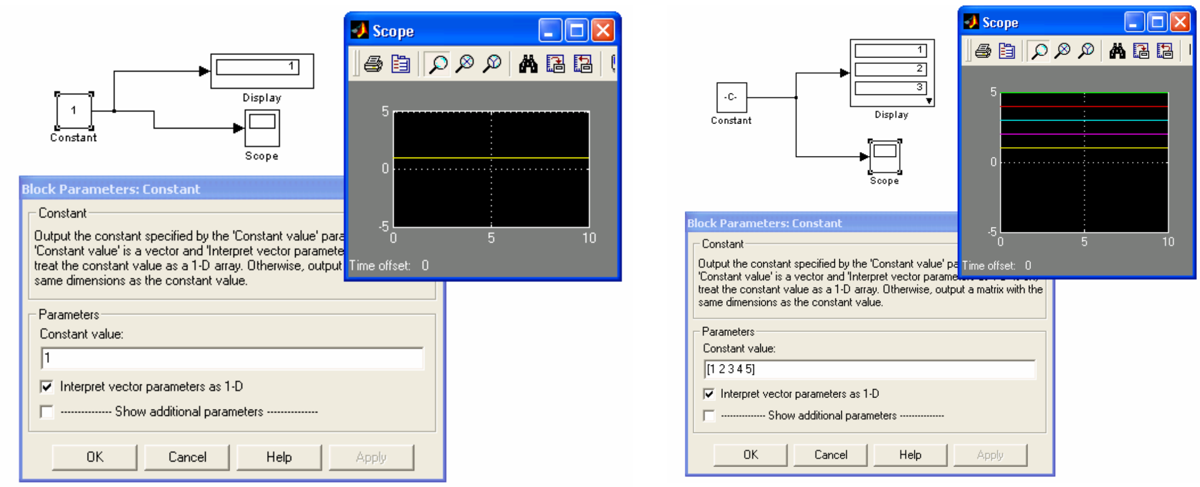


генератор білого шуму з обмеженою смугою частот.

- джерело безперервного сигналу, пропорційного модельному часу (рис. 6.8).



- джерело дискретного сигналу, пропорційного часу.



а) б)

Рис.6.3 Вікно з джерелом постійного впливу з одним значенням (а) та джерелом постійного впливу з багатьма значеннями (б)

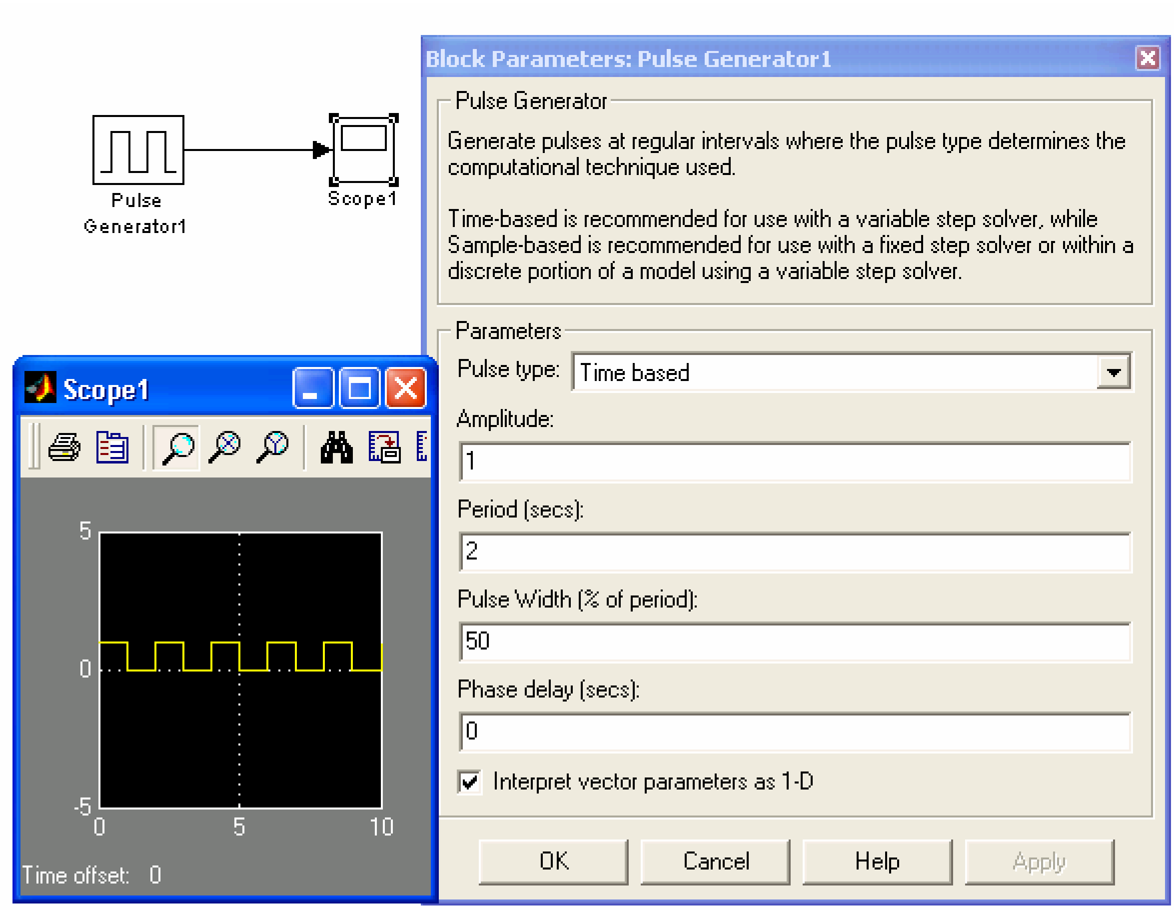


Рис. 6.4 Вікно з джерелом прямокутних імпульсів

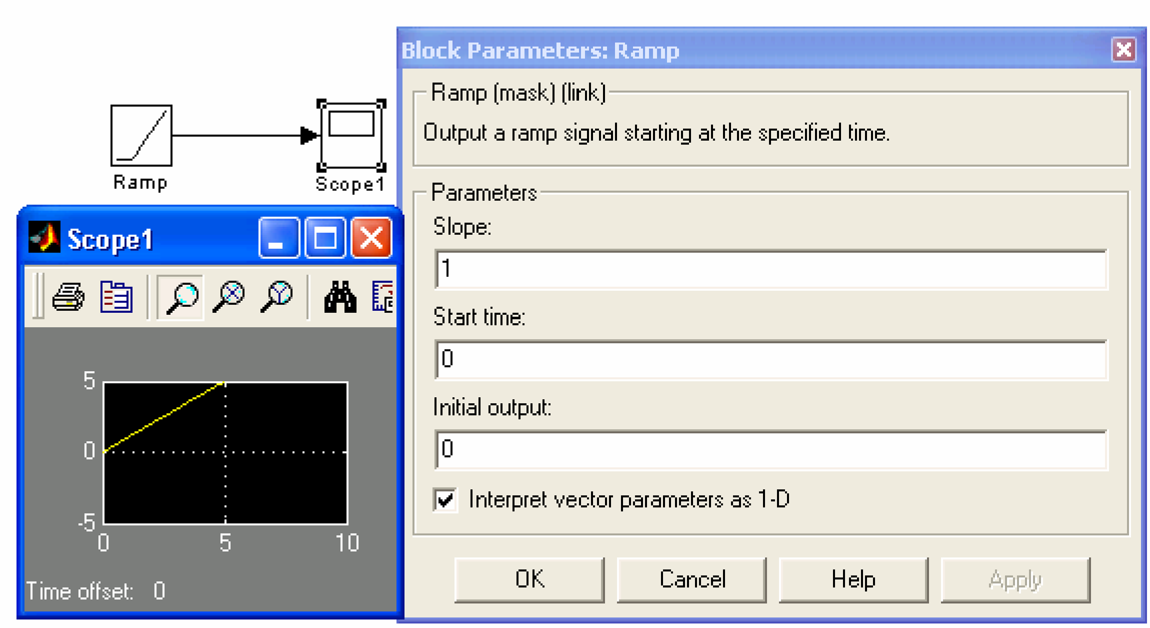


Рис.6.5 Вікно з джерелом лінійно-наростаючого впливу

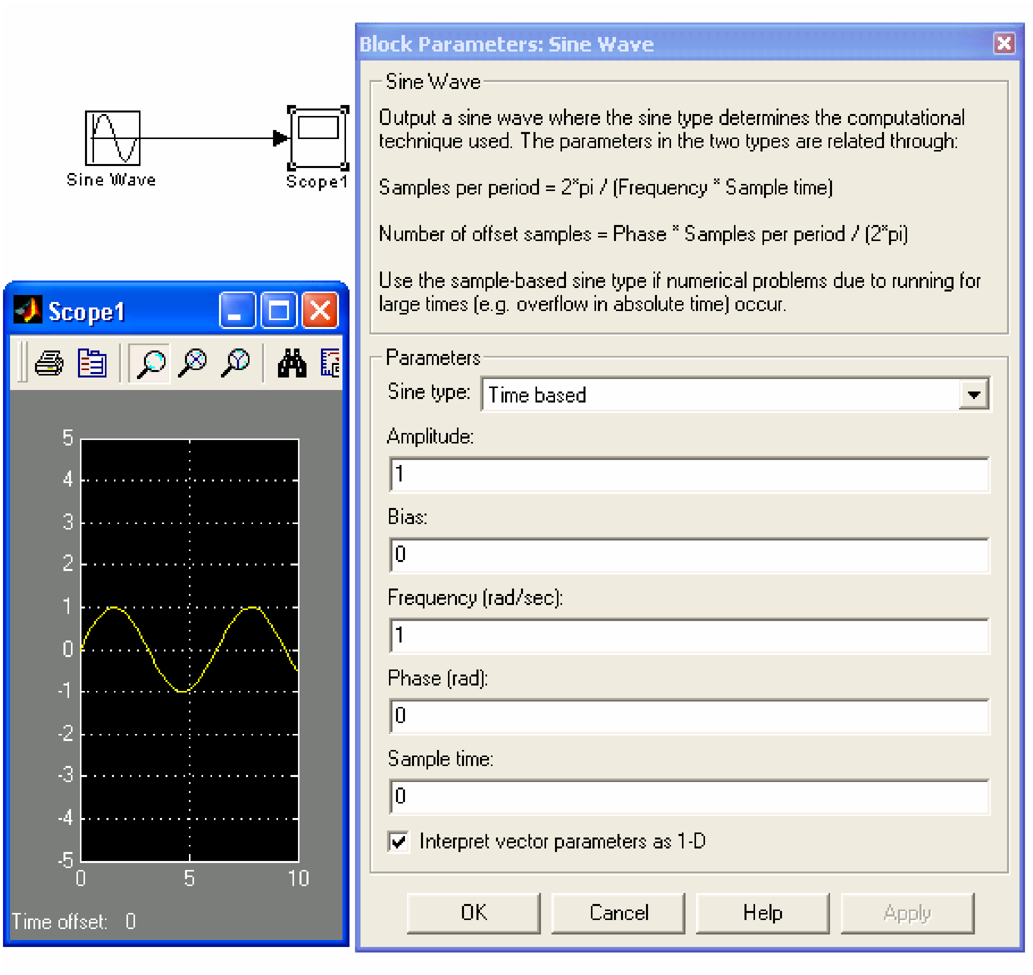


Рис. 6.6 Вікно з джерелом синусоїдального впливу Sine Wave

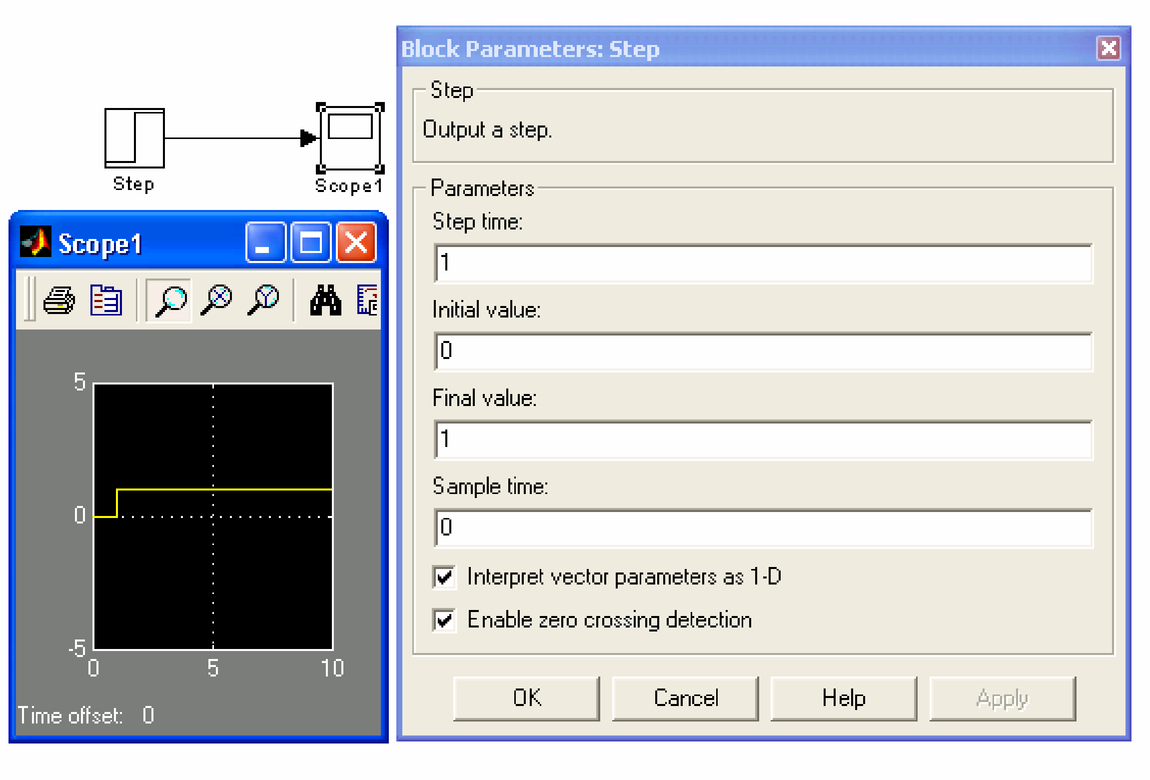


Рис. 6.7 Вікно з джерелом одиночного перепаду

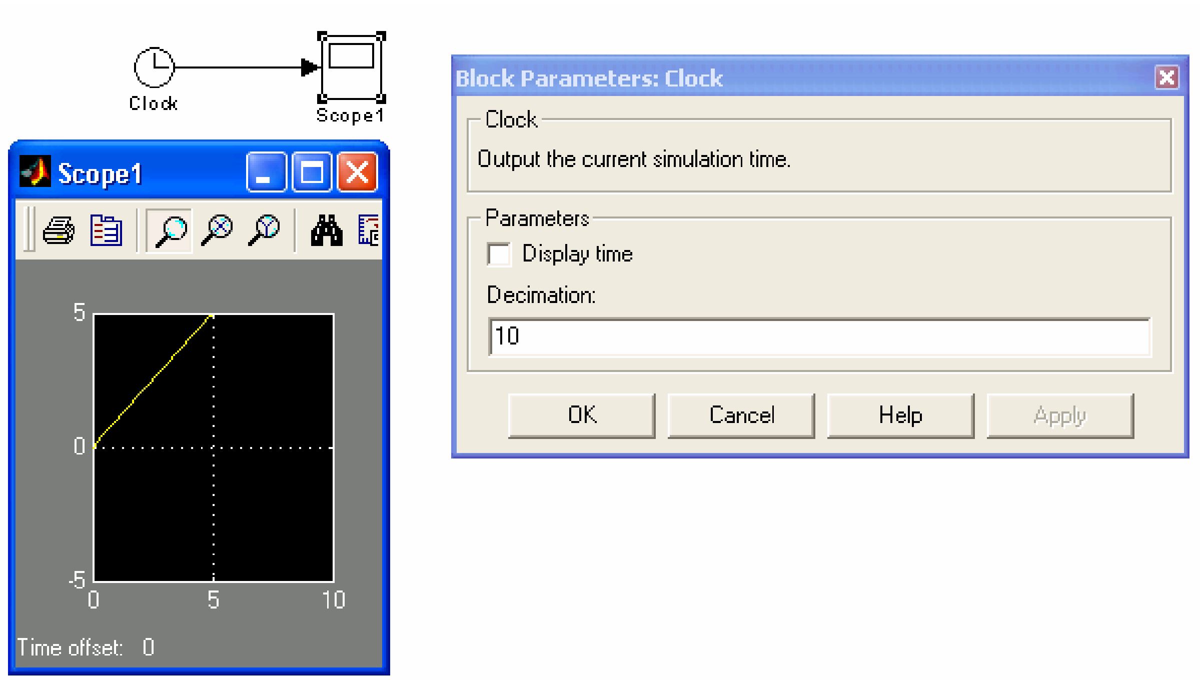


Рис.6.8 Вікно з джерелом безперервного сигналу, пропорційного модельному часу

Завдання на виконання комп’ютерного практикуму

Завдання 1. Створити S-моделі, використовуючи різні джерела, налаштувати параметри джерел (відповідно до варіанту) і зареєструвати сигнали на виходах моделей за допомогою осцилографів (Scope).

1.1. джерело постійного впливу Constant (значення постійної дорівнює номеру варіанту);

1.2. джерело синусоїдального впливу Sine Wave (табл. 6.1);

1.3. джерело наростаючого впливу Ramp (табл. 6.2);

1.4. джерело одиночного перепаду Step (табл. 6.3);

1.5. джерело прямокутних імпульсів Pulse Generator (табл. 6.4).

Завдання 2. Змінити варіант та побудувати нові графіки для кожного типу **джерела.**

Завдання 3. Порівняти отримані графіки та зробити висновки.

Табл.6.1 Варіанти індивідуальних завдань

Варіант	Амплітуда	Частота	Фаза
1	2	3	4
1	0,4	0,4	20
2	0,5	0,2	10
3	0,2	1	15
4	1,5	0,5	100
5	2	0,35	40
6	2,5	1,2	50
7	1,2	1,4	30
8	0,5	1,6	20
9	1,4	0,8	10

1	2	3	4
10	1,6	0,4	1
11	0,8	0,6	20
12	0,5	1,2	50
13	0,7	1,1	40
14	1	1,8	100
15	1,2	2	30
16	1,5	1,2	40
17	1,7	0,5	20
18	0,7	0,6	10
19	1	1,8	5
20	2	0,4	15
21	1,5	0,5	20
22	1,8	0,2	15
23	1,4	1,2	30
24	0,8	0,5	40

Табл.6.2 Варіанти індивідуальних завдань

Варіант	Slope	Start time	Initial value
1	2	3	4
1	0,5	0,4	20
2	0,8	1,0	25
3	1,2	1,3	10
4	1,5	1,0	15
5	1,8	1,35	10
6	2,0	1,25	10

1	2	3	4
7	2,2	0,48	25
8	2,0	2,4	20
9	1,8	1,6	35
10	1,6	1,4	35
11	1,5	1,6	25
12	1,9	1,2	25
13	2,1	1,1	30
14	3,0	1,8	30
15	3,2	0,5	30
16	1,4	1,2	30
17	1,5	0,5	25
18	2,0	0,6	25
19	2,2	1,8	25
20	0,8	0,4	25
21	0,6	0,5	30
22	0,4	0,2	20
23	0,5	1,2	20
24	1,0	0,5	20

Табл.6.3 Варіанти індивідуальних завдань

Варіант	Step time	Start time	Initial value
1	2	3	4
1	1,5	0,4	20
2	1,5	1,0	25

1	2	3	4
3	1,5	1,3	10
4	2,0	1,0	15
5	2,0	1,35	10
6	2,0	1,25	10
7	2,5	0,48	25
8	2,5	2,4	20
9	2,5	1,6	35
10	3,0	1,4	35
11	3,0	1,6	25
12	3,0	1,2	25
13	3,5	1,1	30
14	3,5	1,8	30
15	3,5	0,5	30
16	4,0	1,2	30
17	4,0	0,5	25
18	4,0	0,6	25
19	5,0	1,8	25
20	5,0	0,4	25
21	5,0	0,5	30
22	5,5	0,2	20
23	5,5	1,2	20
24	5,5	0,5	20

Табл.6.4 Варіанти індивідуальних завдань

Варіант	Амплітуда	Період	Фаза
1	0,5	1,0	0
2	0,5	1,0	0
3	0,5	1,0	0
4	1,0	2,0	0
5	1,0	2,0	1,5
6	1,0	2,0	1,5
7	1,5	0,5	1,5
8	1,5	0,5	2,0
9	1,5	0,5	2,0
10	1,5	1,5	2,0
11	2,0	1,5	2,0
12	2,0	1,5	1,0
13	2,5	0,5	1,0
14	2,5	0,5	1,0
15	2,5	0,5	1,5
16	3,0	1,0	1,5
17	3,0	1,0	1,5
18	3,0	1,0	1,5
19	3,5	2,0	2,5
20	3,5	2,0	2,5
21	3,5	2,0	2,5
22	4,0	4,0	2,5
23	4,0	4,0	3,0
24	4,0	4,0	3,0

Контрольні запитання

1. Як запустити пакет Simulink?
2. З яких розділів складається бібліотека Simulink?
3. Які блоки містяться у розділі Sources?
4. Які параметри джерела постійного впливу Constant?
5. Які параметри джерела синусоїдального впливу Sine Wave?
6. Які параметри джерела наростаючого впливу Ramp?
7. Які параметри джерела одиночного перепаду Step?
8. Які параметри джерела прямокутних імпульсів Pulse Generator?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 2.7

Створення S-моделей , використовуючи розділ Continuous

Мета: ознайомлення з розділом Continuous бібліотеки Simulink та набуття навичок створення S-моделей, викоритовуючи розділ Continuous.

Теоретичні відомості

На рис. 7.1 представлено вікно розділу бібліотеки Continuous з неперервними компонентами.

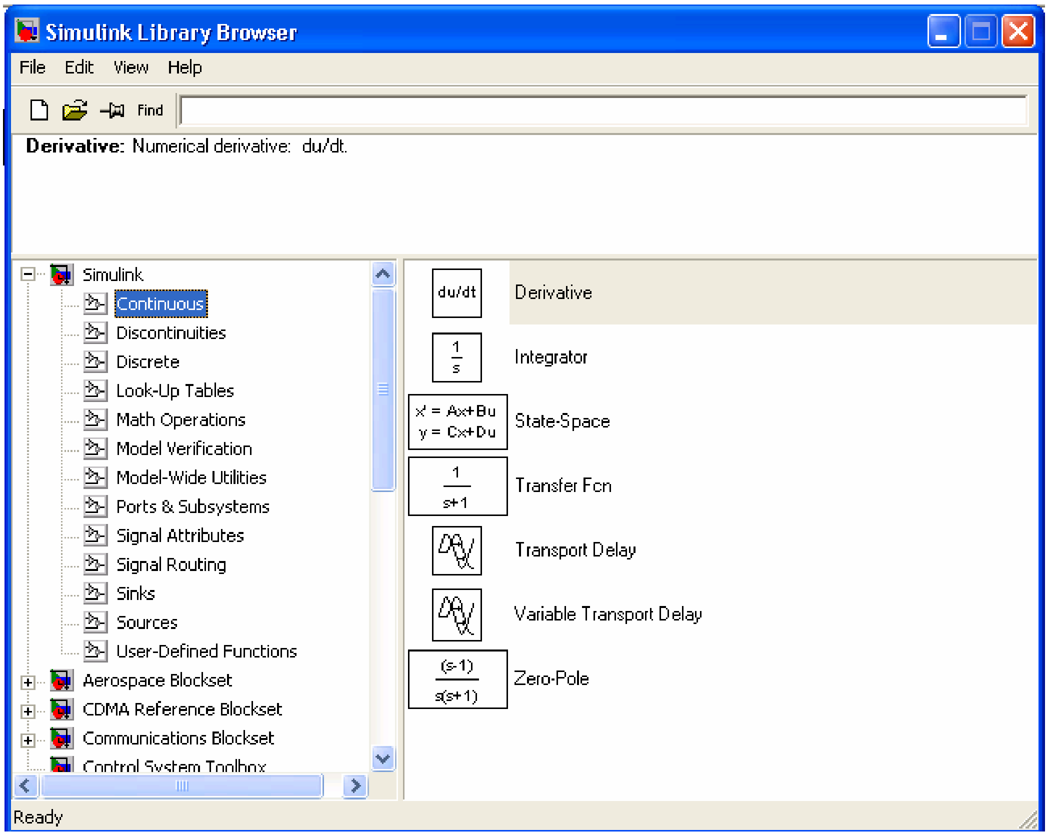
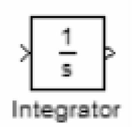


Рис. 7.1 Вікно з блоками розділу Continuous

Зображення блоків, що входять до розділу Continuous бібліотеки Simulink:



- ідеальна інтегруюча ланка. На рис. 7.2 (а) наведено приклад застосування блоку інтегрування для вхідного сигналу у вигляді синусоїдального імпульсу. На рис. 7.2 (б) показано вікно встановлення параметрів інтегруючого блоку.

Вікно параметрів інтегруючого блоку містить такі елементи:

• *External reset* (зовнішній скидання) – тип зовнішнього керуючого сигналу, що обирається зі списку: *none* – ні, *rising* – наростаючий, *falling* – спадаючий, *either* – будь-який;

• *Initial condition Source* – джерело початкового значення вихідного сигналу при інтегруванні. У списку можна вибрати внутрішнє (internal) або зовнішнє (external) джерело;

• *Initial condition* (початковий стан) – установка початкового значення вихідного сигналу при інтегруванні (у вигляді числа, за умовчанням 0);

• *Limit output* – включення / відключення обмеження вихідного сигналу;

• *Upper saturation limit* – верхня межа обмеження вихідного сигналу (за замовчуванням inf, тобто + ∞);

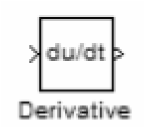
• *Lower saturation limit* – нижня межа обмеження вихідного сигналу (за замовчуванням -Inf, тобто -∞);

• *Show saturation port* – управляє відображенням порту, що виводить рівні обмеження вихідного сигналу;

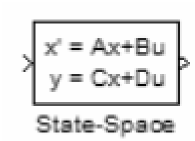
• *Show state port* – управляє відображенням порту стану системи;

• *Absolute tolerance* – абсолютна похибка (за замовчуванням автоматичний вибір – auto).

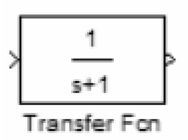
• *Enable zero crossing detections* – включення перевірки переходів через нуль.



- ідеальна диферінцуюча ланка. На рис. 7.3 (а) дано приклад послідовного диференціювання прямокутних імпульсів, а на рис. 7.3 (б) вікно встановлення параметрів блоку диференціювання.



- дозволяє задати лінійну ланку шляхом введення чотирьох матриць його простору станів.



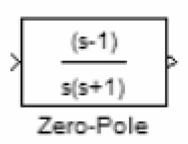
- дозволяє задати лінійну ланку шляхом введення її передавальної функції. На рис. 7.4 показано приклад застосування блоку передатної функції. Вид блоку показаний після встановлення його параметрів у вікні параметрів.

Блок Transfer Fcn має два параметри - вектори коефіцієнтів поліномів чисельника Numerator і знаменника Denominator.

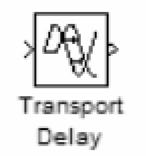
$$W(p) = \frac{b_n p^n + b_{n-1} p^{n-1} + \ldots + b_1 p + b_0}{a_m p^m + a_{m-1} p^{m-1} + \ldots + a_1 p + a_0}.$$

Numerator – значення коефіцієнтів чисельника b_i; їх вводять через пробіл, починаючи з коефіцієнта b_n при старшій похідній;

Denominator – значення коефіцієнтів знаменника a_i; їх вводять через пробіл, починаючи з коефіцієнта a_m при старшій похідній.



- використовується для того, щоб задати ланку допомогою вказівки векторів значень його полюсів і нулів, а також значення коефіцієнта передачі



- забезпечує затримку сигналу на задану кількість кроків модельного часу (не обов'язково ціле число).

- дозволяє задавати керовану ззовні величину затримки.

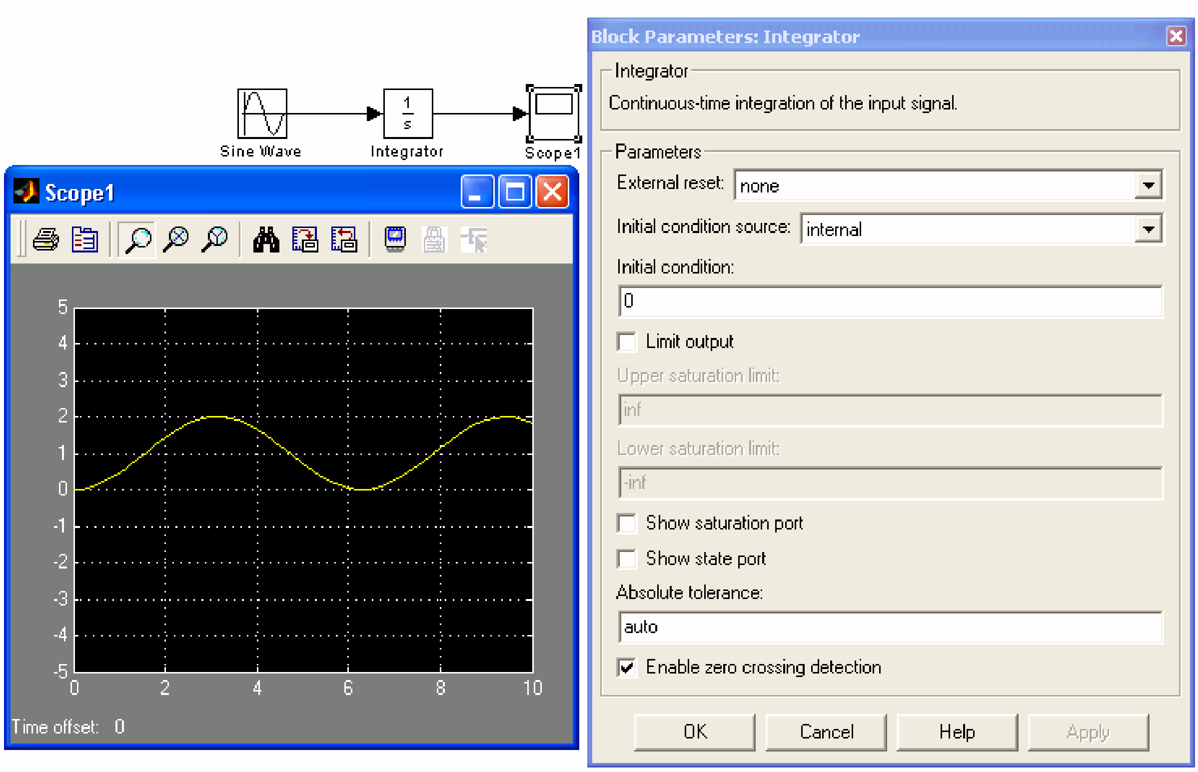


Рис. 7.2 Вікно з прикладом інтегрування синусоїдальних імпульсів та вікно встановлення параметрів інтегруючого блоку

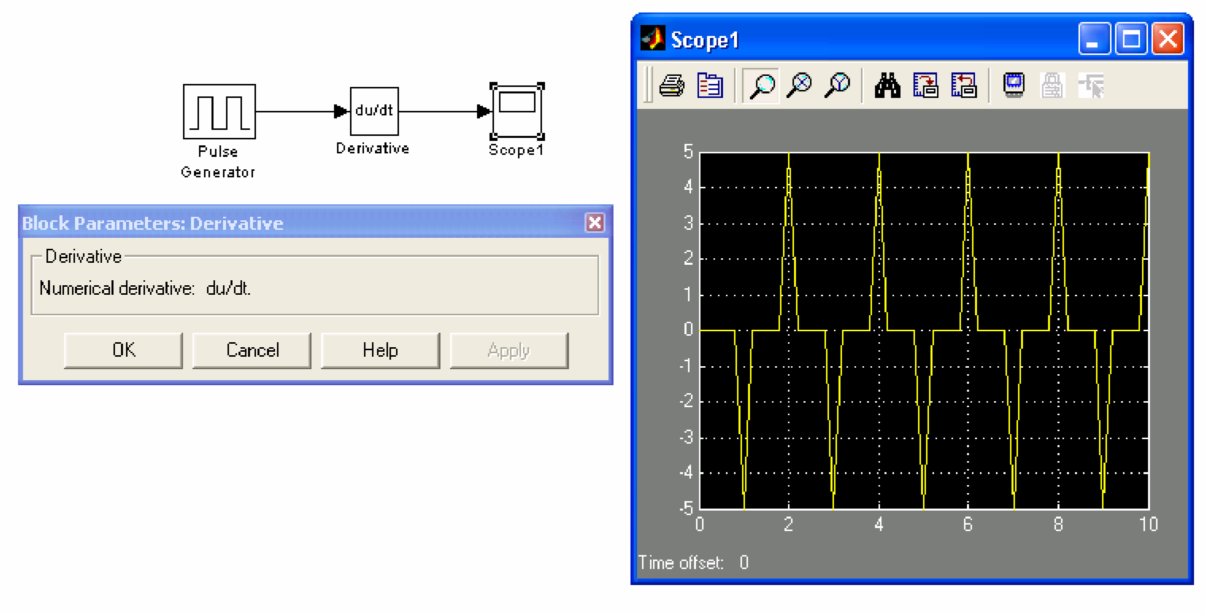


Рис. 7.3 Вікно з прикладом диференціювання прямокутних імпульсів та вікно встановлення параметрів блоку диференціювання

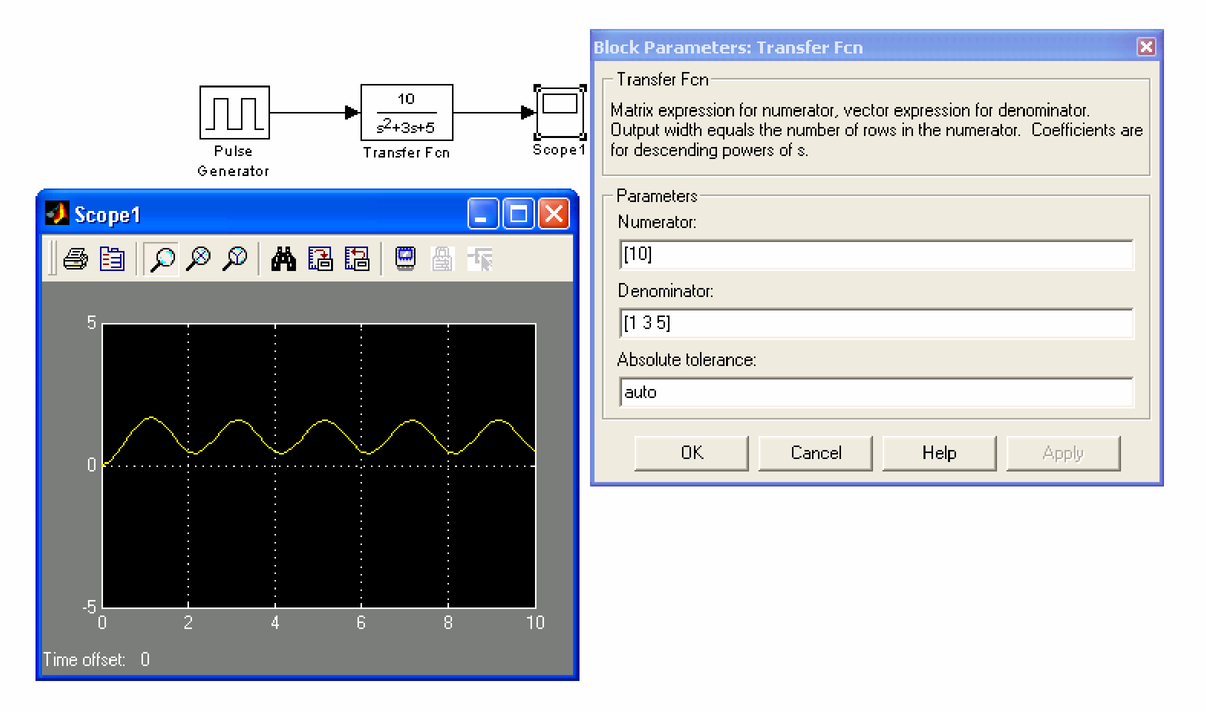


Рис.7.4 Вікно з прикладом застосування блоку Transfer Fcn

Завдання на виконання комп'ютерного практикуму

Завдання 1. Створити S-моделі, використовуючи різні блоки з розділу бібліотеки Continuous та різні джерела, налаштувати параметри блоків (відповідно до варіанту) і зареєструвати сигнали на виходах моделей за допомогою осцилографів.

1.1. Блок інтегрування Integrator

+ джерело постійного впливу Constant (відповідно до номеру варіанту);

+ джерело синусоїдального впливу Sine Wave (табл.6.1);

+ джерело наростаючого впливу Ramp (табл. 6.2);

+ джерело одиночного перепаду Step (табл. 6.3);

+ джерело прямокутних імпульсів Pulse Generator (табл. 6.4).

1.2. Блок диференціювання Derivative

+ джерело постійного впливу Constant (відповідно до номеру варіанту);

+ джерело синусоїдального впливу Sine Wave (табл.6.1);

+ джерело наростаючого впливу Ramp (табл. 6.2);

+ джерело одиночного перепаду Step (табл. 6.3);

+ джерело прямокутних імпульсів Pulse Generator (табл. 6.4).

1.3. Блок передавальної функції Transfer Fcn (табл. 2.1)

+ джерело постійного впливу Constant (відповідно до номеру варіанту);

+ джерело синусоїдального впливу Sine Wave (табл.6.1);

+ джерело наростаючого впливу Ramp (табл. 6.2);

+ джерело одиночного перепаду Step (табл. 6.3);

+ джерело прямокутних імпульсів Pulse Generator (табл. 6.4).

Завдання 2. Змінити варіант та побудувати нові графіки для кожного типу джерела.

Завдання 3. Порівняти отримані графіки та зробити висновки.

Контрольні запитання

1. Які блоки містяться у розділі Continuous**?**
2. Які параметри блоку інтегрування?
3. Які параметри блоку Transfer Fcn?

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 2.8

Створення S-моделі, використовуючи розділ Math Operation

Мета: ознайомлення з розділом Math Operation бібліотеки Simulink та набуття навичок створення S-моделей, викоритовуючи розділ Math Operation.

Теоретичні відомості

На рис. 8.1 представлено вікно розділу бібліотеки Math Operation з неперервними компонентами.

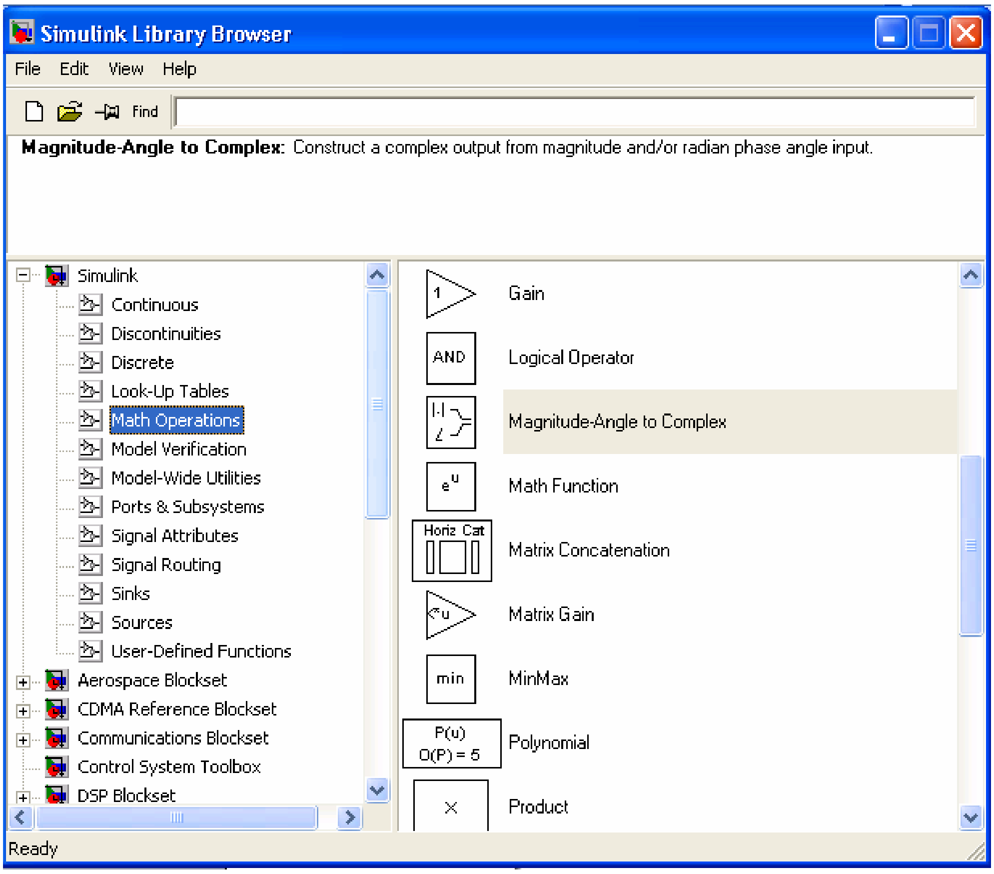
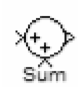
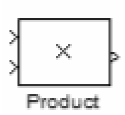


Рис. 8.1 Вікно з блоками розділу бібліотеки Math Operation

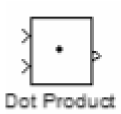
Зображення блоків, що входять до розділу Math Operation бібліотеки Simulink:



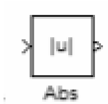
- обчислення суми поточних значень сигналів.



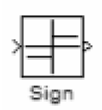
- обчислення звичайного добутку поточних значень сигналів. Блок Product призначений не тільки для множення, а й ділення.



- обчислення скалярного добутку двох векторів

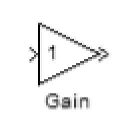


- обчислення абсолютного значення числа

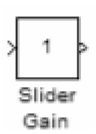


- визначає знака числа.

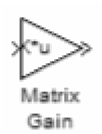
- визначає максимальне або мінімальне значення з усіх сигналів, що надходять на його входи.



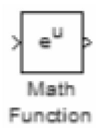
- виконують множення вхідного сигналу на постійний коефіцієнт.



- забезпечує зміну коефіцієнта посилення в процесі розрахунку.



- виконують множення вхідного сигналу на постійний коефіцієнт



- виконує обчислення математичної функції. До параметрів блоку відносяться:

1. ***Function*** – Вид обчислюється функції (вибирається зі списку):

exp – експоненціальна функція;

log – функція натурального логарифма;

10 ^ u – обчислення ступеня 10;

log10 – функція десяткового логарифма;

magnitude ^ 2 – обчислення квадрата модуля вхідного сигналу;

square – обчислення квадрата вхідного сигналу;

sqrt – квадратний корінь;

pow – піднесення до степеня;

conj – обчислення комплексно-спряженого числа;

reciprocal – обчислення частки від розподілу вхідного сигналу на 1;

hypot – обчислення кореня квадратного з суми квадратів вхідних сигналів (гіпотенузи прямокутного трикутника за значеннями катетів);

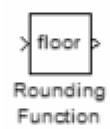
rem – функція, що обчислює залишок від ділення першого вхідного сигналу на другий;

mod – функція, що обчислює залишок від ділення з урахуванням знака;

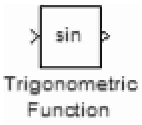
transpose – транспонування матриці;

hermitian – обчислення ермітової матриці.

2. ***Output signal type*** – тип вихідного сигналу (вибирається зі списку): **auto** – автоматичне визначення типу; **real** – дійсний сигнал; **complex** – комплексний сигнал (тобто сигнал складається з комплексних чисел).



- округлення – дозволяє у вікні установки параметрів вибрати зі списку одну з чотирьох функцій округлення: *floor* – до найближчого меншого цілого; *ceil* – до найближчого більшого цілого; *round* – до найближчого цілого; *fix* – до цілого, отриманого відкиданням цілої частини.



- виконує обчислення тригонометричної функції. До параметрів блоку відносяться:

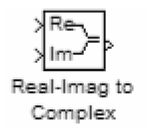
1. ***Function*** – вид обчислюваної функції (вибирається зі списку): **sin**, **cos**, **tan**, **asin**, **acos**, **atan**, **atan2**, **sinh**, **cosh** і **tanh**.

2. ***Output signal type*** – тип вихідного сигналу (вибирається зі списку): **Auto** – автоматичне визначення типу, **Real** – дійсний сигнал; **Complex** – комплексний сигнал.

- обчислює дійсну і (або) уявну частину комплексного числа. До параметрів блоку відноситься:

Output – Вихідний сигнал (вибирається зі списку): *Real* – дійсна частина, *Image* – уявна частина, *RealAndImage* – дійсна і уявна частина.

- обчислює модуль і (або) аргумент комплексного числа. Параметри блоку: **Output** – Вихідний сигнал (вибирається зі списку): *Magnitude* – модуль; *Angle* – аргумент, *MagnitudeAndAngle* – модуль і аргумент.



- обчислює комплексне число за його дійсною та уявною частинами.

- обчислює комплексне число за його модулем і аргументом.

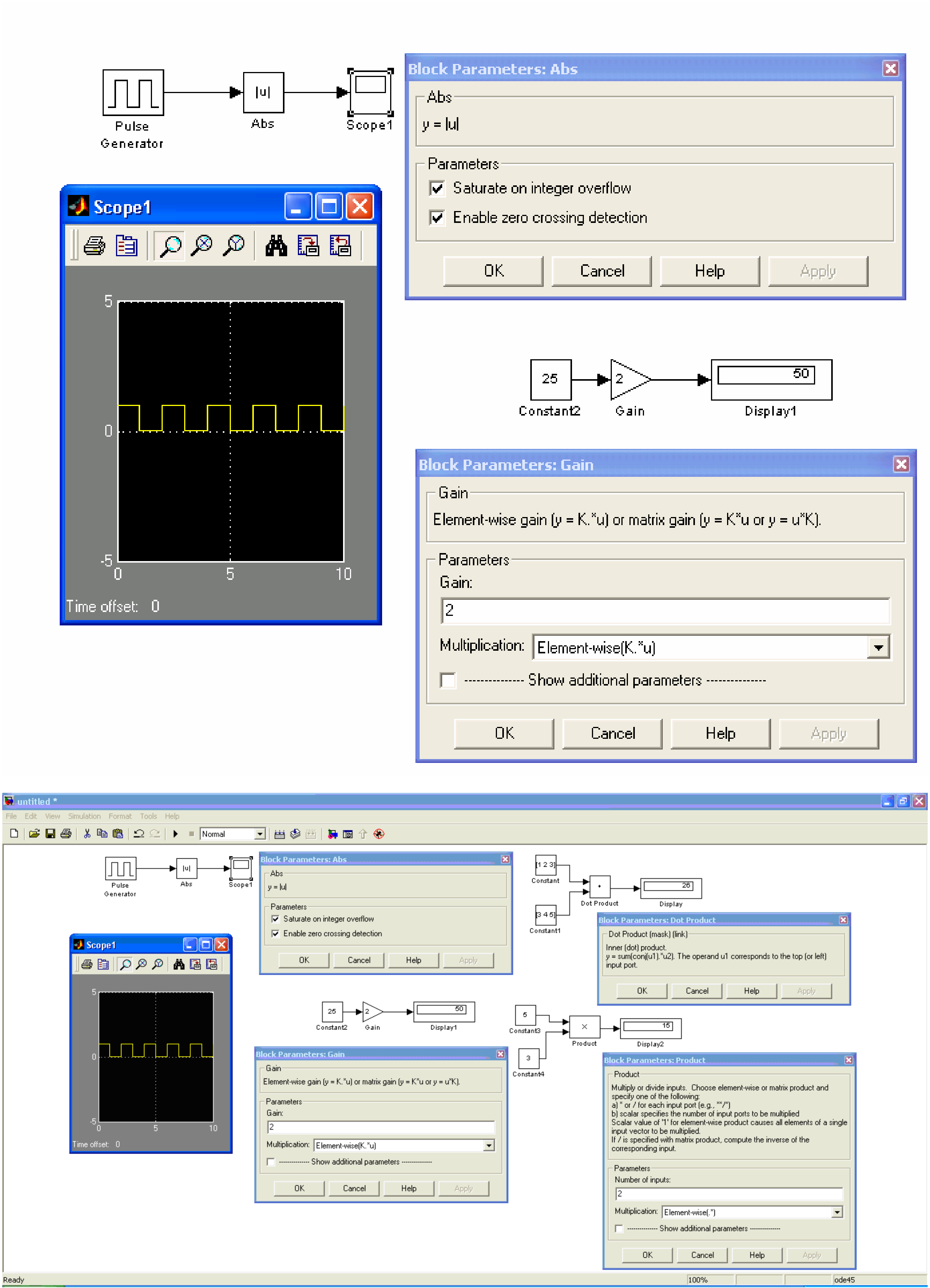


Рис. 8.2 Вікна з прикладами застосування блоків арифметичних операцій

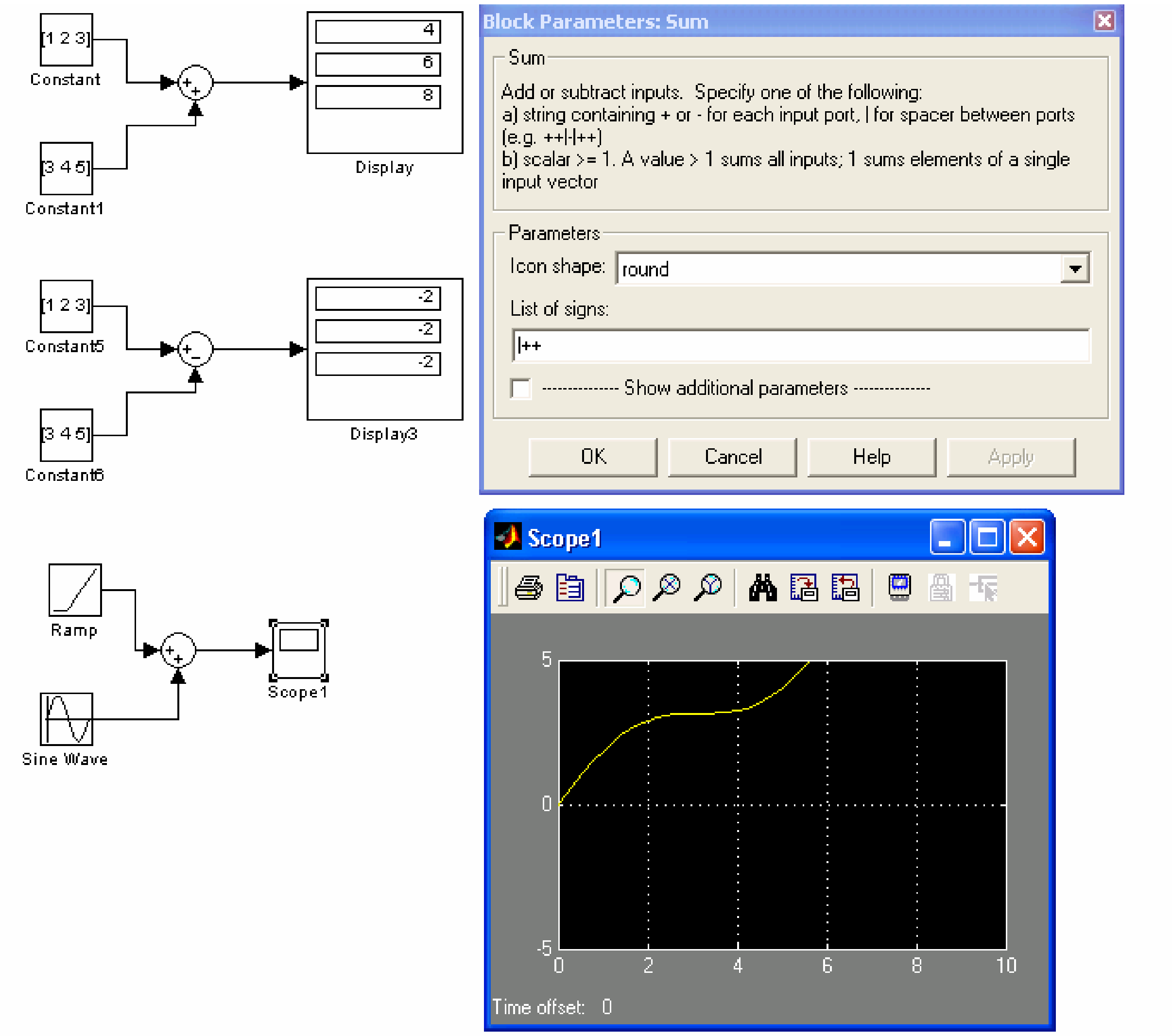


Рис. 8.3 Вікно з прикладами застосування блоку додавання

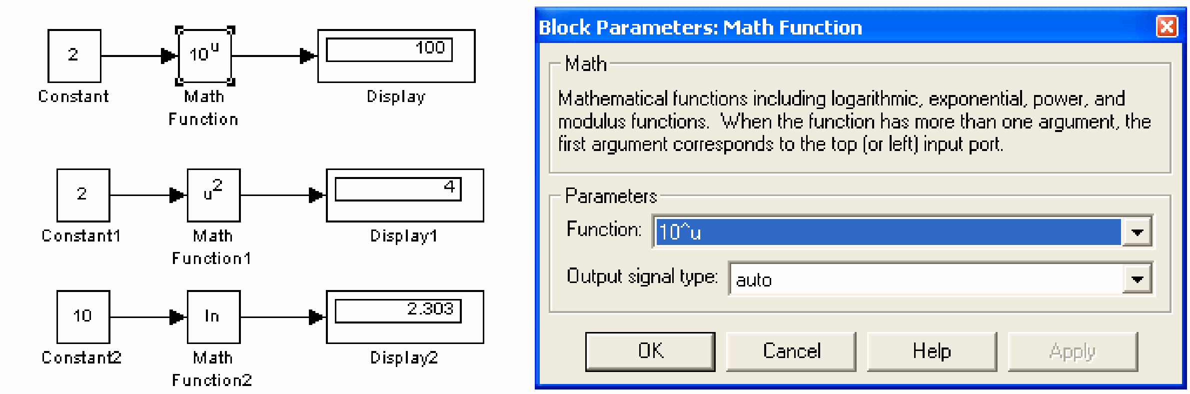


Рис.8.4 Вікно з прикладом використання блоку Math Function

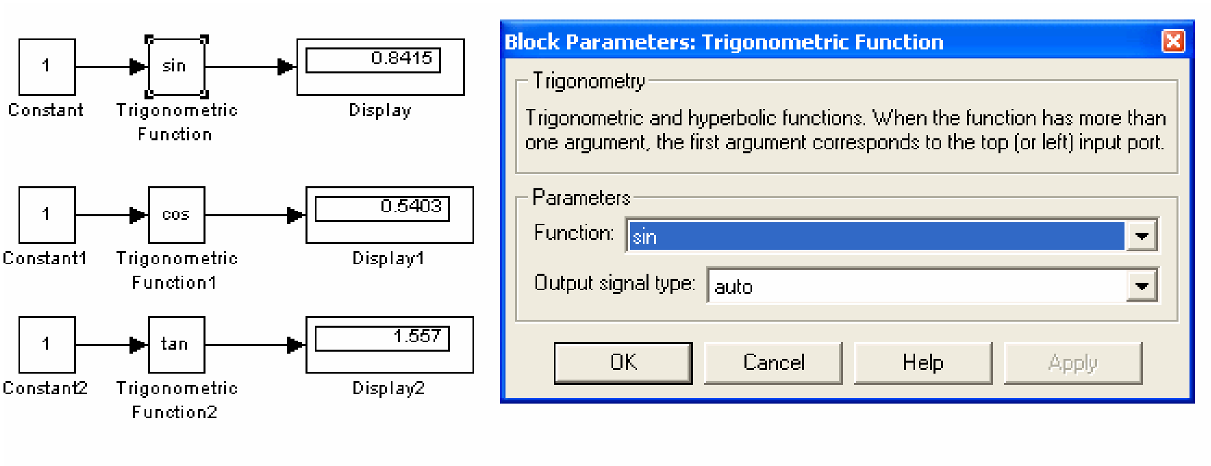


Рис.8.5 Вікно з прикладом використання блоку Trigonometric Function

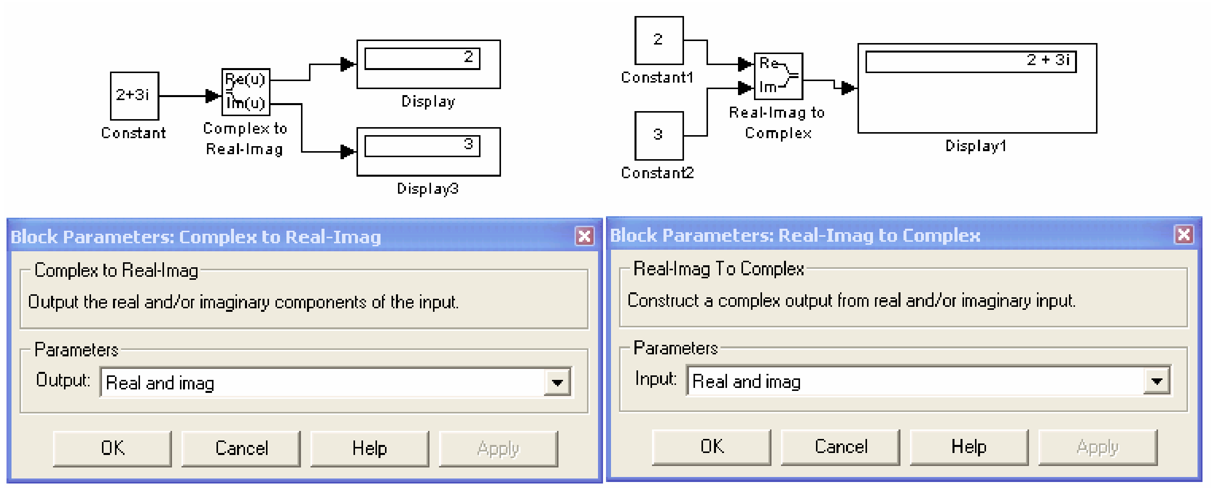
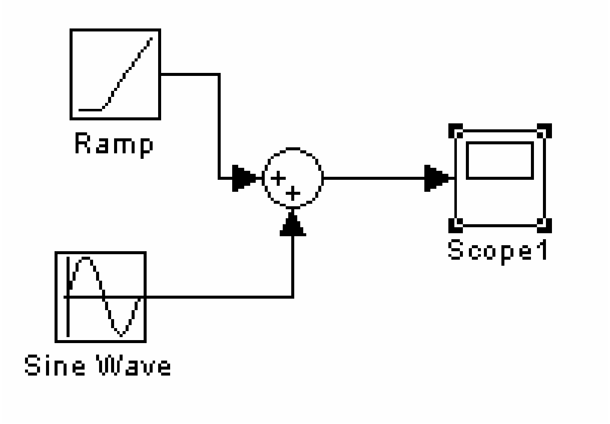
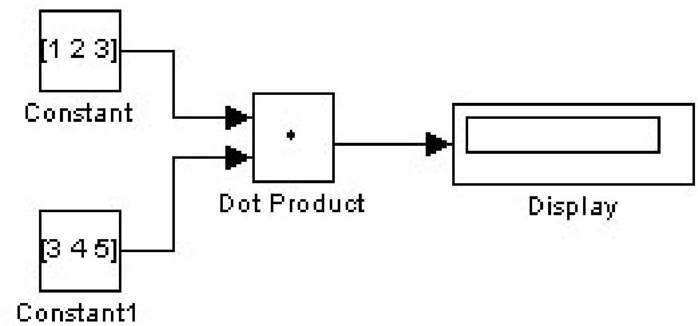
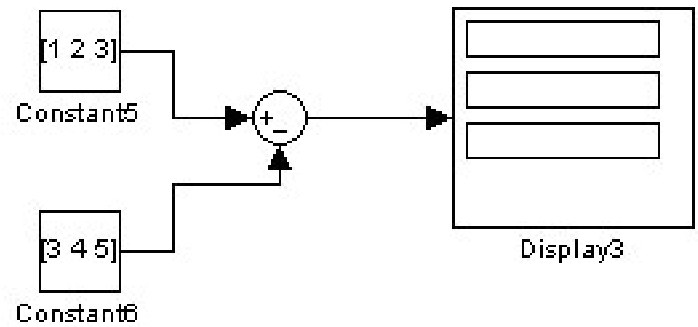


Рис.8.6 Вікно з прикладами використання блоку Complex to Real-Imag

Завдання на виконання комп'ютерного практикуму

Завдання 1. Створити S-моделі, налаштувати параметри блоків, відповідно до варіанту (табл.6.1-6.4), і зареєструвати сигнали на виходах моделей за допомогою осцилографів або дисплею:

1.1.	
1.2.	 Значення векторів – з таблиці 3.1.
1.3.	 Значення векторів – з таблиці 3.1.

Завдання 2. Створити S-моделі, що генерує вказану функцію, налаштувати параметри блоків, відповідно до варіанту (табл. 8.1), і зареєструвати сигнали на виходах кожного з логічно завершених блоків моделі за допомогою осцилографів.

Табл. 8.1 Варіанти індивідуальних завдань

Варіант	Функція	Залежність аргументу від часу моделювання
1	2	3
1	$y=5*\sin(e^{2x^2-\ln\left\|x^3\right\|})$	$x=\cos t+tgt$
2	$y=5*\cos(5^{2x^2-\ln\left\|x^{\sqrt{2}}\right\|})$	$x=arctg\,t$
3	$y=5*\log_5\left\|\sqrt{e^{2x^2}}\right\|$	$x=e^{-t}\cos t$
4	$y=5*\sin(e^{2x^2-\ln\left\|x^3+10\right\|})+\sin(e^{2x^2})$	$x=\cos t+tgt$
5	$y=5*\sin(e^{2x^2})$	$x=floor\left(\cosh t/\left(t^3+1\right)\right)$
6	$y=5*e^{e^{2x^2-\sin\left\|x^3\right\|}}$	$x=t$
7	$y=5*\sin(2x^2-\ln\left\|x^3\right\|)+3*\cos(\ln\left\|x^3\right\|)$	$x=\cos t+tgt$
8	$y=\sqrt{10+\frac{x^5}{\sin(25x+2)+1.5}}+5^x$	$x=2*\sin t$
9	$y=\sqrt{\left\|120-\frac{\sin(25x+2)+1.5}{x^5+0.1}\right\|}$	$x=e^t$
10	$y=5*tg(e^{2x^2-\arcsin\left\|x^3\right\|})$	$x=\cos e^{t+2}$
11	$y=\frac{3*\sqrt[5]{x^3+20x}}{\left(x+1\right)^x+0.5}\ln\left(\left\|x\right\|+0.05\right)$	$x=t+1$
12	$y=\frac{3*\sqrt[5]{x^3}}{\left(x+1\right)^3+e^{0.5x}}\cos\left(20x+0.05\right)$ $+\sin\left(20x-5\right)$	$x=1+\sin t$

1	2	3
13	$y=10*e^{\ln\left(x^5+\sin 1.5x\right)}+\sin(2x^2)$	$x=e^{-t}\cos t$
14	$y=5*\sin(e^{2x^2-\ln\left\|x^3\right\|})$	$x=\arccos t+\sqrt{t}$
15	$y=5\dfrac{\sin(e^x)}{10\cos\left(\sqrt{x}+15\right)+\sin^2\left(x^3/\left(x+1\right)\right)}$	$x=\sin t$
16	$y=12\dfrac{\sin(e^{x^2})}{10\cos\left(\left\|x^{4/9}\right\|+15\right)+\sin\left(x/\left(x+1\right)\right)}$ $+\arcsin(2x)$	$x=t$
17	$y=10\sin(e^{2x^2})+tg\sqrt{\left\|x\right\|}+\dfrac{e^x}{x^3}$	$x=\cos t$
18	$y=\cosh\left(\sin(e^{2x^2})\right)+tgx+x^3e^x$	$x=t^{2/3}$
19	$y=\sinh\left(\cos(e^{2x^2})\right)+ctgx+x^{-3}e^x$	$x=\cos t$
20	$y=arctg(e^{2\sqrt[5]{x^2}-\sin\left\|x^3\right\|})+\cos x$	$x=t$
21	$y=arctg(10e^{2\sqrt[5]{x^2}-\sin\left\|x^3\right\|}+\cos x)$	$x=\sin 2t$
22	$y=5^{\log_{10}\frac{x^2+5x}{\sqrt{\sin 2x}}}+e^{-x}$	$x=ceil\left(100\cos t\right)$
23	$y=5^{\log_8\left\|x^2+5x\right\|}+e^{-\sin x}$	$x=round\left(120\sin t\right)$
24	$y=\sqrt{tg5x^2+e^{\ln 10x^3+arctg10x}}+\dfrac{x^3-1}{x^2+1}$	$x=arcctgt$
25	$y=\sqrt{ctg\dfrac{5x^2}{\sqrt{\left\|x\right\|+1}}+e^{arctgx}}+\dfrac{\ln\left(\left\|x^3\right\|+0.2\right)-1}{x^2+1}$	$x=\sin t$

Контрольні запитання

1. Які блоки містяться у розділі Math Operation?
2. За допомогою яких блоків можна здійснити елементарні арифметичні операції (множення, ділення, додавання, віднімання)?
3. Яку функцію виконує блок Gain?
4. Які параметри блока Math Function?
5. Які параметри блока Trigonometric Function?

ЛІТЕРАТУРА

1. Лазарєв Ю.Ф. Моделювання на ЕОМ: Навчальний посібник / Ю.Ф. Лазарєв. – Київ: Корнійчук, 2007. – 290 с.
2. Лазарєв Ю.Ф. Початки програмування у середовищі MATLAB: Навчальний посібник / Ю.Ф. Лазарєв. – Київ: Корнійчук, 1999. – 160 с.
3. Черных И.В. Simulink: Инструмент моделирования динамических систем / И.В.Черных. – М.: Диалог-МИФИ. 2003. – 252 с.
4. Дьяконов В. П. Simulink 5/6/7: Самоучитель. – М.: ДМК-Пресс, 2008. – 784 с.
5. Барановская Г.Г. Микрокалькуляторы в курсе высшей математики : Практикум : Учеб. пособие для вузов / Г. Г. Барановская, И. Н. Любченко. – К : Вышая шк. Головное изд-во, 1987. – 288 с.
6. Барановская Г.Г., Любченко И.Н. Микрокалькуляторы в курсе высшей математики: Практикум / Г. Г. Барановская, И. Н. Любченко. – Киев, Вышая школа, 1987. – 288с.
7. Бахвалов Н. С. Численные методы / Н. С. Бахвалов. – М.: Наука, 1973. – 632с.
8. Демидович Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон. – М.: Наука, 1970. – 664с.
9. Березин М.С. Методы вычислений / М.С. Березин, Н.П. Жидков. – М.: Наука, 1966. т.1,2. – 620с.