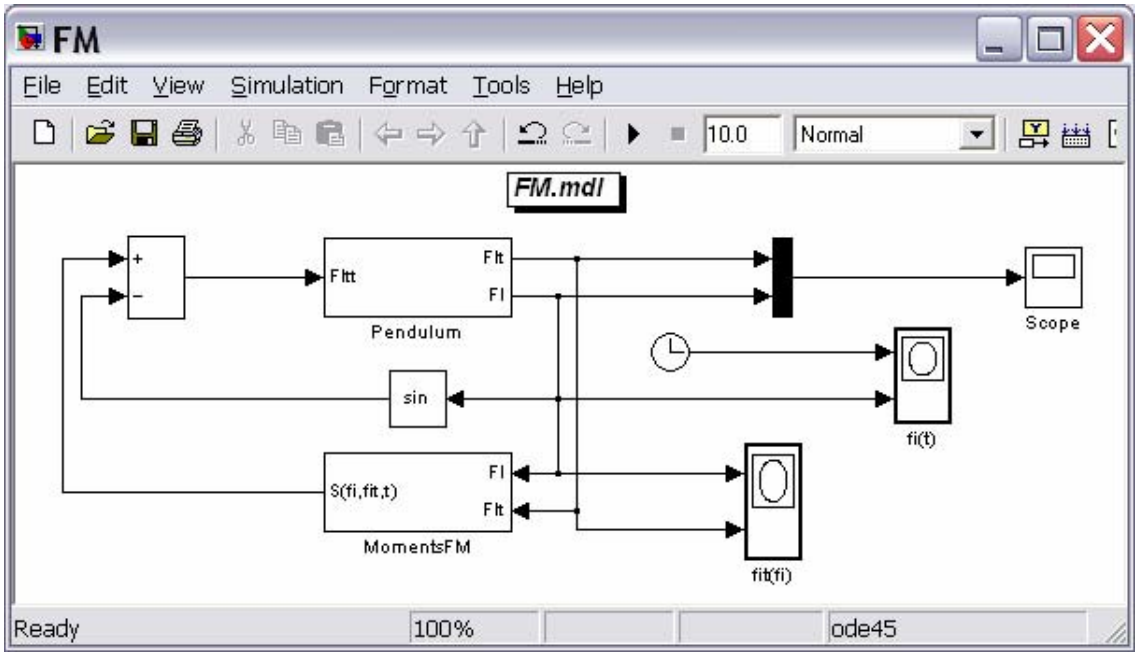
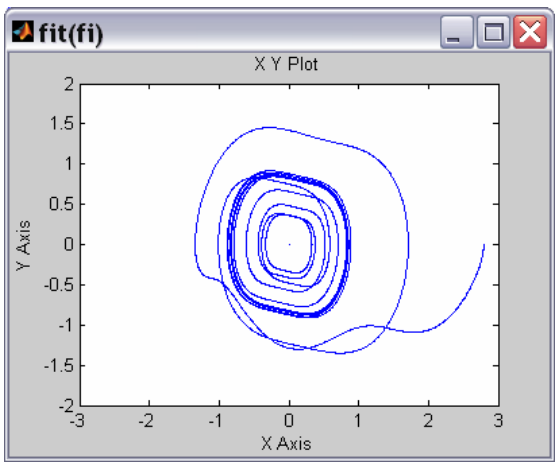
Ю. Ф. Лазарєв

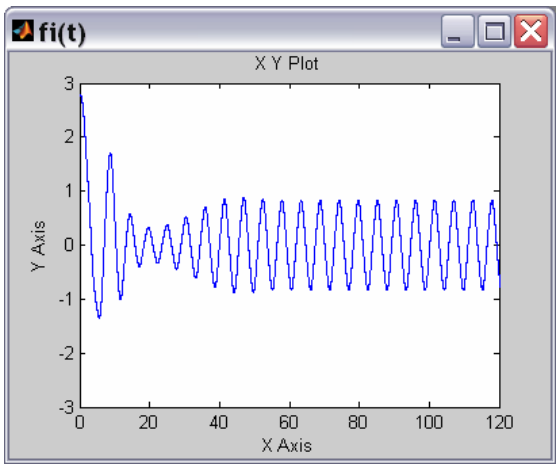
MATLAB і моделювання динамічних систем

Навчальний посібник

Глава 3. Пакет програм Simulink







Київ – 2009

УДК 681.3(0.75)
Л17

Лазарєв Ю. Ф.
Л17 MATLAB і моделювання динамічних систем. Навчальний посібник. Глава 3. Пакет програм Simulink. – Київ: НТУУ "КПІ", 2009. – 79 с.

Зміст

3. Пакет програм SimuLink

Пакет SimuLink дозволяє здійснювати дослідження (моделювання у часі) поводження динамічних нелінійних систем. Утворення чисельної моделі досліджуваної системи здійснюється *шляхом графічного складання* у спеціальному вікні *схеми з'єднань елементарних* візуальних блоків, що містяться в бібліотеках SimuLink. Кожний блок фактично являє собою математичну програму. Лінії з'єднання блоків перетворюються на зв'язки між цими програмами, які дозволяють визначити послідовність виклику програм і пересилання інформації. У результаті такого складання утворюється програмна модель, яку надалі називатимемо S-моделлю і яка зберігається у файлі з розширенням **.mdl**. Такий процес утворення обчислювальних програм прийнято називати *візуальним програмуванням*.

Створення моделей у пакеті SimuLink ґрунтується на використанні технології Drag-and-Drop (*Перетягни й Залиш)*. Як "цеглинки" при побудові S-моделі використовуються модулі (блоки), що зберігаються в бібліотеці SimuLink. S-модель може мати ієрархічну структуру, тобто складатися з моделей більш низького рівня, причому кількість рівнів ієрархії є практично необмеженою. Протягом моделювання є можливість спостерігати за процесами, що відбуваються в системі. Для цього використовуються спеціальні блоки "оглядові вікна", що входять до складу бібліотеки SimuLink. Склад бібліотеки SimuLink може бути поповнений користувачем за рахунок розробки власних блоків.

Використання SimuLink є особливо зручним при моделюванні систем, які складаються із з'єднаних певним чином окремих функціональних пристроїв, поведінка яких описується відомими залежностями. Тоді схема з'єднань візуальних блоків у вікні блок-схеми S-моделі збігається з реальними зв'язками між цими пристроями. Ця обставина суттєво спрощує програмний аналіз і синтез систем автоматичного керування.

3.1. Запуск SimuLink

Запуск SimuLink складається з двох процедур:

1) виклику створеної раніше S-моделі шляхом введення у командному рядку командного вікна MatLAB ймення відповідного MDL-файла або обравши команду **New > Model** у меню **File**, якщо S-модель ще тільки потрібно утворити; у першому випадку при цьому виникне нове вікно з блок-схемою, а у другому - порожнє вікно **untitled** (вікно, де має бути утворена блок-схема нової S-моделі, MDL-файлу, рис. 3.1);
2) виклику браузера бібліотеки SimuLink шляхом натиснення відповідної піктограми у лінійці інструментів командного вікна; виникне вікно **Simulink Library Browser** (рис. 3.2), яке містить перелік основних поділів бібліотеки SimuLink; біль зручним є користування вікном **Library: simulink** (рис. 3.3), яке викликається за допомогою контекстного меню. В ньому представлені графічні позначення поділів бібліотеки

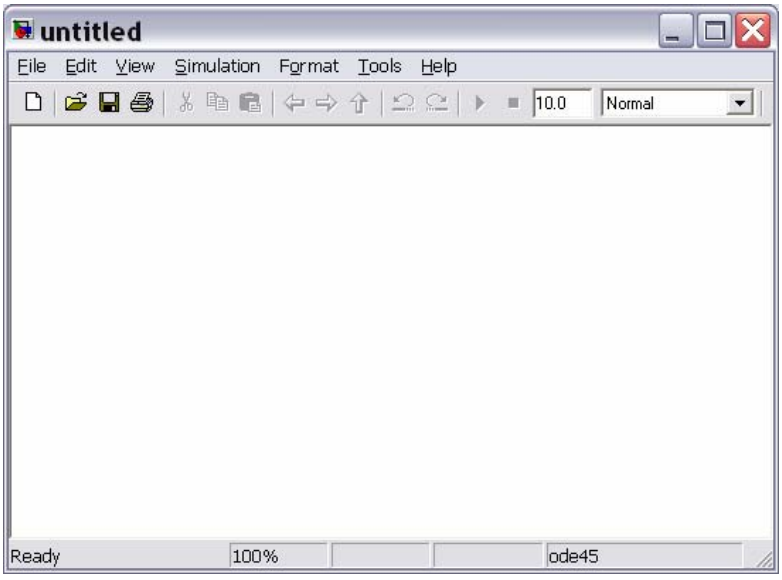


Рис. 3.1. Вікно блок-схеми S-моделі

Усі вікна мають подібну структуру і містять рядок меню і робоче поле.

Меню **File** (Файл) містить команди роботи з MDL-файлами, меню **Edit** (редагування) - команди редагування блок-схеми й роботи з бібліотекою, а меню **View** (Подання) - команди змінювання зовнішнього вигляду вікна.

У меню **Simulation** (Моделювання) містяться команди керування моделюванням, а в меню **Format** (Формат) - команди редагування формату (тобто зовнішнього зображення) блоків схеми й блок-схеми в цілому.

Якщо до складу робочої конфігурації MatLAB включений додаток *Real-Time-Workshop,* то меню доповнюється поділом **Tools** (Інструменти), що містить засоби роботи з ним.

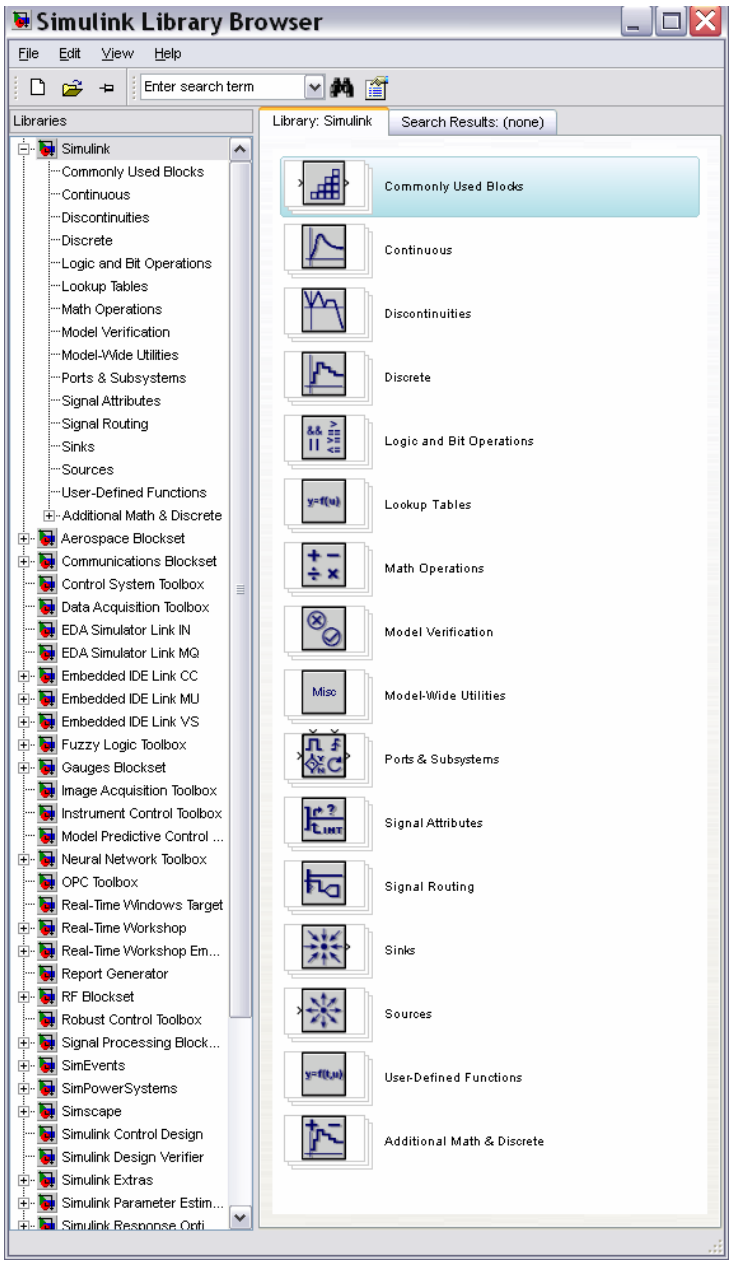


Рис. 3.2. Вікно браузера бібліотеки Simulink

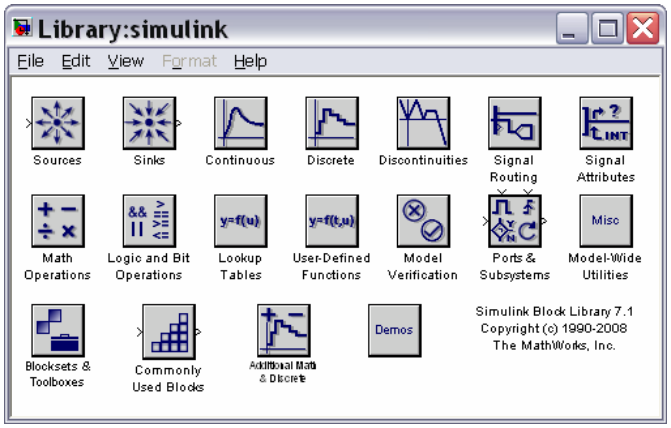


Рис. 3.3. Вікно Library: simulink

3.2. Бібліотека SimuLink

В основі блок-схем S-моделей лежать елементарні блоки, які дозволяють зв’язати блок схему з середовищем Matlab і забезпечити функціонування в ньому S-моделі як програми. Ці блоки містяться у головній бібліотеці пакету ***Simulink***, яка має ту саму назву.Бібліотека блоків SimuLink є набором візуальних об'єктів, використовуючи які, можна, з'єднуючи окремі модулі між собою лініями функціонального зв'язку, складати функціональну блок-схему будь-якого устрою.

Бібліотека блоків складається з 17 поділів. Десять з них є головними і не можуть змінюватися користувачем:

- ***Sources*** (Джерела)*;*
- ***Sinks*** (Приймачі)*;*
- ***Continuous*** (Лінійні неперервні елементи);
- ***Discrete*** (Дискретні елементи)*;*
- ***Discontinuities*** (Розривні елементи)
- ***Signal Routing*** (Пересилання сигналів)*;*
- ***Math Operations*** (Математичні операції);
- ***Logic & Bit Operations*** (Логічні та бітовіоперації);
- ***User-defined Functions*** (Функції, що визначаються користувачем);
- ***Ports & Subsystems*** (Порти та підсистеми).

Блоки, що входять у поділ ***Sources*** (Джерела), призначені для *формування сигналів*, які забезпечують керування роботою S-моделі в цілому або окремих її частин. Усі *блоки-джерела мають по одному виходу і не мають входів.*

Блоки, зібрані в поділі ***Sinks*** (Приймачі)*, мають тільки входи і не мають виходів*. Умовно їх можна поділити на 3 види:

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

Поділ ***Continuous*** (Лінійні неперервні елементи) містить блоки, які можна умовно поділити на дві групи:

- блоки лінійних ланок неперервного часу;
- блоки лінійних стаціонарних ланок із затримкою.

У поділ ***Discrete*** (Дискретні елементи) входять блоки, за допомогою яких у моделі може бути описане поводження дискретних систем. Розрізняють два основних типи таких систем: *системи з дискретним часом* і *системи з дискретними станами.* Блоки, що входять у поділ ***Discrete*** забезпечують моделювання як тих, так і інших.

Поділ ***Discontinuities*** (Розривні елементи) містить 12 блоків, які релізують кусково-лінійні залежності.

У поділі ***Signal Routing*** (Пересилання сигналів) розташовані блоки, що здійснюють об’єднання, роз’єднання сигналів, їх переключення, пересилання тощо.

Блоки поділу ***Math Operations*** (Математичні операції) реалізують перетворення сигналів, що подаються на входи, за різними типовими математичними залежностями, векторно-матричні операції і перетворення комплексних векторів.

У поділі ***Logic & Bit Operations*** (Логічні та бітові операції) зосереджені блоки, що здійснюють логічні і бітові операції з сигналами, які поступають ло їхнього входу.

Блоки, що входять у склад поділу ***User-defined Functions*** (Функції, що визначаються користувачем) призначені для утворення користувацьких блоків.

Нерешті, у поділ ***Ports & Subsystems*** (Порти та підсистеми) входять блоки, яки здійснюють зв'язок між моделями різних рівнів ієрархії, а також дозволяють утворити підсистеми, тобто моделі більш низького рівня ієрархії.

Щоб перейти у вікно відповідного поділу бібліотеки, у якому розташовані графічні зображення блоків, достатньо подвійно клацнути мишкою на піктограмі цього поділу

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

Технологія перетягування зображення блока така: курсор мишки потрібно встановити на зображенні обраного блока у вікні поділу бібліотеки, потім натиснути ліву клавішу мишки і, не відпускаючи її, пересунути курсор на поле складання схеми, після чого відпустити клавішу. Аналогічно здійснюються з'єднання у схемі лініями виходів одних блоків із входами інших блоків: курсор мишки підводять до потрібного виходу певного блока (при цьому курсор має набути форму хрестика), натискують ліву клавішу і, не відпускаючи її, курсор переміщують до потрібного входу іншого блока, а потім відпускають клавішу. Якщо з'єднання зроблено вірно, на вході останнього блоку виникне зображення чорної затушованої стрілки.

3.2.1. Поділ Sinks (Приймачі)

Після переходу до поділу ***Sinks*** на екрані з'являється вікно цього поділу, зображене на рис. 3.4. З його розгляду випливає, що в цьому поділі розміщено три групи блоків, які не мають виходів, а мають тільки входи:

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

- блок ***Scope*** з одним входом, який виводить графік залежності величини, яка подається до його входу, від модельного часу;
- блок ***XYGraph*** із двома входами, який забезпечує побудову графіка залежності однієї модельованої величини (другий зверху вхід) від іншої (перший вхід);
- блок ***Display*** з одним входом, призначений для відображення чисельних значень вхідної величини;

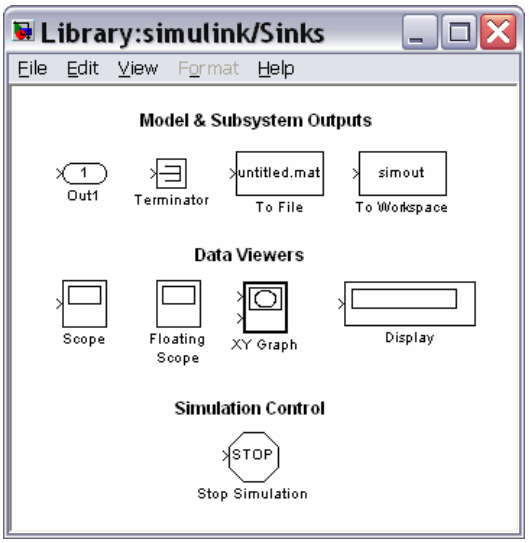


Рис. 3.4. Вміст поділу Sinks

2) блоки для пересилання результатів:

- блок ***To File***, який забезпечує зберігання результатів моделювання на диску в MAT-файлі (із розширенням .mat);
- блок ***To Workspace***, який зберігає результати в робочому просторі;
- блок- заглушка ***Terminator***;
- блок порт виходу ***Out***;

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

Блок ***Scope***

Цей блок дозволяє в процесі моделювання спостерігати на графіку процеси, що цікавлять дослідника. Він має один вход, на який подається сигнал, графік залежності якого від часу потрібно вивести у вікні.

Для настроювання параметрів цього блока потрібно *після встановлення зображення блока у вікно блок-схеми двічі клацнути мишкою на його зображенні*. У результаті на екрані з'явиться вікно **Scope** (рис. 3.5). Розмір і пропорції вікна можна змінювати довільно, користуючись мишкою. По горизонтальній осі відкладаються значення модельного часу, а по вертикальній - значення вхідної величини, які відповідають цим моментам часу. Якщо вхідна величина блока ***Scope*** є вектором, у вікні будуються графіки змінювання всіх елементів цього вектора, тобто стільки кривих, скільки елементів у вхідному векторі, причому кожна - свого кольору. Одночасно у вікні може відображатися до 30 кривих.

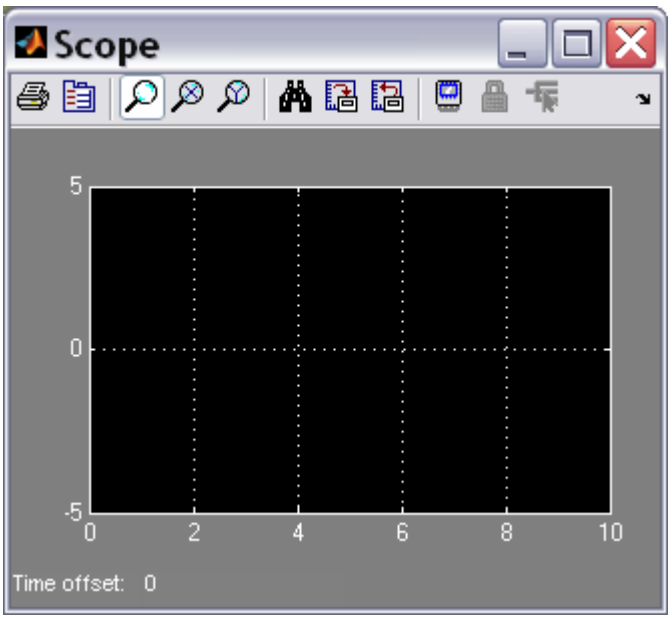


Рис. 3.5. Оглядове вікно Scope

Для керування параметрами вікна в ньому є панель інструментів, що містить 11 піктограм із таким призначенням (зліва праворуч):

- виведення вмісту вікна на прінтер
- виклик вікна настроювання параметрів блоку
- змінювання масштабу одночасно по обох осях графіка;
- змінювання масштабу по горизонтальній осі;
- змінювання масштабу по вертикальній осі;
- автоматичне встановлення оптимального масштабу осей (повний огляд, автошкалювання);
- зберігання встановленого масштабу осей;
- відновлення установок параметрів осей;
- включення холостого під'єднання блоку;
- шлюз селектору сигналів
- селектор сигналів.

Піктограми змінювання масштабу є альтернативними, тобто в кожний момент часу може бути натиснута лише одна з них. Піктограми не активні доти, поки немає графіка у вікні ***Scope.*** Активними із самого початку є лише перші дві, шоста, сьома і девята піктограми. Щиглик на другій піктограмі призводить до появи діалогового вікна настроювання параметрів **'Scope' parameters** (рис. 3.6).

Це вікно має дві вкладинки:

- ***General*** (Загальні), яка дозволяє встановити параметри осей;
- ***Data history*** (Установлювання даних), яку призначено для визначення параметрів подання даних блока ***Scope.***

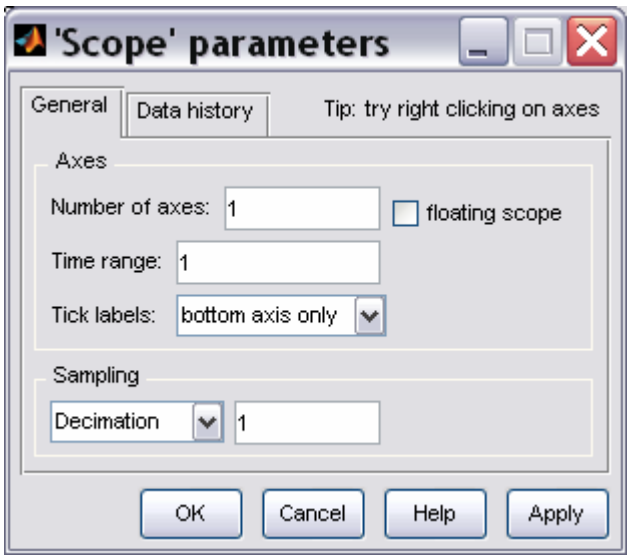


Рис. 3.6. Вікно настроювання блоку Scope

У нижній частині вікна розташовані кнопки: *Apply (Застосувати), Help (Виклик довідки), OK (Підтвердити установку), Cancel (Повернутися назад)*.

В області *Axes* (Осі) вкладинки General містять поля введення *Number of axes (Кількість осей)* і *Time range (Інтервал часу)*, а також список *Tick labels (Мітки осей)*.

У першому полі задається кількість графічних полів у вікні ***Scope*** (одночасно змінюється кількість входів блоку ***Scope***).

У другому полі встановлюється верхня межа модельного часу, що відкладається по осі абсцис. При цьому слід мати на увазі таке. *Якщо розмір заданого інтервалу моделювання ($T_м$) не перевищує встановленого у цьому полі значення (тобто весь процес уміщується у вікні **Scope**), під графіком у рядку Time offset (Зсув за часом) виводиться значення 0. У випадку ж, коли інтервал моделювання перевищує встановлене значення, у вікні **Scope** відображається тільки графік, що відповідає останньому відрізку часу, меншому за розміром, ніж Time range і рівному $T_м$-n*Time range, де n - ціле число. При цьому в рядку Time offset виводиться розмір "схованого" інтервалу часу - n*Time range.* Наприклад, якщо значення *Time range* дорівнює 3, а тривалість інтервалу моделювання встановлена 17, то у вікні ***Scope*** буде виведений графік модельованого процесу за останні 2 одиниці часу, а рядок під графіком буде мати вид: *Time offset: 15*.

За допомогою списку *Tick labels (Мітки осей)* можна задати вид оформлення осей координат в графіках вікна ***Scope***. Зазначений список містить три пункти: *all* (все), *bottom axis only* (лише нижньої осі), *none* (нет). В результаті обрання першого з них поділки по осях будуть нанесені вздовж кожної з осей усіх графіків. Обрання другого означає, що поділки по горизонтальних осях графічних полів (якщо їх декілька) за винятком нижньої будуть відсутні. нарешті, якщо обрати третій пункт, то зникнуть поділки по осях графіків і написи на них, графік займе усе поле вікна і останнє прийме вид, показаний на рис. 7.4.

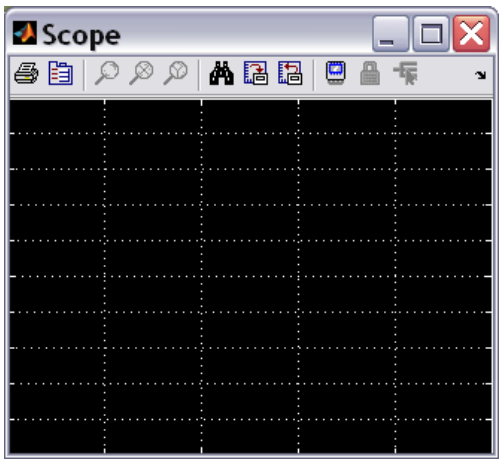


Рис. 3.7. Вікно Scope при встановленні NONE

Якщо в області *Axes* (Осі) встановити прапорець *Floating scope,* то входи у блок ***Scope*** будуть відімкнені. У цьому випадку блок відображується як такий, що не має входу, і якщо від був зв'язаний з іншими блоками, ці зв'язки обриваються. Той самий ефект справляє клацання на кнопці з тою самою назвою, що міститься на панелі інструментів блоку.

В області *Sampling* (Дискретизація) містится список, в якому обраний елемент *Decimation* (Проріджування), і поле, де можна ввести ціле додатне число, яке визначає, через яку кількість проміжків часу (дискретів часу) одержані дані використовуватимуться для побудови графіків в окні ***Scope.***

Вкладинка Data History (Історія даних) вікна **'Scope' parameters** (рис. 3.8) дозволяє задати максимальну кількість (починаючи з кінця) елементів масивів даних, що використовуються для побудови графіків у вікні ***Scope*** (поле поряд з прапорцем Limit data points to last (Максимальна кількість точок даних).

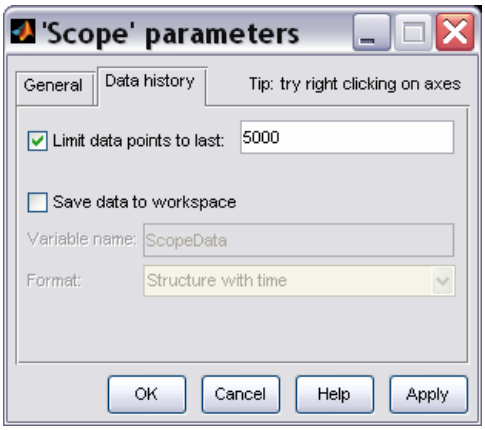


Рис. 3.8. Вкладинка Data History

Якщо встановити прапорець Save data to workspace (Записати дані у робочий простір), виникне можливість записати у робочий простір дані, які виводяться на графікі вікна ***Scope***. При цьому становляться досяжними поле Variable name (Ім'я змінної) і список Format (Формат). В поле можна ввести ймення змінної, під яким зберігатимуться дані у робочому просторі (за замовчуванням ці дані будуть записані під ім'ям ScopeData), а у спискуможна

обрати один з трьох форматів запису даних: Array (Масив, матриця), Structure (Структура) або Structure with time (Структура з часом).

Продемонструємо роботу блока на прикладі. Перетягнемо у вікно блок-схеми з вікна поділу Sources блок Sine Wave, а з вікна поділу Sinks – блок ***Scope*** і з'єднаємо вихід першого блоку зі входом другого. Одержимо схему, показану на рис. 3.9

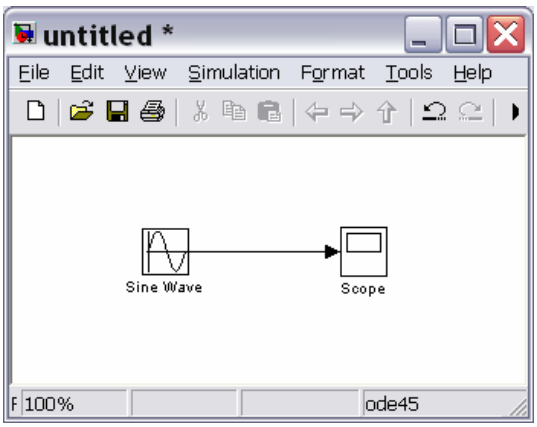


Рис. 3.9. Простіша блок-схема з блоком Scope

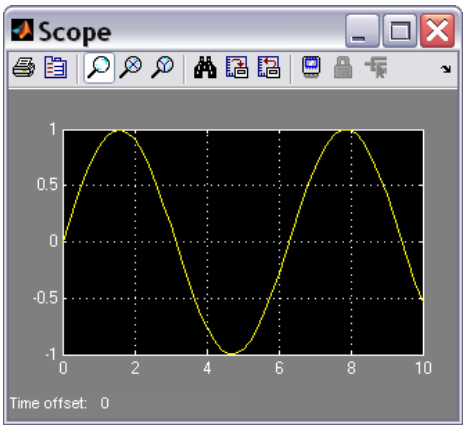


Рис. 3.10. Вікно Scope

Викличемо у вікні цієї блок-схеми команду Simulation > Start (Моделювання > Почати), потім подвійно клацнемо на зображенні блоку ***Scope***. На екрані виникне вікно ***Scope*** цього блоку з зображенням графіка змінювання у часі гармонічного сигналу (рис. 3.10).

*Блок **XYGraph***

Цей блок також є оглядовим вікном. На відміну від ***Scope***, він має два входи: на перший (верхній) подається сигнал, значення якого відкладаються по горизонтальній осі графіка, а на другий (нижній) - по вертикальній осі.

Якщо перетягнути цей блок на поле блок-схеми, а потім на зображенні його подвійно клацнути мишкою, то на екрані виникне вікно настроювання блока (рис. 3.11), яке дозволяє встановити межі змінювання обох вхідних величин, в яких буде побудований графік залежності другої величини від першої, а також задати дискрет за часом.

Наведемо приклад використання блока ***XYGraph***. Для цього перетягнемо у вікно блок-схеми зображення цього блока з вікна *Library simulink/Sinks*, а з вікна *Library simulink/Sources* **-** два блоки-джерела ***Clock*** і ***Sine Wave***. З'єднаємо виходи блоків-джерел із входами блока ***XYGraph.*** Одержимо блок-схему, наведену на рис. 3.12.

Перш ніж запустити процес моделювання цієї схеми, необхідно настроїти блок ***XYGraph***, вводячи у діалоговому вікні *Sink Block Parameters: XY Graph* очікувані діапазони змінювання величин *x* (у розглядуваному випадку – часу) і *y* (синусоїдального сигналу). Вкажено у полях введення x-min, x-max, y-min, y-max відповідно значення 0, 20, -1 і 1, як це відображено на рис. 3.11. Зазначимо, що у випадку використання блоку *Scope* вводити діапазони змінювання величин не потрібно, вони встановлюються автоматично.

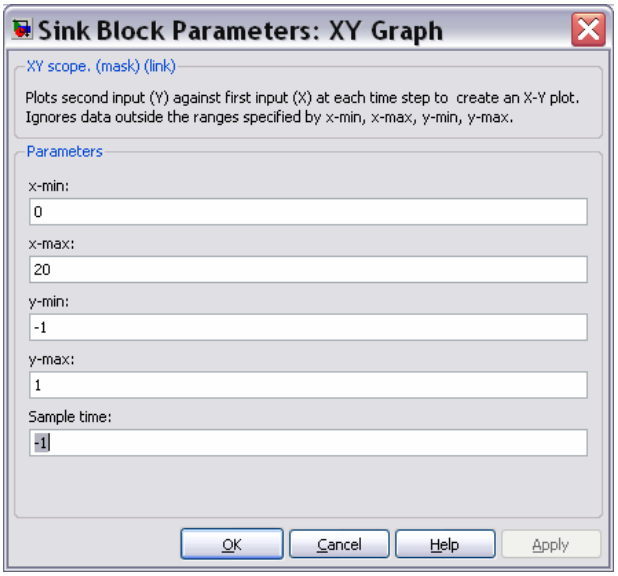


Рис. 3.11. Вікно настроювання блоку XYGraph

Якщо тепер вибрати мишкою меню *Simulation* у рядку меню вікна блок-схеми, а в ньому - команду *Start*, то по закінченні розрахунків у вікні блока ***XYGraph*** з'явиться зображення, подане на рис. 3.13.

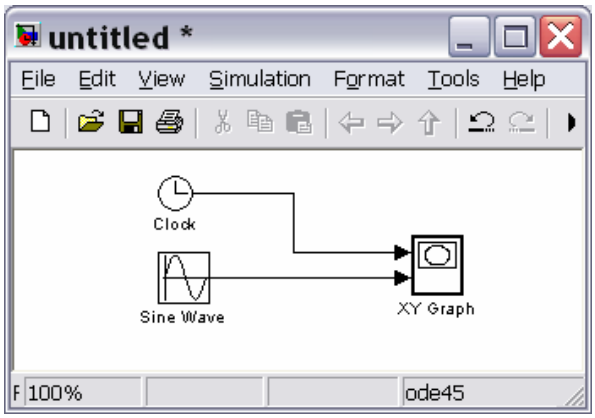


Рис. 3.12. Блок-схема перевірки роботи блоку XYGraph

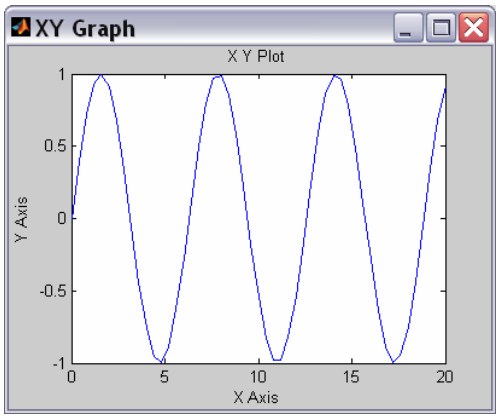


Рис.3.13. Вікно блоку XYGraph з графіком синусоїди

*Блок **Display***

Цей блок призначений для виведення на екран чисельних значень величин, що фігурують у блок-схемі.

Блок має 4 параметри настроювання (рис. 7.9). Список *Format* - задає формат виведення чисел; вид формату обирається за допомогою спадного меню, що містить 5 пунктів: *short, long, short_e, long_e, bank*. Поле введення *Decimation* дозволяє задати періодичність (у дискретах часу) виведення значень у вікні *Display*. Перемикач *Floating display* дозволяє визначати блок ***Display*** як блок без входу, обриваючи його зв'язки.

Блок ***Display*** може використовуватися для виведення як скалярних, так і векторних величин. Якщо відображувана величина є вектором, то вихідне подання блока змінюється автоматично, про що свідчить поява маленького

чорного трикутника в правому нижньому рогові блока. Для кожного елемента вектора створюється своє міні-вікно, але щоб вони стали видимими, необхідно розтягнути зображення блока. Для цього слід виділити блок, підвести курсор мишки до одного з його рогів, натиснути ліву клавішу мишки і, не відпускаючи її, розтягнути зображення блока, поки не зникне чорний трикутник.

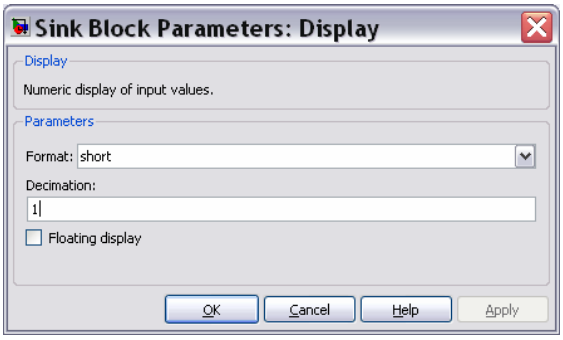


Рис. 3.14. Вікно настроювання блоку Display

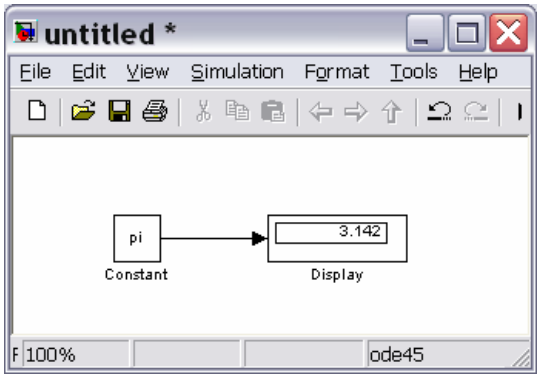


Рис. 3.15. Блок-схема перевірки блоку Display

Для прикладу створимо блок-схему (рис. 3.15) із двох елементів - блока-джерела ***Constant*** і блока-приймача ***Display***.

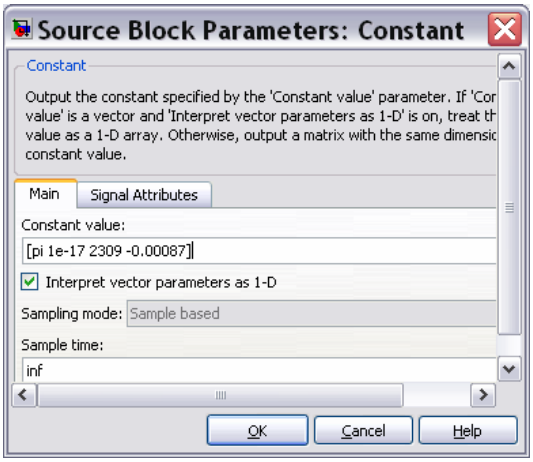


Рис. 3.16. Вікно настроювання блоку Constant

Викликавши вікно настроювання блока ***Constant*** (рис. 3.17), встановимо в ньому значення константи-вектора, який складається із чотирьох елементів [pi 1e-17 2309 -0.00087]. Викликаючи вікно настроювання блока ***Display***,

встановимо з його допомогою формат виведення чисел *short_e*. Після активізації команди *Start* із меню Simulation, розтягуючи зазначеним чином зображення блока ***Display*** на блок-схемі, одержимо картину, подану на рис. 3.17.

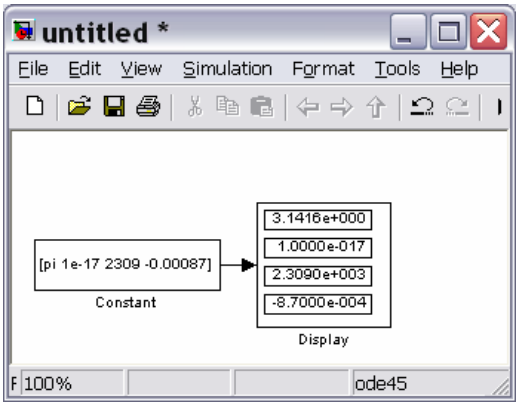


Рис. 3.17. Результат перевірки роботи блоку Display

*Блок **To File***

Цей блок забезпечує запис значень величини, поданої на його вхід, у MAT-файл даних для використання їх у наступному в інших S-моделях.

Блок має такі параметри настроювання (див. рис. 3.18):

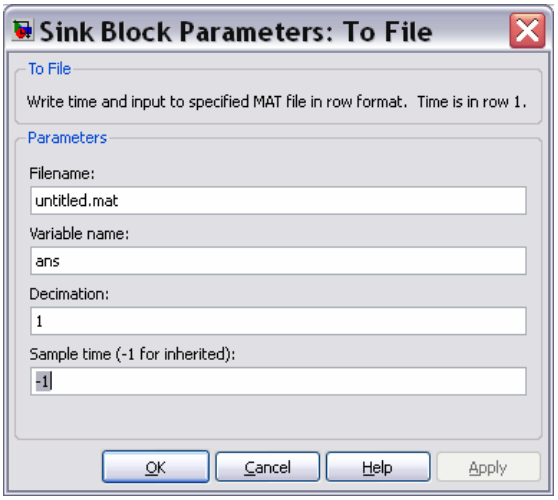


Рис. 3.18. Вікно настроювання блоку To File

Filename - ім'я MAT-файлу, в який записуватимуться значення вхідної величини; за замовчуванням - untitled. mat; ймення файлу виводиться на зображенні блока в блок-схемі;

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

Decimation - кількість дискретів часу, через яке провадитиметься запис даних у файл;

Sample Time - розмір дискрету часу для даного блока.

Слід зазначити, що значення даних, що подаються до входу блока записуються у вихідну змінну (наприклад, ***ans***) у такий спосіб: перший рядок матриці утворюють значення відповідних моментів часу; другий рядок містить відповідні значення першого елемента вхідного вектора, третій рядок - значення другого елемента і т. д. В результаті записується матриця розміром (k+1)*N, де k - кількість елементів вхідного вектора, а N - кількість точок вимірювання (або кількість моментів часу, у які здійснено вимірювання).

*Блок **To Workspace***

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

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

Блок має 4 параметри настроювання (див. рис. 3.19):

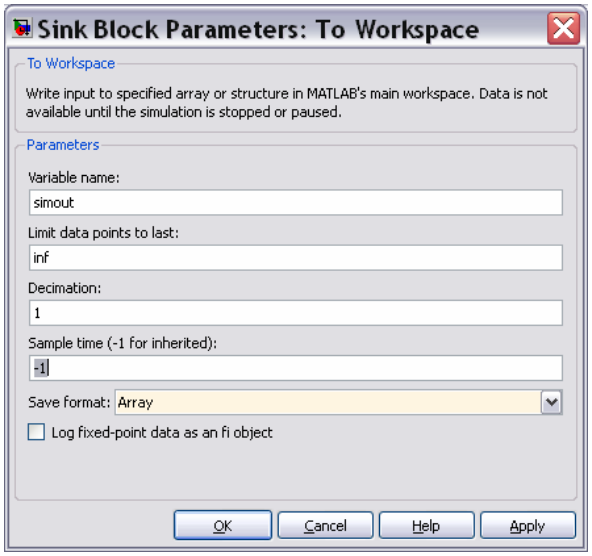


Рис. 3.19. Вікно настроювання блоку To Workspace

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

Limit data points to last - максимально припустима кількість точок, тобто значень даних, що записуються; за замовчуванням вона задається константою ***inf***, тобто дані реєструються на всьому інтервалі моделювання;

Decimation і *Sample Time* мають той самий зміст, що і раніше.

3.2.2. Поділ Sources (Джерела)

Блоки, що входять до поділу ***Sources*** (Джерела), призначені для формування сигналів (послідовності числових даних), які при моделюванні забезпечують роботу S-моделі у цілому або окремих її частин. Усі блоки мають по одному виходу і не мають входів.

Після вибору поділу ***Sources*** бібліотеки SimuLink на екрані з'явиться додаткове вікно, показане на рис. 3.20.

Як бачимо, поділ містить дві групи блоків: ***Model &Subsystem Inputs*** (Входи моделей і підсистем) та ***Signal Generators*** (Генератори сигналів).

Як джерела сигналів передбачені такі блоки:

- ***Constant*** - формує постійну величину (скаляр, вектор або матрицю);
- ***Signal Generator*** - створює (генерує) неперервний коливальний сигнал однієї із хвильових форм на вибір - синусоїдальний, прямокутний, трикутний чи випадковий;
- ***Pulse Generator*** - генератор неперервних прямокутних імпульсів;
- ***Signal Builder*** – побудувач сигналів,
- ***Ramp*** - створює лінійно висхідний (або спадний) сигнал (сходинку за швидкістю);

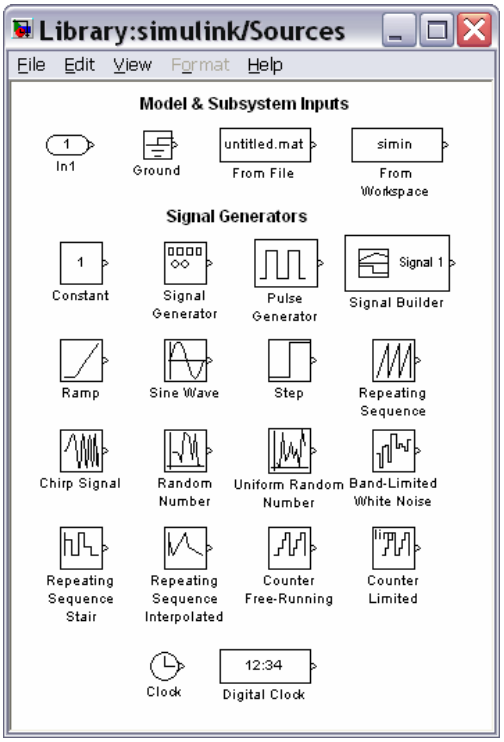


Рис. 3.20. Вміст поділу Sources

- ***Sine Wave*** - генерує гармонійний сигнал;
- ***Step*** - генерує сигнал у виді поодинокої сходинки (східчастий сигнал) із заданими параметрами (початку сходинки і її висоти);
- ***Repeating Sequence*** - генерує періодичну послідовність;
- ***Chirp Signal*** - генератор гармонійних коливань із частотою, яка лінійно змінюється з часом;
- ***Random Number*** - джерело дискретного сигналу, значення якого є випадковою величиною, розподіленою за нормальним (гауссовим) законом;
- ***Uniform Random Number*** - джерело дискретного сигналу, значення якого є випадковою рівномірно розподіленою величиною;

- ***Band-Limited White Noise*** - генератор білого шуму з обмеженою смугою частот;
- ***Clock*** (Годинник) - джерело неперервного сигналу, пропорційного до модельного часу;
- ***Digital clock*** (Цифровий годинник) - формує дискретний сигнал, пропорційний часові.

У групі блоків ***Model &Subsystem Inputs*** передбачені 4 блоки, які забезпечують використовування в моделі даних, отриманих раніше. Перший з них – ***In1*** є вхідним портом, завдяки якому можна завести у модель синал, одержаний в іншій моделі. Завдяки блоку ***Ground*** (Земля) можна ввести порожній (нульовий) сигнал у модель. Блок ***From File*** призначений для введення в S-модель даних, що зберігаються на диску в MAT-файлі. Нарешті блок ***From Workspace*** забезпечує введення в модель даних безпосередньо з робочого простору MatLAB. Нагадаємо, що структура даних у MAT-файлі є багатовимірним масивом із змінною кількістю рядків, яка визначається кількістю змінних, що реєструються. Елементи першого рядка містять послідовні значення модельного часу, елементи в інших рядках - відповідні окремим елементам записаного вектора значення змінних.

Як і інші блоки бібліотеки SimuLink, блоки-джерела можуть настроюватися користувачем, за винятком блока ***Clock***, робота якого ґрунтується на використанні апаратного таймера комп'ютера.

Блок ***Constant***

Блок призначений для генерування процесів, що є незмінними у часі, тобто мають стале значення. Він має один параметр настроювання (рис. 3.16) - *Constant value*, який може бути введений і як вектор-рядок з кількох елементів по загальних правилах MatLAB. Приклад його використання наведений раніше при розгляді блока ***Display***.

Блок ***Signal Generator***

Вікно настроювання цього блока виглядає так, як показано на рис. 3.21. Як видно, у параметри настроювання входять:

- *Wave form* - форма хвилі - дозволяє обрати одну з таких форм періодичного процесу
  1) *Sine* - синусоїдальні хвилі;
  2) *Square* - прямокутні хвилі;
  3) *Sawtooth* - трикутні хвилі;
  4) *Random* - випадкові коливання;

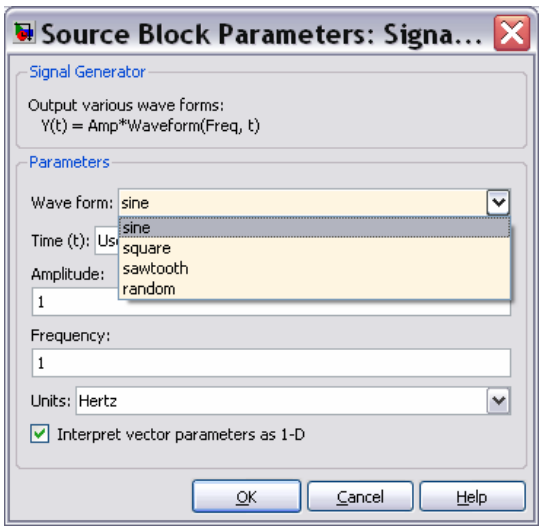


Рис. 3.21. Вікно настроювання блоку Signal Generator

■ *Amplitude* - визначає значення амплітуди коливань, що генеруються;

■ *Frequency* - задає частоту коливань;

■ *Units* - дозволяє обрати одну з одиниць виміру частоти за допомогою в спадного меню - *Hertz* (у Герцах) і *Rad/Sec* (у радіанах у секунду).

На рис. 3.22 показано найпростішу блок-схему S-моделі, що складається з блоку ***Signal Generator*** і оглядового блоку ***XY Graph***.

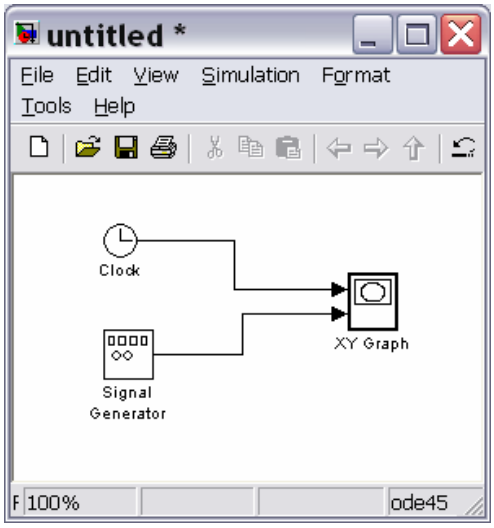
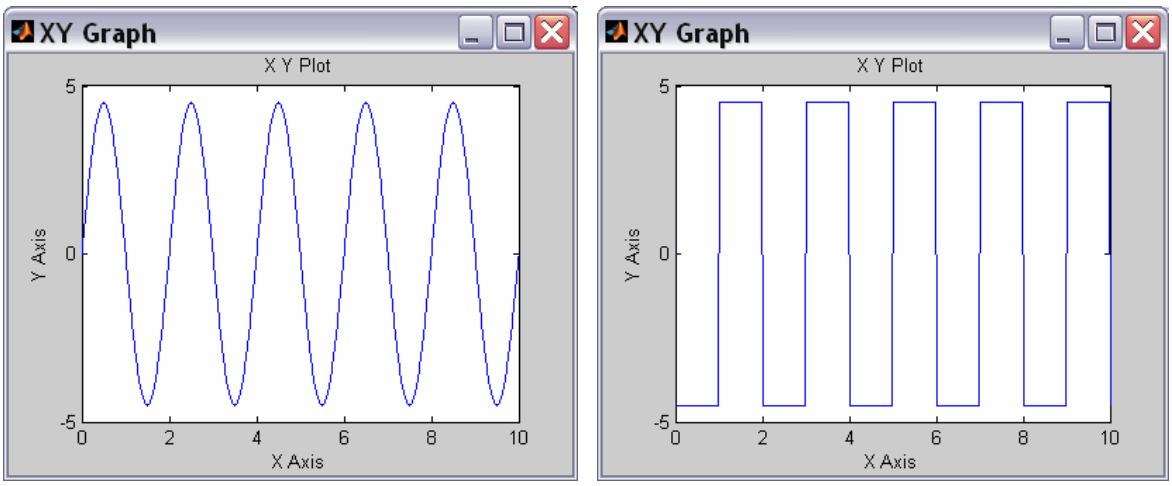


Рис. 3.22. Блок-схема перевірки функціонування блоку Signal Generator

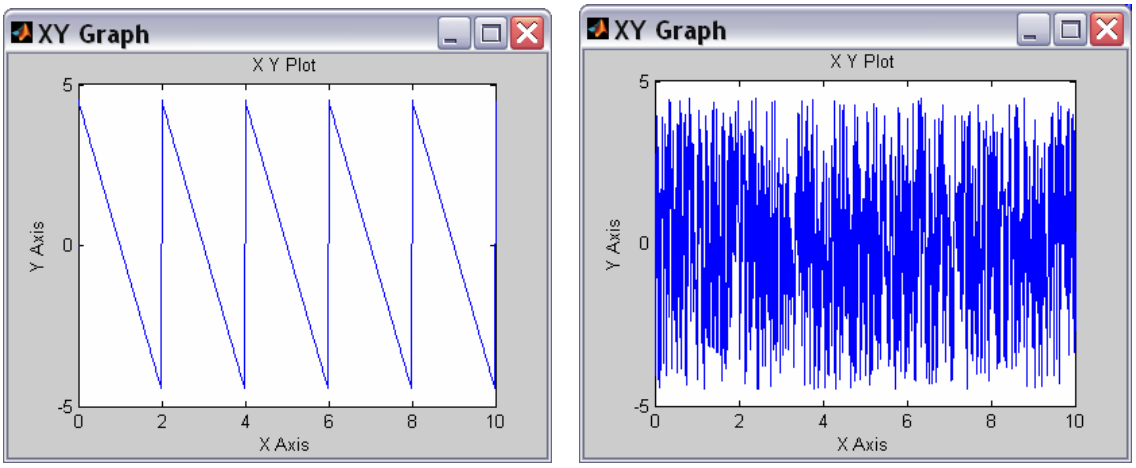
На наступному рис. 3.23а відображений вміст блоку ***XY Graph*** після проведення моделювання при таких параметрах настроювання: вид коливань - *Sine*; амплітуда - 4,5; частота – 0,5 Гц. Рис. 3.23б відбиває результат генерування прямокутної хвилі *Square*.



а) б)

Рис. 3. 23. Результат генерування хвиль Sine (а) і Square (б)

На рис. 3.24 подані результати, відображені у вікні ***XYGraph*** у випадку обрання відповідно трикутних і випадкових хвиль при решті тих самих параметрах настроювання.



а) б)

Рис. 3.24. Результат генерування хвиль Sawtooth (а) і Random (б)

При обранні пункту *Random* у списку *Wave Form* генерується послідовність даних (сигнал), значення яких рівномірно випадково розподілені у діапазоні, вказаному у параметрі *Amplitude*, а значення моментів часу, у які здійснюються стрибкоподібні змінювання сигналу, відділені один від одного на величину кроку моделювання, встановлюваного командою Simulation > Configuration Parameters.

Блок ***Pulse Generator***

Блок генерує послідовності прямокутних імпульсів. У число параметрів цього блока, що настроюються, входять (рис. 3.25):

- тип імпульсів (*Pulse Type*); *Time based* (для неперевного сигналу, аргумент – час), *Sample based* (для дискретного часу, аргумент – кількість дискретів часу);

- амплітуда сигналу (*Amplitude*), тобто висота прямокутного імпульсу;
- розмір періоду сигналу (*Period*) у секундах;
- ширина імпульсу (*Pulse width*), у відсотках до періоду;
- розмір затримки імпульсу щодо *t*=0 (*Phase delay*) - у секундах.

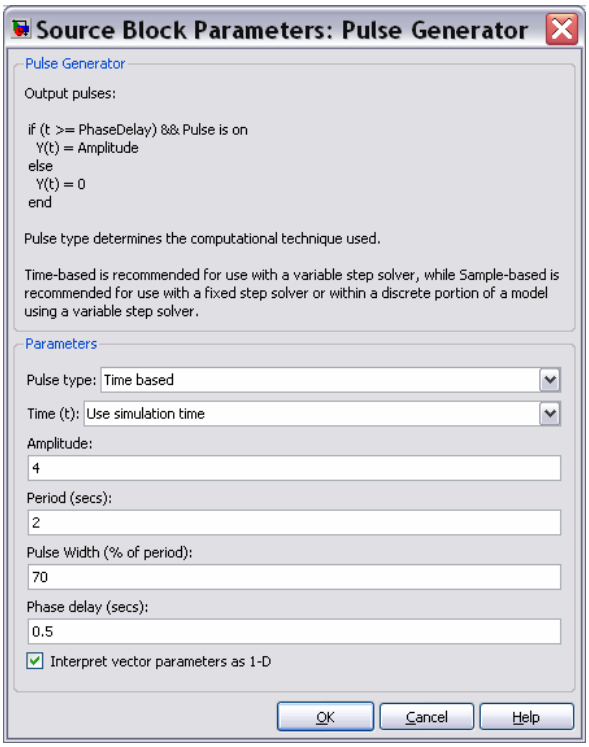


Рис. 3.25. Вікно настроювання блоку Pulse Generator

Приклад застосування цього блока при значеннях параметрів, зазначених на рис. 3.25, наведений на рис. 3.26.

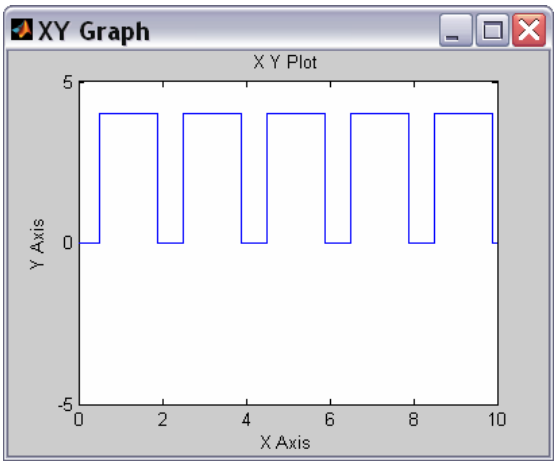


Рис. 3.26. Результат генерування послідовності прямокутних імпульсів блоком Pulse Generator

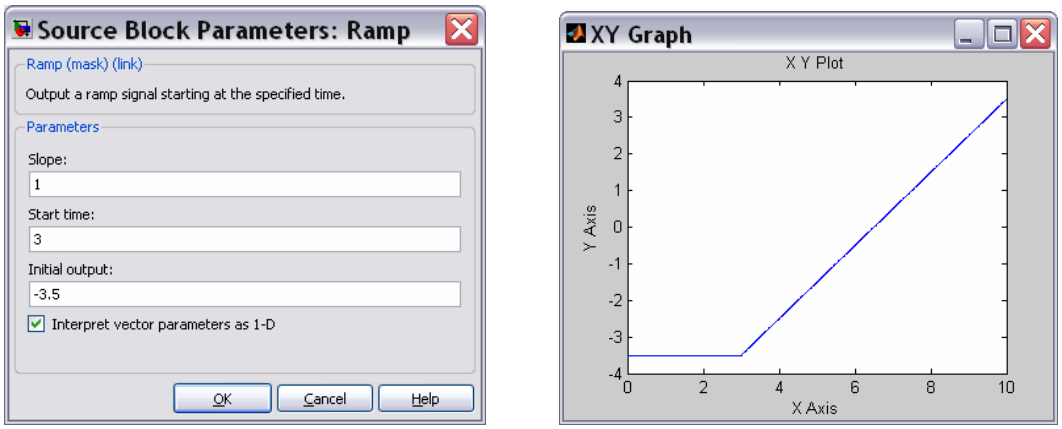
Блок ***Signal Builder***

Блок призначений для утворення поодинокого сигналу довільної форми, яка складається з відрізків прямих ліній.

*Блок **Ramp***

Цей блок формує неперервно висхідний сигнал і має такі параметри настроювання (рис. 3.27а):

- *Slope* - значення швидкості сходження сигналу (тангенса кута нахилу прямої графіка залежності сигнала від часу);
- *Start time* - час початку сходження сигналу;
- *Initial output* - значення сигналу до моменту початку його сходження.



а) б)

Рис. 3.27. Вікно настроювання (а) і результат роботи блоку Ramp

На рис. 3.27б наведений приклад результату застосування блока ***Ramp*** при таких значеннях параметрів: *Slope* = 1; *Start time* = 3; *Initial output* = -3.5.

*Блок **Sine Wave***

Цей блок є генератором гармонічно змінюваного з часом сигнала.

Блок ***Sine Wave*** має такі настройки (рис. 3.28):

- *Sine Type* - тип синусоїдальної хвилі: *Time based* - для неперевного сигналу (аргумент – час); *Sample based* - для дискретного часу, (аргумент – кількість дискретів часу);
- *Amplitude* - амплітуда синусоїдального сигналу;
- *Bias* – зміщення (стала складова сигналу, середнє його значення);
- *Frequensy (rad/sec)* - частота коливань у радіанах у секунду;
- *Phase (rad)* - початкова фаза в радіанах;
- *Sample time* - визначає величину дискрету часу для цього блоку.

Результат застосування блока (при значеннях параметрів настроювання: амплітуда - 2,5; стала складова сигналу – (-1); частота - 1 радіан у секунду і початкова фаза - $\pi./2$ радіан) показаний на рис. 3.29.

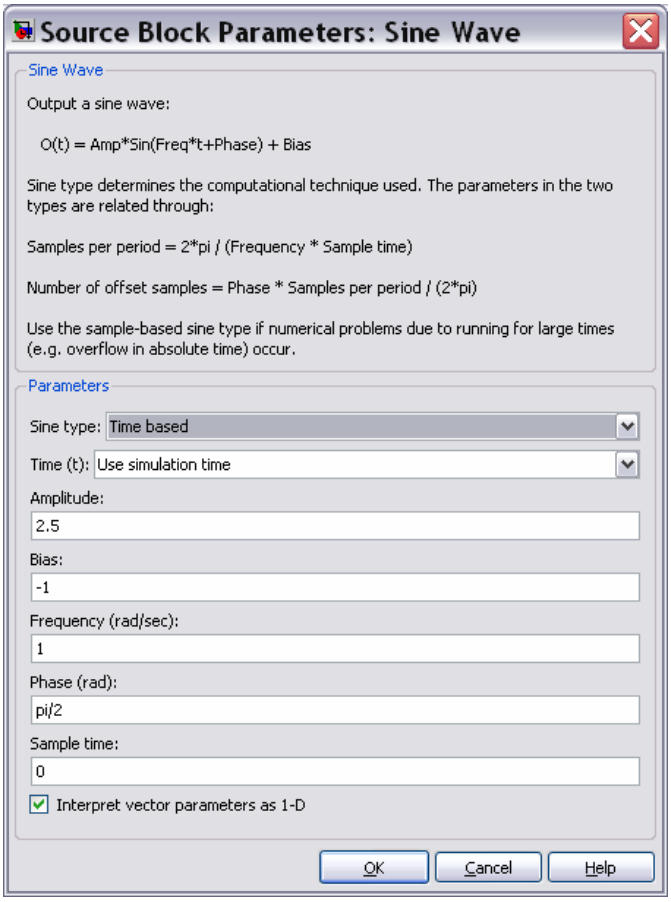


Рис. 3.28. Вікно настроювання блоку Sine Wave

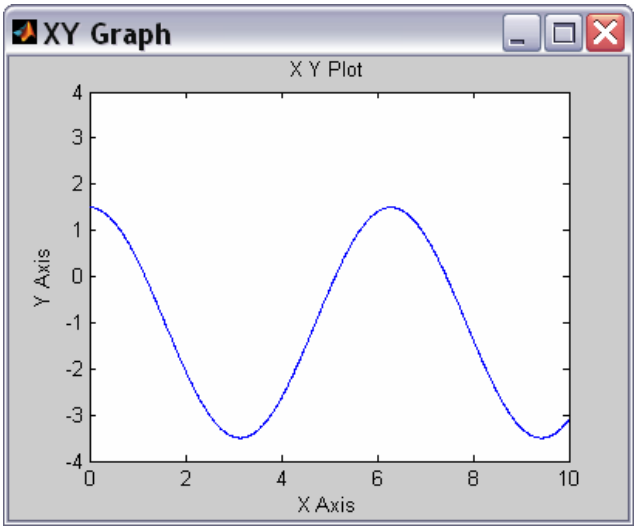


Рис. 3.29. Результат роботи блоку Sine Wave

*Блок **Step***

Блок забезпечує формування керуючого сигналу у формі сходинки (або, як говорять, - східчастого сигналу).

Блок має 3 параметри настроювання (рис. 3.30):

- *Step time* - час початку сходинки (момент стрибка сигналу) - визначає момент часу, у який відбувається стрибкоподібне змінювання сигналу; за замовчуванням приймається рівним 1;

- *Initial value* - початкове значення - задає рівень сигналу до стрибка; значення за замовчуванням - 0;
- *Final value* - кінцеве значення - задає рівень сигналу після стрибка; значення його за замовчуванням - 1.

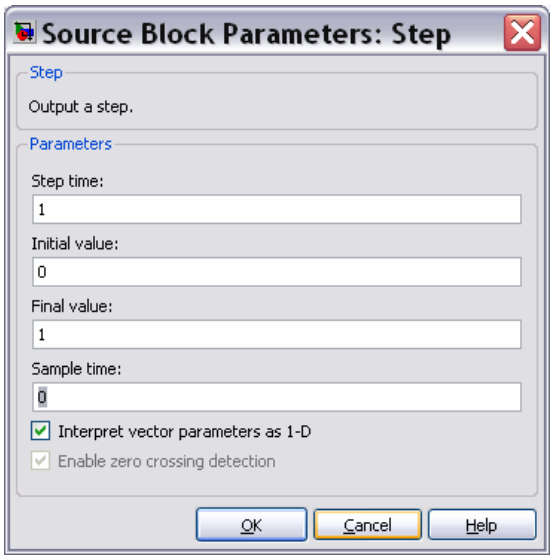


Рис. 3.30. Вікно настроювання блоку Step

Якщо встановити такі параметри настроювання блока: *Step time* - 3,5, *Initial value* - (-2), *Final value* – 3, то після активізації моделювання (*Simulation/Start)* одержимо у вікні ***Scope*** картину, подану на рис. 3.31.

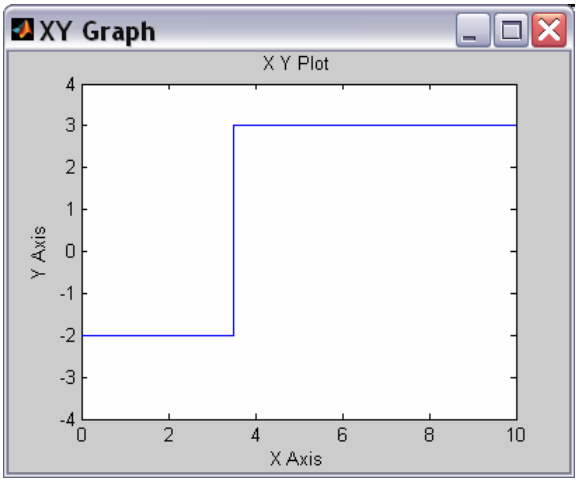


Рис. 3.31. Результат генерування східчастого сигналу

*Блок **Repeating Sequence***

Цей блок являє собою генератор періодичних коливань, хвиля яких складається з відрізків прямих. У вікнімістить дві настройки (рис. 3.32):

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

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

Як приклад на рис. 3.43 наведене зображення процесу, згенерованого блоком ***Repeating Sequence*** при параметрах настроювання, зазначених на рис. 3.32.

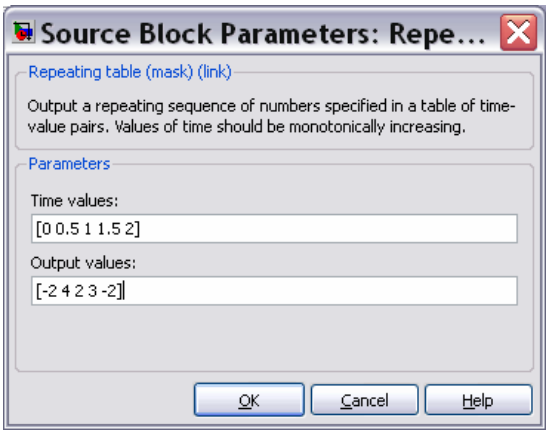


Рис. 3.32. Вікно настроювання блоку Repeating Sequence

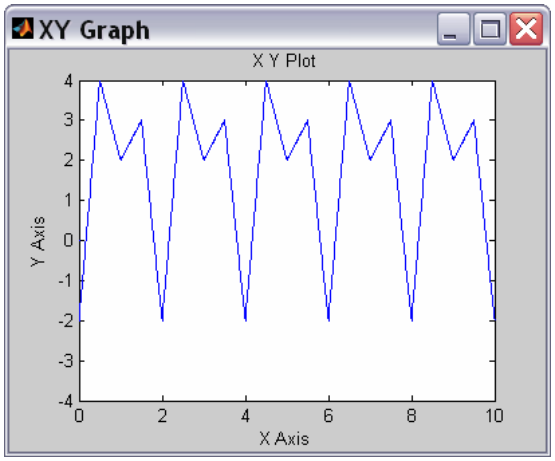


Рис. 3.33. Сигнал, згенерований блоком Repeating Sequence

Блок ***Chirp Signal***

Цей блок генерує синусоїдальний сигнал одиничної амплітуди змінної частоти, причому значення частоти коливань змінюється з часом за лінійним законом. Відповідно до цього в ньому передбачені такі параметри настроювання (рис. 3.34):

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

Рис. 3.35 демонструє результат використання блока при параметрах, зазначених на рис. 3.34.

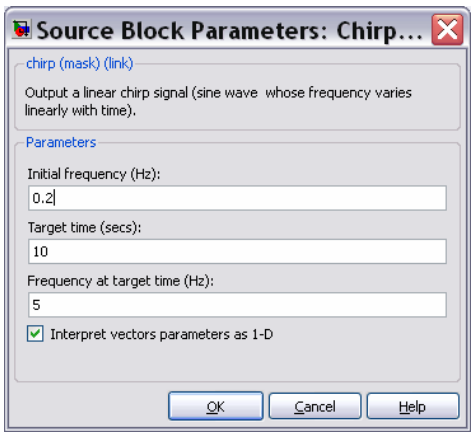


Рис. 3.34. Вікно настроювання блоку Chirp Signal

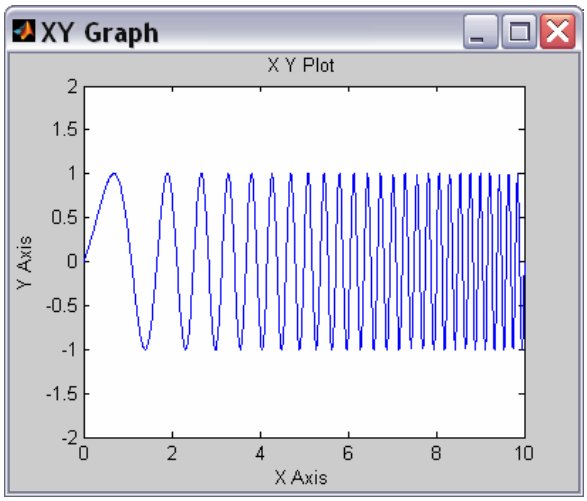


Рис. 3.35. Сигнал, згенерований блоком Chirp Signal

Блоки, що генерують випадкові процеси

Блок ***Random Number*** забезпечує формування сигналів, значення яких в окремі моменти часу є випадковою величиною, розподіленою за нормальним (гауссовим) законом із заданими параметрами.

Блок має чотири параметри настроювання (рис. 3.36). Перші два - *Mean* і *Variance* - є параметрами нормального закону (середнє й середньоквадратичне відхилення від цього середнього), третій - *Initial seed* - задає початкове значення бази для ініціалізації генератора послідовності випадкових чисел. При фіксованому значенні цього параметра генератор завжди виробляє ту саму послідовність. Четвертий параметр (*Sample time*), як і раніше, задає величину дискрету часу.

Блок ***Uniform Random Number*** блок формує сигнали, значення яких в окремі моменти часу є випадковою величиною, що рівномірно розподілена в заданому інтервалі. У число параметрів настроювання блока входять:

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

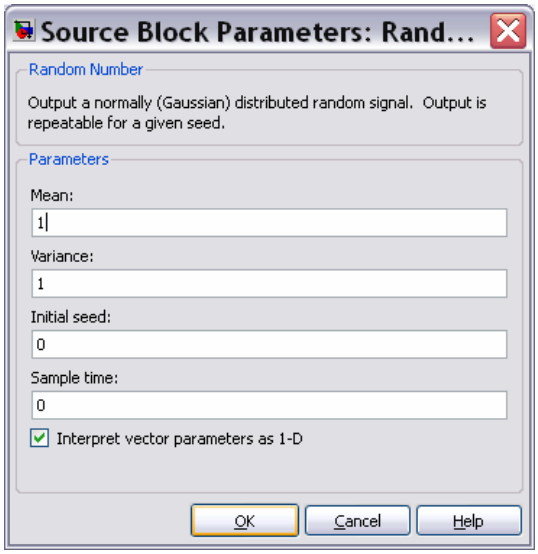


Рис. 3.36. Вікно настроювання блоку Random Number

Блок ***Band-Limited White Noise*** формує процес у виді частотно-обмеженого білого шуму. Параметри настроювання в нього такі:

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

Сформуємо блок-схему, яка ілюструє роботи блоків. Для того, щоб у вікні *Scope* можна було розмістити три графіки, кожен з яких відбивав би окремий процес, необхідно викликати команду Scope > 'Scope' parameters > General і всановити у полі *Number of ax*es кількість осей – 3. Вид блока *Scope* на блок-схемі при цьому зміниться, на ньому виникнуть зображення трьох входів. У списку *Tick Labels* встановимо значення *all*. Після цього над кожним з трьох графіків, які виводяться, можна буде проставити заголовки. Тексти заголовків розміщуються на лініях, що ведуть до входів блоку *Scope* (див. рис. 3.37)

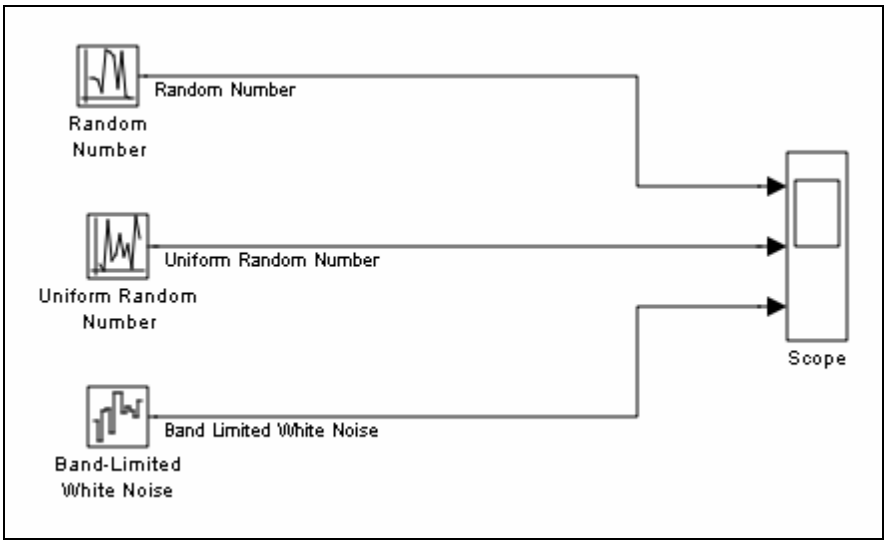


Рис. 3.37. Блок-схема перевірки роботи генераторів випадкових процесів

Перед запуском моделі у вікні блок-схеми викличемо команду *Simulation > Configuration parameters > Solver*, у віконці якого зі спадним списком оберемо значення discrete (no continuous state). У тому самому вікні у віконці з написом *Fixed step size* (Розмір фіксованого кроку) встановимо 0,05.

У вікнах настроювання усіх трьох блоків встановимо дискрет часу (*Sample time*), рівним 0,1.

Здійснюючи тепер моделювання, одержимо процеси, подані на рис. 3.38.

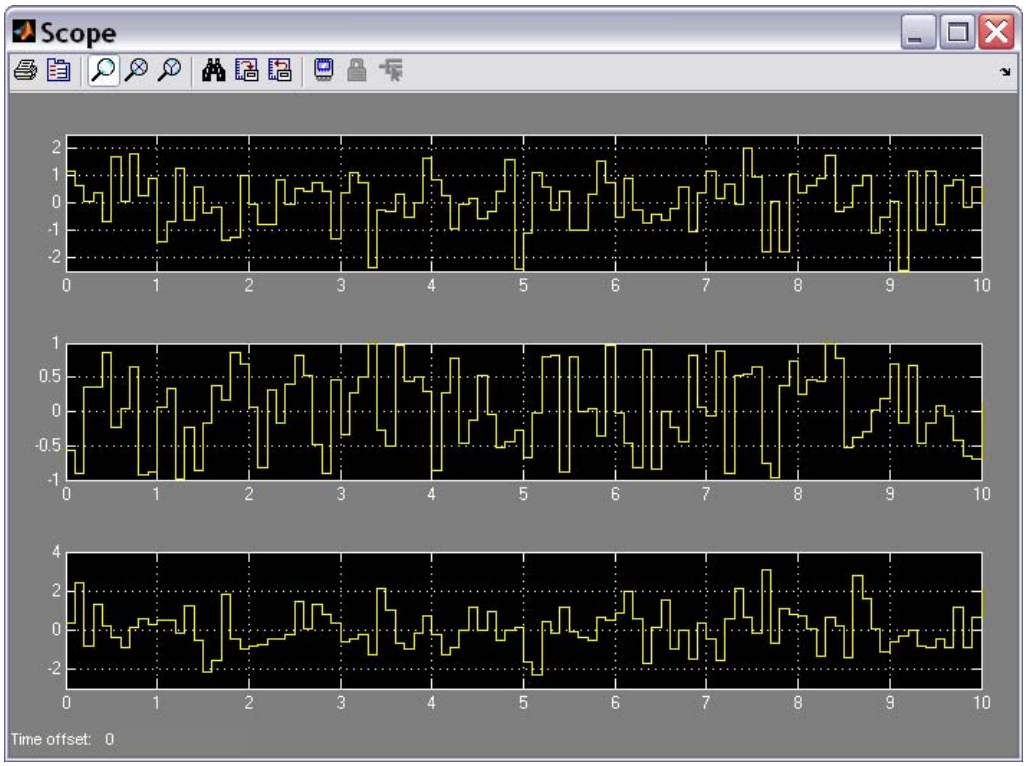


Рис. 3.38. Результати роботи генераторів випадкових процесів при Sample time=0,1

Як бачимо, модельовані процеси зберігають незмінні значення всередині інтервалу часу *Sample time* (Дискрета часу). Змінювання значень процеса здійснюється стрибкоподібно на межі сусідніх дискретів часу.

Примітка. Слід відрізнювати крок змінювання модельного часу, який встановлюється при проведенні моделювання у вікні *Configuration parameters* і визначає інтервали часу, через які здійснюються обчислення окремих станів системи, що моделюється, і параметр *Sample time* (Дискрет часу), який визначає інтервал часу, всередині якого вихідна величина блоку не змінює свого значення.

Варто зазначити, що блок ***Band-Limited White Noise*** суттєво відрізняється від перших двох блоків-генераторів. Для останніх встановлення значення параметра *Sample time*, відмінного від нуля, не є обов'язковим. У блоці ***Band-Limited White Noise*** цей параметр визначає найбільшу частоту у спектрі вихідного сигналу, вона дорівнює величині (у герцах), обернену значенню параметра *Sample time*, тому останній не може бути рівний нулеві.

На рис. 3.39 показаний результат моделювання перших двох блоків при значенні параметра *Sample time,* рівним нулю. У цьому випадку змінювання

величини випадкового процесу здійснюється вже на стику кроків моделювання (0,05), величина яких встановлюється за допомогою команди *Simulation* > *Configuration parameters* у віконці з написом *Fixed step size* (Розмір фіксованого кроку).

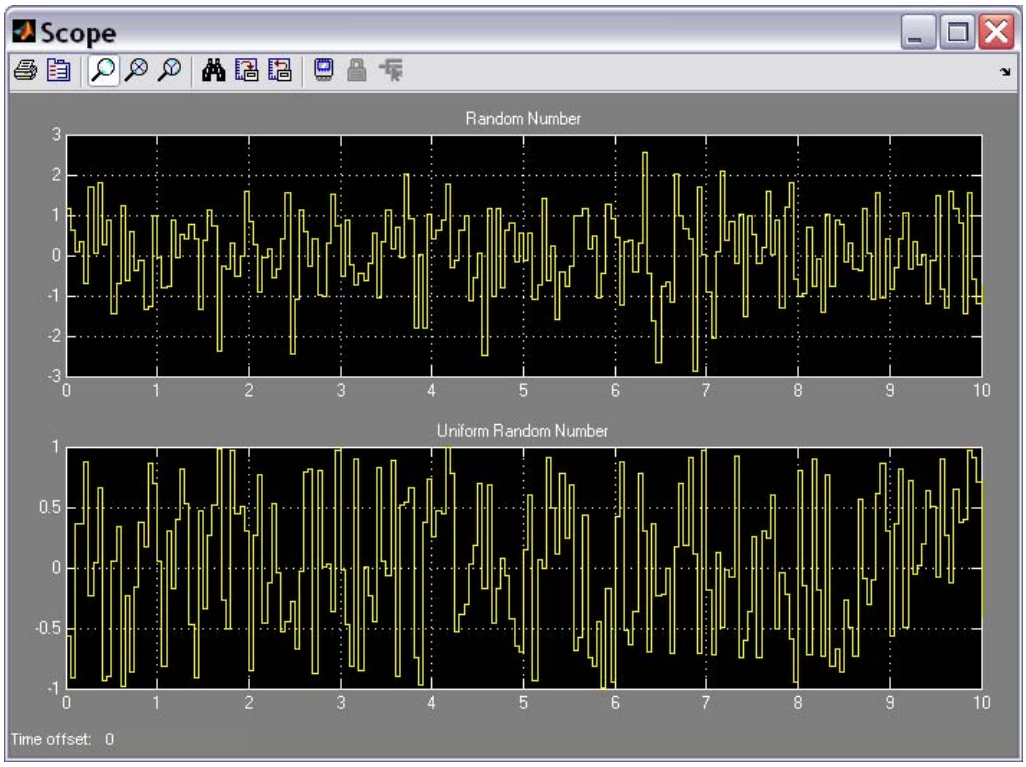


Рис. 3.39. Результати роботи генераторів випадкових процесів при Sample time=0

3.2.3. Поділ Continuous (Неперервні елементи)

Цей поділ бібліотеки містить наступні групи блоків (рис. 3.40):

- *Continuous-Time Linear Systems* (Лінійні системи неперервного часу);
- *Continuous-Time Delays* (Затримки неперервного часу).

Перша група складається з блоків:

- ***Integrator*** - ідеальна інтегруюча ланка (інтегратор);
- ***Derivative*** - ідеальна диференціююча ланка;
- ***State-Space*** - визначення лінійної ланки через завдання чотирьох матриць її простору станів;
- ***Transfer Fcn*** - визначення лінійної ланки через завдання її передатної функції;
- ***Zero-Pole*** - завдання ланки через указівку векторів значень його полюсів і нулів, а також значення коефіцієнта передачі;

Використання більшості блоків-ланок є досить прозорим для тих, хто знайомий з основами теорії автоматичного керування.

*Блок **Integrator***

Блок здійснює інтегрування в неперервному часі вхідної величини. Він має наступні параметри настроювання (рис. 3.41):

- підключення додаткового керуючого сигналу (*External reset*);
- визначення джерела (внутрішнє чи зовнішнє) встановлювання початкового значення вихідного сигналу (*Initial condition source*);
- початкове значення вихідної величини (*Initial condition*); значення вводиться в рядку редагування або як числова константа, або у вигляді виразу, що обчислюється;
- прапорець *Limit output* (*Обмеження вихідного значення)* визначає, чи будуть використовуватися наступні 2 параметри настроювання;

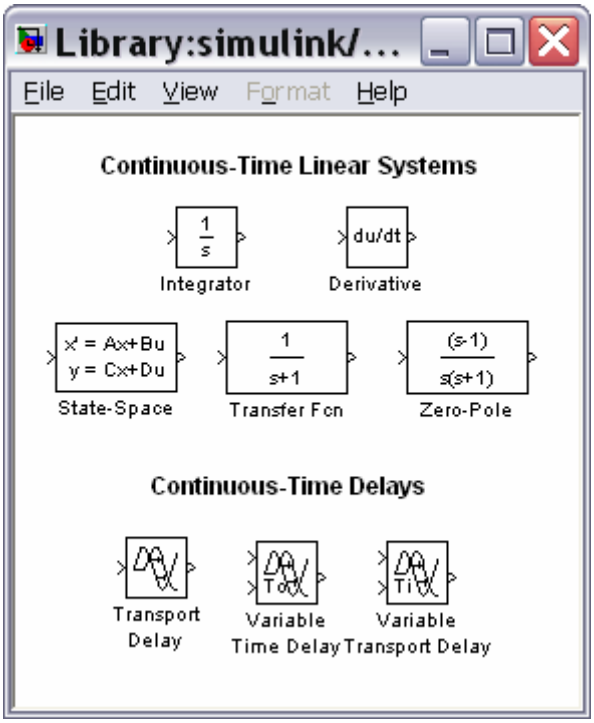


Рис. 3.40. Вміст поділу Continuous

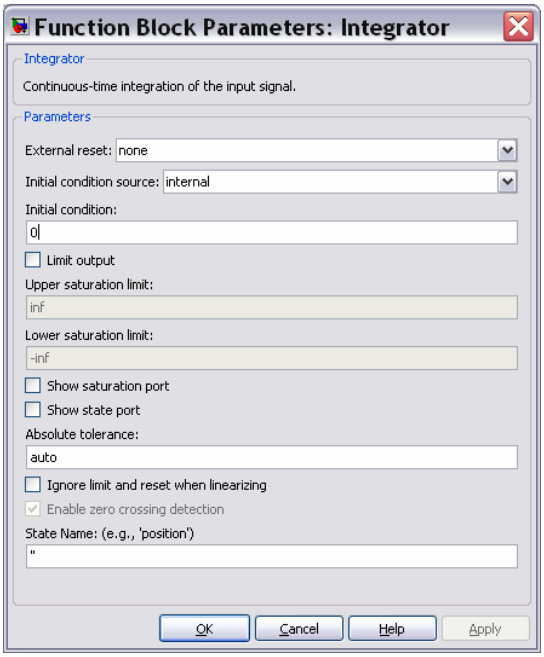


Рис. 3.41. Вікно настроювання блоку Integrator

- верхнє граничне значення вихідної величини (*Upper saturation limit*); за замовчуванням - не обмежене (*inf*);
- нижнє граничне значення вихідної величини (*Lower saturation limit*); за замовчуванням параметр має значення (-*inf*);
- прапорець *Показати порт насичення* (*Show saturation port*);
- прапорець *Показати порт стану* (*Show state port*);
- припустима гранична величина абсолютної похибки (*Absolute tolerance*).

Параметр *External reset* може приймати такі значення (див. його спадне меню): *none* - додатковий керуючий сигнал не використовується; *rising* - для керування використовується висхідний сигнал; *falling* - для керування використовується спадний сигнал; *either* – для керування використовується і висхідний і спадний сигнал.

Параметр *Initial condition source* приймає одне із двох значень:

internal - використовується внутрішнє встановлювання початкового значення вихідної величини;

external - встановлення початкових умов здійснюватиметься ззовні.

Якщо обрані користувачем значення цих двох параметрів припускають наявність додаткових вхідних сигналів, то на графічному зображенні блока виникають додаткові вхідні порти (після натискання кнопки *Apply* у вікні настроювань блока). Якщо прапорець *Limit output* встановлений, то при переході вихідного значення інтегратора через верхню або нижню межу на додатковому виході блока (*saturation port*) формується одиничний сигнал. Щоб цей сигнал можна було використовувати для керування роботою S-моделі, прапорець *Show saturation port* має бути включений. При цьому на графічному зображенні блока з'являється позначення нового вихідного порту на правій стороні зображення блоку-інтегратора.

Встановлення прапорця *Show state port* також приводить до появи додаткового виходу *state port* блока (він виникає звичайно, на нижній стороні зображення блока). Сигнал, який подається на цей порт, збігається з головним вихідним сигналом, але, на відміну від нього, може бути використаний тільки для переривання алгебричного циклу або для узгодження стану підсистем моделі.

*Блок **State-Space** (простір станів)*

Блок ***State-Space*** забезпечує введення лінійної стаціонарної ланки перетворення вхідного сигналу у виді матриць у просторі станів. Його вікно настроювання (рис. 3.42) містить 7 параметрів настроювання:

- матриця A системи зв’язку похідних від змінних стану з самими змінними стану;
- матриця B зв’язку похідних від змінних стану з вхідним сигналом блоку;
- матриця C зв’язку вихідного сигналу зі змінними стану;
- матриця D зв’язку вихідного сигналу з вхідним сигналом;

- вектор *Initial conditions* початкових значень вектору змінних стану;
- вектор Absolute tolerance максимально припустимих похибок обчислення кожної зі змінних стану; за замовчуванням встановлюється автоматично;
- State Name – символьний рядок, що містить ймення стану системи.

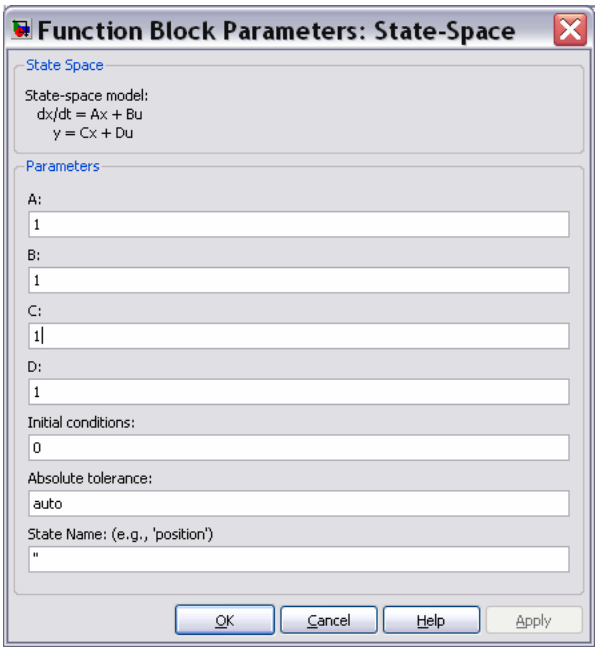


Рис. 3.42. Вікно настроювання блоку State-Space

*Блок **Transfer Fcn** (передатна функція)*

Дозволяє ввести лінійну ланку через завдання її передатної функції. Вікно настроювання показане на рис. 3.43 і містить два основних параметри:

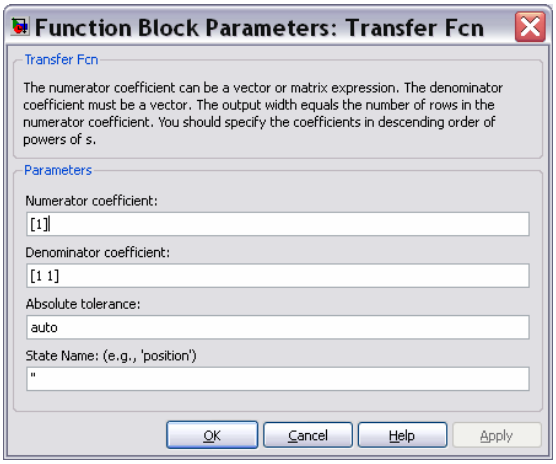


Рис. 3.43. Вікно настроювання блоку Transfer Fcn

- *Numerator coefficients* – вектор значень коефіцієнтів чисельника передатної функції;

- *Denominator coefficients* – вектор значень коефіцієнтів знаменника передатної функції.

*Блок **Zero-Pole** (нулі - полюси)*

Блок ***Zero-Pole*** дозволяє ввести лінійну ланку через завдання нулів і полюсів її передатної функції.

Головні параметри настроювання блоку є наступними (рис. 3.44):

- *Zeros* – вектор нулів передатної функції (коренів поліному її чисельника);

- *Poles* – вектор полюсів передатної функції (коренів поліному її знаменника);

- *Gain* – вектор коефіцієнтів підсилення.

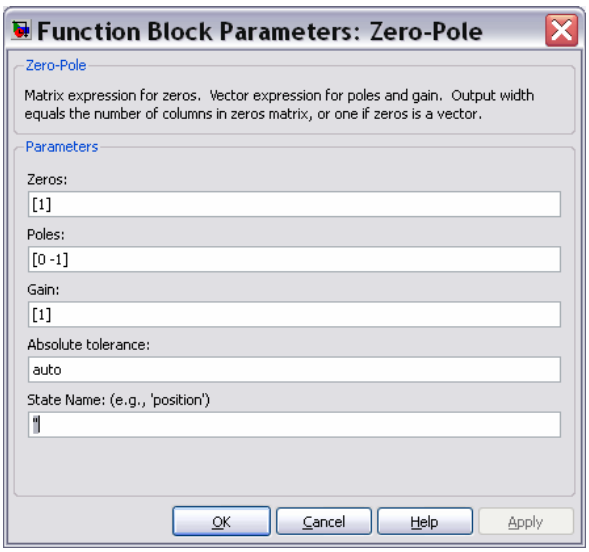


Рис. 3.44. Вікно настроювання блоку Zero-Pole

Друга група блоків здійснює затримку неперервних сигналів, що поступають на їхній вхід.

Блок ***Transport Delay*** забезпечує затримку сигналу на задану кількість кроків модельного часу, причому необов'язково ціле. Настроювання блока відбувається по трьох параметрах:

Time delay (Час затримки) - кількість кроків модельного часу, на який слід затримати сигнал; може вводитися або в числовій формі, або у формі виразу, що обчислюється;

Initial input (Початкове значення входу) - за замовчуванням дорівнює 0;

Initial buffer size (Початковий розмір буфера) - обсяг пам'яті (у байтах), що виділяється в робочому просторі MatLAB для збереження параметрів затриманого сигналу; повинно бути кратним до 8 (за умовчанням - 1024).

Блок ***Variable Transport Delay*** дозволяє задавати керовану ззовні величину затримки. З цією метою блок має додатковий вхід. Подаваний на нього сигнал визначає тривалість затримки.

3.2.4. Поділ Discrete (Дискретні елементи)

Раніше розглянуті розділи бібліотеки дозволяють формувати неперервну динамічну систему. Розділ ***Discrete*** містить елементи (блоки), властиві тільки дискретним системам, а також такі, що перетворюють неперервну систему в дискретну (рис. 3.45). Ці блоки поділені на дві групи

- *Discrete-Time Linear Systems* (Лінійні системи дискретного часу);
- *Sample & Hold Delay*s (Затримки дискретного часу).
-

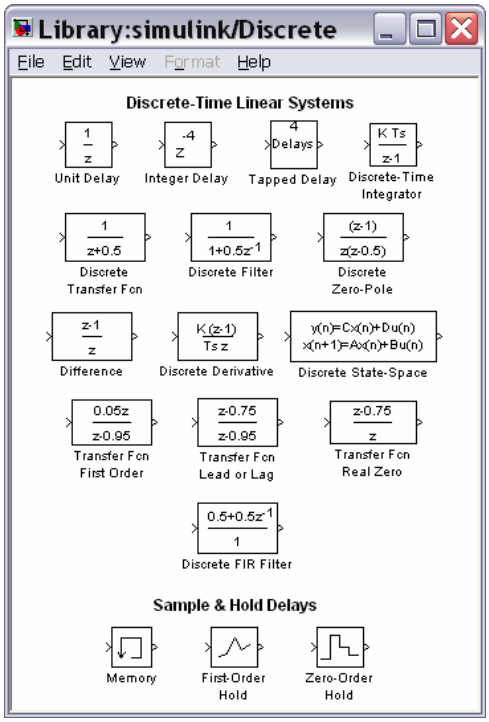


Рис. 3.45. Вміст поділу Discrete

У першу групу входять блоки:

- ***Unit Delay, Integer Delay, Tapped Delay*** - блоки, яки використовуються для затримки сигналу;
- ***Discrete-Time Integrator*** - дискретний інтегратор;
- ***Discrete Transfer Fcn*** - блок завдання лінійної дискретної ланки через дискретну передатну дробово-раціональну функцію щодо z;
- ***Discrete Filter*** - блок завдання дискретної ланки через дискретну передатну дробово-раціональну функцію щодо 1/z;
- ***Discrete Zero-Pole*** - блок завдання дискретної ланки через задання значень нулів і полюсів дискретної передатної функції щодо 1/z.
- ***Discrete Derivative*** – формує дискретну похідну вхідного сигналу
- ***Discrete State-Space*** - блок завдання дискретної лінійної ланки матрицями її стану;
- ***Transfer Fcn First Order, Transfer Fcn Lead or Lag, Transfer Fcn Real Zero*** - блоки, що формують дискретні ланки з спеціальними передатними функціями;
- ***Discrete FIR Filter*** – формує дискретний КІХ –фільтр.

Другу групу складають блоки:

- ***Memory*** – використовується для затримки сигнала на один крок модельного часу;
- ***First-Order Hold*** - екстраполятор першого порядку;
- ***Zero-Order Hold*** - екстраполятор нульового порядку;

Блок ***Unit Delay*** забезпечує затримку вхідного сигналу на задане число кроків модельного часу. Параметрами настроювання для цього блока є: початкове значення сигналу (*Initial condition)* і час затримки *(Sample time)*, який задається кількістю кроків модельного часу.

Блок ***Discrete-Time Integrator*** виконує чисельне інтегрування вхідного сигналу. Більшість параметрів настроювання цього блока збігаються з параметрами блока ***Integrator*** розділу ***Linear***. Відмінності полягають у наступному. У блоці дискретного інтегратора є додатковий параметр - метод чисельного інтегрування (*Integrator method*). За допомогою спадного меню можна вибрати один із трьох методів: прямий метод Ейлера (лівих прямокутників); зворотний метод Ейлера (правих прямокутників); метод трапецій. Друга відмінність - замість параметра *Absolute tolerance* уведений параметр *Sample time,* який задає крок інтегрування в одиницях кроків модельного часу.

Блок ***Memory*** *(Пам'ять)* виконує затримку сигналу тільки на один крок модельного часу. Блок має два параметри настроювання: *Initial condition (Початкова умова)* задає значення вхідного сигналу в початковий момент часу; прапорець *Inherit sample time (Спадкування кроку часу)* дозволяє вибрати величину проміжку часу, на який буде здійснюватися затримка сигналу:

- якщо прапорець знятий, то використовується мінімальна затримка, рівна 0,1 одиниці модельного часу;
- якщо прапорець встановлений, то величина затримки дорівнює значенню дискрету часу блока, що передує блоку ***Memory.***

Блоки ***First-Order Hold*** і ***Zero-Order Hold*** прислуговуються задля перетворення неперервного сигналу у дискретний.

На рис. 3. 46. графічно поданий результат проходження неперервного синусоїдального сигналу через ці два блоки-екстраполятори.

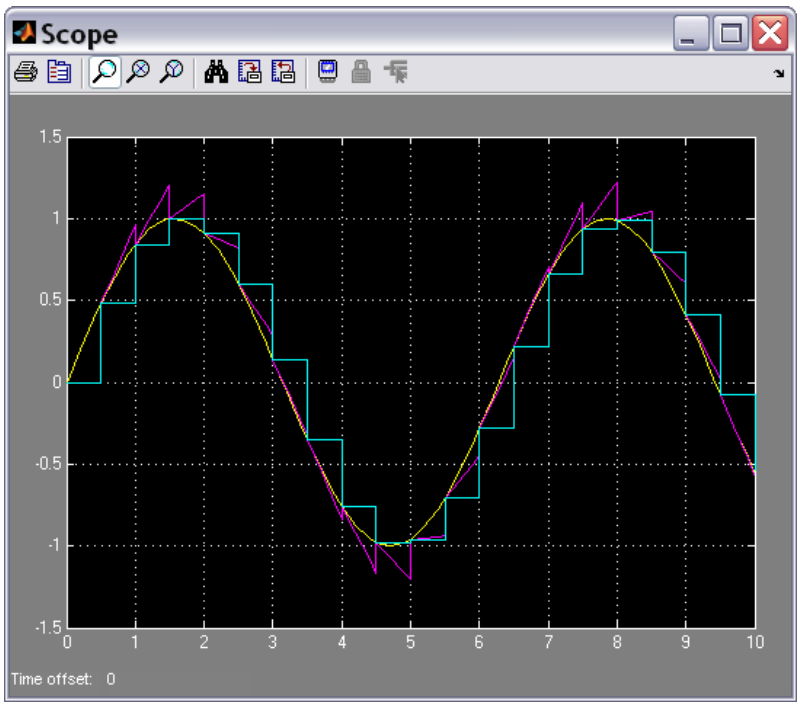


Рис. 3.46. Проходження синусоїдального сигналу через екстраполятори

3.2.5. Поділ Discontinuities (Розривні елементи)

Це, мабуть, найбільш корисний розділ бібліотеки SimuLink. Він включає 12 блоків (рис. 3.47).

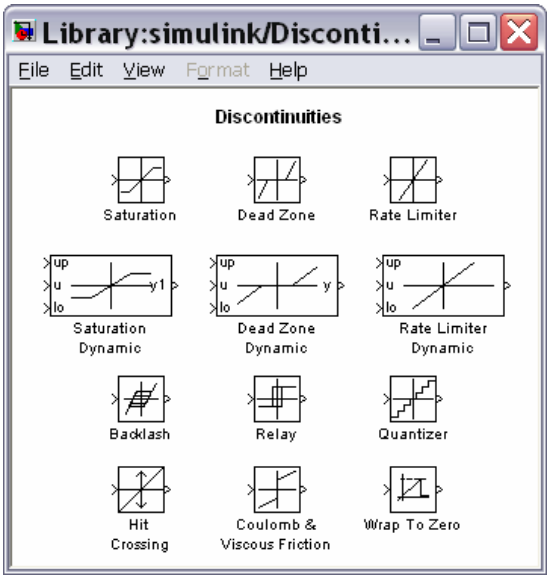


Рис. 3.47. Вміст поділу Discontinuities

Блок ***Saturation*** (*Насичення*) реалізує лінійну залежність із насиченням (обмеженням). Вихідна величина цього блока збігається із вхідною, якщо остання знаходиться усередині зазначеного діапазону. Якщо ж вхідна величина виходить за рамки діапазону, то вихідний сигнал приймає значення найближчої з меж. Значення меж діапазону встановлюються у вікні настроювання блока.

Блок ***Dead Zone*** *(Мертва зона)* реалізує нелінійність типу зони нечутливості. Параметрів настроювання тут два - початок і кінець зони нечутливості.

Блок ***Rate Limiter*** (*Обмежувач швидкості*) забезпечує обмеження згори і знизу швидкості змінювання сигналу, що проходить крізь нього. Вікно настроювання блоку містить два головних параметри: *Rising slew rate* і *Falling slew rate*. Блок працює у такий спосіб: спочатку обчислюється швидкість змінювання сигналу, що проходить кріз нього, за формулою

$$rate = \frac{u(i) - y(i-1)}{t(i) - t(i-1)},$$

де $u(i)$ - значення вхідного сигналу у момент часу $t(i)$; $y(i-1)$ - значення вихідного сигнала у момент $t(i-1)$. Далі, якщо обчислене значення *rate* перевищує значення параметра *Rising slew rate (R)*, вихідна величина визначається за формулою:

$$y(i) = y(i-1) + R \cdot \Delta t .$$

Якщо *rate* є меншим за значення *Falling slew rate (F)*, вихідна величина розраховується так

$$y(i) = y(i-1) + F \cdot \Delta t .$$

У тому випадку, коли значення *rate* міститься проміж значень *R* і *F*, вихідна величина збігається зі вхідною

$$y(i) = u(i) .$$

Блок ***BackLash*** *(Люфт)* реалізує нелінійність типу зазору. У ньому передбачено два параметри настроювання: *Deadband width* - величина люфту і *Initial output* - початкове значення вихідної величини.

Блок ***Relay*** *(Реле)* працює за аналогією зі звичайним реле: якщо вхідний сигнал перевищує деяке граничне значення, то на виході блока формується деякий сталий сигнал. Блок має 4 параметри настроювання:

- *Switch on point (Точка вмикання)* - задає граничне значення, при перевищенні якого відбувається вмикання реле;
- *Switch off point (Точка вимикання)* - визначає рівень вхідного сигналу, при якому реле вимикається;
- *Output when on (Вихід при включеному стані)* - установлює рівень вихідної величини при включеному реле;
- *Output when off (Вихід при виключеному стані)* - визначає рівень вихідного сигналу при виключеному реле.

Блок ***Quantizer*** *(Квантувач)* здійснює дискретизацію вхідного сигналу за його величиною. Єдиним параметром настроювання цього блока є *Quantization interval - Інтервал квантування* - розмір дискрету за рівнем вхідного сигналу.

Блок ***Hit Crossing*** *(Виявити перетинання)* дозволяє зафіксувати стан, коли вхідний сигнал "перетинає" деяке значення. При виникненні такої ситуації на виході блока формується одиничний сигнал. Блок має 3 параметри настроювання (рис. 3.48):

- *Hit crossing offset* - визначає значення, перетинання якого необхідно ідентифікувати;
- *Hit crossing direction* дозволяє зазначити напрямок перетинання, при якому це перетинання повинно виявлятися; значення цього параметра вибирається за допомогою спадного меню, яке містить три альтернативи: *rising (сходження), falling (спадання), either (у будь-якому напрямку);*
- *Show output port (зазначити порт виходу)* - прапорець, за допомогою якого вибирається формат використання блока.

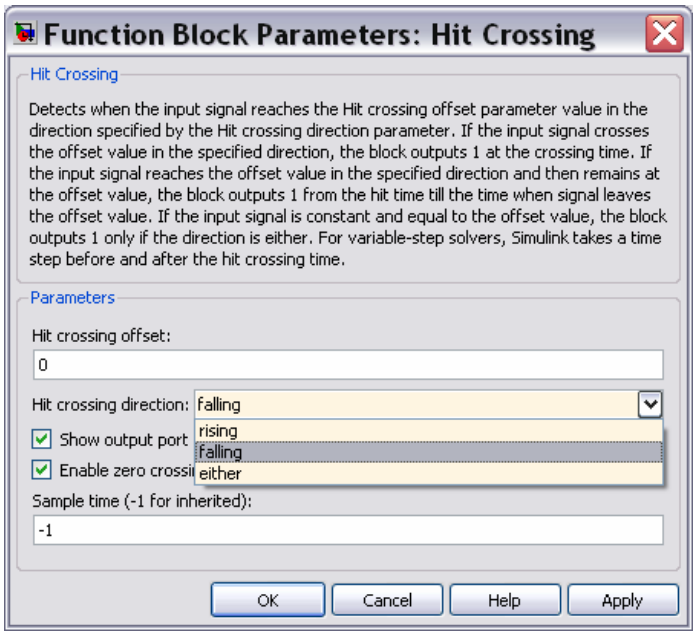


Рис. 3.48. Вікно настроювання блоку Hit crossing

При одночасному виконанні умов, що задаються параметрами *Hit crossing offset* і *Hit crossing direction*, на виході блока формується одиничний сигнал. Його тривалість визначається значенням дискрету часу (параметр *Sample time*) блока, який передує в моделі блоку ***Hit crossing.*** Якщо цей параметр відсутній, то одиничний сигнал на виході блока існує до його наступного спрацьовування.

Блок ***Coulomb & Viscous Friction*** *(Кулонове і в'язке тертя)* реалізує нелінійну залежність типу "лінійна з натягом". Якщо вхід додатний, то вихід пропорційний входові з коефіцієнтом пропорційності - "коефіцієнтом в'язкого тертя" - і збільшений на величину "натягу" ("кулонове, сухе тертя"). Якщо вхід є від'ємним, то вихід також є пропорційним входові (із тим же коефіцієнтом пропорційності) за відрахуванням величини "натягу". При вході рівному нулю вихід теж дорівнює нулю. У параметри настроювання блока входять величини "кулонова тертя" ("натягу") і коефіцієнта "в'язкого тертя".

Блок ***Wrap To Zero*** (Скидання у нуль) забезпечує рівність нулеві вихідного сигналу, якщо значення вхідного сигналу перевищує встановлене значення єдиного параметра настроювання *Threshold (Поріг)*. Якщо ж вхідний сигнал менше за Threshold вихідний сигнал дорівнює вхідному.

3.2.6. Поділ Signal Routing (Пересилання сигналів)

Поділ бібліотеки ***Connections*** *(Зв'язки)* призначений для побудови складних S-моделей, що складаються з інших моделей більш низького рівня. Склад блоків наведений на рис. 3.49.

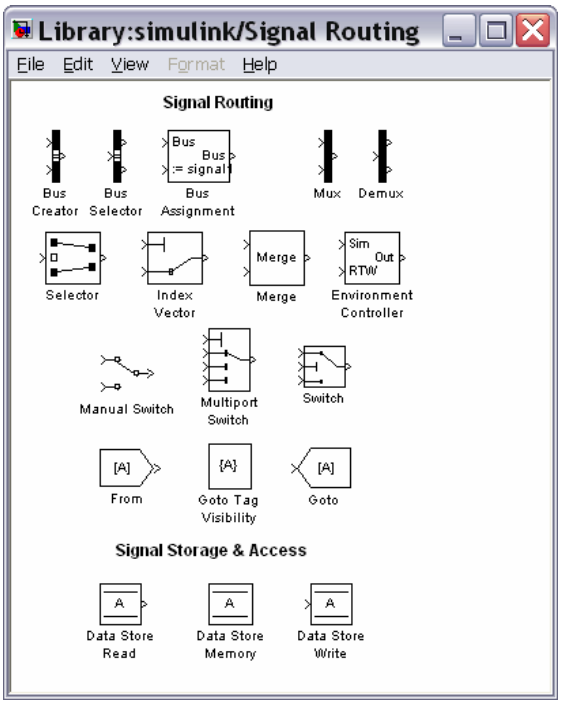


Рис. 3.49. Вміст поділу Signal Routing

Блок ***Mux*** *(Мультіплексор)* виконує об'єднання вхідних величин у єдиний вихідний вектор. При цьому вхідні величини можуть бути як скалярними, так і векторними. Довжина результуючого вектора дорівнює сумі довжин усіх векторів. Порядок елементів у векторі виходу визначається порядком входів (зверху униз) і порядком розташування елементів усередині кожного входу. Блок має один параметр настроювання - *Number of inputs (Кількість входів).*

Блок ***Demux*** *(Подільник, Демультіплексор)* виконує зворотну функцію - *поділяє вхідний вектор на задану кількість компонентів*. Він також має єдиний параметр настроювання *Number of outputs (Кількість виходів).* У випадку, коли зазначене число виходів (N) задається меншим довжини вхідного вектора (M), блок формує вихідні вектори в такий спосіб. Перші (N-1) виходів будуть векторами однакової довжини, рівної цілій частині відношення M/(N-1). Останній вихід буде мати довжину, рівну залишку від ділення.

Блоки ***Bus Creator*** (Побудувач шини) і ***Bus Selector*** (Роздільник шини) взагалі виконують ті самі функції, що й відповідно блоки ***Mux*** і ***Demux***, але мають більші можливості щодо перерозподілу сигналів всередині шини.

Блоки ***From*** *(Прийняти від)*, ***Goto Tag Visibility*** *(Ознака видимості)* і ***Goto*** *(Передати до)* використовуються спільно і призначені для обміну даними

між різноманітними частинами S-моделі з урахуванням досяжності (видимості) цих даних.

Блоки ***Data Store Read*** *(Читання даних),* ***Data Store Memory*** *(Запам'ятовування даних)* і ***Data Store Write*** *(Запис даних)* також використовуються спільно і забезпечують не тільки передачу даних, але і їхнє збереження на інтервалі моделювання.

Блок ***Merge*** *(Злиття)* виконує об'єднання сигналів, що надходять до його входів, у єдиний.

Група блоків-перемикачів, що керують напрямком передачі сигналу, складається з трьох блоків: ***Switch*** *(Перемикач)*, ***Manual Switch*** *(Ручний перемикач)* і ***Multiport Switch*** *(Багатовходовий перемикач).*

Блок ***Switch*** має три входи: два (1-й і 3-й) інформаційні й один (2-й) – керуючий - і один вихід. Якщо величина керуючого сигналу, що надходить до другого входу, не менше за деяке задане межове значення (параметр *Threshold - Поріг*), то на вихід блока передається сигнал з 1-го входу, у противному разі - сигнал із 3-го входу.

Блок ***Manual Switch*** не має параметрів настроювання. У нього два входи й один вихід. На зображенні блока показано перемичкою, який саме із двох входів залучений до виходу. Блок дозволяє "вручну" перемикати входи. Для цього необхідно двічі клацнути мишкою на зображенні блока. При цьому зміниться і зображення блока - на ньому вихід уже буде сполучений перемичкою з іншим входом.

Блок ***Multiport Switch*** має не менше трьох входів. Перший (горішній) з них є керуючим, інші - інформаційними. Блок має один параметр настроювання *Number of inputs (Кількість входів)*, який визначає кількість інформаційних входів. Номер входу, що з'єднується з виходом, дорівнює значенню керуючого сигналу, що надходить на верхній вхід. Якщо це значення є дробовим числом, то воно округлюється до цілого по звичайних правилах. Якщо воно менше за одиницю, то воно вважається рівним 1; якщо воно більше кількості інформаційних входів, то воно приймається рівним найбільшому номеру (входи нумеруються зверху униз, крім самого верхнього - керуючого).

3.2.7. Поділ Math Operations (Математичні операції)

У поділі ***Math Operations*** *(Математичні операції)* містяться блоки, які реалізують деякі вбудовані математичні функції системи Matlab. Вони об’єднані у три групи (рис. 3.50)

- *Math Operations (Математичні операції)*,
- *Vector/Matrix Operations (Векторно-матричні операції)*
- *Complex Vector Conversions (Перетворення комплексного вектора)*.

У першу групу *Math Operations* входять блоки, які здійснюють математичні перетворення вхідного сигналу у вихідний.

Блоки ***Sum***, ***Add***, ***Subtract*** і ***Sum of Elements*** здійснюють підсумовування сигналів, що надаються до їхніх входів.

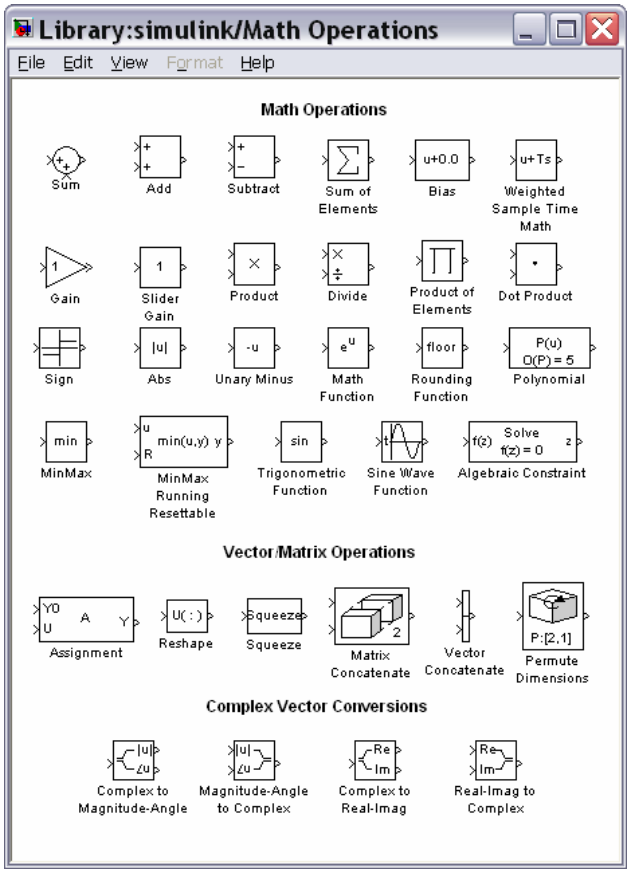


Рис. 3.50. Вміст поділу Math Operations

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

- додавання вхідних сигналів (у тому числі з різними знаками);
- підсумовування елементів вектора, що надходить на вхід блока.

Для керування режимами роботи блока використовується два параметри (рис. 3.51):

- *Icon Shape* (Форма зображення),
- *List of signs* (Список знаків).

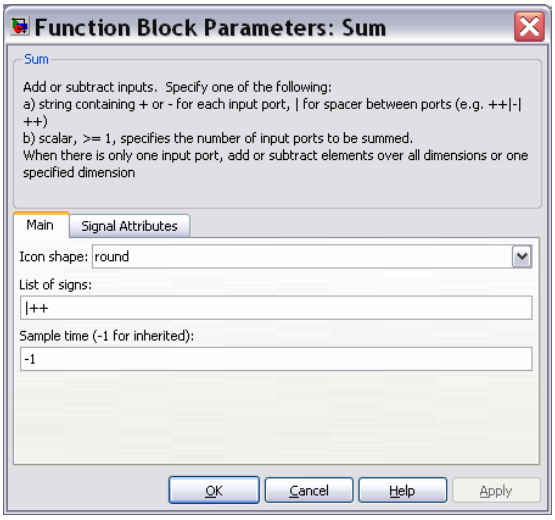


Рис. 3.51. Вікно настроювання блоку Sum

Першій параметр може приймати два значення: *round* (круглий) і rectangular (прямокутний).

Значення другого параметра можуть задаватися одним із трьох способів:

- у виді послідовності знаків "+" або "-"; при цьому *кількість знаків визначає кількість входів блока, а самий знак - полярність відповідного вхідного сигналу*;
- у виді цілої додатної і більше 1 константи; *значення цієї константи визначає кількість входів блока, а усі входи вважаються додатними*;
- у виді символу "1", який указує, що *блок використовується в другому режимі*

Блок ***Bias*** додає постійну величину (параметр настроювання *Bias*) до вхідного сигналу.

Блок ***Gain*** є лінійною підсилювальною ланкою, тобто здійснює множення вхвдного сигналу на постійне число або вектор, значення якого задається параметром настроювання і вказується на зображенні блоку. Вхідна величина блоку (u) може бути скалярною, векторною або матричною. У випадку, коли вхідний сигнал є вектором довжиною N елементів, коефіцієнт підсилювання має бути вектором тої самої довжини. У списку *Multiplication* (Множення) обирається один з наступних способів множення вхідної величини на вектор K коефіцієнтів підсилювання: *Element wise(K.*u)* – поелементне множення вхідного вектора на вектор коефіцієнтів підсилювання; *Matrix(K*u)* - матричне множення вектора коефіцієнтів підсилювання на матрицю вхідних величин; *Matrix(u*K)* - матричне множення матриці вхідних величин на вектор коефіцієнтів підсилювання; *Matrix(K*u) (u vector)* - матричне множення векторів K і u.

Блок ***Slider Gain*** є різновидом підсилювальної ланки і одним з елементів взаємодії користувача з моделлю. Він дозволяє в зручній діалоговій формі змінювати значення деякого параметра в процесі моделювання. Щоб відчинити вікно з "повзунковим" регулятором (рис. 3.52), необхідно подвійно клацнути мишкою на зображенні блока.

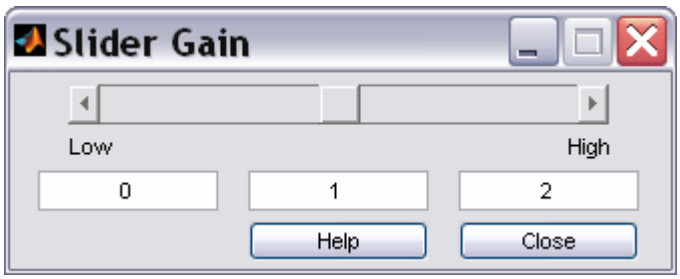


Рис. 3.52. Вікно настроювання блоку Slider Gain

Вікно ***Slider Gain*** має три поля введення інформації:

- для вказівки нижньої межі змінювання параметра (*Low*);
- для вказівки верхньої межі змінювання параметра (*High*);

■ для вказівки поточного значення.

Поточне значення має лежати усередині діапазону [*Low, High*]. Його можна вводити, записуючи значення у середнє поле, або переміщуючи мишкою повзунок у верхній частині вікна настроювання. Проте при виборі нового діапазону необхідно спочатку зазначити нове значення параметра, а потім змінити межі діапазону.

Блок ***Product*** виконує множення або ділення кількох вхідних сигналів. У параметри настроювання входять кількість входів блока (*Number of inputs*) й вид виконуваної операції (*Multiplication*). Завдання значень цих параметрів аналогічно настроюванню блока ***Sum.*** Якщо як значення параметра настроювання блока ввести "1", буде обчислюватися добуток елементів вхідного вектора. При цьому на зображенні блока виводиться символ ***P.*** Вхідні сигнали можуть бути векторними або матричними. У списку *Multiplication* обирається спосіб множення вхідних величин: *Element wise* – поелементне множення вхідних векторів або матриць; *Matrix* – матричне множення вхідних векторів або матриць. У випадку, коли результат виконання має містити ділення на деякі вхідні величини, у полі *Number of inputs* (Кількість входів) слід ввести послідовність символів "*" або "/" (за кількістю входів блоку). Якщо обрано матричне множення, то символ "/" означає множення на матрицю, обернену по відношенню до матриці відповідної вхідної величини.

У блоці ***Dot Product*** (Скалярний добуток) є лише два входи. Вхідні сигнали блоку мають бути векторами однакової довжини. Вихідна величина блоку у кожний момент часу дорівнює сумі добутків відповідних елементів цих двох векторів. Якщо вектори є комплексними, то перед множенням першій вектор (верхній вхідний порт) замінюється комплексно-спряженим.

Блок ***Sign*** реалізує нелінійність типу сигнум-функції. У ньому немає параметрів настроювання. Блок формує вихідний сигнал, що приймає тільки три можливих значення: "+1" - у випадку, коли вхідний сигнал є додатним, "-1" - при від'ємному вхідному сигналі і "0" при вхідному сигналі, рівному нулю.

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

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

Блок ***Math Function*** дозволяє обрати для перетворення вхідного сигналу елементарні не тригонометричні і не гіперболічні функції, такі як ***exp***, ***log***, ***10^u***, ***log10***, ***square*** (u^2), ***sqrt***, ***pow***, ***reciprocal*** (1/u), ***hypot***, ***rem***, ***mod***. Потрібна функція обирається за допомогою спадного меню у вікні настроювання.

Блок ***Rounding Function*** містить різноманітні функції округлення, передбачені в MatLAB. Він здійснює округлення значення вхідного сигналу. Вибір конкретного методу округлення здійснюється також за допомогою спадного меню у вікні настроювання.

Для зазначених вище блоків ім'я обраної функції виводиться на графічному зображенні блока.

Блок ***MinMax*** здійснює пошук мінімального або максимального елемента вхідного вектора. Якщо входом є скалярна величина, то вихідна величина збігається із вхідною. Якщо входів декілька, шукається мінімум або максимум серед входів. У число настроювань входить вибір методу й кількість входів блока.

Блоки третьої групи - *Complex Vector Conversions* (Перетворення комплексного вектора) здійснюють перетворення комплексних вхідних сигналів у дійсні і навпаки. Блоки ***Complex to Magnitude-Angle*** та ***Complex to Real-Imag*** здіюснюють перетворення комплексного вхідного сигнала у два дійсних сигнали, що є модулем і аргументом вхідного сигналу у першому випадку і дійсною і уявною частинами – у другому випадку. Блоки ***Magnitude-Angle to Complex*** та ***Real-Imag to Complex*** перетворюють два вхідних дійсних сигнали у єдиний комплексний.

3.2.8. Поділ Logic & Bit Operations (Логічні та бітові операції)

Як вказано у назві поділу, він містить блоки, що забезпечують певну логічну обробку вхідних величин, або перетворення їх двійкового подання (рис. 3.53).

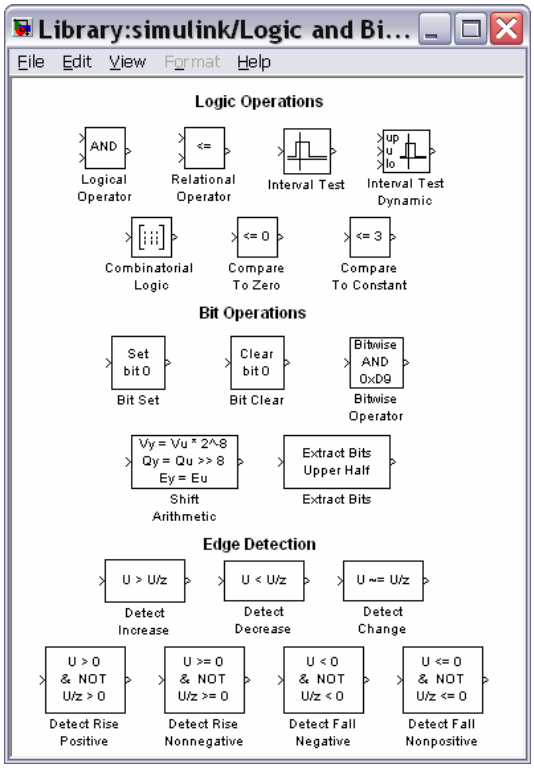


Рис. 3.53. Вміст поділу Logic & Bit Operations

Загальним для всіх блоків групи Logic Operations є те, що вихідна величина в усіх них є бульовою, тобто може приймати лише два значення: "1"

("істина") або "0" ("хибність"). У багатьох з них бульовими мають бути й усі вхідні величини.

Блок ***Relational Operator*** реалізує операції відношення між двома вхідними сигналами >, <, ≤, ≥, ==, ~= (відповідно: більше, менше, менше або дорівнює, більше або дорівнює, тотожно, не дорівнює). Конкретна операція обирається при настроюванні параметрів блока за допомогою спадного меню. Знак операції виводиться на зображенні блока.

Блок ***Logical Operator*** містить набір основних логічних операцій AND, OR, NAND, NOR, XOR, NOT. Вхідні величини мають бути бульовими. обрання необхідної логічної операції здійснюється у вікні настроювання блока за допомогою спадного меню. Другим параметром настроювання є кількість вхідних величин (портів) блока (*Number of input ports*), тобто кількість аргументів логічної операції.

Блок ***Combinatorial Logic*** забезпечує перетворення вхідних бульових величин у вихідну у відповідності із заданою таблицею істинності. Блок має єдиний параметр настроювання - *Truth table* (таблиця істинності).

3.2.9. Поділ User-defined Functions (Функції, що визначаються користувачем)

У поділі User-defined Functions (Користувацькі функції) містяться блоки, призначення яких визначає користувач (рис. 3.54).

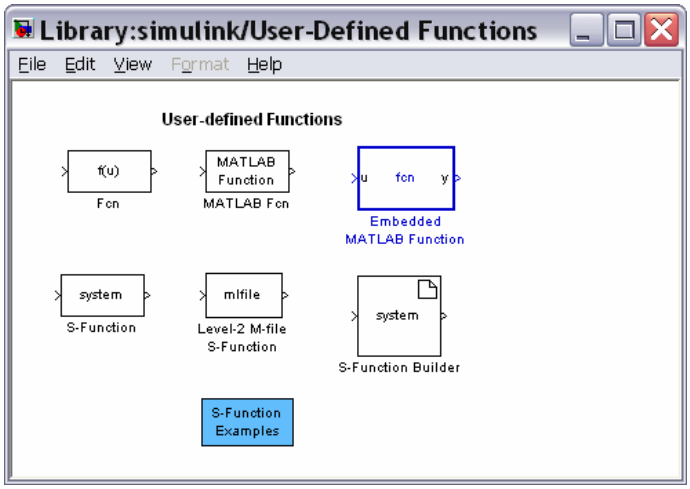


Рис. 3.54. Вміст поділу User-defined Functions

Блок ***Fcn*** дозволяє користувачеві ввести будь-яку скалярну функцію від одного (скалярного або векторного) аргументу, яка виражається через стандартні функції MatLAB. Вираз функції вводиться у вікні настроювання блока згідно правил М-мови. Для позначення вхідного сигналу (аргументу функції) використовується символ ***u***.

Блок ***MATLAB Fcn*** дозволяє застосувати до вхідного сигналу будь-яку підпрограму обробки, реалізовану у виді М-файлу. На відміну від попереднього блока, тут до числа параметрів настроювання доданий параметр *Output width (Ширина вихідного сигналу)*, який визначає кількість елементів вихідного

вектора. Окремий *i*-й елемент вихідного вектора у вікні настроювання ***MATLAB Fcn*** задається у виді функції, що записана на М-мові, якій передує запис **u(i)=**.

За допомогою блоку *S-function* користувач має змогу реалізувати у виді візуального блоку Simulink будь-яку програму обробки вхідного сигналу, включаючи створення складних моделей систем, що описані нелінійними диференційними або кінцево-різницєвими рівняннями, і обробку дискретних у часі сигналів. Більш детально робота з S-функціями описана далі, у главі 4.

Блок ***S-function Builder*** (Побудувач S-функцій) дає можливість користувачеві утворювати S-функцію у діалоговому режимі.

3.2.10. Поділ Ports & Subsystems (Порти та підсистеми)

Більшість блоків поділу *Ports & Subsystems* призначені для розробки складних за ієрархією S-моделей, що містять моделі більш низького рівня (підсистеми), а також забезпечують встановлення необхідних зв'язків між кількома S- моделями (рис. 3.55).

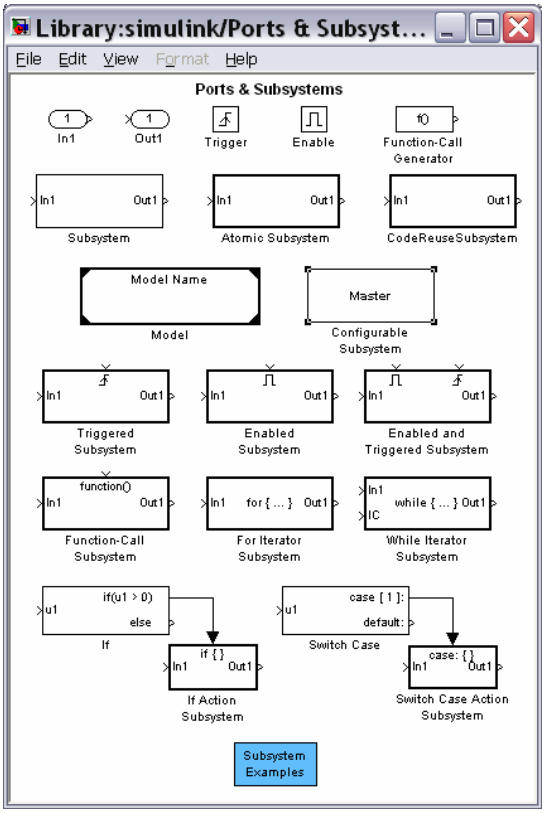


Рис. 3.55. Вміст поділу Ports & Subsystems

Підсистема являє собою S-модель більш низького рівня, у яку можуть бути вкладені підсистеми різних рівнів. підсистеми можуть функціонувати лише у складі основної S-моделі, зв'язок з якою здійснюється через вхідні (***In***) і вихідні (***Out***) порти підсистеми.

Роль підсистеми у S-моделі така сама, що й роль функцій (процедур) в основній M- програмі, яка їх викликає. При цьому вхідні і вихідні величини підсистеми визначаються відповфідно її вхідними і вихідними портами. Результати виконання дій у підсистемі (вихідні величини) у подальшому можуть бути використані у S-моделі, яка її викликає (або у підсистемі більш високого рівня).

Застосування підсистем дозволяє звести складання складної S-моделі до утворення сукупності вкладених простих підсистем більш низького рівня, що робить моделювання більш наочним і спрощує налагодження моделі.

Опишемо основні блоки поділу.

Блоки ***In*** (Вхідний порт) і ***Out*** (Вихідний порт) забезпечують інформаційний зв'язок між підситемою і S-моделі, що її викликає.

Блоки ***Enable*** *(Дозволити)* і ***Trigger*** *(Засувка)* призначені для логічного керування роботою підсистем S-моделі.

Раніше розглянути блоки ***Ground*** (Земля) і ***Terminator*** (Обмежувач) можуть використовуватися як "заглушки" для тих портів, які з якоїсь причини не були з'єднані з іншими блоками S-моделі. при цьому блок ***Ground*** застосовується для вхідних портів, а блок ***Terminator*** – для вихідних.

Блок ***Subsystem*** *(Підсистема)* є "заготівкою" для створення підсистеми. Подвійне клацання на його зображенні приводить до появі на екрані вікна, в якому розміщена блок-схема, що складається лише з одного вхідного порта, який з'єднаний з одним вихідним портом (рис. 3.56). Це є нагадуванням, що утворювана користувачем підситема має обов'язково містити з'єднані між собою (можливо, через інші блоки) вхідні і вихідні порти.

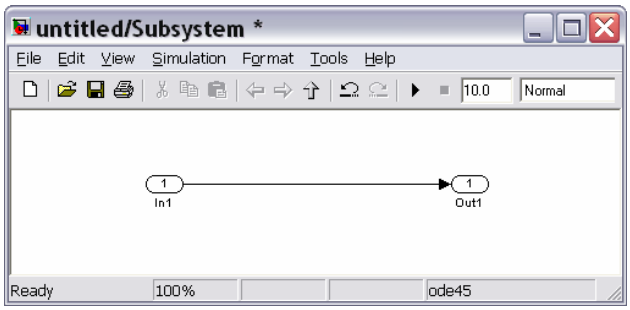


Рис. 3.56. Вікно заготівки блоку Subsystem

У вікні, що відчинилося, користувач будує блок-схему підситеми за звичайними правилами побудови блок-схем, а потім записує її на диск. Розміщення додаткових вхідних і вихідних портів приведе до почви на зображенні блоку ***Subsystem*** додаткових входів і виходів. При цьому поряд з відповідними входами і виходами блока ***Subsystem*** виникнуть написи, що зроблені на вхідних і вихідних портах підсистеми.

3.3. Побудова блок-схем

Розглянемо операції, за допомогою яких можна формувати блок-схеми складних динамічних систем.

3.3.1. Виділення об'єктів

При створенні й редагуванні S-моделі потрібно виконувати такі операції, як копіювання або вилучення блоків і ліній, для чого необхідно спочатку виділити один чи кілька блоків і ліній (об'єктів).

Щоб ***виділити окремий об'єкт,*** потрібно клацнути на ньому один раз. При цьому з'являються маленькі кола по рогах блока, або на початку й кінці лінії. *При цьому стають невиділеними усі інші попередньо виділені об'єкти*. *Якщо клацнути по блоку другий раз, він стає невиділеним*.

На рис. 3.57 наведений результат виділення лінії, що з'єднує блоки Constant і XYGraph, а на рис. 3. 58 - блоку ***Clock***.

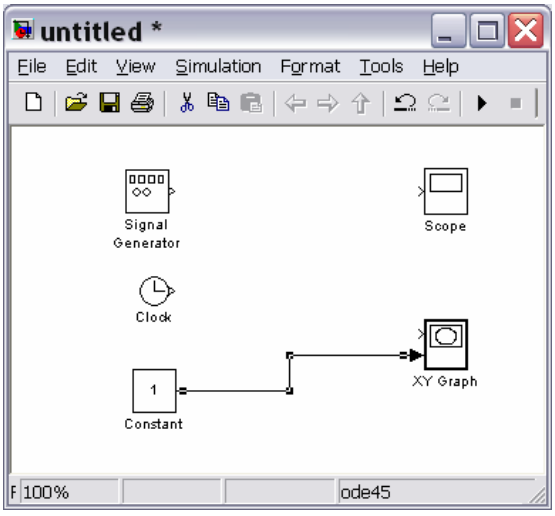


Рис. 3.57. Виді лена лінія

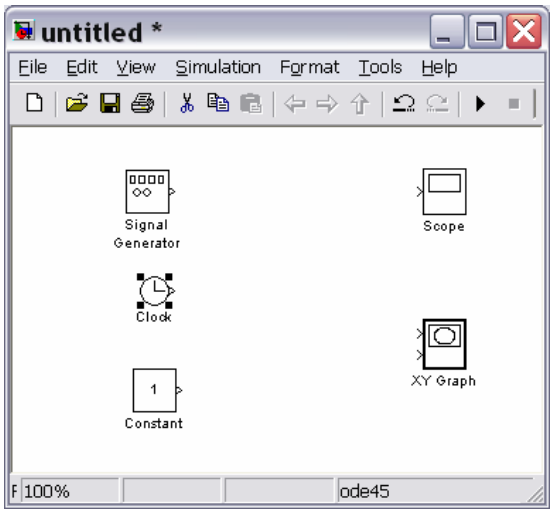


Рис. 3.58. Виділений блок Clock

Щоб виділити кілька об'єктів, слід, утримуючи натисненою клавішу Shift, клацнути на кожному з них, а потім відпустити клавішу .

Саме у такий спосіб на рис. 3.59 виділені блоки ***Signal Generator***, ***Constant*** і ***XYGraph***.

Кілька об'єктів можна виділити також за допомогою прямокутної рамки. Для цього потрібно клацнути мишкою у точці, яка буде прислуговуватися рогом рамки, а потім, утримуючи кнопку миші натисненою, протягнути курсор у напрямку діагоналі прямокутника. В результаті навколо об'єктів, що виділяються має виникнути пунктирна рамка (рис. 3.60). Коли усі потрібні об'єкти будуть охоплені рамкою, потрібно відпустити кнопку миші.

Виділення усієї моделі, тобто усіх об'єктів в активному вікні блок-схеми, здійснюється одним із двох шляхів:

1) обранням команди *Select All* у меню *Edit* вікна блок-схеми;

2) натисканням сукупності клавіш <Ctrl>+<A>.

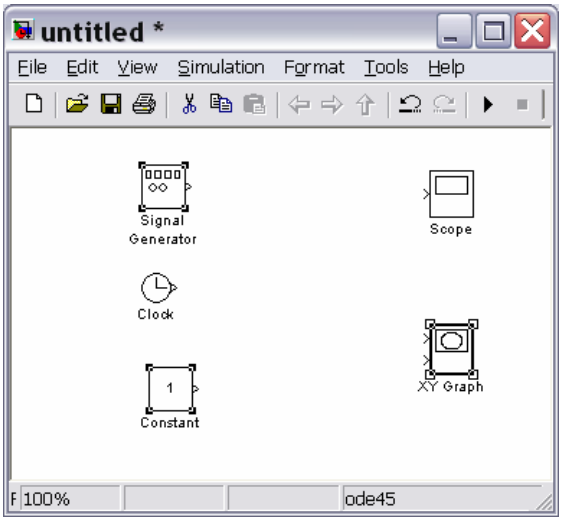


Рис. 3.59. Виділення кількох блоків

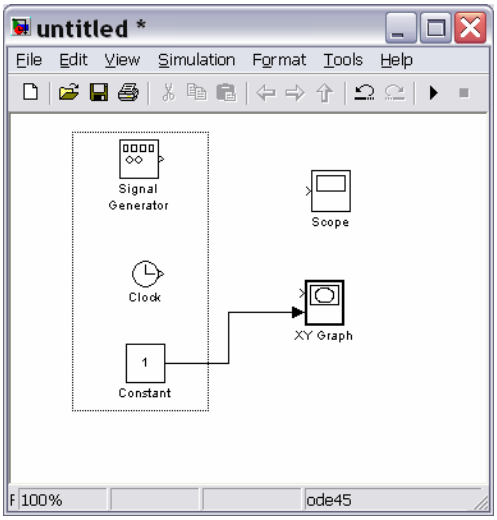


Рис. 3.60. Виділення за допомогою рамки

3.3.2. Операції з блоками

Копіювання блоків з одного вікна у інше

Під час створення й редагування моделі потрібно копіювати блоки з бібліотеки або іншої моделі у поточну модель. Для цього достатньо:

- відчинити потрібний розділ бібліотеки чи вікно моделі - прототипу;
- перетягнути мишкою потрібний блок у вікно утворюваної (редагованої) моделі.

Інший спосіб є таким:

1) виділити блок, який потрібно скопіювати;
2) обрати команду *Copy (Копіювати)* з меню *Edit (Редагування)*;
3) зробити активним вікно, в яке потрібно скопіювати блок;
4) обрати в ньому команду *Paste (Встановити)* з меню *Edit*.

Кожному зі скопійованих блоків автоматично присвоюється ім'я. *Першій скопійований блок матиме те саме ім'я, що й блок у бібліотеці*. *Кожний наступний блок того ж типу матиме те саме ім'я з додаванням порядкового номера*. Користувач може перейменувати блок (див. далі).

При копіюванні блок одержує ті самі значення настроюваних параметрів, що й блок - оригінал.

Переміщення блоків у моделі

Щоб перемістити блок усередині моделі з одного місця у інше, достатньо перетягнути його у це положення за допомогою мишки. При цьому будуть автоматично перерисовані лінії зв'язків інших блоків з тим, якого переставлено.

Переставити кілька блоків одночасно, включаючи з'єднувальні лінії можна у такий спосіб:

- виділити блоки й лінії (див попередній розділ);
- перетягнути мишкою один із виділених блоків на нове місце; решта блоків, зберігаючи всі відносні відстані, посядуть нові місця.

На рис. 3.61 показаний результат таких дій з блоками, виділеними на рис. 3.59.

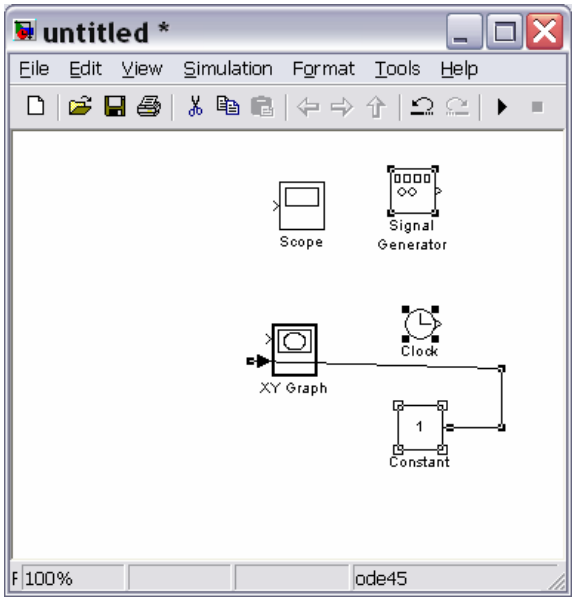


Рис. 3.61. Результат переміщення блоків, виділених на рис. 3.60

Дублювання блоків усередині моделі

Щоб ***скопіювати блоки усередині моделі*** потрібно зробити наступне:

1) натиснути клавішу <Ctrl>;
2) не відпускаючи клавішу <Ctrl>, встановити курсор на блок, який необхідно скопіювати й перетягнути його у нове положення.

Того самого результату можна досягти, якщо просто *перетягти* мишкою блок у нове положення, але за допомогою *правої* клавіші мишки.

На рис. 3.62 подано результат копіювання блоків ***Scope*** і ***XYGraph***.

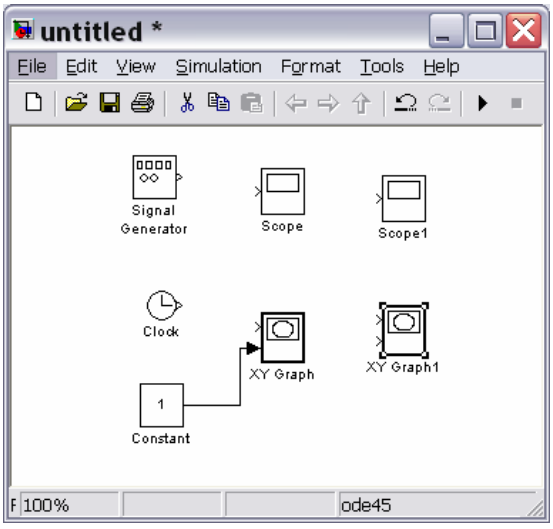


Рис. 3.62. Результат копіювання блоків

Встановлення параметрів блока

Функції, які виконує блок, залежать від значень параметрів блока. Встановлення цих значень здійснюється у вікні настроювання блока, яке виникає, якщо подвійно клацнути на зображенні блока у блок-схемі.

Вилучення блоків

Для ***вилучення непотрібних блоків*** із блок-схеми достатньо виділити ці блоки так, як було зазначено раніше, і натиснути клавішу <Delete> або <Backspace>. Можна також використати команду *Clear* або *Cut* із розділу *Edit* меню вікна блок-схеми. Якщо використано команду *Cut*, то у подальшому вилучені блоки можна скопіювати зворотно у модель, якщо скористатися командою *Paste* того ж розділу меню вікна схеми.

Від'єднання блока

Для ***від'єднання блоку від з'єднуючих ліній*** достатньо натиснути клавішу <Shift> і, не відпускаючи її, перетягнути блок у деяке інше місце.

Змінювання кутової орієнтації блока

У звичайному зображенні сигнал проходить крізь блок зліва направо (ліворуч містяться входи блока, а праворуч - виходи). Щоб ***змінити кутову орієнтацію блока*** потрібно:

- виділити блок, який потрібно повернути;
- обрати меню *Format* у вікні блок-схеми;
- у додатковому меню, яке з'явиться на екрані обрати команду *Flip Block* - поворот блока на 180 градусів, або *Rotate Block* - поворот блока за годинниковою стрілкою на 90 градусів.

На рис. 3.63 показаний результат застосування команди *Rotate Block* до блоку ***Constant*** і команд *Rotate Block* і *Flip Block* - до блока ***Signal Generator***.

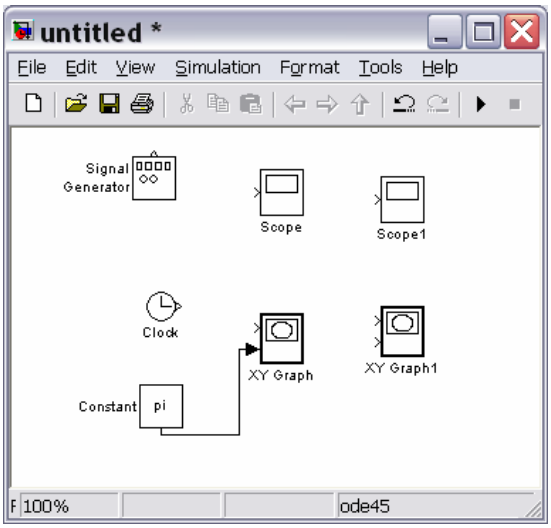


Рис. 3.63. Результат зміненяя орієнтації блоків

Змінювання розмірів блока

Щоб змінити розміри блока, потрібно зробити наступне:

- виділити блок, розміри якого треба змінити;
- навести курсор мишки на одну з рогових міток блоку; при цьому на екрані у цієї мітки повинен з'явитися новий курсор у вигляді обопільної стрілки під нахилом 45 градусів;

- захопити цю мітку мишкою і перетягнути у нове положення; при цьому протилежна мітка цього блоку залишиться нерухомою.

На рис. 3.64 показаний результат процесу збільшення розмірів блока ***XYGraph1***.

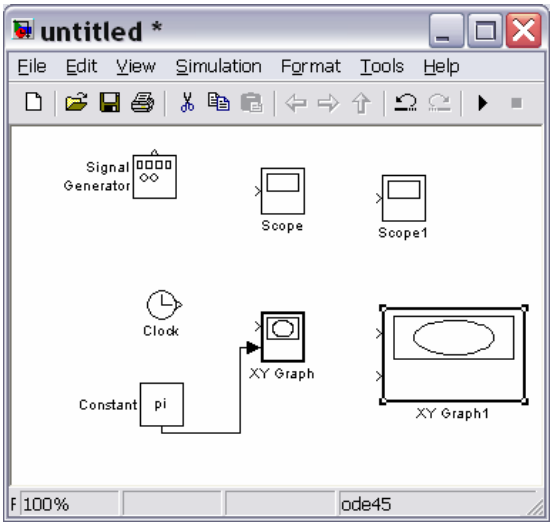


Рис. 3.64. Результат розтягування блоку XYGraph1

Змінювання імен блоків та маніпулювання з ними

Усі імена блоків у моделі мають бути унікальними і мати, як мінімум один символ. Якщо блок орієнтований зліва направо, то ім'я, за замовчуванням, міститься під блоком, якщо справа наліво - понад блоком, якщо ж зверху униз або знизу уверх - праворуч блоку (див. рис. 3.64).

Змінювання йменння блоку здійснюється у такий спосіб: треба клацнути на існуючому імені блока, потім, використовуючи клавіші звичайного редагування тексту, змінити це ім'я на потрібне.

Для ***змінювання шрифту*** слід виділити блок, потім обрати команду *Font* (Шрифт) із меню *Format* вікна моделі і обрати потрібний шрифт із поданого переліку.

Щоб ***змінити місце розташування йменння виділеного блока,*** існують два шляхи:

- перетягнути ім'я на протилежний бік мишкою;
- скористатися командою *Flip Name* (Розвернути ім'я) із поділу *Format* меню вікна моделі - вона теж переносить ім'я на протилежний бік.

Вилучити ім'я блоку можна,використовуючи команду *Hide Name* (Сховати ім'я) з меню *Format* вікна моделі. Щоб ***відновити*** потім ***відображення імені*** поряд із зображенням блоку, слід скористатися командою *Show Name* (Показати ім'я) того самого меню.

3.3.3. Проведення з'єднувальних ліній

Сигнали у моделі передаються по лініях. Кожна лінія може передавати або скалярний, або векторний сигнал. Лінія з'єднує вихідний порт одного блоку

із вхідним портом іншого блоку. Лінія може також з'єднувати вихідний порт одного блоку із вхідними портами кількох блоків через розгалужування лінії.

Створення лінії між блоками

Для сполучення вихідного порту одного блока із вхідним портом іншого блоку слід виконати таку послідовність дій:

- встановити курсор на вихідний порт першого блока; при цьому *курсор має перетворитися на перехрестя*;
- утримуючи натисненою ліву кнопку миші, пересунути перехрестя до вхідного порту другого блоку;
- відпустити кнопку миші; Simulink замінить символи портів з'єднувальною лінією з поданням напрямку передавання сигналу.

Саме таким чином з'єднано на рис. 3.60 вихід блока ***Constant*** із входом блоку ***XYGraph***.

Лінії можна рисувати як от вхідного порту до вихідного, так і навпаки.

Simulink малює з‘єднувальні лінії, використовуючи лише горизонтальні й вертикальні сегменти *Для утворення діагональної лінії натисніть і утримуйте клавішу <Shift> протягом рисування.*

Створення розгалуження лінії

Лінія, що розгалужується, починається з існуючої і передає її сигнал до вхідного порту іншого блоку. Як існуюча, так і відгалужена лінія передають той самий сигнал. Розгалужена лінія дає можливість передати той самий сигнал до кількох блоків.

Щоб ***утворити відгалуження*** від існуючої лінії, потрібно:

- установити курсор на точку лінії, від якої має відгалужуватися інша лінія;
- натиснувши й утримуючи клавішу <Ctrl>, натиснути й утримувати ліву кнопку миші;
- провести лінію до вхідного порту потрібного блоку; відпустити клавишу <Ctrl> і ЛКМ (див. рис. 3.65).

Те саме можна зробити, використовуючи праву кнопку миші, при цьому не потрібно користатися клавишою <Ctrl>.

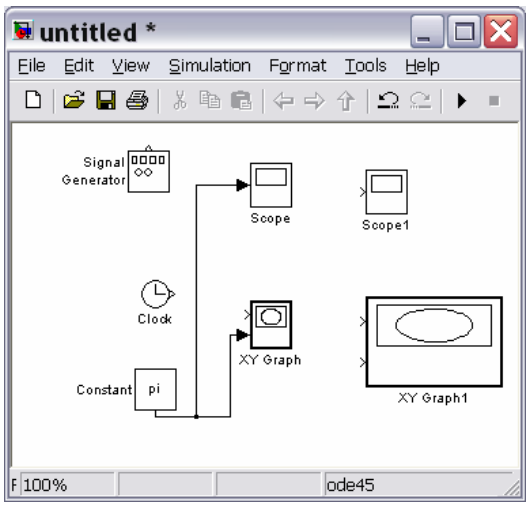


Рис. 3.65. Утворення розгалуження лінії

Створення сегмента лінії

Лінії можуть бути нарисовані по сегментах. У цьому разі для створення наступного сегмента слід установити курсор у кінець попереднього сегмента і нарисувати (за допомогою миші) наступний сегмент. У такий спосіб, наприклад, з'єднані на рис. 3.66 блоки ***Clock*** з ***XYGraph1*** та ***Signal Generatorr*** з ***Scope1***.

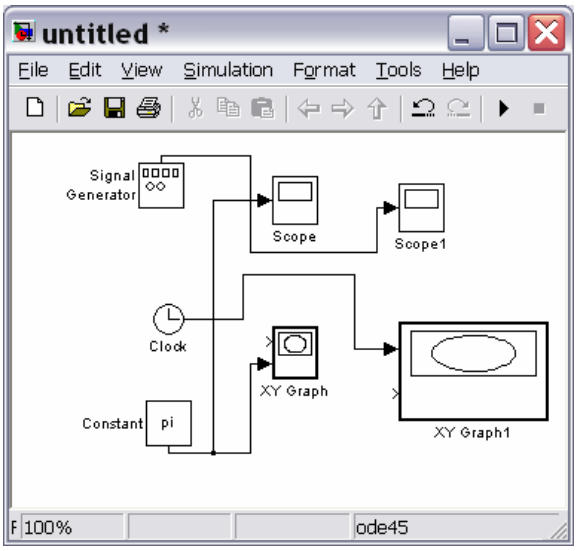


Рис. 3.66. Лінії, нарисовані по сегментах

Пересування сегмента лінії

Щоб ***пересунути окремий сегмент*** лінії, необхідно виконати наступне:

- установити курсор на сегмент, який потрібно пересунути;
- натиснути й утримувати ліву кнопку миші; при цьому курсор має перетворитися на "хрест";
- пересунути "хрест" до нового положення сегмента;
- відпустити кнопку миші.

На рис. 3.67 показаний результат пересування вертикального сегменту лінії, що з'єднує блоки ***Random Number*** з ***XYGraph1***.

Не можна пересунути сегмент, який безпосередньо прилягає до порту блока.

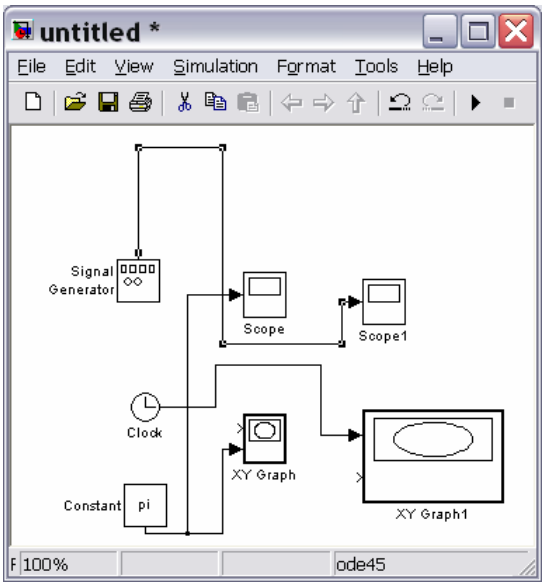


Рис. 3.67. Сегмент лінії переміщений угору

Поділення лінії на сегменти

Щоб ***поділити лінію на два сегменти,*** потрібно:

- виділити лінію;
- установити курсор у ту точку виділеної лінії, у якій лінія має бути поділеною на два сегменти;
- утримуючи натисненою клавішу <Shift>, натиснути й утримувати ліву кнопку миші; курсор перетвориться на маленьке коло; на лінії утвориться злам;
- пересунути курсор у нове положення зламу;
- відпустити кнопку миші і клавішу <Shift>.

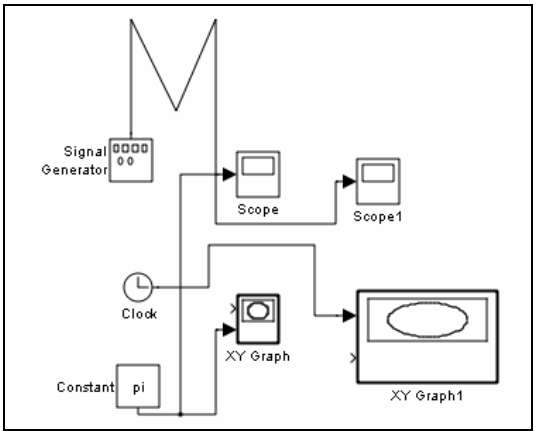


Рис. 3. 68. Верхня горизонтальна лінія поділена на два сегменти

Приклад проведення цих дій подано на рис. 3.68, де лінія, яка з'єднує блоки ***Sine Wave*** і ***XYGraph1*** поділена на два сегменти.

Пересування зламу лінії

Для ***пересування зламу*** лінії достатньо перетягнути точку цього зламу у нове положення на блок-схемі.

3.3.4. Мітки сигналів і коментарів

Задля наочності оформлення блок-схеми й зручності користування нею можна супроводжувати лінії мітками сигналів, що прямують по ній. Мітка розміщується під або над горизонтальною лінією, ліворуч або праворуч вертикальної лінії. Мітка може бути розташована на початку, у кінці або посередині лінії.

Створення й маніпулювання мітками сигналів

Щоб ***створити мітку сигналу,*** треба подвійно клацнути на сегменті лінії і ввести мітку (рис. 3.69). *При створенні мітки сигналу необхідно подвійно клацнути саме точно на лінії, бо інакше буде створений коментар до моделі.*

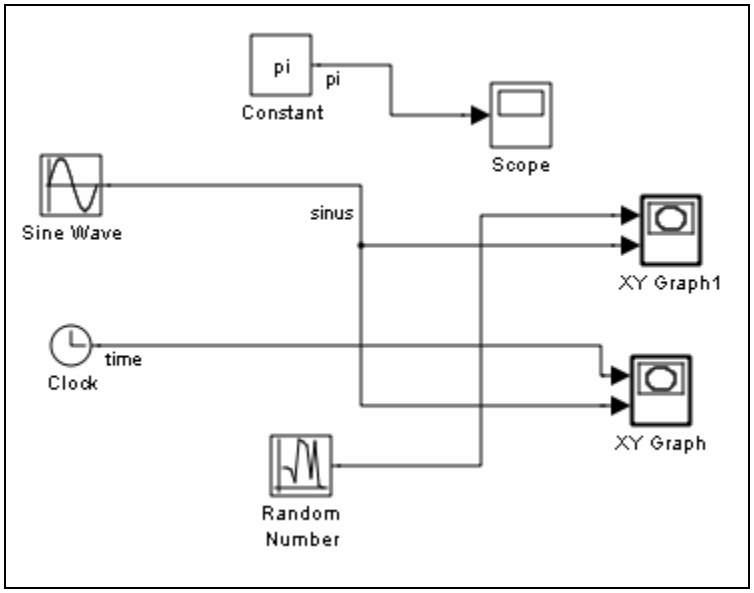


Рис. 3. 69. Створені мітки pi, sinus і time

Для ***пересування мітки*** треба її просто перетягнути на нове місце мишкою.

Щоб ***скопіювати мітку,*** слід натиснути й утримувати клавішу <Ctrl> і перетягнути мітку до нового положення на лінії, або обрати інший сегмент лінії, на якому потрібно встановити копію мітки і подвійно клацнути по цьому сегменту лінії.

Відредагувати мітку можна клацнувши на ній і здійснюючи потім відповідні змінювання як у звичайному текстовому редакторі.

Щоб ***вилучити мітку***, натисніть і утримуйте клавішу <Shift>, виділить мітку і знищить її, використовуючи клавіші <Delete> або <Backspace>. При цьому будуть вилучені усі мітки цієї лінії

Розповсюдження міток лінії

Розповсюдження міток лінії - це процес автоматичного переносу імені мітки до сегментів однієї лінії, що розірвані блоками ***From/Goto***, ***Mux***.

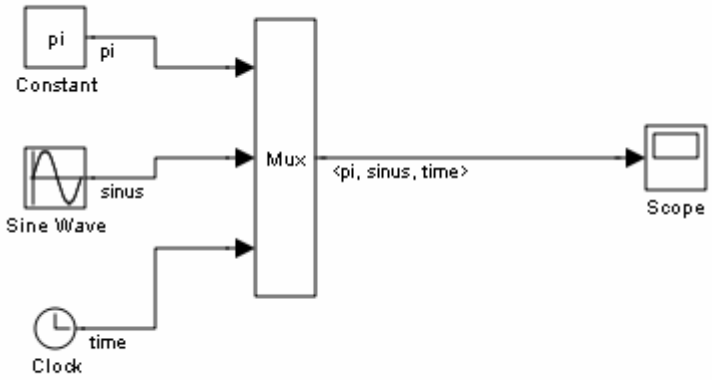


Рис. 3.70. Розповсюдження міток

Щоб ***розповсюдити мітки,*** створіть у другому й наступних сегментах лінії мітки з ім'ям '<'. Після повернення до вікна блок-схеми у цих сегментах автоматично будуть проставлені мітки (див. рис. 3.70)

Створення коментаря й маніпулювання ним

Коментарі дають можливість опоряджувати блок-схеми текстовою інформацією про модель та окремі її складові. Коментарі можна проставляти у будь-якому вільному місці блок-схеми (див. рис. 3.71).

Для ***створення коментаря*** двічі клацніть у будь-якому вільному місці блок-схеми, а потім уведіть коментар у прямокутнику, що виник.

Для ***переміщення коментарю*** у інше місце його потрібно перетягнути на це місце за допомогою миші.

Щоб ***скопіювати коментар,*** достатньо натиснути клавішу <Ctrl> і, утримуючи її у цьому положенні, перетягти коментар у нове місце.

Для ***редагування коментарю*** треба клацнути на ньому, а потім внести потрібні зміни як у звичайному текстовому редакторі. Щоб ***змінити*** при цьому ***шрифт***, його розмір або стиль, слід виділити текст коментарю, який потрібно змінити, а потім обрати команду *Font* із меню *Format* вікна блок-схеми, вибрати у вікні, що виникне, назву шрифту, його розмір, атрибути й стиль і натиснути кнопку <Ok> у цьому вікні.

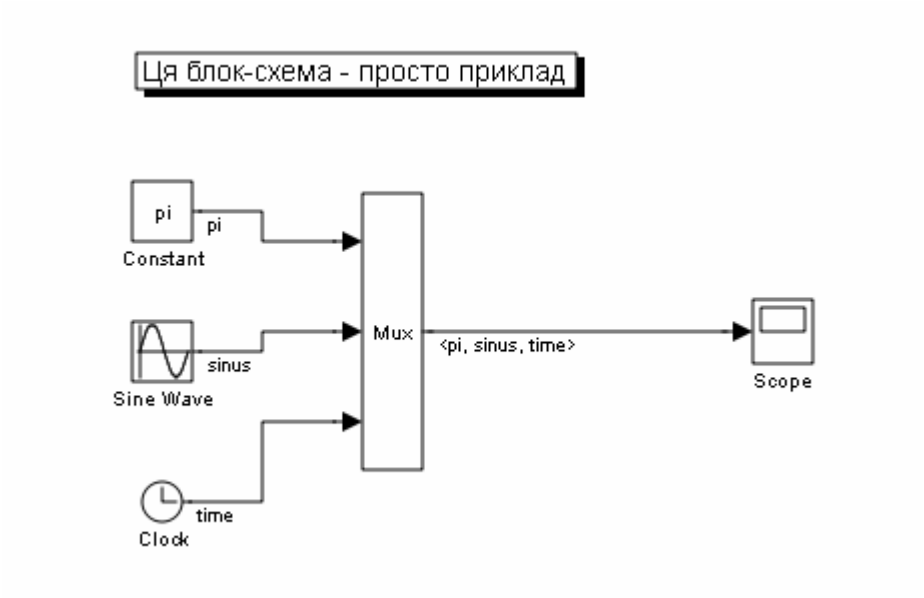


Рис. 3.71. Приклад розміщення коментаря на блок-схемі

Щоб ***вилучити коментар,*** натисніть і утримуйте клавішу <Shift>, виділить коментар і натисніть клавішу <Delete> або <Backspace>.

3.3.5. Створення підсистем

Якщо складність і розміри блок-схеми моделі стають надто великими, її можна суттєво спростити, групуючи блоки у підсистеми. Використання підсистем дає наступні переваги:

- скорочення кількості блоків, що виводяться у вікні моделі;
- об'єднання у єдину групу (підсистему) функціонально пов'язаних блоків;
- можливість створення ієрархічних блок-схем.

Існують три можливості створення підсистем:

- додати блок ***Subsystem*** у модель, потім увійти у цей блок і створити підсистему у вікні підсистеми, яке при цьому виникне;
- виділити частину блок-схеми моделі й об'єднати її у підсистему.

Створення підсистеми через додавання блока ***Subsystem***

У цьому разі слід зробити наступне:

- скопіювати блок ***Subsystem*** у вікно моделі, перетягнувши його з поділу ***Ports&Subsystems***;
- розкрити вікно блока ***Subsystem***, подвійно клацнувши на зображенні блока у блок-схемі;
- у порожньому вікні моделі створити підсистему, використовуючи блоки ***In*** і ***Out*** для створення входів і виходів підсистеми.

Створення підсистеми через групування існуючих блоків

Якщо блок-схема вже містить блоки, які потрібно об'єднати у підсистему, то останню можна зробити у такий спосіб:

- виділити рамкою блоки і з'єднувальні лінії, які потрібно включити у склад підсистеми (див. рис. 3.72);
- обрати команду *Create Subsystem* із меню *Edit* вікна моделі; при цьому SimuLINK замінить виділені блоки одним блоком ***Subsystem*** (див. рис. 3.73)

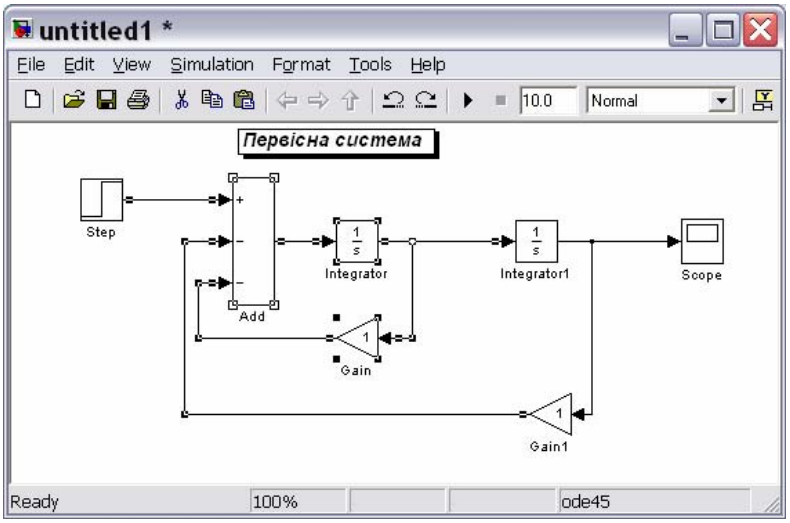


Рис. 3.72. Первісна система з виділеними блоками

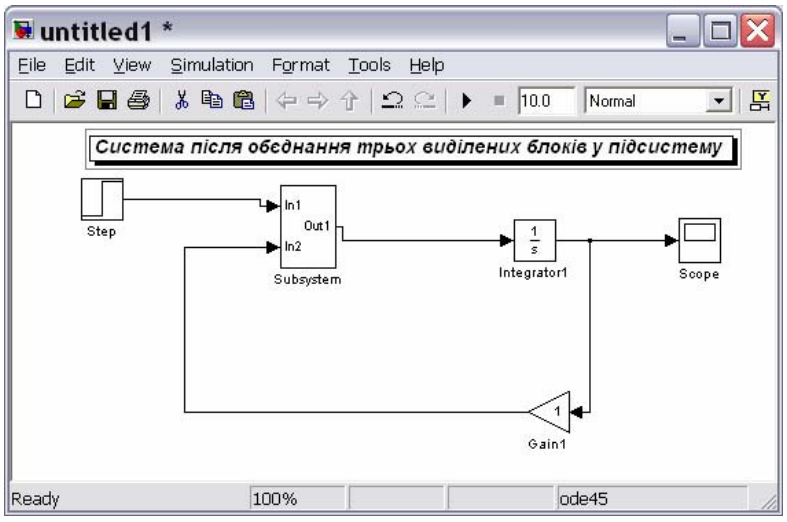


Рис. 3.73. Система після об'єднання виділених блоків у підсистем

Якщо розкрити вікно блоку ***Subsystem***, подвійно клацнувши на ньому, то SimuLINK відобразить блок-схему створеної підсистеми (рис. 3.74). Як видно, SimuLINK додав блоки ***In*** і ***Out*** для відображування входів і виходів у систему вищого рівня.

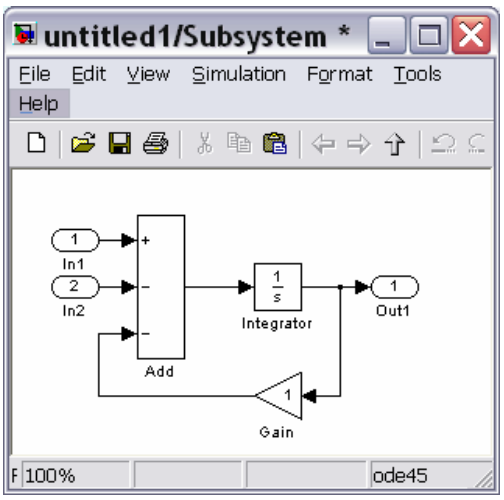


Рис. 3.74. Утворена підсистема

3.3.6. Зберігання і виведення до друку блок-схеми

Для запису моделі (блок-схеми) на диск потрібно викликати команду *Save* (Зберегти) або *Save As* (Зберегти як) з меню *File* (Файл) вікна моделі. Simulink записує у вказану директорію файл з заданим (введеним з клавіатури) ім'ям, присвоюючи йому розширення **.mdl**. При цьому у цей файл автоматично записуються усі вложені підсистеми моделі.

Щоб роздрукувати блок-схему на прінтері, потрібно скористуватися командою File > Print (Файл > Друк) вікна моделі.

Доволі цікавою і важливою є можливість вставлення блок-схеми у документ Word. Для цього слід використати команду Copy model to clipboard (Копіювати модель у буфер) меню Edit (Правка) вікна моделі, яка поміщає у буфер обміну вміст вікна моделі. Якщо після цього перейти у вікно текстового редактора і натиснути клавиші Ctrl+V, у відкритому документі редактора виникне зображення блок-схеми моделі. Саме у такий спосіб отримані рис. 3.68-3.71.

3.4. Приклади утворення S-моделей

При утворенні власних S- моделей користувач має вирішити ряд проблем, що виникають у процесі побудови блок-схеми заданої математичної моделі

3.4.1. Модель поводження фізичного маятника

Опишемо процес створення S-моделі на прикладі задачи моделювання поводження фізичного маятника при гармонічній вібрації точки його підвісу.

Користуючись результатами раніше проведених перетворень (див. п. 2.6.2 глави 2), вихідне рівняння руху маятника подамо у такій безрозмірній формі:

$$\varphi'' + \sin\varphi = S(\tau, \varphi, \varphi'), \tag{3.1}$$

де функція $S(\tau,\varphi,\varphi')$ має наступний вид

$$\begin{aligned} S(\tau,\varphi,\varphi') &= -2\cdot\zeta\cdot\varphi' - \\ &-[n_{mx}\cdot\sin(\nu\cdot\tau+\varepsilon_x)\cdot\cos\varphi + n_{my}\cdot\sin(\nu\cdot\tau+\varepsilon_y)\cdot\sin\varphi], \end{aligned} \tag{3.2}$$

а безрозмірні величини ζ і ν визначаються виразами

$$\zeta = \frac{R}{2\cdot\sqrt{mgl\cdot J}}; \qquad \nu = \frac{\omega}{\omega_0}; \qquad \left(\omega_0 = \sqrt{\frac{mgl}{J}}\right).$$

Вхідними (тими, що задаються) параметрами для моделювання вважатимемо:

1) параметри самого маятника; до них у розглядуваному випадку відноситься лише відносний коефіцієнт загасання ζ;
2) параметри, що характеризують зовнішню дію; сюди входять: амплітуди віброперевантажень точки підвісу маятника у вертикальному n_{ym} ш горизонтальному n_{xm} напрямках; відносна (по відношенню до частоти власних малих коливань маятника) частота вібрації основи ν; початкові фази ε_x і ε_y вібрації основи;
3) початкові умови руху маятника, тобто початкове відхилення φ_0 маятника від вертикалі і його безрозмірну кутову швидкість $\varphi'_0 = \dot{\varphi}_0 / \omega_0$.

До вихідних (модельованих) величин віднесемо поточний кут відхилення маятника від вертикалі $\varphi(\tau)$ і його безрозмірну кутову швидкість $\varphi'(\tau)$.

У подальшому застосовуватимемо наступні позначення первісних і шуканих величин у робочому просторі Matlab:

fi0 –початкове значення φ_0 кута відхилення маятника від вертикалі;

fit0 –початкова безрозмірна кутова швидкість φ'_0 маятника;

dz - відносний коефіцієнт загасання ζ;

nmx – амплітуда n_{xm} віброперевантаження у горизонтальному напрямку;

nmy – амплітуда n_{ym} віброперевантаження у вертикальному напрямку;

nu - відносна частота ν вібрації основи;

ex - початкова фаза ε_x віброприскорення у горизонтальному напрямку;

ey - початкова фаза ε_y віброприскорення у вертикальному напрямку.

Відмітимо одну вельму важливу особливість взаємодії робочого простору Matlab і середовища Simulink.

УВАГА. *Увесь робочий простір системи Matlab є досяжним для виконуваних S-моделей середовища Simulink. Тому при запису значень параметрів настроювання S-блоків можна замість числових значень вводити імена (ідентифікатори) змінних, існуючих на цей час у робочому просторі Matlab, і навіть вирази, записані М-мовою.*

Це значно полегшує складання блок-схем, оскільки ви можете у робочому просторі формувати масив змінних, а потім при складанні блок-схем виражати через ці змінні значення параметрів настроювання блоків.

Перш ніж приступитися до складання S-моделі, запишемо рівняння (3.1) в іншій формі:

$$\varphi'' = S(\tau, \varphi, \varphi') - \sin\varphi. \qquad (3.3)$$

При побудові блок-схеми, що відповідає цьому рівнянню, розмірковуватимемо у такий спосіб.

Припустимо, що процеси $\varphi(\tau)$ і $\varphi'(\tau)$ є відомими. Тоді, здійснюючи операції з ними у відповідності до правої частини рівняння (3.3), можна віднайти й кутове прискорення $\varphi''(\tau)$. Якщо потім проінтегрувати прискорення, можна отримати кутову швидкість $\varphi'(\tau)$. Нарешті, проінтегрувавши і її, можна одержати кут $\varphi(\tau)$ як функцію часу. Дві останні величини (процеси) можна тепер використовувати для формування правої частини рівняння (3.3).

Тому при формуванні бло-схеми, що здійснює чисельне інтегрування диференційного рівняння (3.3), вчинимо у такий спосіб. В основу блок-схеми покладемо два послідовно з'єднаних інтегратори (блоки ***Integrator***). нав хід першого інтегратора подамо кутове прискорення, а у якості початкової умови використаємо початкове значення кутової швидкості φ'_0. Виходом першого інтегратора буде поточна кутова швидкість $\varphi'(\tau)$. Цю величину слід подати на вхід другого інтегратора з початковою умовою у виді початкового значення кута φ_0. Виходом другого інтегратора буде шуканий процес $\varphi(\tau)$.

Оформимо описану частину блок-схеми у виді підсистеми. для цього у поронє вікно блок-схеми (яку назвемо FM) перетягнемо з поділу *Ports&Subsystems* блок ***Subsystem*** і двічі клацнемо на ньому. У вікні, що виникне, складемо блок-схему підсистеми (рис. 3.75) й назвемо її ***Pendulum***.

При настроюванні першого блоку ***Integrator*** у якості значення його параметра *Initial condition* вкажемо fit0. У другому інтеграторі цьому параметру присвоїмо значення fi0.

Вихідними величинами створеної підсистеми є поточний кут $\varphi(\tau)$ відхилення маятника від вертикалі і його поточна безрозмірна кутова швидкість $\varphi'(\tau)$.

Використовуючи їх, можна тепер аналогічно сформувати підсистему, яка обчислює (у відповідності до виразу (3.2)) значення безрозмірного "моменту" $S(\tau,\varphi,\varphi')$ зовнішніх сил, що діють на маятник при вібрації точки його підвісу.

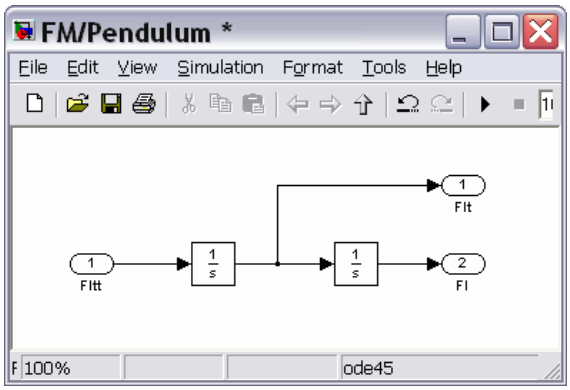


Рис. 3.75. Підсистема Pendulum

Назвемо нову підсистему ***MomentsFM*** (рис. 3.76).

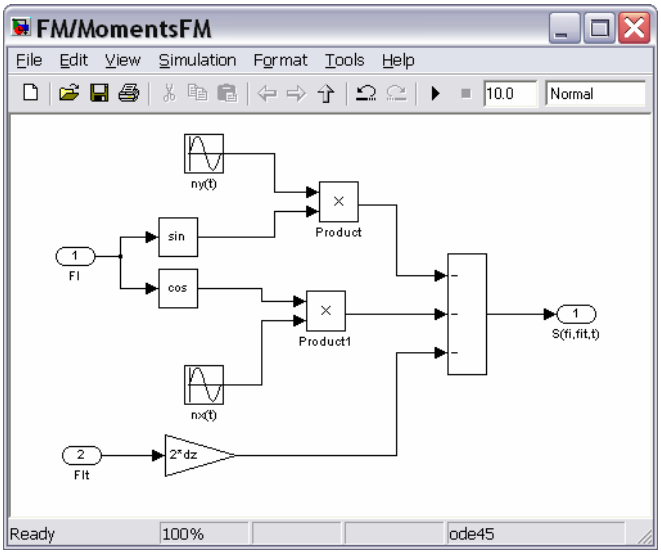


Рис. 3.76. Підсистема MomentsFM

Вона формує поточні значення виразу $S(\tau,\varphi,\varphi')$ по поточних значеннях кута й кутової швидкості маятника, які подаються на її вхід. Її вихідний порт видає величину $S(\tau,\varphi,\varphi')$ моменту сил, що діють на маятник.

Можна помітити, що величина dz (відносний коефіцієнт загасання ζ) використаний у блоці ***Gain***, що розташований внизу блок-схеми. Окрім того, при формуванні перевантажень вдовж горизонтальної і вертикальної осей (блоки ***nx(t)*** і ***ny(t)***) у параметрах настроювання використані змінні nmx, nmy, nu, ex і ey (рис 3.77).

Тепер, використовуючи дві створені підсистеми можна скласти головну блок-схему, яка реалізує моделювання шляхом чисельного інтегрування диференційного рівняння (3.3). Її зображено на рис. 3.78.

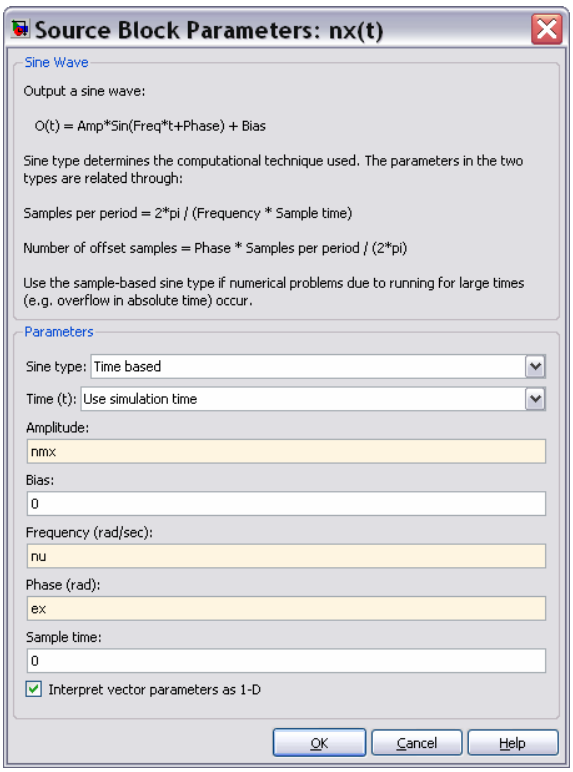


Рис. 3.77. Вікно настроювання блоку nx(t)

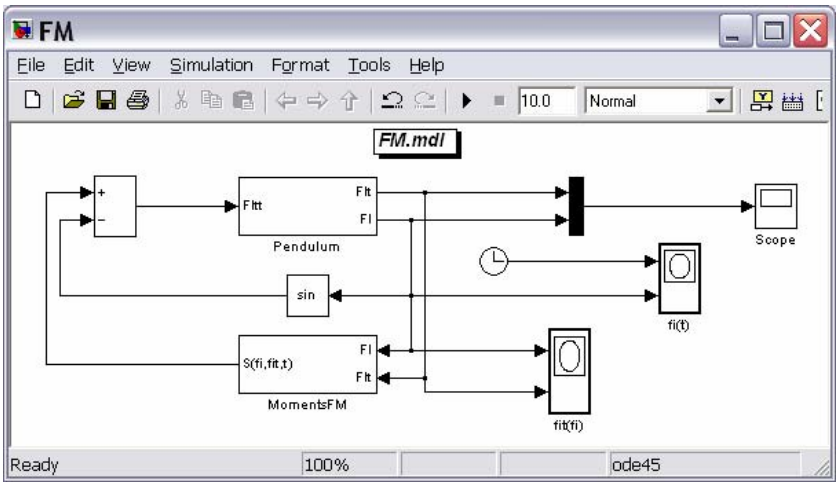


Рис. 3.78. Головна блок-схема FM

Але для здійснення чисельного інтегрування попередньо потрібно встановити програму розв'язувача, яка реалізовуватиме той чи інший метод чисельного інтегрування.

Для цього перед початком моделювання викличемо з вікна блок-схеми командою *Simulation > Configuration Parameters* вікно з тією самою назвою і встановимо в ньому вкладинку *Solver*. Встановимо у полі Stop time (Час завершення) значення 2*pi*20, що відповідає інтервалу безрозмірного часу у двадцять повних періодів власних малих коливань маятника. У полі Type задамо тип розв'язувача з фіксованим кроком інтегрування (Fixed-step), і у полі Fixed-step size (Розмір фіксованого кроку) – значення кроку 0,01. У полі *Solver* оберемо конкретний метод інтегрування – ode4 (Runge-Kutta). У цілому це вікно набуде виду, поданому на рис. 3.79.

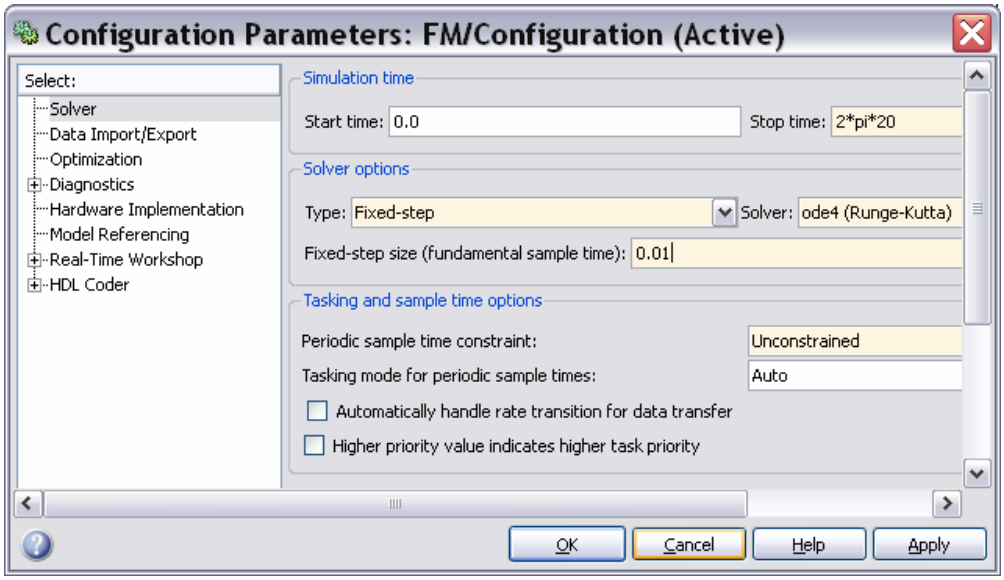


Рис. 3.79. Вкладинка Solver вікна Configuration Parameters

Тепер практично усе готово до моделювання. Залишилося лише забезпечити присвоювання значень усім змінним, які були використані упараметрах настроювання блоків. Для цього складемо невеличку програму FM_dat.m:

```
% FM_dat
% Програма присвоювання значень даним, що використовуються
% у S-моделі FM.mdl

clear all, clc
dz = 0.1;   fi0 = 1*pi/180;   fit0 = 0;
nmy =1; nmx = 0; ex = 0;  ey = 0; nu = 2.3;
```

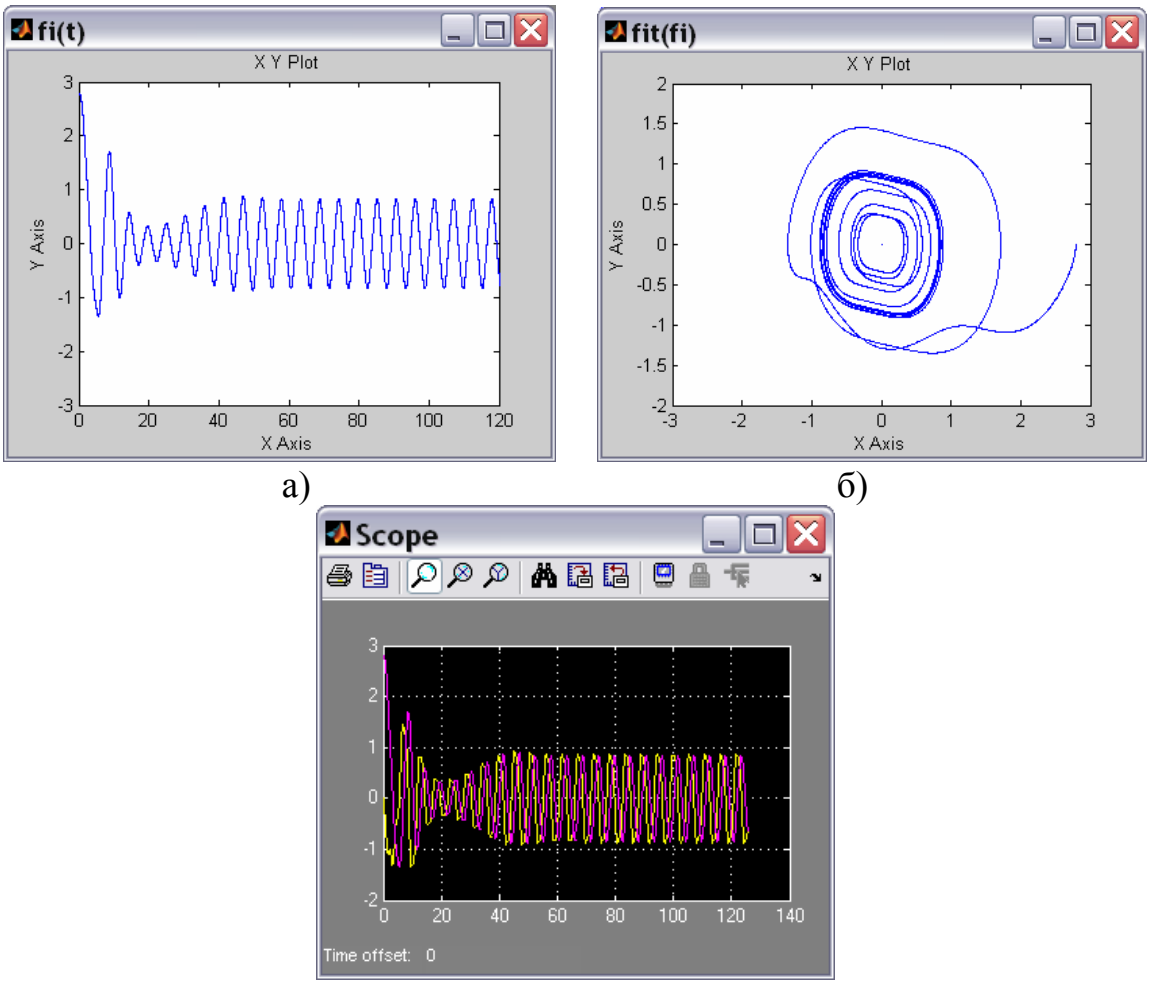
Дані, що введені у файлі FM_dat, відповідають вертикальній вібрації основи з частотою, яка у 2,3 більша за частоту власних малих коливань маятника, амплітудою 1g по прискоренню і при початковому відхиленні маятника від вертикалі 160^{o}.

Викливавши програму FM_dat з командного вікна Matlab, а потім й S-модель ***FM*** на моделювання з вікна її блок-схеми, отримаємо у наглядових вікнах моделі результати, подані на рис. 3.80.

У вікні блоку ***fi(t)*** (див. рис. 3.80а) виведений графік залежності кута повороту маятника від часу. У вікні блоку ***fit(fi)*** (див. рис. 3.80б) зображений фазовий портрет маятника при обраних параметрах маятника і збурень, а у вікні блоку ***Scope*** (див. рис. 3.80в) подані графіки залежності кута й кутової швидкості від часу.

Результати повністю відповідають одержаним раніше (п. 2.7.1, рис. 2.21) і ілюструють виникнення параметричних незгасаючих коливань маятника при вертикальній вібрації точки його підвісу.

Змінюючи дані настроювання у файлі FM_dat, можна проводити дослідження поводження маятника при різних значеннях вхідних параметрів.



а) б)

в)

Рис. 3.80. Результати моделювання у різних оглядових вікнах моделі FM: а) у вікні блоку fi(t); б) у вікні блоку fst(fi); в) у вікні блоку Scope

З цього прикладу стають наочними переваги й недоліки моделювання динамічних систем за допомогою пакету Simulink у порівнянні з аналогічними дослідженнями з використанням лише програм Matlab.

До переваг застосування Simulink відносяться такі:

- складання блок-схеми часом є набагато більш простим і наочним процесом, аніж написання програм, що реалізують ті самі операції; внаслідок цього виникнення помилок є менш вірогідним; це посилюється наявністю добре відпрацьованих і надійних блоків-програм, які реалізують типові, часом доволі складні, математичні операції і процедури, які часто зустрічаються при практичному моделюванні і відповідають реальним особливостям поводження динамічних систем;

- блок-схема рівнянь руху є значно більш наочною, аніж код процедури обчислення правих частин диференційних рівнянь і не потребує приведення рівнянь до форми Коші; завдяки цьому стає більш зрозумілим фізичний сенс окремих блоків і їх взаємозв'язків;

- при проведенні моделювання у середовищі Simulink щезає потреба в організації процесу чисельного інтегрування диференціальних рівнянь і

виведення результатів у графічній формі, - цей процес у Simulink автоматизований;

- особливо корисним є наявність значної кількості розв'язувачів – програм різних методів чисельного інтегрування, - і можливість довільного їх обрання.

Недоліки Simulink пов'язані, головним чином, з недостатніми можливостями і гнучкістю виведення результатів у графічній формі:

- неможливо додати власні написи у заголовок графіків у наглядових вікнах і по осях графіків;
- неможливо встановити сітку координатних ліній у вікні блоку ***XYGraph***;
- незручністю застосування оглядового вікна ***XYGraph*** є необхідність попереднього встановлення діапазонів змінювання обох вхідних величин по осях графіка; якщо ці діапазони встановлені невірно, в оглчдовому вікні може взагалі не виникнути зображення графіка, або виникне такий його фрагмент, за яким неможливо зробити правильний висновок щодо поводження системи; а при дослідженні системи часто неможливо заздалегідь передбачити діапазони змінювання величин;
- відсутні засоби виведення текстової інформації на поле графіка – це робить графічне подання безадресним.

Ці недоліки є суттєвими. Вони можуть бути подолані поєднанням можливостей Simulink і Matlab. Наприклад, можна записати отримані значення вихідних величин у MAT-файл (надсилаючи їх на блок ***To File***), потім скласти і використати програму, яка здійснювала би зчитування даних, записаних у MAT-файлі, і формування на цій основі графічного зображення у вікні фігури на зразок того, як це зроблено у п. 2.4. Такий спосіб моделювання реалізований у наступному прикладі.

3.4.2. Моделювання руху трьох тіл під дією сил гравітації

Поглянемо на класичну задачу небесної механіки – визначення руху трьох матеріальних тіл під дією сил гравітації – з точки зору чисельного моделювання цього руху у середовищі Simulink.

Розглянемо три ізольовані матеріальні точки з масами відповідно m_1, m_2 і m_3. Позначимо радіус-вектори цих точок відносно деякої нерухомої в інерціальному просторі точкі O через $\mathbf{R}_1$, $\mathbf{R}_2$ і $\mathbf{R}_3$. Диференційні рівняння руху цих трьох точок можуть бути записані у наступному виді:

$$\begin{cases} m_1 \dfrac{d^2\mathbf{R}_1}{dt^2} = \gamma \dfrac{m_1 m_2}{R_{21}^3}\mathbf{R}_{21} - \gamma \dfrac{m_1 m_3}{R_{31}^3}\mathbf{R}_{13} \\ m_2 \dfrac{d^2\mathbf{R}_2}{dt^2} = -\gamma \dfrac{m_1 m_2}{R_{21}^3}\mathbf{R}_{21} + \gamma \dfrac{m_2 m_3}{R_{32}^3}\mathbf{R}_{32} \\ m_3 \dfrac{d^2\mathbf{R}_3}{dt^2} = -\gamma \dfrac{m_3 m_2}{R_{32}^3}\mathbf{R}_{32} + \gamma \dfrac{m_1 m_3}{R_{31}^3}\mathbf{R}_{13} \end{cases} \qquad (3.4)$$

де позначено $\mathbf{R}_{21} = \mathbf{R}_2 - \mathbf{R}_1$; $\mathbf{R}_{32} = \mathbf{R}_3 - \mathbf{R}_2$; $\mathbf{R}_{13} = \mathbf{R}_1 - \mathbf{R}_3$.

Дослідження рівнянь руху (і особливо чисельне) завжди зручніше здійснювати за рівняннями, приведеними до безрозмірної форми, - у цьому випадку скорочується кількість параметрів, від яких залежить розв'язок.

Приведемо рівняння (3.4) до безрозмірної форми. Для цього у якості базових параметрів використаємо такі фізичні величини:

- γ - гравітаційна стала, що має розмірність $L^3M^{-1}T^{-2}$ (L - одиниця довжини, M - одиниця маси, T - одиниця часу);

- m_1 - маса першого тіла, яке вважатимемо основним; ним зазвичай є найбільш масивне тіло (наприклад, Сонце, яке знаходиться під дією гравітаційного тяжіння Землі (друге, середнє за масою, тіло) і Місяця (третє, найменш масивне тіло)); розмірність - M;

- R_{210} - початкове значення відстані між першим і другим тілами, розмірність - L.

Тепер введемо безрозмірні величини:

1) безрозмірна маса першого тіла:

$$\mu_1 = \frac{m_1}{m_1} = 1;$$

2) безрозмірна маса другого тіла:

$$\mu_2 = \frac{m_2}{m_1}; \qquad (3.5)$$

3) безрозмірна маса третього тіла:

$$\mu_3 = \frac{m_3}{m_1}; \qquad (3.6)$$

4) безрозмірні довжини радіус-векторів:

$$\rho_1 = \frac{R_1}{R_{210}}; \quad \rho_2 = \frac{R_2}{R_{210}}; \quad \rho_3 = \frac{R_3}{R_{210}}; \qquad (3.7)$$

5) безрозмірні радіус-вектори

$$\mathbf{r}_i = \frac{\mathbf{R}_i}{R_{210}}; \qquad \mathbf{r}_{ij} = \frac{\mathbf{R}_{ij}}{R_{210}}; \qquad (3.8)$$

6) безрозмірний час

$$\tau = t\sqrt{\frac{\gamma\, m_1}{R_{210}^3}}. \qquad (3.9)$$

Остання формула означає, що у якості одиниці вимірювання часу використовується величина

$$T_0 = 2\pi\sqrt{\frac{R_{210}^3}{\gamma\, m_1}}\,. \tag{3.10}$$

Для системи Сонце-Земля-Місяць вона дорівнює рокові.

Прийнятий безрозмірний час є таким, що безрозмірний період кругового обертання другого тіла навколо першого дорівнює 2π.

З врахуванням цього рівняння (3.4) у беррозмірній формі набудуть такого виду:

$$\begin{cases} \dfrac{d^2\mathbf{r}_1}{dt^2} = \dfrac{\mu_2}{r_{21}^3}\mathbf{r}_{21} - \dfrac{\mu_3}{r_{31}^3}\mathbf{r}_{13} \\ \dfrac{d^2\mathbf{r}_2}{dt^2} = -\dfrac{1}{r_{21}^3}\mathbf{r}_{21} + \dfrac{\mu_3}{r_{32}^3}\mathbf{r}_{32} \\ \dfrac{d^2\mathbf{r}_3}{dt^2} = -\dfrac{\mu_2}{r_{32}^3}\mathbf{r}_{32} + \dfrac{1}{r_{31}^3}\mathbf{r}_{13} \end{cases} \tag{3.11}$$

Три вектори $\mathbf{r}_{21}, \mathbf{r}_{32}$ і $\mathbf{r}_{13}$ пов'язані між собою наступним співвідношенням

$$\mathbf{r}_{21} + \mathbf{r}_{32} + \mathbf{r}_{13} = 0\,. \tag{3.12}$$

Тому з трьох рівнянь (3.11) незалежними є лише двое. Щоб їх одержати, віднімемо з другого рівняння перше, а з третього рівняння – друге і вилучимо з отриманих рівнянь вектор $\mathbf{r}_{13}$, користуючись (3.12). Одержимо шукані два рівняння

$$\begin{cases} \dfrac{d^2\mathbf{r}_{21}}{dt^2} = -\left(\dfrac{1+\mu_2}{\rho_{21}^3} + \dfrac{\mu_3}{\rho_{31}^3}\right)\mathbf{r}_{21} + \mu_3\left(\dfrac{1}{\rho_{32}^3} - \dfrac{1}{\rho_{31}^3}\right)\mathbf{r}_{32} \\ \dfrac{d^2\mathbf{r}_{32}}{dt^2} = -\left(\dfrac{\mu_3+\mu_2}{\rho_{32}^3} + \dfrac{1}{\rho_{31}^3}\right)\mathbf{r}_{32} + \left(\dfrac{1}{\rho_{21}^3} - \dfrac{1}{\rho_{31}^3}\right)\mathbf{r}_{21} \end{cases} \tag{3.13}$$

Тут використані позначення:

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j\,; \qquad \rho_{ij} = \left|\mathbf{r}_{ij}\right|. \tag{3.14}$$

До рівнянь (3.13) слід додати співвідношення, що випливає з (3.12):

$$\rho_{13} = \left|\mathbf{r}_{21} + \mathbf{r}_{32}\right|. \tag{3.15}$$

Рівняння (3.13) у сукупності з (3.15) покладемо в основу майбутньої моделі. При чисельному інтегруванні слід взяти до уваги, що $\rho_{21}(0) = 1$.

Використовуватимемо наступні позначення у робочому просторі:

- mu2 – відношення мас другого і першого тіла (μ_2);
- mu3 – відношення мас третього і першого тіла (μ_3);
- X210, Y210, Z210 – початкові безрозмірні координати другого тіла відносно першого ($x_{21}(0)$, $y_{21}(0)$, $z_{21}(0)$);
- X320, Y320, Z320 – початкові безрозмірні координати третього тіла відносно другого ($x_{21}(0)$, $y_{21}(0)$, $z_{21}(0)$);

- V21x, V21y, V21z – початкові безрозмірні проекції вектора швидкості другого тіла відносно першого ($V_{21x}(0)$, $V_{21y}(0)$, $V_{21z}(0)$);
- V32x, V32y, V32z – початкові безрозмірні проекції вектора швидкості третього тіла відносно другого ($V_{32x}(0)$, $V_{32y}(0)$, $V_{32z}(0)$).

Варіант блок-схеми S-моделі під назвою GR_3tel.mdl наведений на рис. 3.81:

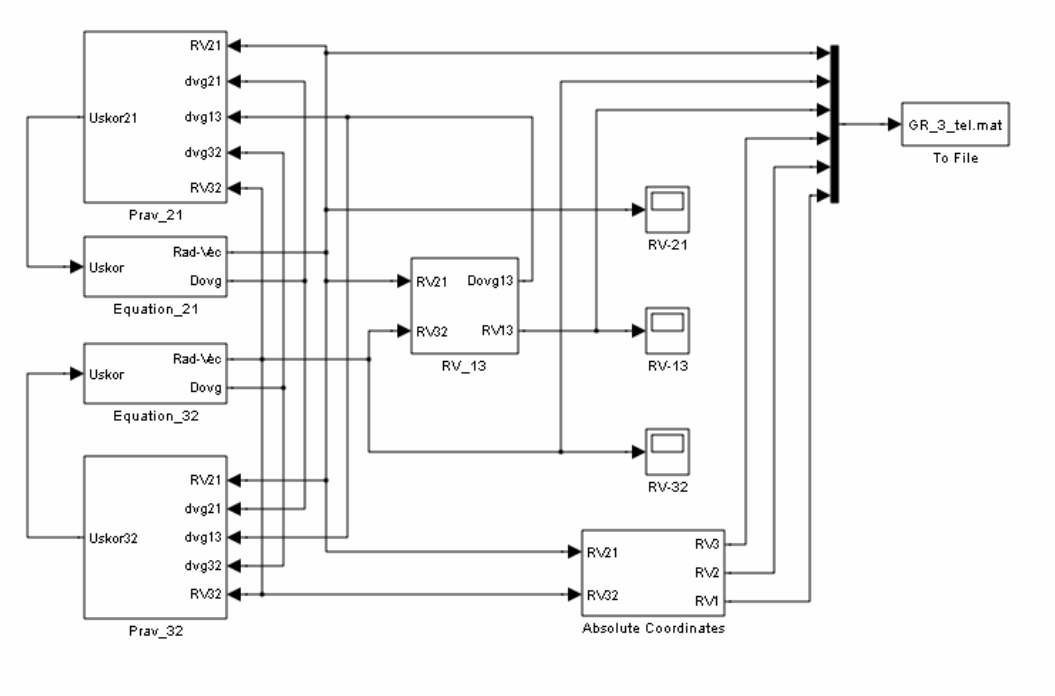


Рис. 3.81. Блок-схема S-моделі GR_3tel.mdl

Вона включає у себе шість підсистем – ***Equation_21***, ***Equation_32***, ***Prav_21***, ***Prav_32***, ***RV_13***, ***Absolute Coordinates*** – і блок ***To File***. Останній забезпечує запис результатів моделювання у MAT-файл, де вони зберігаються у матриці за ім'ям RV (рис. 3.82).

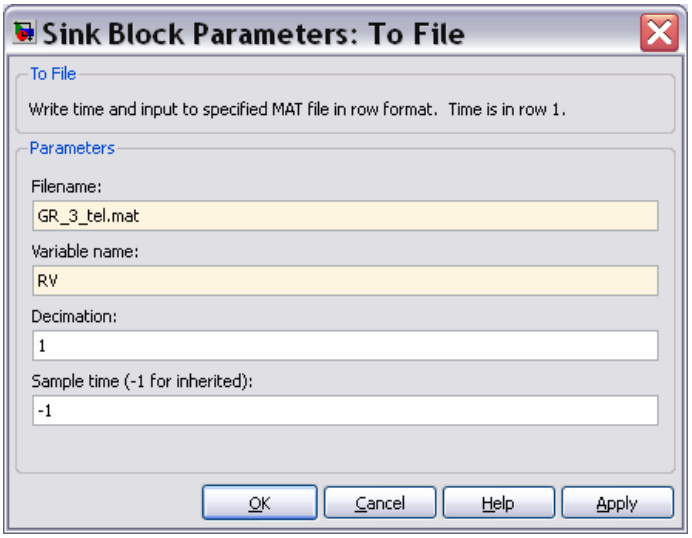


Рис. 3.82. Вікно настроювання блоку To File

Основні дії з чисельного інтегрування диференційних рівнянь (3.13) зосереджені у підсистемах ***Equation_21*** і ***Equation_32***.

Тут (рис. 3.83) здійснюється подвійне інтегрування прискорень тіл, а також обчислюється поточна відстань між тілами.

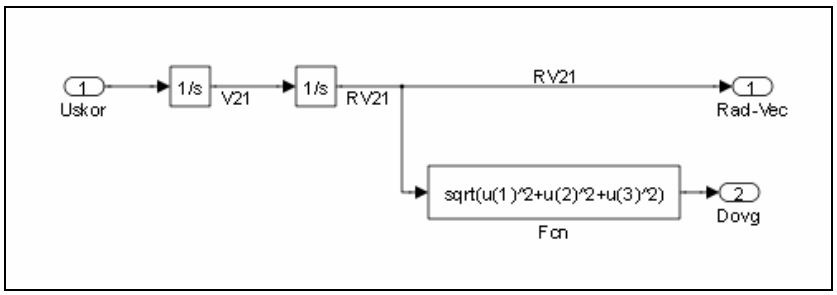


Рис. 3.83. Блок-схема підсистеми Equation_21

Вхідною величиною обох підсистем є векторна величина, елементи якої являють собою проекції вектора прискорення на осі X, Y і Z. Вихідних величин дві. Одна з них – вектор з трьох елементів, що є поточними проекціями радіуса-вектора на ті самі осі. Другий вихід – скалярна величина, довжина цього радіуса-вектора. Блок, який здійснює обчислення довжини реалізований на основі блоку ***Fcn*** з поділу *User-defined Functions* бібліотеки Simulink.

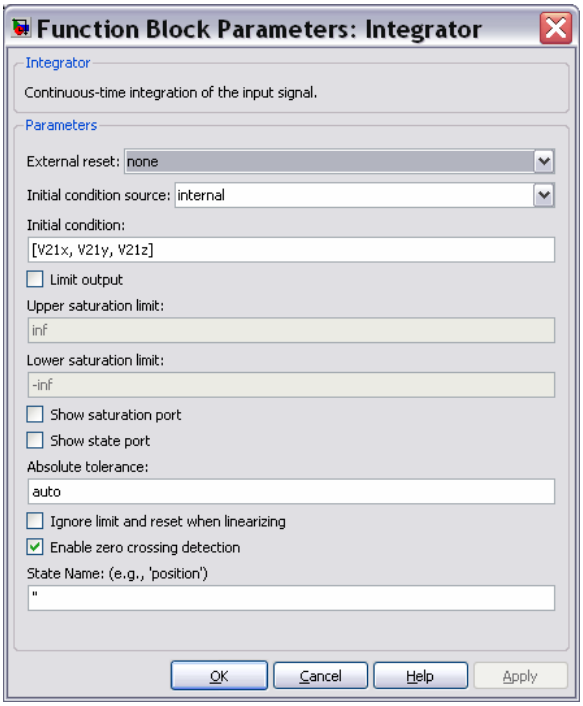


Рис. 3.84. Вікно настроювання першого інтегратора блоку Equation_21

Характерною і важливою особливістю цих підсистем, яка відрізняє їх від використаної раніше підсистеми ***Pendulum*** (див. рис. 3.75) є те, що в них інтегрується не скалярна функція часу, а векторна, яка складається одразу з трьох скалярних величин. Наприклад, у підсистемі ***Equation_21*** здійснюється одночасне подвійне інтегрування прискорень x''_{21}, y''_{21} і z''_{21}, яке досягається за рахунок лише двох блоків типу ***Integrator***. Для цього достатньо у блоці

настроювання інтеграторів ввести у якості початкової умови не одну величину, а вектор з трьох величин, кожна з яких є початковою умовою відповідного елемента вектора, який надається на вхід інтегратора. наприклад, вектором початкових умов для першого інтегратора є [V21x, V21y, V21z] (рис. 3.84). Аналогічно, у другому інтеграторі встановлений вектор [X210, Y210, Z210] початкових умов.

Те саме зроблено у підсистемі ***Equation_32*** , яка обчислює поточні складові вектора $\mathbf{r}_{32}$, одержані подвійним інтегруванням прискорення $\mathbf{r}''_{32}$. Тут в інтеграторах початкові умови встановлені у виді векторів [V32x, V32y, V32z] і [X320, Y320, Z320].

Підсистеми ***Prav_21*** і ***Prav_32***, формують праві частини першого і другого рівнянь (3.13), тобто прискорення у виді вектора з трьох елементів. Вихідна величина у цих підсистемах єдина. Наприклад, у підсистемі ***Prav_21*** (рис. 3.85) виходом є вектор Uskor21, складові якого – проекції прискорення $\mathbf{r}''_{21}$ на осі X, Y і Z. Входами є два обчислених раніше вектори RV21 і RV32, що сладені з проекцій радіус векторів $\mathbf{r}_{21}$ і $\mathbf{r}_{32}$, і три скалярні величини, що дорівнюють поточним значенням довжин трьох радіусів-векторів, які з'єднують три тіла.

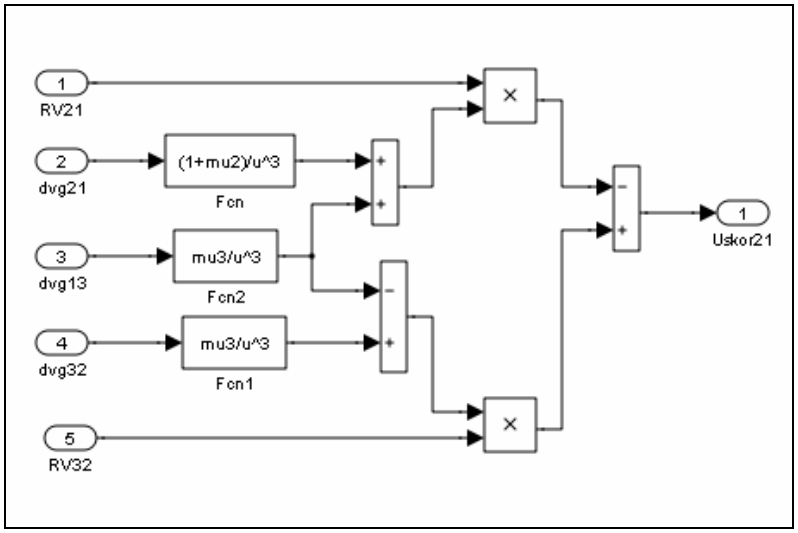


Рис. 3.85. Блок-схема підсистеми Prav_21

У підсистемі ***Prav_21*** використовуються три блоки типу ***Fcn*** для обчислення коефіцієнтів при векторах у правій частині першого рівняння (3.13) і два блоки Product задля перемножування вектора на скалярний коефіцієнт.

Блок-схема підсистеми ***RV_13***, яка обчислює вектор $\mathbf{r}_{13}$ і його довжину за відомими векторами $\mathbf{r}_{21}$ і $\mathbf{r}_{32}$, наведена на рис. 3.86.

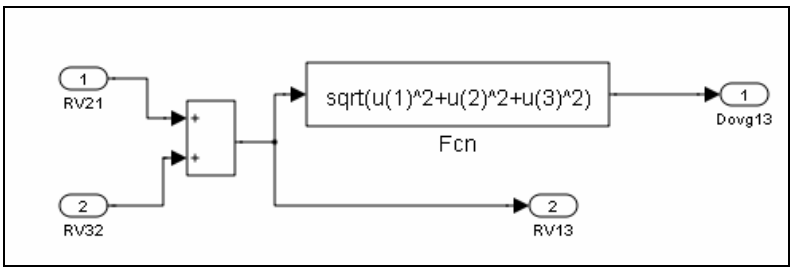


Рис. 3.86. Блок-схема підсистеми RV_13

На рис. 3.87 показана блок схема підсистеми ***Absolute Coordinates***, яка обчислює абсолютні координати першого, другого і третього тіл відносно центру мас системи трьох тіл за формулами:

$$\mathbf{r}_1 = \frac{-(\mu_2+\mu_3)\mathbf{r}_{21} - \mu_3\mathbf{r}_{32}}{1+\mu_2+\mu_3}; \qquad \mathbf{r}_2 = \frac{\mathbf{r}_{21} - \mu_3\mathbf{r}_{32}}{1+\mu_2+\mu_3}; \qquad \mathbf{r}_3 = \frac{\mathbf{r}_{21} + (1+\mu_2)\mathbf{r}_{32}}{1+\mu_2+\mu_3}.$$

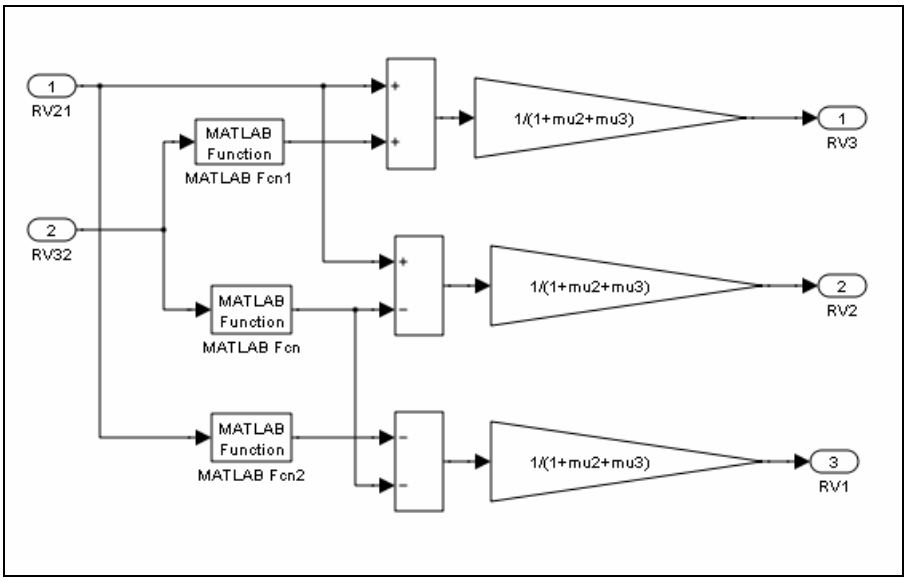


Рис. 3.87. Блок-схема підсистеми Absolute Coordinates

Дані, що обчислюються у блок-схемі подаються у блок ***To File***. Ці дані записуються на диск у MAT-файл GR_3_tel.mat у виді матриці за ім'ям RV (див. рис. 3.82) у наступному порядку:

- перший рядок утворює масив значень модельного часу, у які обчислені величини, що подаються на вхід;
- другий рядок утворюють значення першого елемента вхідного векторного сигналу (йому відповідає верхній сигнал, що поступає на блок ***Mux***., вихідний сигнал з якого поступає на вхід блоку ***To File***), відповідні моментам часу, що записані у першому рядку;
- решта рядків заповнюються значеннями наступних елементів вектора сигнала входу у порядку слідування елементів цього вектора.

Ці особливості варто мати на увазі при зчитуванні даних зі сформованого MAT-файла.

Для того, щоб виконати моделювання по створеній S-моделі і графічно оформити отримані результати, потрібно здійснити наступне.

1. Ввести первісні дані у робочий простір.
2. Запустити S-модель GR_3tel.mdl на моделювання. При цьому результати моделювання будуть записані у MAT-файл GR_3_tel.mat.
3. Завантажити вміст MAT-файла GR_3_tel.mat у робочий простір, ввівши у командному вікні Matlab команду **load GR_3_tel**. При цьому усі дані, що зберігаються у цьому MAT-файлі, будуть записані у робочий простір у виді матриці під ім'ям RV розміром $(1+6\cdot 3)\times n$, де n - розмір вектора значень модельного часу.
4. Використовуючи дані матриці RV, програмно реалізувати виведення результатів у графічне вікно ***figure*** у графічному виді з відповідним текстовим оформленням.

Усі ці операції можна автоматизувати, запрограмувавши їх у окремій M-програмі, яку зручно оформити у виді M-файла *GR_3tel_upr.m* (керуюча програма для моделі *GR_3tel*. При цьому викликати на моделювання S-модель з програми можна за допомогою команди

sim('ім'я S-моделі**')**

Нижче наведений текст керуючої програми.

```
% GR_3TEL_upr
% Програма керування запуском і обробки результатів для S-моделі GR_3TEL

% Лазарев Ю. Ф. 2-08-2009
clear all, clc
        %   1. Введення даних у  робочий простір
mu2=0.1;   mu3=0.01;
X210=1; Y210=0; Z210=0;
V21x=0; V21y=1; V21z=0;
X320=0.1; Y320=0.0; Z320=0;
V32x=0; V32y=0.0; V32z=1.2;
        %   2. Запуск S-моделі
sim('GR_3TEL')
        %   3. Завантажування MAT-файла
load GR_3_TEL;
        %   4. Присвоювання значень з матриці RV MAT-файла новим змінним
t=RV(1,:);
X21=RV(2,:);   Y21=RV(3,:);   Z21=RV(4,:);
X32=RV(5,:);   Y32=RV(6,:);   Z32=RV(7,:);
X13=RV(8,:);   Y13=RV(9,:);   Z13=RV(10,:);
X3=RV(11,:);   Y3=RV(12,:);   Z3=RV(13,:);
X2=RV(14,:);   Y2=RV(15,:);   Z2=RV(16,:);
X1=RV(17,:);   Y1=RV(18,:);   Z1=RV(19,:);
n=length(t);
        %   5. Побудова графіка залежності координат второго тіла від часу
subplot(2,2,1)
plot(t,X21,t,Y21,'--',t,Z21,'.'), grid
set(gca, 'FontSize',14)
title('Рух другого тіла відносно першого')
xlabel('Час (безрозмірний)')
ylabel('Координати (безрозмірні)')
legend('X_2_1','Y_2_1','Z_2_1',0)
        %   6. Побудова графіка залежності координат третього тіла від часу
subplot(2,2,3)
n1=round(n);   set(gca, 'FontSize',14)
plot(t(1:n1),X32(1:n1),t(1:n1),Y32(1:n1),'--',t(1:n1),Z32(1:n1),'.'),grid
```

```
title('Рух третього тіла відносно другого')
xlabel('Час (безрозмірний)')
ylabel('Координати (безрозмірні)')
legend('X_3_2','Y_3_2','Z_3_2',0)
        %     7. Побудова просторових траекторий другого і третього тіл
subplot(4,4,[3,4,7,8,11,12])
plot3(X1,Y1,Z1,'o',X2,Y2,Z2,'.-',X3,Y3,Z3), grid
axis('square')
set(gca, 'FontSize',14)
title('Рух двох тіл у системі трьох гравітуючих тіл','FontSize',16)
xlabel('Координата X')
ylabel('Координата Y')
zlabel('Координата Z')
legend('перше тіло','друге тіло','третє тіло',0)
        %     8. Текстове оформления
subplot(4,4,[15,16]), axis('off')
h= text(-0.1,0.8,'Маси тіл (відносні):', 'FontSize',14);
h= text(0.5,0.8,['\mu-1 = 1;   ',sprintf('\\mu_2 = %g;   ',mu2),sprintf('\\mu_3 = %g',mu3)], 'Font-
Size',14);
h= text(-0.1,0.650,'Початкові координати (безрозмірні):', 'FontSize',14);
h= text(-0.1,0.50,['другого тіла відносно першого:   ',sprintf('X_2_1 = %g;   ',...
   X210),sprintf('Y_2_1 = %g;   ',Y210),sprintf('Z_2_1 = %g;   ',Z210)], 'FontSize',14);
h= text(-0.1,0.35,['третього тіла відносно другого:   ',sprintf('X_3_2 = %g;   ',...
   X320),sprintf('Y_3_2 = %g;   ',Y320),sprintf('Z_3_2 = %g;   ',Z320)], 'FontSize',14);
h= text(0,0.20,'Початкові швидкості (безрозмірні):', 'FontSize',14);
h= text(-0.1,0.05,['другого тіла відносно першого:   ',sprintf('V_2_1_x = %g;   ',...
   V21x),sprintf('V_2_1_y = %g;   ',V21y),sprintf('V_2_1_z = %g;   ',V21z)], 'FontSize',14);
h= text(-0.1,-0.1,['третього тіла відносно другого:   ',sprintf('V_3_2_x = %g;   ',...
   V32x),sprintf('V_3_2_y = %g;   ',V32y),sprintf('V_3_2_z = %g;   ',V32z)], 'FontSize',14);
h= text(-0.1,-0.2,' ____________________________________________________________ ');
h= text(-0.1,-0.3,'Програма GR-3TEL-upr        Автор - Лазарєв Ю. Ф.    2-08-2009');
h= text(-0.1,-0.4,' ____________________________________________________________ ');
```

Запустивши цю програму, можна одразу одержати результати моделювання у такому виді, як показано на рис. 3.88-3.90.

Подані результати відповідають трьом випадкам руху:

1) орбітальні рухи другого тіла відносно першого і третього відносно другого здійснюються в одній площині (XY) і в одному напрямку (орбітальні моменти спрямовані однаково – вздовж осі Z);
2) вказані орбітальні рухи здійснюються в одній площині, але у протилежних напрямках (орбітальні моменти спрямовані протилежно);
3) вказані орбітальні рухи здійснюються у перпендикулярних площинах.

В усіх трьох випадках відношення мас першого, другого і третього тіл дорівнює 100:10:1. Відносні початкові швидкості тіл у всіх випадках прийняті однаковими за величиною ($|V_{21}(0)| = 1$ і $|V_{32}(0)| = 1$). Зверніть увагу на відмінність у параметрах орбіти середнього (другого) тіла і періодах його орбітального руху.

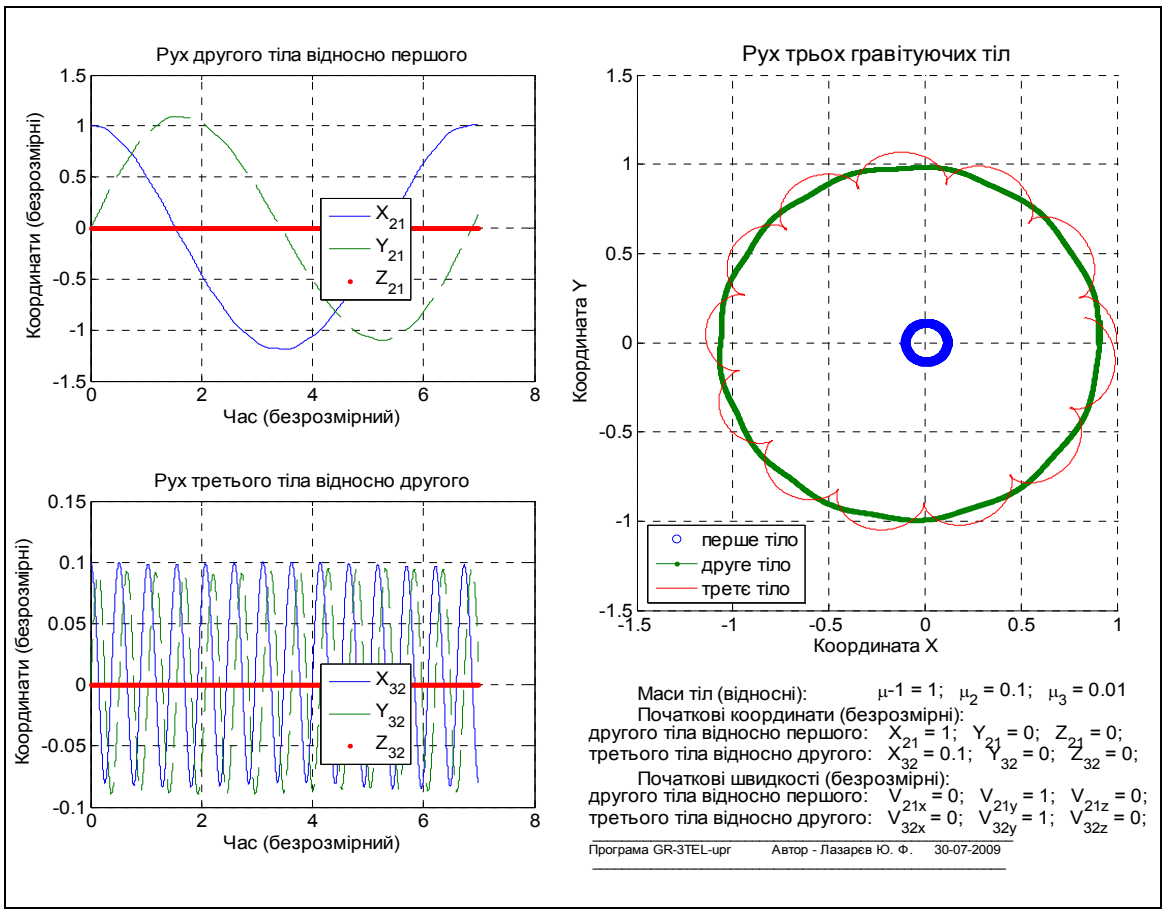


Рис. 3.88. Плаский орбітальний рух, коли тіла обертаються в одному напрямку

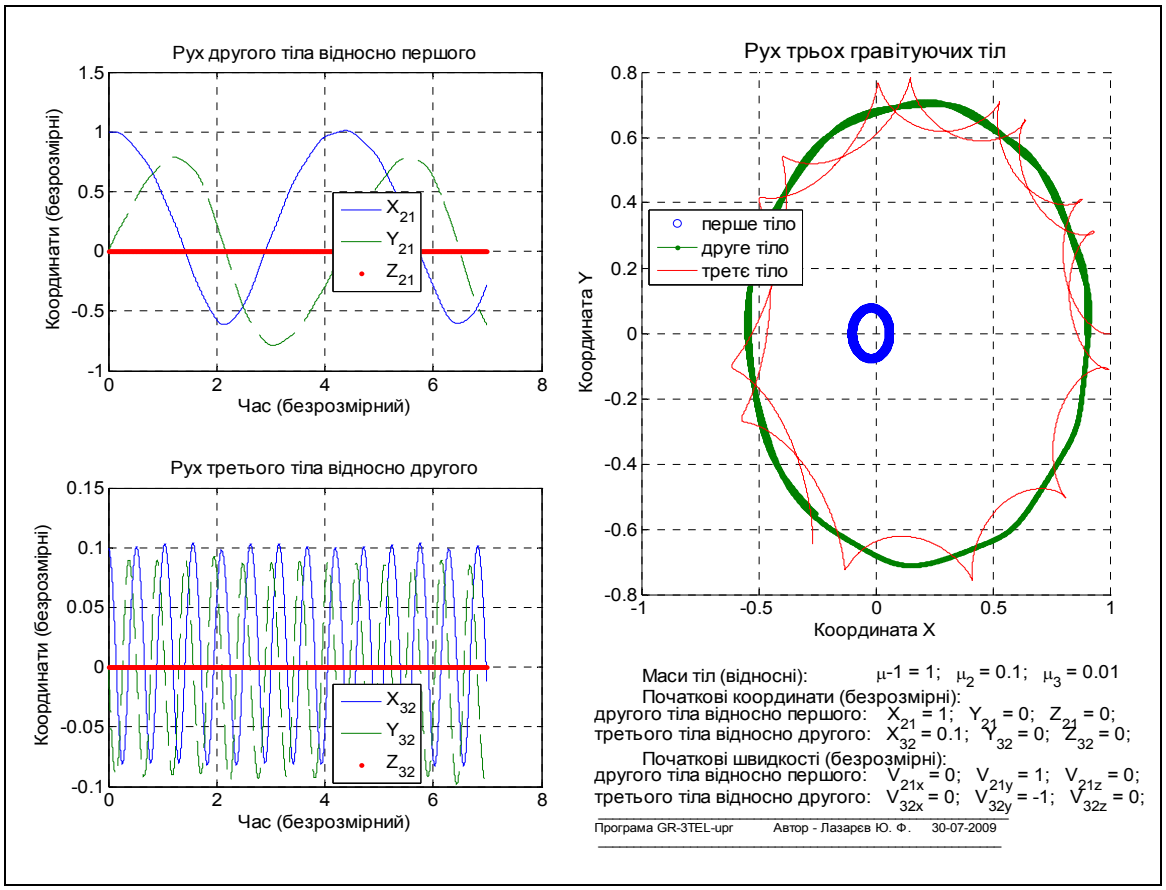


Рис. 3.89. Плаский орбітальний рух, коли тіла обертаються у протилежному напрямку

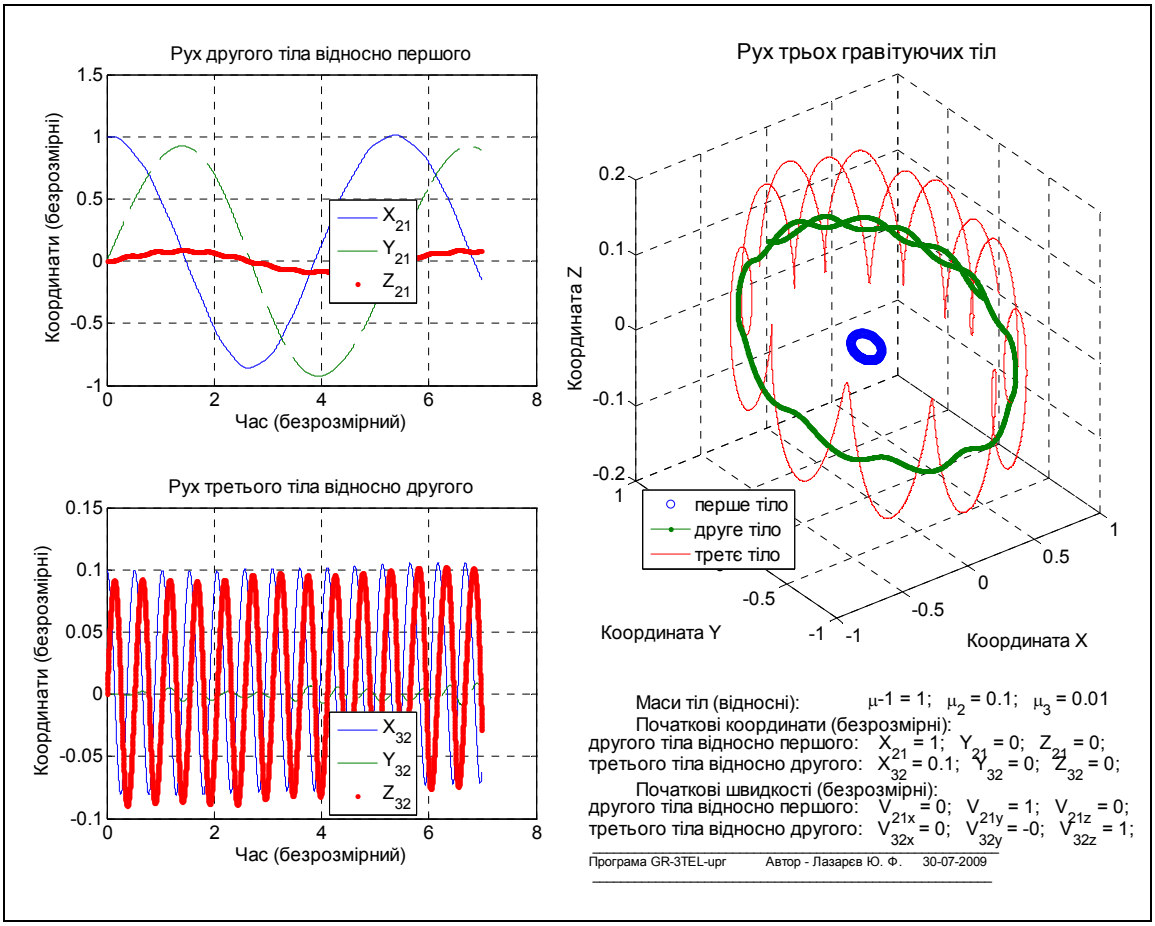


Рис. 3.90. Орбітальний рух, коли площини обертання перпендикулярні

3.5. Контрольні запитання

1. Які переваги має використання пакету Simulink для вирішення обчислювальних задач у порівнянні з застосуванням програмування безпосередньо у середовищі Matlab?

2. Блоки яких поділів бібліотеки Simulink мають обов'язково бути присутними у блок-схемі будь-якої моделі?

3. Яке головне призначення блоків поділу Source бібліотеки Simulink?

4. Яке головне призначення блоків поділу Sinks бібліотеки Simulink?

5. Блоки якого поділу бібліотеки Simulink забезпечують користувачеві можливість створювати власні блоки?

6. Що таке підсистема і як її утворити?

7. У чому полягають переваги у використанні підсистем?

3.6. Література

1. Лазарєв Ю. Ф. Початки програмування у середовищі MatLab. Навч. посібник. – К.: Корнійчук, 1999. – 160 с.
2. Лазарев Ю. Ф. MatLAB 5.x. – К.: Издат. группа BHV, 2000. - 384 с.
3. Лазарев Ю. Ф. Моделирование процессов и систем в MATLAB. Учебный курс. – СПб.: Питер; Киев: Издат.группа BHV, 2005. – 512 с.
4. Лазарєв Ю. Ф. Моделювання на ЕОМ. Навч. посібник. – К.: Корнійчук, 2007. = 290 с.