

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

Кафедра автоматизації електромеханічних систем та електроприводу

МОДЕЛЮВАННЯ ТА АНАЛІЗ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ В MATLAB

Методичні вказівки до виконання лабораторних робіт

для студентів

спеціальності 141 – “Електроенергетика, електротехніка та електромеханіка”
спеціалізації – “Електромеханічні системи автоматизації та електропривод”

Рекомендовано Вченою радою ФЕА НТУУ “КПІ”

Київ
НТУУ «КПІ»
2017

Моделювання та аналіз електромеханічних систем в MATLAB. Методичні вказівки до лабораторних робіт для студентів спеціальності 141 – "Електроенергетика, електротехніка та електромеханіка" спеціалізації "Електромеханічні системи автоматизації і електропривод") / Укл.: О.І.Толочко – Київ: КПІ ім. Ігоря Сікорського, 2017. – 235 с.

Гриф надано Методичною радою

ФЕА НТУУ «КПІ»

(Протокол № 4 від 27 листопада 2017 р.)

Навчальне видання

МОДЕЛЮВАННЯ ТА АНАЛІЗ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ В MATLAB

Методичні вказівки до виконання лабораторних робіт

для студентів денної та прискореної форм навчання
спеціальності 141 – "Електроенергетика, електротехніка та електромеханіка"
спеціалізація – "Електромеханічні системи автоматизації та електропривод"

Укладач:

О.І. Толочко, д.т.н., проф.

Відповідальний редактор:

С.М. Пересада, д.т.н., проф., зав. каф.

Рецензент

М.Я. Островерхов, д.т.н., проф., зав. каф.

ВВЕДЕННЯ

Математичне моделювання систем електропривода є важливим етапом їх проектування. На цьому етапі виконується аналіз динамічних властивостей електроприводу з точки зору відповідності технологічним вимогам, уточнюється структура системи керування, типи регуляторів, їх параметри. При моделюванні можна безбоязно досліджувати поведінку системи в аварійних ситуаціях, що неможливо на лабораторних і тим паче на діючих промислових установках. Іноді моделювання проводять для того, щоб оцінити коректність прийнятих при математичному опису системи спрощень.

Математичне моделювання на ЕОМ зводиться до розв'язання системи диференціальних або різницевих рівнянь, що описують динамічні властивості досліджуваного об'єкта, в результаті чого отримують графіки перехідних процесів об'єкта в характерних режимах його роботи.

При дослідженні систем автоматичного регулювання (САР) і, зокрема, систем керування електроприводами вихідним матеріалом для моделювання дуже часто є структурна схема, що уявляє собою графічну інтерпретацію математичного опису системи. В структурну схему, крім лінійних динамічних ланок можуть входити арифметичні ланки, "типові нелінійності" та нелінійності довільного вигляду, задані аналітично або графічно, різноманітні ключі, логічні елементи та т.і.

Перехідні процеси можна розраховувати як за допомогою програм, написаних будь-якою алгоритмічною мовою, що потребує від дослідника достатньо високої кваліфікації в галузі програмування та обчислювальної математики, так і за допомогою спеціалізованого програмного забезпечення, що дозволяє користувачу задавати моделі у вигляді математичних рівнянь або у вигляді структурних схем, обирати методи розв'язання диференціальних рівнянь та їх параметри в діалоговому режимі та отримувати результати у зручній формі.

Одним із найзручніших програмних засобів структурного математичного моделювання на теперішній час є додаток *Simulink* пакета *MATLAB* фірми *Mathwork* [1-5].

Основою для розробки моделей в *Simulink* є бібліотеки блоків, з котрих складаються структурні схеми систем автоматичного управління, що повинні бути дослідженими. Розрахунок перехідних процесів може бути виконаний за допомогою відповідних операцій *Simulink*-меню або в програмному режимі (з використанням функцій пакета *MATLAB*).

Наразі все більшу актуальність при дослідженні складних електромеханічних об'єктів набуває віртуальне фізичне моделювання, яке не потребує від користувача знання математичного опису досліджуваного процесу, потребують менше часу для налаштування, є більш наочними, мають розвинені засоби візуалізації та анімації результатів досліджень. Такі моделі можна створювати і в *Simulink* шляхом використання віртуальних механічних блоків бібліотек *SimPowerSystems (SPS)* та *SimMechanics (SM)*. Блоки бібліотеки *SPS*, призначені для моделювання електричних та електромагнітних кіл, електронних пристроїв, електродвигунів і ліній електропередач, подано у вигляді позначень відповідних елементів на принципових електричних схемах [5-8]. Блоки бібліотеки *SM*, призначені для моделювання складних механічних об'єктів зі складною кінематикою та багатьма ступіннями свободи, подано у вигляді механічних тіл обертового і поступального руху та різноманітних кінематичних передач [9-11]. Математичний опис окремих елементів цих бібліотек приховано від користувача, завдяки чому створюється ілюзія віртуального фізичного моделювання.

Маючи структурну математичну або віртуальну фізичну модель досліджуваного об'єкта, складену в програмному середовищі пакета *MATLAB*, можна не тільки отримати графіки перехідних процесів, а і всебічно проаналізувати його властивості у просторі часу, у просторі змінної Лапласа, побудувати частотні характеристики та карти розташування нулів-полісів [12-16].

Отже, метою даного лабораторного практикуму є придбання практичних навичок структурного математичного та віртуального фізичного моделювання неперервних стаціонарних динамічних систем та їхнього автоматизованого аналізу у названому вище середовищі. У якості об'єктів дослідження обрано електричні кола, електродвигуни, механічні та електромеханічні системи.

Завдання та питання до лабораторних робіт складені у такий спосіб, щоб моделювання не стало самоціллю, а було одним із методів вивчення властивостей електротехнічних та електромеханічних об'єктів.

Автор методичних вказівок вважає, що читач обізнаний з основами *Windows*-технології та має навички роботи і програмування в середовищі пакета *MATLAB*. Методичні вказівки, враховуючи можливості існуючого апаратного та програмного забезпечення, складено для версії *MATLAB-9b (R12.1)*, *Simulink-5.1*, *SimPowerSystems-5.2* і *SimMechanics-3.1.1*. Але більшість матеріалів може бути використано і при роботі з більш пізніми версіями.

У часи, передбачені для самостійної роботи, студенти повинні скласти звіт з попередньої лабораторної роботи та підготуватися до наступної шляхом пророблення відповідного лекційного матеріалу, рекомендованої літератури та методичних вказівок. Для перевірки готовності до виконання лабораторної роботи викладач може провести контрольне опитування.

Під час виконання лабораторної роботи студенти збирають математичні моделі, вводять текстові програмні файли, призначені для ініціалізації моделей та виконання модельного експерименту, і фіксують результати.

Звіт з лабораторної роботи повинен отримувати таку інформацію:

- 1) назва роботи;
- 2) завдання згідно з варіантом;
- 3) схеми моделей з параметрами блоків;
- 4) тексти програмних модулів;
- 5) результати досліджень.

1 Лабораторна робота №1

ЗНАЙОМСТВО З СЕРЕДОВИЩЕМ ПРОГРАМИ СТРУКТУРНОГО МОДЕЛЮВАННЯ *Simulink* СИСТЕМИ ПРОГРАМУВАННЯ *MATLAB*

1.1 Запуск *Simulink*. Перелік бібліотек та демонстрацій

Simulink можна запустити з командної строки пакета *MATLAB* одною-менною командою

```
>>simulink
```

(>> – запрошення *MATLAB* до вводу команди) або спеціальною кнопкою на панелі інструментів, що показана на рис. 1.1.

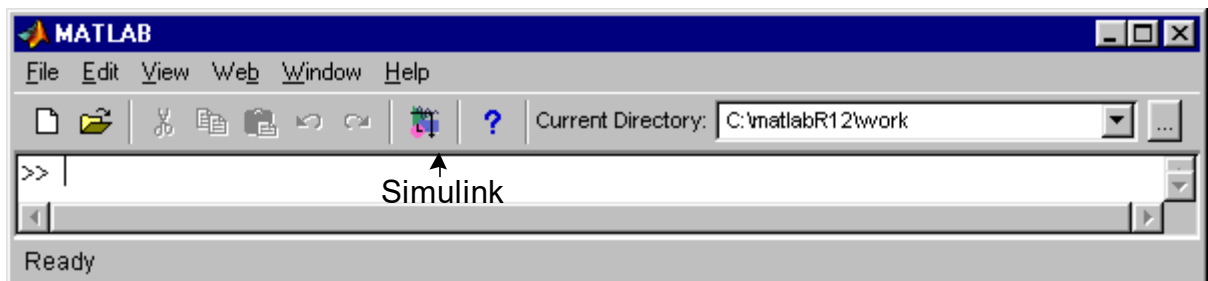


Рис. 1.1 – Командне вікно системи *MATLAB*

При цьому на екрані відкривається вікно з бібліотеками блоків, представлене на рис. 1.2, яке має назву *Simulink Library Browser (Навігатор Бібліотек)*.

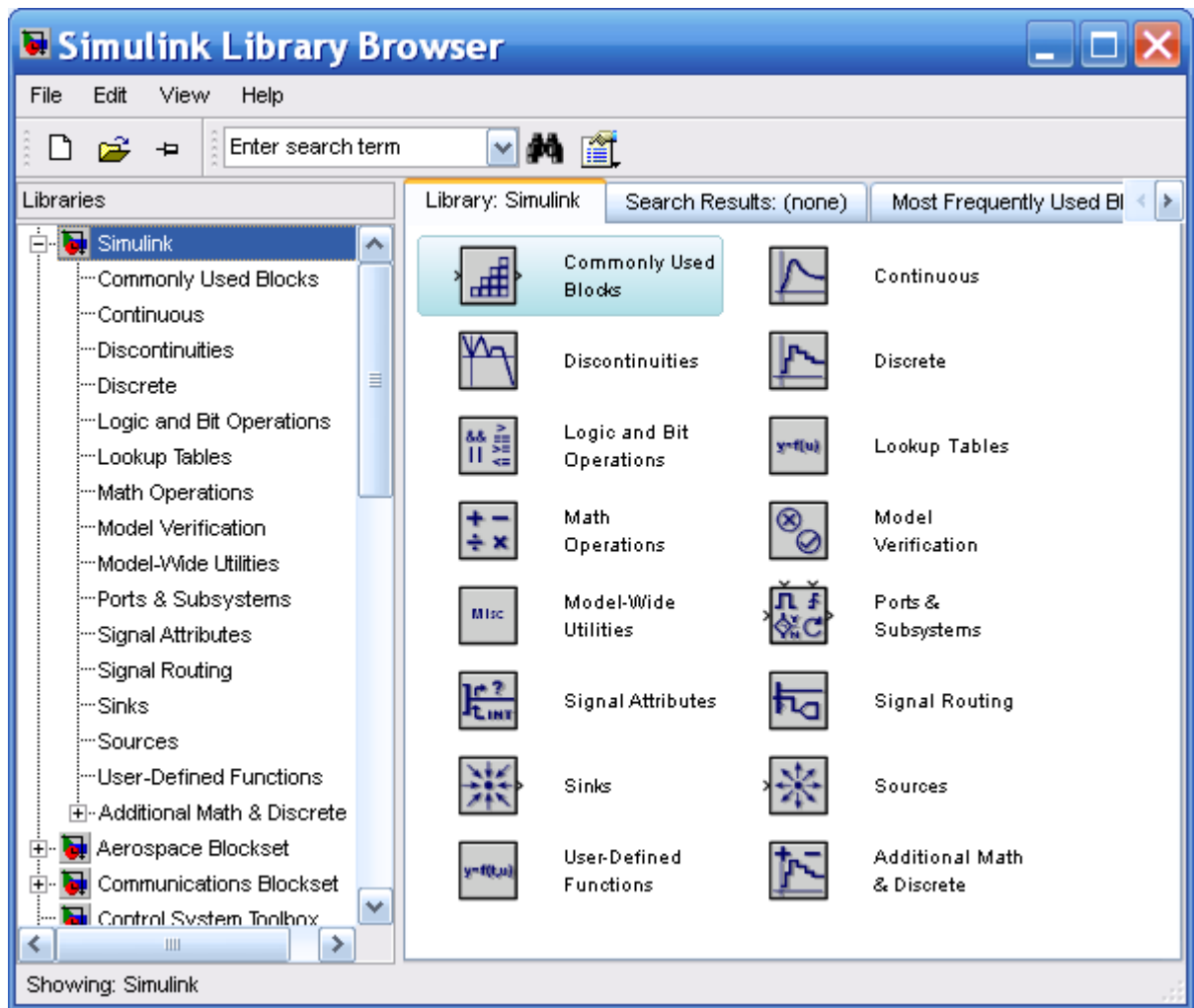
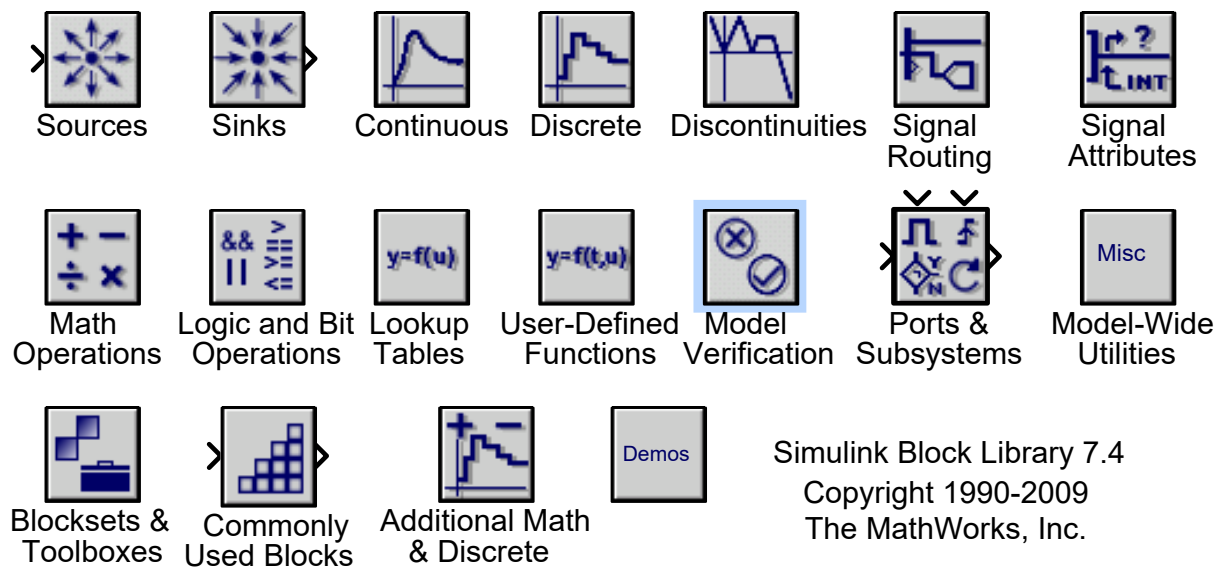
Кожну бібліотеку можна розкрити подвійним натиском лівої кнопки миші на піктограмі бібліотеки в правій частині вікна або вибором назви бібліотеки в лівій частині вікна. При цьому піктограми блоків обраної бібліотеки заміщають піктограми бібліотек.

Натиском правою кнопкою миші на імені бібліотеки через меню, що випадає, можна відкрити будь-яку бібліотеку у вигляді, звичному для користувачів більш старих версій.

Simulink зі старим інтерфейсом можна також активізувати з командної строки командою

```
>>simulink3
```

При цьому відкривається вікно, зображене на рис. 1.3.

Рис. 1.2 – Вікно *Simulink Library Browser*Рис. 1.3 – Вікно бібліотек блоків *Simulink*

З цього вікна кожен бібліотеку можна розкрити подвійним натиском миші по її піктограмі. Вікна бібліотек будуть показані нижче (при розгляді їхніх блоків).

До основних *Simulink-бібліотек* відносяться

<i>Source</i>	– джерела входних сигналів;
<i>Sinks</i>	– блоки реєстрації та візуалізації;
<i>Continuous</i>	– неперервні лінійні динамічні ланки;
<i>Discrete</i>	– дискретні лінійні динамічні ланки;
<i>Discontinuities</i>	– типові нелінійності;
<i>Math Operations</i>	– математичні операції;
<i>Logic & Bit Operations</i>	– логічні та бітові операції;
<i>Lookup Tables</i>	– кусково-лінійна апроксимація табличних функцій;
<i>User Define Functions</i>	– функції користувача;
<i>Ports & Subsystems</i>	– порти та підсистеми;
<i>Commonly Used Blocks</i>	– блоки, що використовуються найчастіше;
<i>Blocksets & Toolboxes</i>	– комплектні блоки та комплекти інструментів.

Елементи розділу *Blocksets & Toolboxes* показано на рис. 1.4. Перелічимо деякі з них:

<i>Signal Processing Toolbox</i>	– створення, обробка та аналіз сигналів;
<i>Robust Control Toolbox</i>	– інструменти робастного керування;
<i>Fuzzy Logic Toolbox</i>	– інструменти нечіткої логіки;
<i>System Identification</i>	– ідентифікація систем;
<i>MPC Toolbox</i>	– інструменти прогнозного керування (<i>MPC Model Predictive Control Toolbox</i>);
<i>Neural Network Blockset</i>	– елементи нейрональних мереж;
<i>Simpower Systems</i>	– елементи електричних схем і систем, силова електроніка, електричні машини;
<i>Report Generator</i>	– генератор звітів;

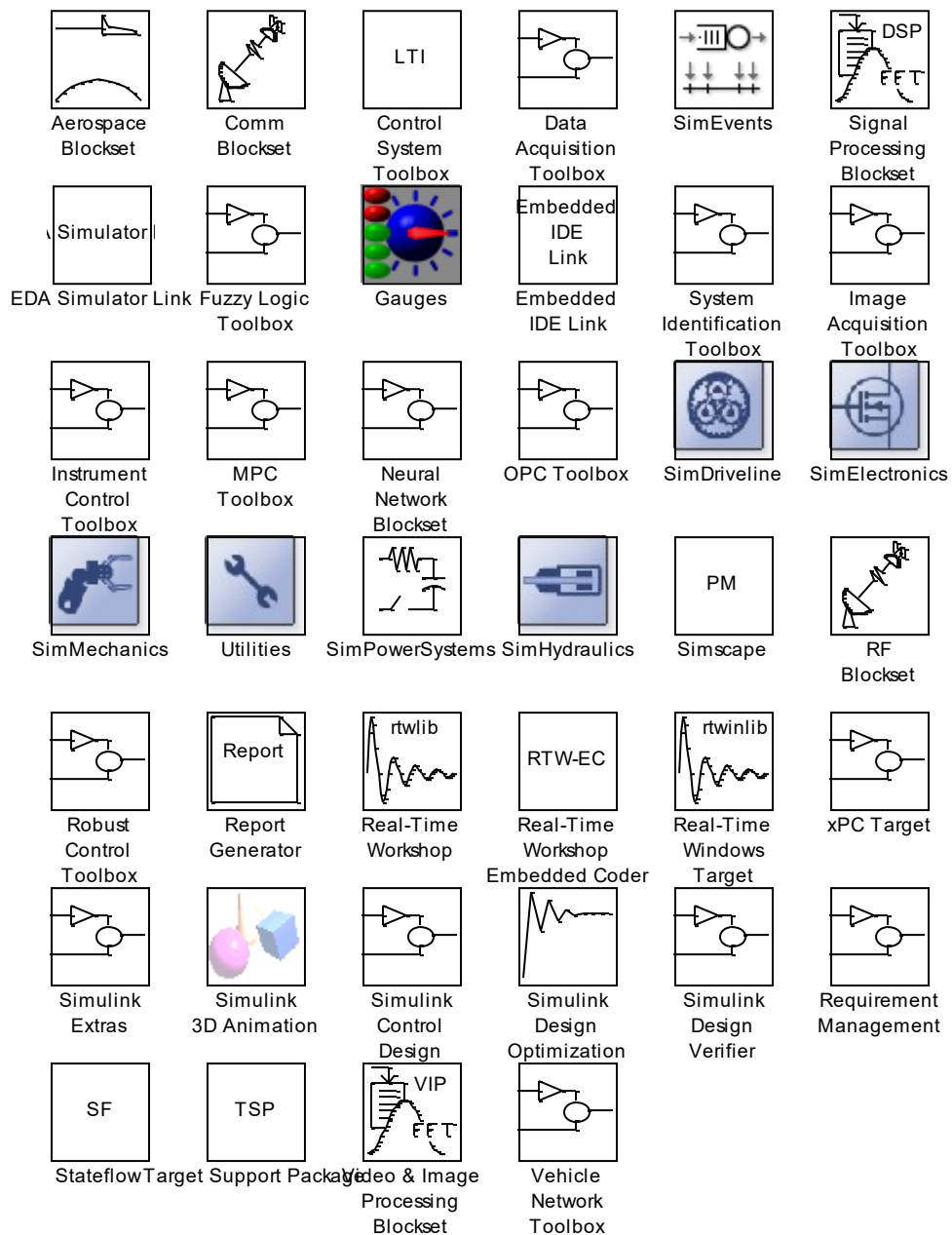


Рис. 1.4 – Вікно бібліотек *Blocksets & Toolboxes*

- Simulink design optimisation* – оптимізація синтезу в середовищі *Simulink*;
Real-Time Workshop – майстерня реального часу;
Simulink extras – спеціальні блоки *Simulink*.

Ретельний опис кожного з цих розділів вимагає окремої книги.

Окремо зупинимося на додатку віртуального фізичного моделювання , *SimScape*, до складу якого, як видно з рис. 1.5 поряд з фундаментальною бібліотекою *Foundation Library* та бібліотекою механічних блоків *SimMechanics*

входять бібліотеки гідравлічних блоків *SimHydraulics* (насоси), електронних пристроїв *SimElectronics*, трансмісій *SimDriveline*.

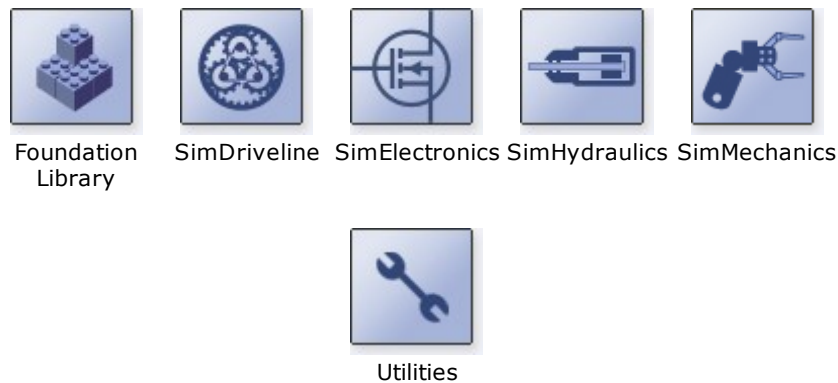


Рис. 1.5 – Вікно бібліотек *SimScape*

При подвійному натиску мишею по піктограмі блоку відкривається вікно введення його параметрів, у якому відображуються ім'я блоку, його тип, короткий опис і рядки введення параметрів із супроводжуючими їх запитами. Більш докладну інформацію про блок можна побачити, звернувшись до функції *Help*. Піктограма блоку найчастіше відображує його динамічні або статичні властивості і реагує на зміну параметрів. Як приклад, на рис. 1.6 наведено вікно введення параметрів блоку *Transfer Function*.

Параметри блоків можуть бути константами, перемінними, функціями або виразами, припустимими в *MATLAB*. Будь-які перемінні, від яких залежить параметр, повинні бути визначені в робочому середовищі до початку процесу моделювання, інакше *Simulink* сигналізує про помилку в цьому блоці.

Клавіша *Apply* дозволяє оцінити результат зміни параметрів, не закриваючи вікна їхнього введення, клавіша *OK* закриває вікно введення параметрів із запам'ятовуванням виконаних змін, клавіша *Cancel* закриває вікно введення параметрів зі скасуванням внесених змін.

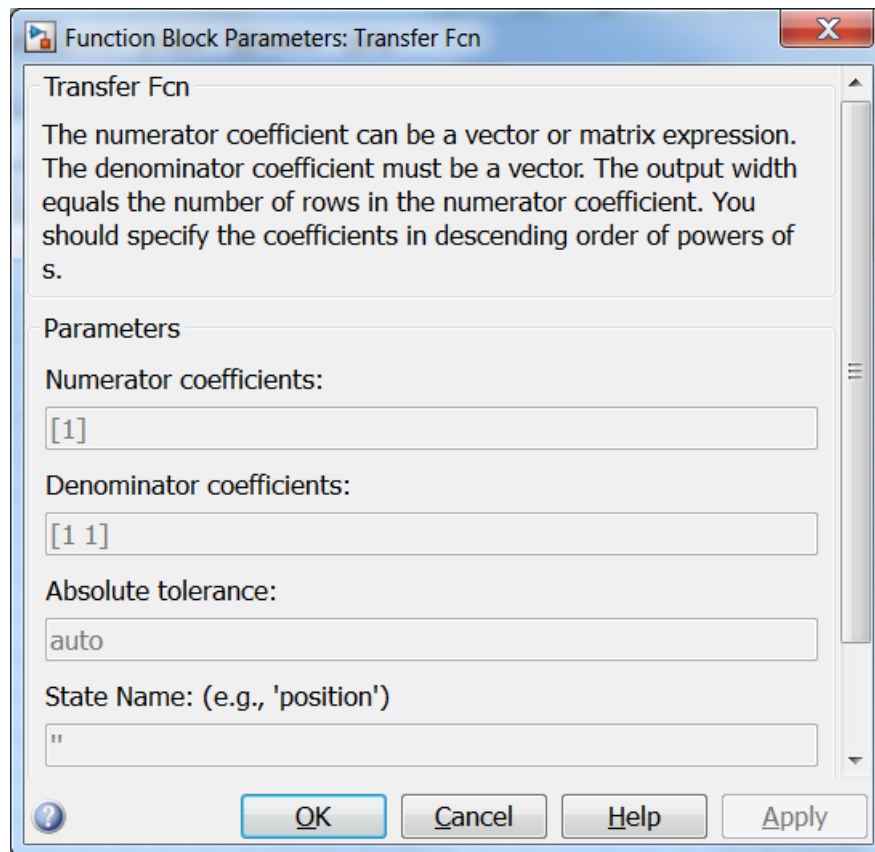


Рис. 1.6 – Вікно введення параметрів блока *Transfer Fcn*

Для знайомства з можливостями додатку *Simulink* та особливостями роботи деяких блоків можна скористуватися демонстраціями, які зручно запускати з вікна *MATLAB Demo Window*. Воно з'являється на екрані при виконанні команди

```
>>demo simulink
```

та пропонує наступні демонстрації:

1) *General applications* (основні демонстрації):

- *Bangbang – State event handling* (побудова фазового портрету),
- *Bounce – Bouncing ball using reset integrator* (підстрибування м'яча з використанням інтегратора зі скиданням вихідного сигналу),
- *Countersdemo – Counter demonstration* (демонстрація лічильників),
- *dblcart1 – Double cart demonstration* (двомасова система, що складається з двох возиків, з'єднаних між собою пружиною),
- *dblpend1 – Pendulum with one joint* (маятник з одним суглобом),

- *dbllpend2 – Pendulum with two joints* (маятник з двома суглобами),
 - *Hardstop – Friction model with hard stops* (модель тертя),
 - *Hydcyl – Hydraulic cylinder model* (модель гідравлічного циліндра),
 - *hydcyl4 – Four hydraulic cylinders* (чотири гідравлічних циліндри),
 - *hydrod – Two hydraulic cylinders interconnected with a stiff rod* (два гідравлічних циліндри з жорстким шатуном),
 - *onecart – Single cart/spring demonstration* (візок з пружиною, до якої прикладається періодичне зусилля, з анімацією),
 - *penddemo – Inverted pendulum (on cart) with animation* (інверсний маятник, встановлений на візку, з дискретним спостерігачем стану),
 - *simppend – Simple pendulum demonstration* (простий маятник),
 - *simquat – Quaternion rotation with animation* (обертання кватерніона),
 - *slprimes – Simulink prime number generator* (генератор простих чисел),
 - *thermo – Thermodynamic model of a house* (термодинамічна модель будинку),
 - *toilet – Toilet tank flushing animation* (злив туалетного бачка),
 - *vdp – Van der Pol system of equations* (розв'язання рівнянь Ван дер Пола);
- 2) *Feature demonstrations and samples* (особливі демонстрації):
- *busdemo – Bus block* (блоки створення шини та її селектора),
 - *combfilter – Data Typing with filter* (фільтрація сигналів одиночної та подвійної точності),
 - *datatypedemo – Data Typing demonstration* (демонстрація використання різних типів арифметичних даних),
 - *enablesub – Enabled subsystems* (підсистеми, які виконуються при додатному значенні керуючого сигналу),
 - *enabsubs – Advanced Enabled subsystems* (підсистеми з керуючим входом, що видає дозвіл на їх роботу, з додатковими можливостями),
 - *ifsub – If block* (демонстрація умовного блоку, що керує виконанням підсистем),

- *logdemo* – *Flip-flop blocks* (тригери),
 - *mergedemo* – *Merge block* (блок поєднання),
 - *prioritydemo* – *Block priority* (блок пріоритету),
 - *triggeredsub* – *Triggered subsystems* (підсистеми, які виконуються при перетині керуючим сигналом нуля та утримують вихідний сигнал до наступного перетину),
 - *zeroxing* – *Zero-crossing detection* (реєстрація перетину нуля),
 - *matrixdemos* – *Matrix signals* (матричні сигнали),
 - *sl_subsys_semantics* – *Subsystem semantics* (семантика підсистем),
 - *sfundemos* – *S-Functions* (демонстрація S-функцій);
- 3) *Automotive model demonstrations and samples* (автомобільні моделі);
- 4) *Aerospace model demonstrations and samples* (аерокосмічні моделі);
- 5) *Power System Blockset Demos* (електричні джерела, машини, трансформатори, лінії електропередач, силова електроніка):
- *psbfilter* – *Steady-state operation of linear circuit filter* (операції у просторі стану з лінійними фільтрами),
 - *psbtransient* – *Transient analysis of a linear circuit* (аналіз у часі лінійних електричних кіл),
 - *psbtransfo* – *Three-winding distribution transformer* (трансформатор з трьома обмотками),
 - *psbtransfosat* – *Energization of a three-phase Saturable Transformer* (збудження трифазного трансформатора з насиченням),
 - *psbsurgnetwork* – *AC network Surge Arrester* (сіть змінного струму з громовідводом),
 - *psbrectifier* – *Three-phase diode rectifier* (трифазний діодний випрямляч),
 - *psbconverter* – *Thyristor Converter* (тиристорний перетворювач),
 - *psbswitching* – *Ideal Switch in switching circuit* (ідеальний ключ),
 - *psbmosconv* – *Mosfet converter* (транзисторний Mosfet-перетворювач),

- *psbbuckconv* – *Gto Thyristor in buck converter topology* (Gto-тиристор у складі перетворювача-компенсатора реактивної потужності),
- *psbloadshed* – *Load shedding on a simplified alternator* (зниження навантаження генератора змінного струму),
- *psbturbine* – *Synchronous generator powered by hydraulic turbine* (синхронний енергатор із гідравличною турбіною),
- *psbpwm* – *Asynchronous machine fed by PWM inverter* (асинхронна машина з живленням від ШІМ-інвертора),
- *psbptmotor* – *PM synchronous motor fed by PWM inverter* (явнополюсна синхронна машина з живленням від ШІМ-інвертора),
- *psbtriplhaseline* – *Single phase energization of a three-phase line* (однофазне живлення трифазної лінії),
- *psbmonophaseline* – *Switching of a single phase line* (переключення однофазної лінії),
- *psbmachines* – *Machines and load flow* (машини та навантаження),
- *psbcompensated* – *Case study 1 Series compensated network* (послідовна компенсована сіть),
- *psbdcdrive* – *Case study 2 Chopper fed DC motor drive* (двигун постійного струму з імпульсним керуванням),
- *psbregulator* – *Case study 3 Synchronous machine and regulator* (синхронна машина і регулятор),
- *psbfrequency* – *Case study 4 Variable frequency induction motor drive* (асинхронний електропривод зі змінною частотою),
- *psbhvdc* – *Case study 5 HVDC transmission system* (високовольтна трансмісивна система постійного струму).

Імена демонстрацій співпадають з іменами *mdl*-файлів, що знаходяться відповідно у папках

Matlab\toolbox\simulink\simdemos\ simgeneral

Matlab\toolbox\simulink\simdemos\ simnew

Matlab\toolbox\simulink\simdemos\ simfeatures

Matlab\toolbox\simulink\simdemos\ automotive

Matlab\toolbox\simulink\simdemos\ aerospace

Matlab\toolbox\powersys\powerdemo

Так що кожен окрему демонстрацію можна виконати з командної строки введенням імені файлу моделі без поширення.

Для спілкування користувача з ЕОМ під час роботи з поширенням *Simulink* у середовищі системи *MATLAB* передбачені такі засоби:

- меню командного вікна *MATLAB*;
- основні меню вікон *Simulink*-моделей, *Simulink*-бібліотек та вікна *Simulink Library Browser*;
- контекстуальні меню цих вікон, що викликаються правою кнопкою миші при виділеному будь-якому елементу вікна;
- кнопки панелі інструментів;
- команди, що набираються та вводяться користувачем безпосередньо у вікні *Matlab Comand Window* або у програмному режимі при виконанні програми, що складається з відповідних команд.

1.2 Меню вікна *Simulink Library Browser*

Меню вікна *Simulink Library Browser* містить наступні функції:

<u>F</u> ile	—	робота з файлами;
<u>E</u> dit	—	редагування;
<u>V</u> iew	—	вигляд вікна;
<u>H</u> elp	—	допомога.

Кожну функцію цього меню можна вибрати не тільки мишею або клавішами $\langle \rightarrow \rangle$, $\langle \leftarrow \rangle$ і $\langle Enter \rangle$, але й комбінацією клавіш “ $\langle Alt \rangle + s$ ”, де s — підкреслений символ імені обраної функції.

Функція *File* містить наступні операції:

- New...*  – створити вікно для введення нової моделі (*Model* (^N)) або бібліотеки (*Library*);
- Open...* (^O) – відкрити модель;
- Close* – закрити вікно *Simulink Library Browser*;
- Preferences...* – переваги (настроювання середовища).

Операції підменю можна вибрати мишею, клавішами <↓>, <↑> та <Enter>, комбінацією клавіш, зазначеної в дужках, і клавішею <s>, де s – підкреслений символ імені обираної операції.

Для збереження моделей і інших файлів користувача в *Windows* передбачена папка *C:\Users\...\Documents\Matlab*. У ній можна створити нову папку, наприклад, *C:\Users\...\Documents\Matlab\models*. Шлях до файлів-моделей необхідно додати у список доступних директорій за допомогою операції *Set Path* → *Add Folder...* / *Add with Subfolder*. Нову папку можна установити в початок списку (*Move to Top*), у кінець списку (*Move to Bottom*), перемістити на одну позицію вниз (*Move Down*) або догори (*Move Up*). Якщо не виконати запис нового списку у файл, то установка діє тільки протягом поточного сеансу роботи. Для того, щоб використовувати нову установку в майбутніх сеансах, потрібно записати її у файл (*MATLAB \ toolbox \ loc l \ pathdef.m*) за допомогою клавіші *Save*.

Файли моделей і бібліотек додатка *Simulink* повинні мати маску *.*mdl*. Вони є текстовими файлами особливої структури і містять інформацію про структуру та параметри моделей, достатню для їхнього графічного відображення і моделювання. З вікон *Simulink* вони відкриваються в графічному вигляді, а з будь-якого текстового редактора (наприклад, вбудованого текстового редактора системи *MATLAB medit.exe*) – у текстовому. Починаючи з версії *MATLAB-2012a*, розширення *mdl* замінено на *slx*.

Текстовий редактор системи *MATLAB* відкривається командою *File* → *New* → *M-file* меню командного вікна *MATLAB*.

Відкрити модель можна не тільки з меню *File* будь-якого *Simulink*-вікна, але й з меню *File* пакета *MATLAB* за допомогою функції *Open...*, а також з командного рядка *MATLAB* введенням у ній імені файлу-моделі без розширення. Оскільки в такий же спосіб (з командного рядка) у *MATLAB* викликаються на виконання *script-файли* (головні програми, написані алгоритмічною мовою системи *MATLAB*) і виводяться значення перемінних [1, 2], тож файлам потрібно давати оригінальні і не занадто короткі імена.

Тут слід зазначити, що при використанні імен перемінних, функцій, файлів, програм, що збігаються з іменами, визначеними в робочому просторі або у файловій системі *MATLAB* раніше, вона не виводить попереджень про збіг імен. При виконанні команди, що містить яке-небудь ім'я, *MATLAB* шукає це ім'я серед перемінних, потім серед файлів з розширеннями *m*, *mex*, *mdl* і серед убудованих функцій в указаному порядку. Перше з виявлених імен сприймається як об'єкт для виконання команди. Отже, використання неоригінальних імен файлів може призвести до казусів.

Якщо при спробі відкрити модель або виконати яку-небудь програму з командного рядка на екран виводиться повідомлення *Undefined function or variable 'name'* (невизначена функція або перемінна), то це означає, що користувач або помилився при наборі імені файлу, або цей файл знаходиться в недоступній для системи *MATLAB* директорії (у більш пізніх версіях *MATLAB* приєднання нових директорій до списку доступних виконується системою програмування автоматично при зверненні до будь-якого файла цієї директорії).

Функція *Preferences* дозволяє виконувати налаштування середовища *MATLAB* і *Simulink*.

Параметр *Window reuse* (повторне використання вікна) визначає, чи повинен *Simulink* використовувати старе вікно або відкрити нове для показу змісту підсистеми. Він може приймати одне з 4-х значень: *none*, *reuse*, *replace* і *mixed*.

Дії, виконувані при відкриванні й закриванні підсистеми з різними значеннями цього параметра, перераховані в табл. 1.1.

Таблиця 1.1

Значення параметра	Дії при відкриванні підсистеми	Дії при закриванні підсистеми
<i>none</i>	Зміст підсистеми відображається у новому вікні на передньому плані	Батьківське вікно переміщається на передній план, а вікно підсистеми – на задній
<i>reuse</i>	Зміст підсистеми заміщає батьківську модель у поточному вікні	Вихідна (батьківська) модель заміщає зміст підсистеми в поточному вікні
<i>replace</i>	Зміст підсистеми відображається у новому вікні, а батьківське вікно зникає	Батьківське вікно з'являється, а вікно підсистеми зникає
<i>mixed</i>	Зміст підсистеми відображається в її власному вікні	Батьківське вікно переміщається на передній план, а вікно підсистеми зникає

Установкою або скиданням прапорців у полях *Browser initially visible*, *Show library links* і *Look under masks*, що визначають властивості *Навігатора Моделей (Model Browser)*, можна змінювати режими перегляду структури моделі.

При установці першого з перерахованих прапорців знов відкрите *Simulink*-вікно поділяється на дві частини. Ліва частина приділяється для навігатора, а права – для моделі. Це має сенс при наявності в моделі звичайних і замаскованих підсистем (*Subsystems*). Два інших прапорці дозволяють вмикати і вимикати зв'язок підсистеми з бібліотекою і переглядати зміст замаскованих блоків. Розрив зв'язку з бібліотекою передбачено для коректування структури підсистеми або вікна маскування.

Для зображення векторних сигналів стовщеними єднальними лініями слід установити прапорець у поле *Wide nonscalar lines*, а для виводу типів вихідних сигналів – у поле *Port data types* опції *Display*.

При установці прапорця в поле *Callback tracing* на екран виводяться команди, виконувані в специфічних режимах роботи блоків чи блокових діаграм. Як приклад можна навести команди, записані в поле параметра *Open function* вікна *Block Properties*, які виконуються при подвійному натисканні мишею на піктограмі цього блоку.

Крім цього, у вікні *Preferences* можна установити шрифти (*Simulink Fonts*), що використовуються для позначення блоків (*Blocks*), єднальних ліній (*Lines*) і коментарів (*Annotations*), а також параметри моделювання (*Simulink Simulation Preferences*).

Функція *Edit* містить наступні команди:

- Add to the current model (^I)* – скопіювати обраний блок або бібліотеку в активне вікно моделі чи бібліотеки;
- Find block... (^F)* – знайти блок за його ім'ям;
- Find next block (<F3>)* – знайти наступний блок.

При використанні команд функції *View* слід звернути увагу на можливість відкриття тільки основних бібліотек блоків *Simulink* (*Collapse entire Browser*) і повного їхнього комплексу (*Expand entire Browser*), а також на режим *Stay on top*, що у вимкненому стані забезпечує статус вікна *Simulink Library Browser* «поверх усіх вікон».

1.3 Меню вікон *Simulink*-моделей

Меню вікон *Simulink*-моделей відрізняється від меню вікна *Simulink Library Browser* і отримує функції та кнопки, які наведені на рис. 1.8.

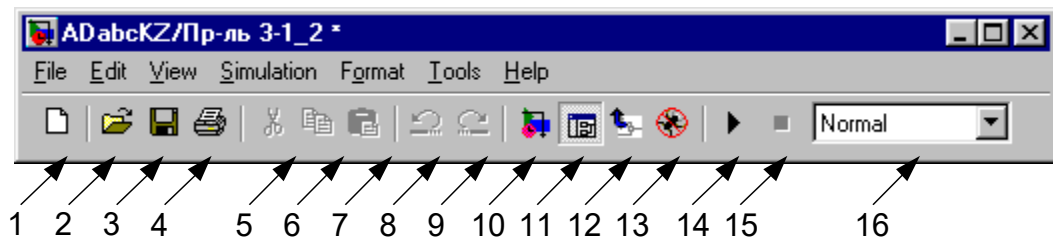


Рис. 1.8 – Меню та кнопки вікна *Simulink*-моделі

Перші дев'ять кнопок виконують стандартні операції *Windows* (1 - відкрити нове *Simulink*-вікно, 2 - відкрити існуючу модель, 3 - зберегти модель у файлі, 4 - надрукувати модель, 5-7 - вирізати (5), скопіювати (6) виділений фрагмент моделі в буфер та причепити (7) зміст буфера до активного вікна, 8,9 - скасувати / відновити результат останнього редагування), а інші - специфічні операції *Simulink*:

- 10 – *Library Browser* (активізувати „навігатор бібліотек”);
- 11 – *Toggle model browser* (увімкнути „навігатор моделі”);
- 12 – *Go to parent system* (активізувати батьківську модель);
- 13 – *Debug* (увімкнути відлагоджувач);
- 14 – *Start simulation* (почати моделювання);
- 15 – *Stop simulation* (припинити моделювання);
- 16 – вибір режиму моделювання.

Установку випадаючого меню 16 не слід змінювати починаючим користувачам.

У полі заголовка вікна рис. 1.8 відображено шлях до моделі у файловій системі. Символ “*” у кінці шляху означає, що після редагування моделі не виконано її запис у файл.

Функція *File* містить стандартні операції роботи з файлами:

- New ▸ – створити вікно для введення нової моделі (*Model ^N*) або бібліотеки (*Library*);
- Open... (^O) – відкрити модель;
- Close (^W) – закрити модель;
- Save (^S) – зберегти модель у колишньому файлі;

<i>Save As...</i>	– зберегти модель у новому файлі;
<i>Source Control...</i>	– керування джерелами сигналів;
<i>Model Properties</i> ►	– властивості моделі;
<i>Preferences...</i>	– переваги (настроювання середовища);
<i>Print...</i> (^P)	– вивід на принтер;
<i>Print Setup...</i>	– установки принтера;
<i>Exit MATLAB</i>	– вихід із системи <i>Matlab</i> .

Вікно *Model Properties*, яке відкривається при виборі відповідної операції, містить вкладки *Summary*, *Callbacks* і *Hystory*. Вкладка *Summary* дозволяє увести творця (*Creator*), дату створення (*Created:*) і опис моделі (*Model description:*).

Вкладка *Callbacks* дозволяє визначити команди, що будуть автоматично виконані перед завантаженням моделі (*Model pre-load function:*), при її ініціалізації (*Model initialization function:*), перед стартом моделювання (*Simulation start function:*), після його закінчення (*Simulation stop function:*) і перед збереженням моделі у файлі (*Model pre-save function:*). Вкладка *Hystory* містить дані про модифікацію моделі.

Вікно друку моделі *Print Model*, наведене на рис. 1.9, дає можливість, крім звичайних для *Windows* параметрів, установлювати глибину роздруківки блоків моделі при наявності в ній підсистем: *Current system*, *Current system and above*, *Current system and bellow*, *All systems*. Зміст цих установок наочно продемонстровано у графічній формі.

Функція *Edit* містить як стандартні *Windows-операції* редагування, так і деякі специфічні операції:

<i>Undo</i> (^Z)	– відмінити останню зміну;
<i>Redo</i> (^Y)	– відновити останню зміну;
<i>Cut</i> (^X)	– переміщення обраних об'єктів у буфер;
<i>Copy</i> (^C)	– копіювання обраних об'єктів у буфер;
<i>Paste</i> (^V)	– копіювання змісту буфера в обране місце графічного вікна;

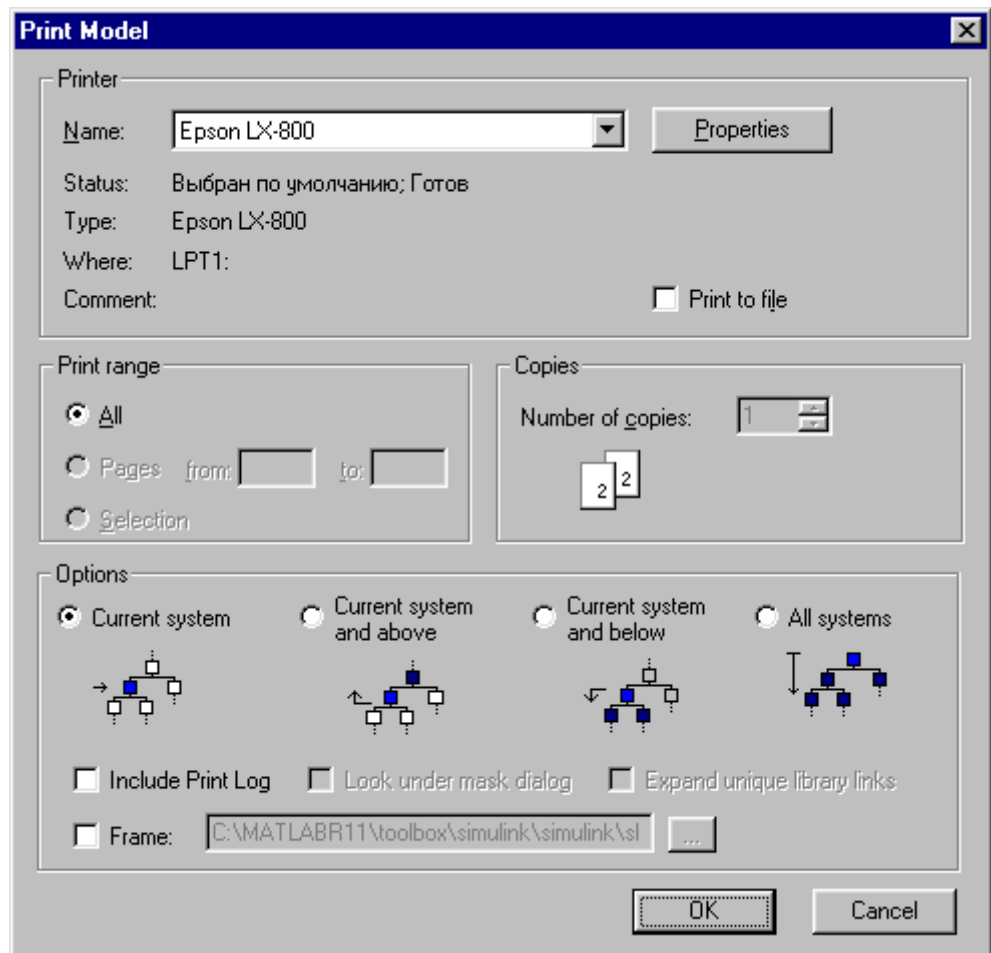
<i>C</i> <u>lear</u> (<i>Delete</i>)	– вилучення обраних об'єктів;
<i>S</i> <u>elect All</u> (^A)	– вибір всіх об'єктів в активному вікні
<i>C</i> <u>opy model to clipboard</u>	– копіювання моделі в буфер;
<i>F</i> <u>ind...</u>	– пошук блоків, сигналів, коментарів та інших об'єктів;
<i>O</i> <u>pen block</u>	– відкрити підсистему;
<i>M</i> <u>ask parameters...</u>	– параметри замаскованої підсистеми;
<i>B</i> <u>lock p</u> <i>arameters...</i>	– параметри блоку;
<i>B</i> <u>lock p</u> <i>roperties...</i>	– властивості блоку;
<i>M</i> <u>ask subsystem</u> (^M)	– маскування підсистеми;
<i>C</i> <u>reate subsystem</u> (^G)	– об'єднання обраних об'єктів у підсистему (<i>subsystem</i>);
<i>M</i> <u>ask subsystem</u> (^M)	– маскування підсистеми;
<i>L</i> <u>ook undo mask</u> (^U)	– заглянути під маску;
<i>L</i> <u>ink options</u> ►	– опції зв'язку підсистеми або замаскованого блоку з бібліотекою;
<i>U</i> <u>ppdate d</u> <i>iagram...</i> (^D)	– оновити блок-діаграму.

Команду *Update Diagram* необхідно використовувати в наступних випадках:

- після зміни бібліотечних блоків, копії яких використовуються в моделі;
- після зміни кількості вхідних або вихідних аргументів у *S-функції*, що описує модель.

Функція *View*, крім звичайних для *Windows* прапорців *Toolbar* і *Statusbar* та команд керування масштабом зображення *Zoom in*, *Zoom out*, *Fit system to view* і *Normal (100%)*, містить наступні операції:

<i>G</i> <u>o to parent</u>	– перехід до батьківської моделі;
<i>M</i> <u>odel browser</u> <i>options</i> ►	– властивості „навігатора моделі”;
<i>B</i> <u>lock d</u> <i>ata tips</i> <i>options</i> ►	– властивості типів даних блоків;
<i>S</i> <u>how Library</u> <i>Browser</i>	– перегляд „навігатора бібліотек”.

Рис. 1.9 – Вікно друку *Simulink*-моделі

Навігатор моделі відкривається при установці прапорця *Model browser options* ► → ✓ *Model browser* або при натисканні кнопки 11 (рисунок 1.8). При цьому модель займає праву (більшу) частину вікна, а *Model browser* – ліву. Навігатор моделі відображає її структуру (вкладеність підсистем).

Функція *Format* містить операції форматування моделі. Частина з них діє тільки після виділення фрагмента моделі. Тільки для виділених блоків діють такі операції:

- Flip name* — змінити положення імені блоку (для горизонтально розташованих блоків воно може розташовуватися вгорі або внизу, а для розташованих вертикально – праворуч або ліворуч),
- Hide/Show name* — сховати/показати ім'я блоку,

- Flip block (^I)* – повернути блок із праворуч на ліворуч або зверху вниз,
- Rotate block (^R)* – повернути блок на 90° проти годинної стрілки,
- Schow/Hide port labels* – показати/сховати імена портів (для підсистем);

для виділених блоків і коментарів можна виконати операції

- Text alignment* – спосіб вирівнювання тексту в текстовому блоці,
- Schow/Hide drop shadow* – показати/сховати тінь від блоку,
- Foreground Color* ▶ – установка кольору переднього плану (пиктограма й ім'я блоку),
- Background Color* ▶ – установка кольору заднього плану (фон блоку);

для виділених блоків, ліній і коментарів дійсна операція

- Font...* – установка шрифту для текстових написів (для зміни шрифту імені блоку повинний бути виділений блок, а не його ім'я).

Дія інших операцій поширюється на всю модель. До них відносяться

- Screen Color* ▶ – колір екрана,
- Library link display* ▶ – відображення зв'язків блоків з бібліотеками,
- Sample time colors* – установка кольору блоку індикації часу,
- Wide nonscalar lines* – зображення векторних сигналів стовщеними лініями,
- Signal dimensions* – відображення розмірності сигналів,
- Port data types* – виведення інформації про типи сигналів,
- Storage class* – клас пам'яті;
- Execution order* – виведення порядкового номера блоку в послідовності розрахунку їхніх вихідних сигналів при моделюванні.

Якщо фоновий колір блоку не збігається з кольором екрана, то вихідні лінії зв'язку цього блоку приймають колір його фону (*Background*), у против-

ному випадку вони приймають колір піктограми й імені блоку (*Foreground*). Вищесказане відноситься і до контурних ліній блоку.

Функція *Simulation* керує наступними режимами:

- Start* (^T) — старт моделювання;
- Stop* — переривання моделювання;
- Simulation Parameters...*(^E) — установка і редагування параметрів моделювання.

Вікно параметрів моделювання має декілька вкладок (панелей): *Solver*, *Workspace I/O*, *Diagnostics*, *Advanced*, та *Real-Time Workshop*.

Найбільш важливою є панель *Solver* (розв'язання рівнянь, що описують математичну модель), яка передбачає введення таких параметрів:

- Start time* — час початку моделювання;
- Stop time* — час закінчення моделювання;
- Type* — тип розв'язання: *Fixed/Variable-Step* (з фіксованим/змінним кроком).

Серед методів розв'язання ДР з фіксованим кроком в *Simulink* пропонуються методи Рунге-Кутта першого-п'ятого порядків, а саме:

- ode5* — метод Рунге-Кутта 5-го порядку (*Dormand-Prince formula*);
- ode4* — метод Рунге-Кутта 4-го порядку (*fourth-order Runge-Kutta formula*);
- ode3* — метод Рунге-Кутта 3-го порядку (*Bogacki-Shampine formula*);
- ode2* — модифікований метод Ейлера (*improved Euler's formula / Heun's method*);
- ode1* — метод Ейлера (*Euler's method*).

Серед методів розв'язання ДР зі змінним кроком, або, вірніше, методів з заданою точністю, в *Simulink* пропонуються комбіновані методи Рунге-Кутта (4-5)-го (*ode45*) і (2-3)-го (*ode23*) порядків та деякі методи, більш пристосовані для розв'язання жорстких ДР:

- ode45* – комбінований метод Рунге-Кутта (4-5)-го порядку з автоматичним вибором кроку, призначений для розв’язання нежорстких ДР;
- ode23* – комбінований метод Рунге-Кутта (2-3)-го порядку з автоматичним вибором кроку, призначений для розв’язання нежорстких та слабо жорстких ДР;
- ode113* – багатокроковий метод Адамса-Мілтона-Бешфорта (метод прогнозу та корекцій змінного порядку), призначений для розв’язання нежорстких ДР; може бути більш ефективним, ніж *ode45*, при високій точності;
- ode15s* – багатокроковий метод (1-5)-го порядку (за замовчанням 5-го), що базується на формулах чисельного диференціювання (*NDFs*), але може використовувати і формули зворотного диференціювання (*BDFs*), у відповідності з методом Гіра, спрямований на розв’язання жорстких ДР;
- ode23s* – однокроковий метод, що базується на модифікованій формулі Розенброка 2-го порядку; для деяких жорстких ДР виявляється при невисокій точності більш ефективним, ніж *ode15s*;
- ode23t* – метод трапецій з використанням „вільного” інтерполянта для розв’язання помірно жорстких ДР;
- ode23tb* – комбінований метод, що використовує на першому етапі формулу трапецій, а на другому – зворотну диференціальну формулу другого порядку; як і метод *ode23s*, може бути при невисокій точності більш ефективним, ніж *ode15s*, для розв’язання жорстких ДР.

Якщо модель має тільки дискретні змінні стану, або зовсім не має динамічних ланок, то для моделювання використовують метод з фіксованим або змінним кроком. Змінний крок обирають, коли модель має у своєму

складі дискретні динамічні ланки з різними періодами переривання, або коли треба фіксувати моменти часу перетину деякими сигналами нульового рівня.

Для методів з фіксованим кроком треба задати значення цього параметру в полі *Fixed step size*.

Для методів зі змінним кроком задаються такі параметри:

- Max step size* – максимальний крок; за замовченням він розраховується за формулою $h_{\max} = (t_{\text{stop}} - t_{\text{start}}) / 50$;
- Min step size* – мінімальний крок;
- Initial step size* – початковий крок;
- Relative tolerance* – відносна точність (за замовченням 10^{-3});
- Absolute tolerance* – абсолютна точність (за замовченням 10^{-6});
- Refine factor* – коефіцієнт збільшення кількості точок моделювання (ціле число); для методу *ode45* рекомендовано значення 4, для інших методів – 1.

Операції функції *Tools* призначені для взаємодії *Simulink* з іншими інструментами пакета *MATLAB*, наприклад,

- Data explorer...* – перегляд імен, розмірів і типів змінних,
- Simulink debugger...* – завантаження *Simulink*-відлагоджувальника,
- Model difference* ▶ – відмінність моделей (має два режими: *Merge/Compare two models...* – об'єднання / порівняння двох моделей і *Compare to last saved model* ... – порівняння з останньою записаною моделлю),
- Linear analysis...* – лінійний аналіз,
- Report generator...* – завантаження генератора звітів.

Інші операції функції *Tools* не слід застосовувати без особливих на те підстав, тому що вони вимагають високої кваліфікації користувача і застосовуються при розв'язанні нетривіальних задач.

1.4 Меню вікон *Simulink*-бібліотек

Меню вікон *Simulink*-бібліотек має багато спільного з меню вікон *Simulink*-моделей.

Між ними існує така різниця:

- у меню бібліотек відсутні функції *Simulation* та *Tools*;
- у заблокованому стані у вікні бібліотек не можливо виконання операцій редагування, тобто усіх операцій функції *Format*, операцій *Clear*, *Cut*, *Paste*, *Create Subsystem*, *Mask Subsystem*, *Mask Parameters*, *Edit mask*, *Link options*, *Update diagram*, *Undo*, *Redo* функції *Edit* та операцій редагування, які здійснюються за допомогою “миші” (пересування блоків, зміна їх розміру, назви, тощо);
- для розблокування бібліотек функція *Edit* їхніх вікон має операцію *Unlock Library*.

Стан *Unlock Library* діє з моменту виконання відповідної операції до закриття вікна бібліотеки і поширюється на вкладені бібліотеки. При наступному відкритті бібліотеки вона буде знову заблокованою.

Користувач може редагувати існуючі бібліотеки (у розблокованому стані) та створювати власні.

Для створення нової бібліотеки треба відкрити її вікно через меню будь-якого вікна *Simulink*-моделі, *Simulink*-бібліотеки або *Simulink Library Browser*: *File* → *New* → *Library*, занести в нього необхідні блоки і закрити зі зберіганням у файлі.

Файли бібліотек, як і файли моделей, мають поширення *mdl*.

1.5 Типи *Simulink*-блоків

Simulink-блоки можуть створюватися різними способами.

Основу стандартних *Simulink*-бібліотек складають вбудовані блоки, які не доступні для перегляду користувачем у вигляді текстового файлу або структурної схеми.

Деякі стандартні блоки створені за допомогою вже існуючих блоків і являють собою або модифікацію деякого блоку з іншими параметрами і зміненою піктограмою, або комбінацію декількох блоків, об'єднаних в одну підсистему. Решта блоків створені програмно за допомогою апарата *MATLAB*-функцій або *S*-функцій. Після імені такого блоку у вікні введення його параметрів стоять позначки (*mask*) і (*link*), що означають, що блок замаскований і зв'язаний з однією з бібліотек. У такий же спосіб може створювати нові блоки і користувач. Заглянувши під маску (*Edit* → *Look undo Mask* = ^U), можна побачити в окремому вікні, з чого вони складаються (цю ж операцію можна виконати через контекстуальне меню, що відкривається щигликом правої кнопки миші по піктограмі блоку). Для перегляду і редагування параметрів маскування (*Edit* → *Edit mask...* = ^M) необхідно попередньо тимчасово вивести з ладу зв'язок з бібліотекою (*Edit* → *Link options* → *Disable link*). Після перегляду і внесення змін в установки цього вікна зв'язок з бібліотекою можна або відновити (*Restory link*) або перервати (*Break link*). При відновленні зв'язку можливі два варіанти: ігнорувати результати редагування (*Use library block*) чи поширити ці результати на бібліотечний блок (*Update library*). При опису імена цих блоків будуть позначені символом “*”.

S-функції можуть бути написані за особливими правилами на мовах *MATLAB* (*.m), або C++ (*.cpp), або C (*.c), або Ada (*.adb), або Fortran (*.f), а потім конвертовані в блоки *Simulink*.

1.6 Створення та редагування моделей

Перед початком формування структурної схеми необхідно відкрити для неї нове вікно (*Simulink* → *File* → *New* → *Model* (^N) або *MATLAB* → *File* → *New* → *Model*). При цьому відкривається вікно з заголовком “*Untitled*”. При запису в файл (*File* → *Save* / *Save as...*) у якості заголовку вікна буде фігурувати призначене користувачем ім'я файла, яке за замовченням отримає поширення *mdl*.

Вже існуючу модель можна завантажити з меню *Simulink* або *MATLAB* (*File* → *Open*) та із командного рядка *MATLAB* введенням в ній імені файла, в якому збережена модель, без поширення.

В основу створення та редагування моделей покладено, як і в більшості графічних *Windows*-додатків, принцип *drag-and-drop* (перетягни та покинь).

Копіювання блоків із бібліотек або з будь-якої вже існуючої моделі у вікно створеного файла після їх відкриття виконується мишею за допомогою операції *drag* (тягнути при натиснутій лівій клавіші миші). Аналогічно здійснюється переміщення блоків всередині вікна. Копіювання блоків всередині вікна виконується операцією *drag right* (тягнути при натиснутій правій клавіші миші). При цьому до імені нового блока додається цифра, що відображає порядковий номер копіювання, чим забезпечується унікальність імені кожного блока однієї моделі.

Імена блоків можна змінювати безпосереднім редагуванням. При цьому не можна їх дублювати або залишати блок без імені (ім'я – пустий рядок), але можна сховати ім'я (*Format* → *Hide Name*). Зворотна операція здійснюється командою *Format* → *Show Name*. Для завершення редагування імені треба зробити щиглик мишею зовні поля введення тексту. Після цього ім'я аналізується системою та чи приймається, чи відкидається нею з виведенням повідомлення. Для зміни шрифту імені (*Format* → *Font...*) необхідно попередньо виділити сам блок, а не його ім'я.

В довільній точці активного *Simulink*-вікна, що виділяється щигликом лівої клавіші миші, можна вставити будь-які коментарі (*Annotations*). Введення та редагування коментарю виконуються точно так, як відповідні операції з ім'ям блока. В результаті утворюється новий специфічний блок *Note*, який відрізняється від інших блоків тим, що він не має ні піктограми, ні портів, ні вікна введення параметрів, а складається з одного імені, яке є текстом анотації. Рештою цей блок схожий на інші: його можна копіювати, пересувати, знищувати, виділяти і т.п.

Іменами можна відмічати і лінії зв'язку. Для цього необхідно клацнути двічі лівою кнопкою миші по обраній з'єднувальній лінії та ввести текст в утворене поле. Отриманий у такий спосіб надпис буде, на відміну від блока *Note*, прив'язаний до лінії зв'язку.

При пересуванні блока, до якого вже приєднані лінії зв'язку, останні витягуються або скорочуються для збереження зв'язків; кінці зв'язків, не приєднані до блоку, що пересовується не змінюють своє положення. Перемістити блок в межах одного вікна без ліній зв'язку (висунути блок з моделі) можна операцією *<Shift>+drag*.

Для виконання якої-небудь дії з будь-яким об'єктом моделі (блоком, лінією зв'язку, сукупністю блоків та/або зв'язків, тобто фрагментом моделі, усією моделлю) його треба спочатку відмітити. Поодинокий об'єкт виділяється щигликом лівої клавіші миші (*click*). Фрагмент моделі можна виділити такими способами:

- помітити вибірково блоки або зв'язки, не відпускаючи клавішу *Shift* (*<Shift> + click*);
- помітити підряд, тобто. заключити за допомогою мишки у прямокутник;
- помітити все (*Edit* → *Select All*=*^A*).

Про те, що виділення відбулося, свідчать маленькі незафарбовані квадратики, розташовані в кутах обраних блоків та поблизу кінців обраних зв'язків.

З поміченим блоком можна виконувати такі операції:

- зміна розміру — *drag* за вугол;
- поворот на 90° — *Options* → *Rotate* або *^R*;
- поворот на 180° — *Options* → *Flip Horizontal* або *^I*;
- знищення — *<Delete>*;
- всі операції функцій *Edit*, *Options*, *Style* меню *Simulink*.

Подвійний щиглик по піктограмі блока розкриває текстове вікно для введення його параметрів.

Параметри блока можуть бути подані як в чисельному вигляді, так і у вигляді імен перемінних або математичних виразів. В останньому випадку до початку моделювання перемінним повинні бути присвоєні значення. Це можна зробити з командного рядка *Matlab* або виконанням командного файлу.

Зв'язки встановлюються між вхідними та вихідними портами блоків. Стрілка на зв'язку показує напрямок потоку даних. Утворюються зв'язки у такі способи:

- довільно кусочно-неперервно (*drag* від порту з перериванням цієї операції в бажаних точках зламу);
- з автоматичним розташуванням точок зламу.

Автоматичне розташування точок зламу забезпечується при виконанні операції *drag* від порту до порту без переривання, а також при швидкому з'єднанні блоків, яке виконується у такий спосіб: помічається початковий блок або блоки, натискається клавіша *Ctrl* і помічається кінцевий блок.

Для розгалуження лінії зв'язку використовують операцію *drag right* або *drag left* або *drag* будь-якою клавішею але у зворотному напрямку, тобто від вхідного порту до точки розгалуження.

При проведенні ліній зв'язку не варто намагатися влучити точно в порт; лінія приєднається до порту, якщо відпустити клавішу миші при знаходженні графічного курсора всередині блока або поблизу порту (на відстані менше 5 піксел).

З поміченими лініями зв'язку можливі виконувати такі операції:

- зміна напрямку – *Options* → *Rerout Lines*, або *^L*;
- паралельне пересування – *drag*, ухопившись за середину сегменту зв'язку;
- зміна кута між сегментами зв'язку – *drag*, ухопившись за вузол;
- додатковий злам сегмента зв'язку – *<Shift> + drag*, ухопившись за бажану точку зламу;
- ділення зв'язку – *drag*, ухопившись за першу ліву мітку зв'язку або *drag right* від будь-якої точки зв'язку;

- знищення – *<Delete>*.

З поміченими фрагментами моделі або з усією моделлю можна виконувати всі ті операції, що і з окремими блоками та/або зв'язками.

Для того, щоб зробити розташування об'єктів моделі більш зручним, вікна *Simulink* мають невидиму сітку 5x5 піксел, до вузлів або до ліній якої прив'язуються всі об'єкти. Дискретне пересування виділених об'єктів можна виконати клавішами *<→>*, *<←>*, *<↑>*, *<↓>*.

При редагуванні моделі графічний курсор змінює свою форму, сигналізуючи користувачу про ту дію, до виконання якої підготовлена система:

-  – готовий для наступних дій;
-  – готовий для нанесення обмежувального прямокутника (виділення групи поруч розташованих об'єктів);
-  – готовий для пересування блока, сегмента лінії, або виділеного фрагмента;
-  – готовий до проведення лінії зв'язку;
-  – готовий для перенесення або для створення нової точки зламу лінії;
-  – готовий до зміни розміру блока.

1.7 Завдання

1) ознайомитися з основними та спеціальними бібліотеками *Simulink* на рівні їхніх можливостей;

2) ознайомитися з основними демонстраціями *Simulink* та демонстраціями поширення *Power System Blockset*, не вдаючись до зайвих подробиць;

3) створити задану викладачем папку в директорії *C:\Пользователи\Мои документы\MATLAB* та забезпечити доступ до неї з середовища пакета *MATLAB*;

4) відкрити незамасковану підсистему та зазирнути під маску замаскованої підсистеми;

5) випробувати описані вище функції настроювання середовища *Simulink* у вікнах демонстраційних моделей, створених при виконанні попереднього пункту завдання;

6) відкрити одну з моделей, створених при виконанні пункту 5, у середовищі текстового редактора та ознайомитися зі структурою відкритого файлу.

1.8 Контрольні питання

1. Поясніть призначення команд меню середовища *MATLAB*, кнопок панелі інструментів та окремих вікон (*Command Window*, *Command History*, *Current Directory* та ін.).

2. Яким чином можна виконати встановлення за власним бажанням поточного каталогу та доступних каталогів *MATLAB*? Навіщо це робити?

3. Поясніть призначення команд меню та кнопок панелей інструментів вікон *Simulink Library Browser* та *Simulink*.

4. Дайте загальну характеристику бібліотек та блоків *Simulink*.

5. Перелічите основні прийоми створення та редагування моделей. Як виконується встановлення потрібних параметрів моделювання, яка мета цього встановлення?

6. З яких міркувань обирають метод і параметри чисельного інтегрування при моделюванні?

7. Які засоби моделювання у середовищі *Simulink* Ви знаєте?

Лабораторна робота №2

ФОРМУВАННЯ ВХІДНИХ СИГНАЛІВ ТА РЕЄСТРАЦІЯ ВИХІДНИХ СИГНАЛІВ В СЕРЕДОВИЩІ *Simulink*

2.1 Основні засоби формування вхідних сигналів

Вхідні сигнали, що діють на системи автоматичного керування, здебільш є функціями часу. Їх можна розподілити на такі групи:

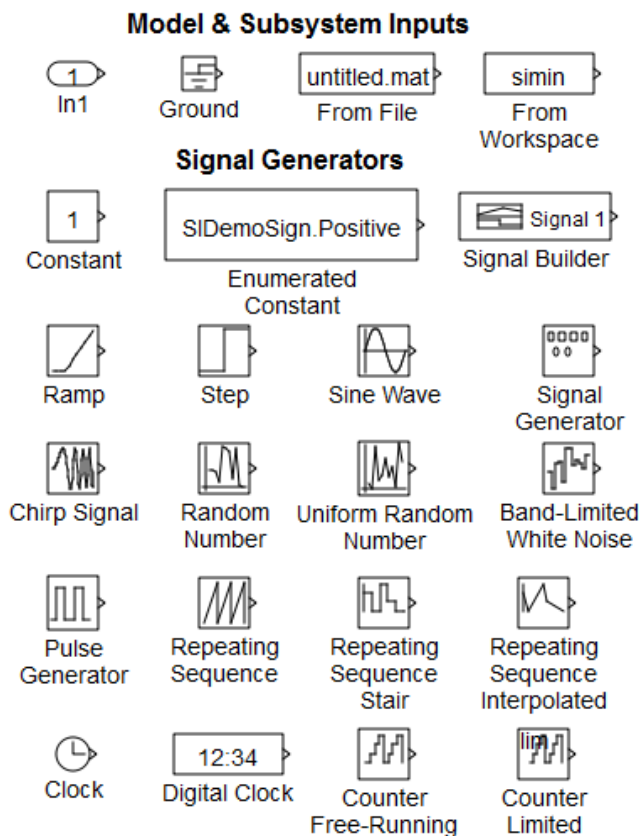


Рис. 2.1. Бібліотека джерел вхідних сигналів *Source*

- 1) постійні;
- 2) ступеневі;
- 3) лінійні;
- 4) нелінійні неперіодичні, що описуються аналітичними виразами;
- 5) гармонічні;
- 6) нелінійні періодичні;
- 7) випадкові.

Основу формування вхідних сигналів складають блоки бібліотеки джерел *Source*. У цю бібліотеку входить група блоків, що не мають вхідних портів, а мають тільки виходи. Їхні піктограми подані на рис. 2.1.

Блоки цієї бібліотеки можна розділити на 3 групи:

- 1) блоки, що не формують вхідних сигналів, а тільки помічають входи (блок *In*) або заземляють їх (блок *Ground*), щоб при старті моделювання в командному вікні *MATLAB* не виводилося попередження про наявність у моделі неприєднаного вхідного порту (Warning: Input port 1 of block ... is not connected...);

2) блоки, що можуть формувати тільки один вихідний сигнал (*Ramp*, *Clock*, *Digital Clock* і *Repeating Sequence*);

3) блоки, що можуть формувати як один, так і декілька вихідних сигналів (*Const*, *Step*, *Signal Generator*, *Sine Wave*, *Pulse Generator*, *Chirp Signal*, *Rundom Number*, *Uniform Rundom Number* і *Band Limited White Noise*).

Для того, щоб блоки 3-ої групи формували один вихідний сигнал (скалярний) у вікні настроювання блоку повинен бути встановленим прапорець *Interpret vector parameters as 1-D*. Якщо цей прапорець не встановлено, то блок може формувати одночасно кілька вихідних сигналів, об'єднаних в одинірний масив (вектор).

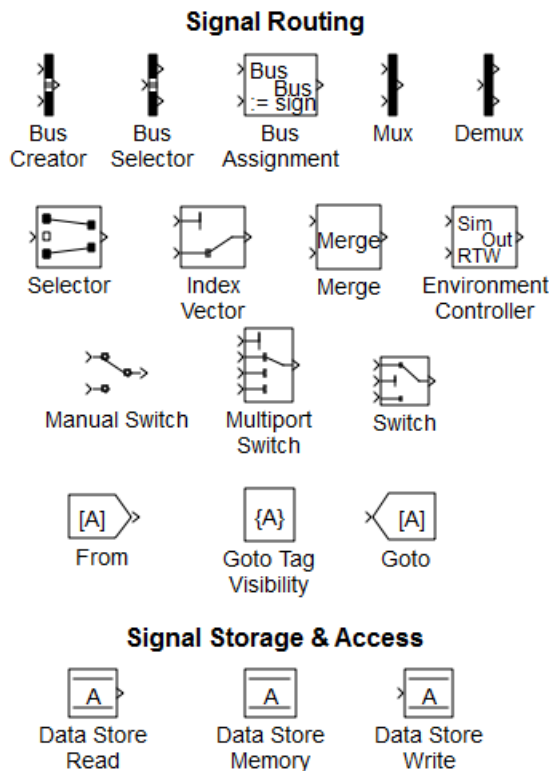


Рис. 2.2. Бібліотека *Signals Routing* (Маршрутизація сигналів)

Векторні параметри джерел можуть бути задані у вікнах настроювання блоків як рядками, так і стовпцями за правилами алгоритмічної мови системи *MATLAB*. Векторний сигнал може бути розділений на скалярні сигнали або на векторні сигнали з меншою кількістю компонентів за допомогою блока *Demux* бібліотеки *Signal Routing* (див. рис. 2.2). Зворотну операцію виконує блок *Mux*.

Такі джерела як *Step*, *Sine Wave*, *Pulse Generator*, *From File*, *From Workspace*, *Rundom Number* і *Uniform Rundom Number* можуть

змінювати значення своїх вихідних сигналів як неперервно, так і дискретно, тобто в моменти часу, кратні періоду дискретності, що задається параметром *Sample Time* (T_s). Для того, щоб джерело працювало у неперервному режимі, значення цього параметра необхідно покласти рівним нулю ($T_s=0$).

Якщо джерело повинне формувати дискретний сигнал, то в поле параметра *Sample Time* може вводиться одне додатне значення, що інтерпретується як період дискретності T_s , або два додатних значення, перше з яких сприймається як період дискретності T_s , а друге – як зсув (запізнення) *offset*. У цьому випадку зміна вихідних сигналів ланок будуть здійснюватися в дискретні моменти часу

$$t_d = nT_s + \text{offset}; \quad |\text{offset}| \leq T_s. \quad (2.1)$$

У дискретному режимі кожен блок працює так, як працював би цей же блок у неперервному режимі з приєднанням до його виходу екстраполятора нульового порядку (блок *Zero-Order Hold* бібліотеки *Discrete*) з тим же значенням параметра *Sample Time*. Якщо установити $T_s = -1$, то джерело працює в тій же режимі, що і блок, приєднаний до його виходу.

Виходом блоку **Constant (Константа)** є скалярне постійне значення c , визначене параметром *Constant value* і не залежне від часу $y = c = \text{const}$ або вектор констант у вигляді рядка $y = [c_1 \ c_2 \ \dots \ c_n]$ або стовпця $y = [c_1 \ c_2 \ \dots \ c_n]^T$.

Джерело **Step (Стрибок, Сходінка)** забезпечує стрибкоподібну зміну вихідного сигналу між двома постійними рівнями c_1 (*Initial value*) і c_2 (*Final value*) у заданий момент часу t_3 (*Step time*):

$$y(t) = \begin{cases} c_1 & \text{при } t \leq t_3, \\ c_2 & \text{при } t > t_3. \end{cases} \quad (2.2)$$

Декілька блоків *Step*, приєднаних до входів алгебраїчного суматора *Sum* з математичної бібліотеки *Math operations* (рис. 2.3) створюють багатосходишковий сигнал.

Блок **Sum (Суматор)** може мати круглу (*round*) або прямокутну (*rectangular*) форму, яка визначається параметром *Icon shape*.

Він підсумовує вхідні сигнали з відповідними знаками, які встановлюються в полі параметра *List of signs* (входи нумеруються зверху вниз для прямокутного блоку та проти годинної стрілки для круглого блоку). Комбінація

знаків “+”, “-” описує параметри кожного конкретного порту, де кількість портів відповідає кількості знаків, використаних у комбінації. Для пропуску позиції чергового порту у списку знаків використовують символ „|”.

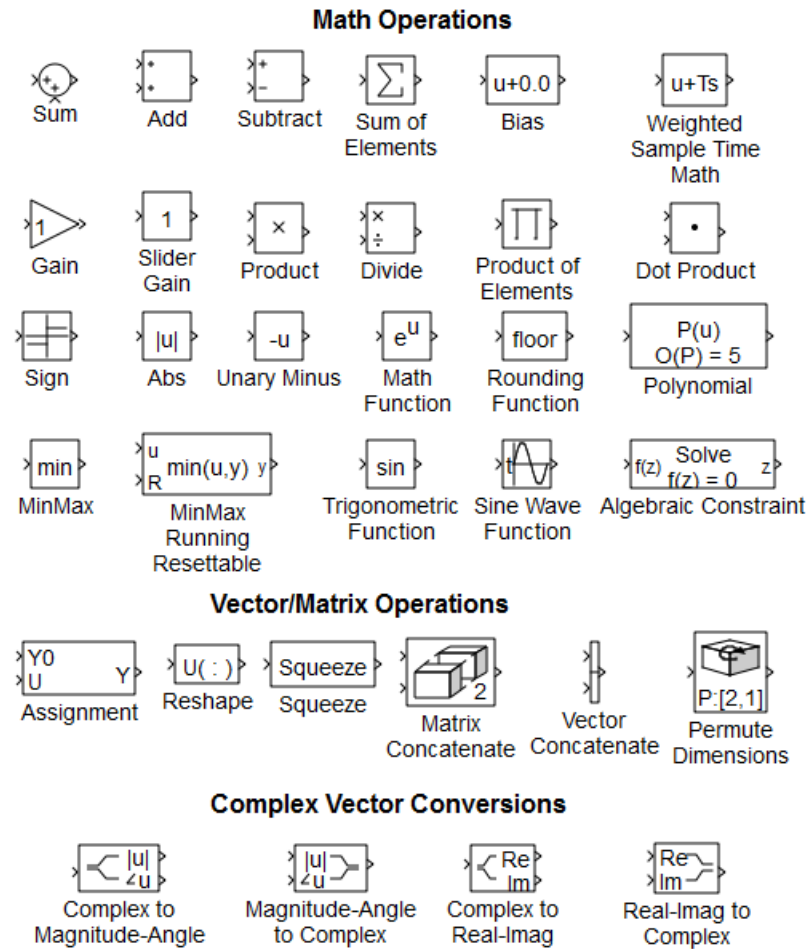


Рис. 2.3. Бібліотека математичних блоків *Math operations*

Якщо замість списку знаків визначити кількість входів, то усі вони будуть підсумовуючими. При одному вході блок підсумовує з відповідним знаком елементи вхідного вектора:

$$y = \text{sum}(u) = \sum_{i=1}^k u_i . \quad (2.2)$$

При цьому в піктограмі блоку відображається не знак вхідного сигналу, а символ Σ . Можливі різновиди блоку в залежності від значень його параметрів наведені на рис. 2.4.

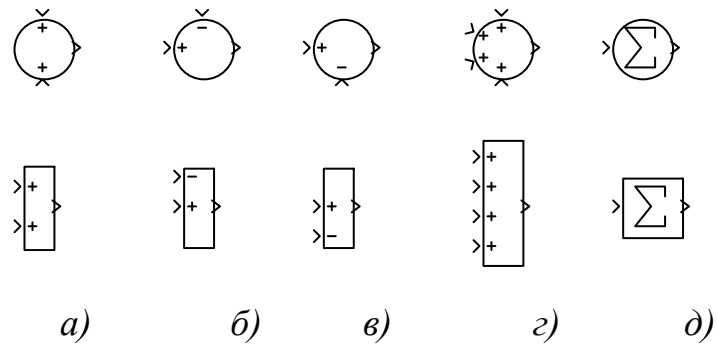


Рис. 2.4 – Блоки *Sum* при таких значеннях параметру *List of signs*:

а) ++; б) -+; в) |+-; г) 4; д) 1 або +

Блок **Clock (Годинник)** забезпечує відлік і відображення часу моделювання: $y(t) = t$. Параметр *Decimation*, що може приймати цілочисельні додатні значення d , має сенс тільки при включеному прапорці *Display time* і при використанні для рішення диференціальних рівнянь методів чисельного інтегрування з постійним кроком h . У цьому випадку в піктограмі блоку, що зображує прямокутну форму, відображаються по черзі чисельні значення часу моделювання, кратні $d \cdot h$.

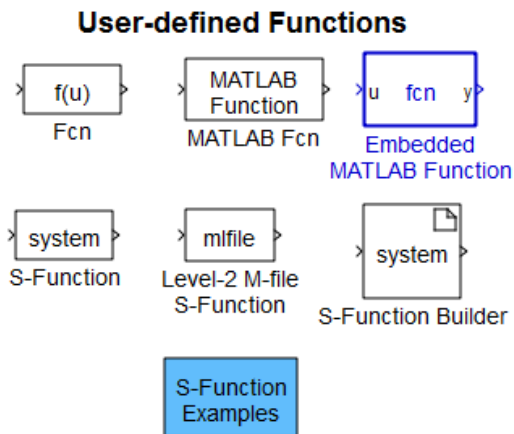


Рис. 2.5 – Бібліотека *User defined functions*

Цей блок укупі з блоками *Math Function*, *Trigonometric Function* і *Polynomial* бібліотеки *Math operations* (див. рис. 2.3), а також укупі з блоками *Fcn*, *MATLAB Fcn* бібліотеки *User defined functions*, наведеною на рис. 2.5, і блоком *Lookup Table* бібліотеки *Lookup Tables* використовують для формування сигналів з відповідним відомим законом їхньої зміни у функції часу.

Приєднанням *Годинника* до блоку *To Workspace (У пам'ять)* бібліотеки *Sinks* можна запам'ятати вектор часу з метою використання його згодом для побудови графіків перехідних процесів.

Блок **Math Function** може відтворювати елементарні одно- та двоаргументні функції від вхідних сигналів. Вибір функції здійснюється через меню. Він впливає на вигляд піктограми блоку. На рис. 2.6 наведені можливі різновидності цього блоку. Для наочності імена блоків замінені іменами функцій випадаючого меню.

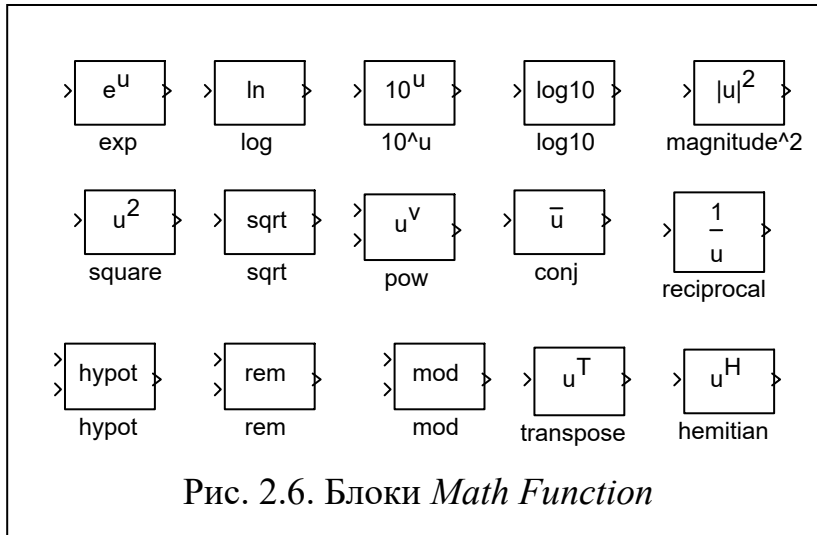


Рис. 2.6. Блоки **Math Function**

Блок **Trigonometric Function** аналогічним способом відтворює елементарні тригонометричні ($\sin$, $\cos$, $\tan$), зворотні тригонометричні (asin , acos , atan , $\operatorname{atan2}$), гіперболічні ($\sinh$, $\cosh$, $\tanh$), та зворотні гіперболічні функції (asinh , acosh , atanh) від вхідних сигналів.

Блок-функція **Polynomial (Поліном)** обчислює значення степеневого полінома, в якому незалежною змінною є вхідний сигнал, а вектор коефіцієнтів задається у полі параметра *Polynomial coefficients*.

Вихідний сигнал блоку **Fcn (Функція)** є аналітичною функцією, при будівництві якої можуть бути використані наступні компоненти:

- $u[i]$ або $u(i)$ – значення i -го вхідного сигналу (u без індексу еквівалентно $u[1]$);
- числа;
- операції $+$, $-$, $*$, $/$, $^$;
- дужки $()$;
- елементарні функції $\sin$, $\cos$, $\tan$, asin , acos , atan , $\sinh$, $\cosh$, $\tanh$, $\exp$, $\ln$, $\log10$, sqrt , floor , ceil , abs , $\operatorname{atan2}$, rem ;
- операції порівняння $==$, $\sim$, $>$, $<$, $>=$, $<=$;
- скалярні неіндексовані змінні.

Наприклад, $\sin(4*u).*\exp(-u/k)$, $\text{rem}(u[1], u[2])$, $u[1]*u[2]>=u[3]+u[4]$.

Блок **MATLAB Fcn (MATLAB-Функція)** обчислює задану *MATLAB*-функцію (стандартну або створену користувачем) від вхідного сигналу u (скалярного або векторного). Цей блок є пасивним двостороннім інтерфейсом між середовищами *Simulink* і *MATLAB*. Наприклад, якщо задана функція $\sin$, то вихідний сигнал y буде обчислюватися по формулі $y=\sin(u)$. Якщо вхідних сигналів декілька, то вони позначаються як $u(1)$, $u(2)$, $u(3)$ і т.д.

Блок *MATLAB Fcn* підтримує векторні входи і виходи, які перетворюються у скалярні блоком *Demux*, а навпаки – блоком *Mux*. Кількість входів повинна співпадати з кількістю вхідних аргументів *Matlab*-функції, а розмірність виходу (*Output width*) – з кількістю її вихідних аргументів. Якщо користувач задає параметр *Output width* рівним -1, то *Simulink* установлює розмірність вихідного сигналу рівною розмірності вхідного сигналу.

Блок **1D-Look-Up Table (Функціональний Перетворювач)** забезпечує кусочно-лінійну апроксимацію табличної функції, заданої векторами аргументів (*Breakpoints*) і значень (*Table data*) однакової довжини: $y = f(u)$;

$$y = \frac{y_i - y_{i-1}}{u_i - u_{i-1}} \cdot (u - u_{i-1}) + y_{i-1}, \quad (2.3)$$

$i=1\dots n$ – номери вузлових точок.

Вектор аргументів повинен бути монотонно зростаючим. Вихідні значення для значень входу, що попадають за межі заданих, знаходять за допомогою екстраполяції за першими (останніми) двома точками.

Піктограма блоку *Look-Up Table* відображає графік залежності вхід-вихід. При зміні значень параметрів у діалоговому вікні блоку графік автоматично перемальовується. Якщо ж параметр заданий перемінною, значення якої змінюється в робочому середовищі пакета *MATLAB*, то характеристика блоку змінюється відразу, але перемальовується тільки після відкриття його діалогового вікна.

Блок **Digital clock (Цифровий Годинник)** формує сигнал, який збігається з часом моделювання у моменти часу, кратні періоду дискретності *Sample*

Time. Між цими моментами вихідний сигнал має постійне значення, що дорівнює останньому обмірюваному часу моделювання.

Джерело ***Ramp (Рампа, Наклонна Площина)*** формує сигнал, що лінійно наростає або спадає від значення y_0 (*Initial output*), починаючи з моменту часу t (*Start time*) з коефіцієнтом пропорційності k (*Slope*):

$$y(t) = \begin{cases} y_0 & \text{при } t \leq t_S, \\ y_0 + kt & \text{при } t > t_S. \end{cases} \quad (2.4)$$

Блок ***Sine Wave (Синусоїда)*** може працювати в двох режимах (*Time-Based Mode* і *Sample-Based Mode*), котрі встановлюються у вікні налаштування блоку за допомогою випадаючого меню (*Parameters, Sine type*). У режимі *Time-Based* можливе формування як неперервного, так і дискретного вихідного сигналу. У неперервному режимі ($T_S=0$) блок генерує синусоїду у функції часу відповідно до рівняння:

$$y(t) = A \sin(\omega t + \varphi_0) + y_0, \quad (2.5)$$

де

A – амплітуда (*Amplitude*);

ω – кругова частота, рад/с (*Frequency*);

φ_0 – фазовий зсув, рад. (*Phase*);

y_0 – вертикальний зсув (*Bias*).

У дискретному *Time-Based* режимі ($T_S>0$) при обчисленні вихідного сигналу використовуються формули:

$$\begin{cases} \sin(t + T_S) = \sin(t)\cos(T_S) + \sin(T_S)\cos(t), \\ \cos(t + T_S) = \cos(t)\cos(T_S) - \sin(t)\sin(T_S), \end{cases} \quad (2.6)$$

які дозволяють формувати вихідне значення значно швидше, ніж у неперервному режимі. Недоліком цього методу є нагромадження похибок округлення, тому що для обчислення вихідного сигналу в кожний наступний момент часу використовується значення вихідного сигналу в попередній момент часу. Для ліквідації цього недоліку ланка *Sine Wave* передбачає можливість роботи в

режимі *Sample-Based*. При цьому значення вихідного сигналу обчислюється за формулою

$$y = A \sin(2\pi(k + o)/p) + y_0, \quad (2.7)$$

де

p – кількість розрахункових крапок на періоді хвилі синусоїди (*Samples per period*);

o – зсув (відносне фазове зрушення) сигналу (*Number of offset Samples*);

k – цілочисельний параметр, що приймає на періоді синусоїди значення $[0, 1, 2, \dots, p-1]$...

Для узгодження методів *Time-Based discrete* і *Sample-Based* необхідно забезпечити наступні співвідношення між їхніми параметрами:

$$p = \frac{2\pi}{\omega T_s}; \quad o = \frac{p\varphi_0}{2\pi}. \quad (2.8)$$

Ліквідуючи нагромадження похибок округлення, формула (2.7) має схований недолік, який виявляється при використанні цього блоку в підсистемі, що виконується за умовою (*Conditionally Executed Subsystem*). Цей недолік полягає в можливості неузгодженої роботи блоку *Sine Wave* з іншими блоками моделі після виконання підсистемою паузи з наступним поновленням її роботи. Отже, при виникненні потреби формування дискретної синусоїди в складі підсистеми, виконуваної за умовою, блок *Sine Wave* необхідно використовувати в режимі *Time-Based*.

Блок ***Puls Generator (Генератор Прямокутних Імпульсів)*** також може працювати в неперервному (*Time-Based*) або дискретному (*Sample-Based*) режимах, вибір одного з яких здійснюється за допомогою параметра *Pulse Type*. В обох режимах блок формує періодичний сигнал, що складається з однополярних прямокутних імпульсів. В неперервному режимі він має наступні параметри:

ht – висота імпульсу (*Amplitude*);

T – період в одиницях виміру часу (*Period(secs)*);

du – ширина імпульсу у відсотках від періоду (*Puls width (in % of period)*);

stt – час початку пульсацій (*Phase delay (secs)*).

У дискретному режимі у вікні налаштування блоку з'являється додатковий параметр *Sample Time*, а період і час початку пульсацій задаються у відносних одиницях (частках періоду дискретності – *Number of samples*).

Якщо при моделюванні обрано метод рішення диференціальних рівнянь з фіксованим кроком, то *Simulink* автоматично використовує блок *Puls Generator* у режимі *Sample-Based* з періодом дискретності, рівним кроку інтегрування. У цьому випадку крок інтегрування повинний бути обраний таким, щоб параметри T , $du*T/100$ і stt були кратні йому.

Блок ***Repeating Sequence (Повторювана Послідовність)*** виконує повторення циклу, заданого таблицею двох параметрів: *Time values* (t_v) – вектор часу і *Output values* (y_v) – вектор вихідних сигналів. Вектор часу повинен бути монотонно зростаючим. Різниця між значеннями його останнього і першого елементів визначає період вихідного сигналу: $T = t_v(end) - t_v(1)$. Блок перемальовує свою піктограму відповідно до вигляду вихідного сигналу. Він реалізований за допомогою замаскованої підсистеми зображеної на рис. 2.7.

Блок *Fcn* моделі рис. 2.7 визначає залишок від цілочисельного ділення часу моделювання на період, а блок *Look-Up Table* виконує кусочно-лінійну апроксимацію табличної функції, заданої параметрами *Time values* і *Output values*.

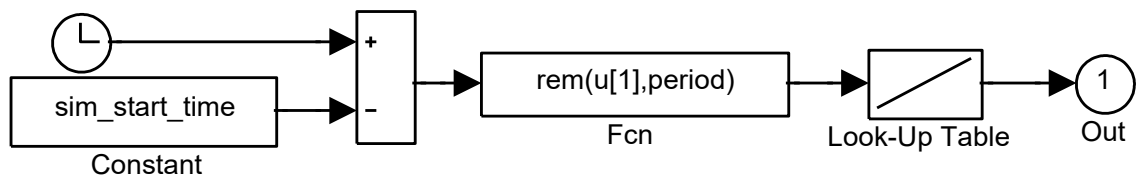


Рис. 2.7 – Модель макроблока *Repeating Sequence*

Джерело ***Chirp Signal (Синусоїда зі Змінною Частотою)***, реалізоване за допомогою замаскованої підсистеми рис. 2.8, формує синусоїду з одиничною амплітудою і частотою, що змінюється за лінійним законом:

$$y = \sin(At^2 + Bt), \quad (2.9)$$

де

$$B=2 \pi f_1, \quad A=2 \pi (f_2 - f_1) / T$$

за вхідними параметрами:

f_1 (*Initial frequency [Hz]*) – початкова частота [Гц];

T (*Target time [secs]*) – заданий час [сек];

f_2 (*Frequency at target time [Hz]*) – значення частоти [Гц] в момент часу T

Піктограма джерела зображує графік функції $y=\sin(t^2)$.

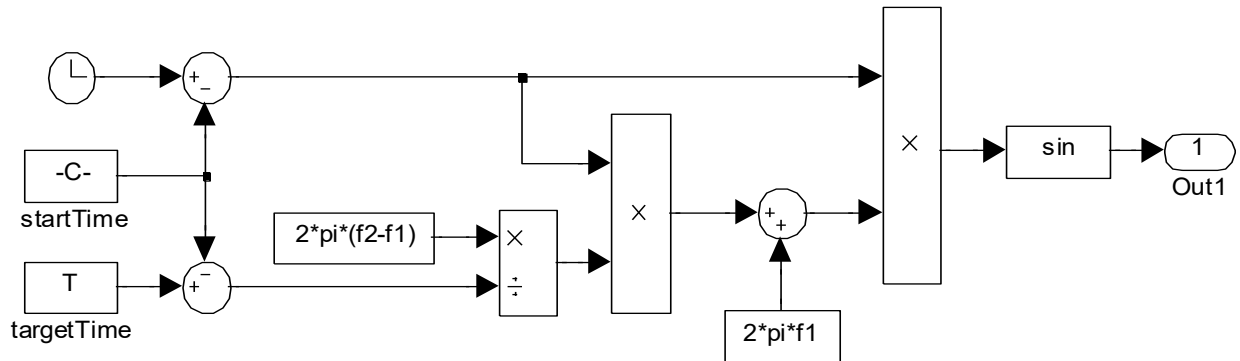


Рис. 2.8 – Модель блоку *Chirp Signal*

Комплексне джерело ***Signal Generator (Генератор Сигналів)*** генерує один з 4-х сигналів: синусоїду (*Sine*), прямокутний (*Square*) періодичний сигнал зі шпаруватістю 50%, пилкоподібний (*Sawtooth*) періодичний сигнал, а також випадковий сигнал (*Random*).

Тип сигналу вибирається за допомогою випадального меню (*Parameters, Wave form*). Чисельними параметрами являються амплітуда A (*Amplitude*) і частота (*Frequency*). Частота може задаватися як у Гц (*Hertz*), так і в рад/с (*rad/sec*) за допомогою параметра *Units*. Значення всіх сигналів змінюються від $+A$ до $-A$. Вихідні установки блоку можуть бути змінені протягом процесу моделювання, що особливо ефективно у сполученні з фіксацією результатів блоком *Scope (Осцилограф)*, коли реакцію системи на різні типи вхідного сигналу потрібно одержати швидко.

Випадковий сигнал створюється генератором випадкових чисел алгоритмічної мови *C*.

Джерела випадкових чисел з нормальним (***Random Number***) та рівномірним (***Uniform Random Number***) розподілами генерують відповідно нор-

мально розподілений (Гаусовський) і рівномірно розподілений випадкові сигнали для деяких заданих стартових чисел *Initial seed*, ініціалізуючих генератори випадкових чисел.

Специфічними параметрами ланки *Random Number* є середнє значення сигналу *Mean* і середньоквадратичне відхилення *Variance*, а ланки *Uniform Random Number* – мінімальне *Minimum* і максимальне *Maximum* значення випадкового сигналу. Обидва блоки можуть працювати як в неперервному, так і у дискретному режимах.

Джерело ***Band-Limited White Noise (Білий Шум з Екстраполяцією)*** забезпечує формування “білого шуму” для неперервних систем за допомогою послідовно з'єднаних блоків *Random Number*, що працює в дискретному режимі, та *Gain* (*Пропорційна ланка*) на основі таких параметрів:

- Cov* (*Noise Power*) – потужність шуму (коваріація);
- T_s* (*Sample Time*) – період дискретності;
- Seed* – стартове число генератора випадкових чисел.

Коефіцієнт підсилення ланки *Gain* обчислюється за формулою

$$k = \sqrt{Cov} / \sqrt{T_s} . \quad (2.10)$$

Вектори *Noise Power* і *Seed* можуть бути однієї довжини. Для більш швидкого протікання процесу моделювання необхідно параметр *Sample Time* установлювати максимально великим, узгоджуючи однак його величину з мінімальної сталою часу системи.

Блок ***From Workspace (З пам'яті)*** забезпечує читання даних з матриці, що повинна мати два або більш стовпці. Перший стовпець повинний містити монотонно зростаючі значення часу, а кожен додатковий стовпець – табличні значення функцій часу:

$$\begin{bmatrix} t_1 & y_1 & z_1 & \dots & w_1 \\ t_2 & y_2 & z_2 & \dots & w_2 \\ \dots & \dots & \dots & \dots & \dots \\ t_k & y_k & z_k & \dots & w_k \end{bmatrix} . \quad (2.11)$$

Наприклад, для того, щоб блок *From Workspace* сформував на інтервалі $t \in [0, 2\pi]$ сигнал $y(t) = \sin(t) + \cos(t^2)$, у середовищі *MATLAB* до моделювання необхідно виконати послідовність операцій

```
Time = (0 : pi/50 : 2*pi)';
Out=sin(Time)+cos(Time.^2);
D=[Time Out];
```

і установити в поле параметра *Data* у вікні налаштування блоку *From Workspace* ім'я матриці *D*.

Значення часу використовуються блоком для обчислення вихідного сигналу за допомогою лінійної інтерполяції або екстраполяції.

Піктограма блоку відображає ім'я матриці, з якої зчитуються його дані.

За допомогою випадаючого меню *Form output after final data value by* можна вибрати один з чотирьох способів зміни вихідного сигналу після перевищення часом моделювання значення останнього елемента першого стовпця матриці даних (максимального значення аргументу табличної функції):

- Extrapolation* — лінійна екстраполяція;
- Setting To Zero* — установка в нуль;
- Holding Final Value* — затримка останнього табличного значення;
- Cyclic Repetition* — циклічне повторення.

Для використання першого способу прапорець інтерполяції *Interpolate data* повинний бути обов'язково встановлений, а для останнього способу — скинутий. Для інших способів стан цього прапорця не має значення. При останньому способі (*Cyclic Repetition*) дані, що зчитуються, повинні мати формат не матриці, а структури. Наприклад, щоб розглянутий у попередньому прикладі сигнал циклічно повторювався при $t > 2\pi$, необхідно сформувати структуру даних у такий спосіб:

```
D.Time = (0 : pi/50 : 2*pi)';
D.signals(1).Out=sin (Time)+cos (Time.^2);
D.signals(1).demensions = 1;
```

Останній параметр визначає розмірність вихідного сигналу.

Джерело ***From File (З Файлу)*** забезпечує читання даних з файлу, чиє ім'я занесено в поле параметра *Filename* і відображається в піктограмі блоку.

Файл даних повинен мати розширенням *mat (*.mat)* і містити матрицю з двома або більш рядками і мінімум двома стовпцями. Перший рядок інтерпретується як монотонно зростаючі значення часу, а інші рядки – як відповідні цим моментам часу значення вихідних сигналів джерела:

$$\begin{bmatrix} t_1 & t_2 & \dots & t_k \\ y_1 & y_2 & \dots & y_k \\ z_1 & z_2 & \dots & z_k \\ \dots & \dots & \dots & \dots \end{bmatrix}. \quad (2.12)$$

Запис даних у *mat*-файл у пакеті *MATLAB* виконує команда

`save FileName var1 var2 ...`

наприклад,

`save data1 D`

збереже значення змінної *D* у файлі *data1.mat*. Для визначення вихідних сигналів у моменти часу, відсутні у файлі даних, використовується лінійна інтерполяція або екстраполяція. Файл даних для блоку *From File* може бути сформований блоком *To File* при моделюванні іншої системи.

Блок ***In (Вхідний Порт)*** помічає входи моделі. Якщо у вкладці *Workspace I/O* вікна *Simlation Parameters* установити прапорець *Input* функції *Load from Workspace*, то зовнішній вхідний сигнал можна задавати в полі параметру *Input* у такий же спосіб, як за допомогою блоку *From Workspace*.

Основним параметром блоку *In* є номер порту (*Port number*), що відображається в його піктограмі. Усі порти в межах одного вікна повинні мати послідовну нумерацію. Номера портів формуються автоматично в порядку їхньої появи у вікні. При видаленні порту з вікна порти, що залишилися в ньому, перенумеровуються таким чином, щоб у послідовності номерів не було пропусків. Номера портів можна змінювати і вручну.

Параметром *Port dimensions* можна визначити розмірність вхідного сигналу. Значення за замовчуванням (-1) відповідає автоматичній установці цього параметра.

Тип сигналу (*real*, *complex*), тип представлення даних (*double*, *single*, *int8*, ..., *boolean*), прапорець *Interpolate data*, а також параметр *Sample time* мають сенс тільки при використанні зовнішніх джерел у моделях верхнього рівня.

2.2 Основні засоби реєстрації та візуалізації сигналів

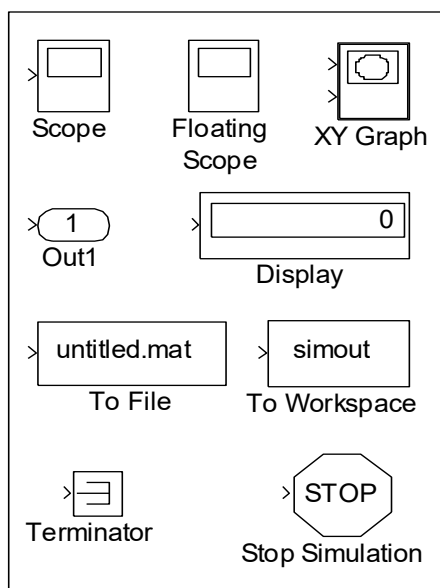


Рис. 2.9 – Бібліотека вихідних блоків *Sinks*

Для реєстрації та візуалізації сигналів у *Simulink* існує бібліотека *Sinks*, блоки якої подані на рис. 2.9.

Більшість з них призначені для запам'ятовування результатів моделювання (*To File*, *To Workspace*), а також для їх графічного (*Scope*, *Floating Scope*, *XY Graph*) або чисельного (*Display*) відображення. Усі вони мають тільки входи і не мають виходів.

Блок *To Workspace* (*У Пам'ять*) запам'ятовує дані, що надходять на його вхідний порт у перемінній з ім'ям, завданим у поле параметра *Variable name*.

Параметр *Decimation* (*Проріджування*) визначає дискретність запам'ятовування або відображення інформації і може приймати тільки додатні цілочисельні значення d , що вказують через скількох кроків чисельного інтегрування варто фіксувати результати моделювання. При $d=1$ (значення за замовчуванням) блоком запам'ятовується інформація, отримана на кожному кроці ЧІ, а при $d=5$ – тільки інформація, отримана на кожному п'ятому кроці. При інтегруванні з постійним кроком параметр d – це відношення кроку фіксації до кроку інтегрування: $h_{out} = dh_{int}$.

Якщо в процесі моделювання для рішення диференціальних рівнянь обрано метод ЧІ зі змінним кроком, а крок фіксації повинний бути постійним, то для дискретизації виведення замість параметра *Decimation* варто використовувати параметр *Sample Time*, обираючи його значення рівним бажаному кроку реєстрації: $T_s = h_{out}$. При $T_s = -1$ блок успадковує період дискретності від блоку, вихідні сигнали якого він реєструє.

Параметри *Limit data point to last* і *Save format* визначають максимальну кількість точок для збереження реєструемого сигналу (відлік ведеться від останньої розрахованої точки) і формат запам'ятовування даних. Для того, щоб не втратити інформацію, значення параметра *Limit data point to last* можна установити рівним ∞ (константа *Inf*).

Інформація може бути збережена в наступних форматах: *Structure with time* (Структура з часом моделювання), *Structure* (Структура без часу моделювання) і *Array* (Масив).

Найпростішим типом даних є *Array*. У цьому випадку кожен сигнал записується в окремий стовпець матриці. Один рядок матриці містить стан усіх сигналів у конкретний момент часу. Кількість рядків при $d=1$ дорівнює числу кроків моделювання. Якщо дані збережені в матриці y , то виділити з неї j -ий сигнал можна операцією $y(:,j)$, а всі сигнали в i -й зареєстрований момент часу – операцією $y(i,:)$.

Якщо дані збережені у змінній y , що має формат *Structure with time*, то вектор-стовпець часу визначається звертанням до поля *time* змінної y ($y.time$), звертання $y.blockName$ формує шлях до блоку (ім'я-моделі / ім'я підсистеми / ім'я-блоку), звертання $y.signals.dimensions$ – кількість вхідних сигналів, звертання $y.signals.label$ – мітку лінії зв'язку, а звертання $y.signals.values$ – чисельні значення матриці вхідних сигналів, організованої так само, як при використанні формату *Array*. Структура без часу (*Structure*), відрізняється від структури з часом тим, що її поле *time* має значення порожньої матриці $[]$.

Усякий раз при старті процесу моделювання дані, що зберігаються, об'єднуються, так що цей блок не може об'єднати результати кількох послідов-

них розрахунків. Результати не доступні в робочому середовищі доти, поки не закінчено або не зупинено розрахунок перехідних процесів.

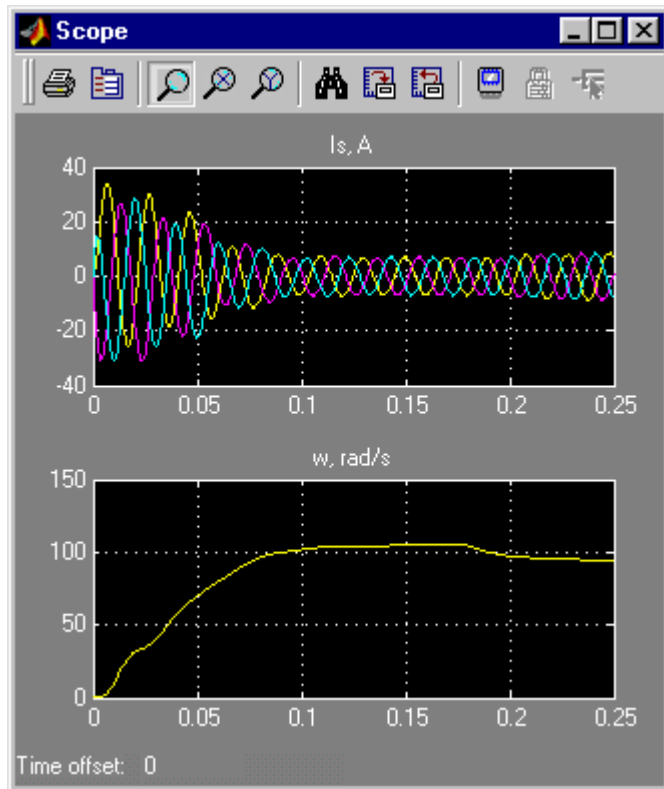


Рис. 2.10 – Вікно блока *Scope*

Блок *Scope* (*Осцилограф*) у процесі моделювання відображає вихідні сигнали приєднаних до нього блоків. Для того, щоб побачити графіки перехідних процесів, необхідно попередньо відкрити його вікно (див. рис. 2.10), що поряд з полем виведення графіків (на відміну від попередніх версій їх може бути декілька) містить наступні «кнопки» (нумерація зліва направо):

- 1 – *Print* (друкування);
- 2 – *Parameters* (настроювання параметрів);
- 3 – *Zoom* (рівномірна зміна масштабу по обох осях);
- 4 – *Zoom X-axis* (зміна масштабу по осі *X*);
- 5 – *Zoom Y-axis* (зміна масштабу по осі *Y*);
- 6 – *Autoscale* (автоматичне масштабування);
- 7 – *Save current axes settings* (запам'ятовування системи координат);
- 8 – *Restore saved axes settings* (відновлення системи координат, наприклад, після зміни масштабу однієї з клавiш *Zoom*);
- 9 – *Floating Scope* (плаваючий осцилограф);
- 10 – *Lock/Unlock axes selection* (заблокувати/розблокувати вибір ліній зв'язку для плаваючого осцилографа);
- 11 – *Signal Selection* (вибір сигналів для плаваючого осцилографа).

Звичайно після закінчення процесу моделювання для того, щоб побачити графіки в нормальному масштабі, натискають кнопку *Autoscale*, а потім запам'ятовують діапазони систем координат кнопкою *Save current axes settings*.

Якщо результати автоматичного масштабування за якимись причинами не влаштовують користувача, то він може змінити межі координатних осей графіків вручну за допомогою кнопок 3-5 візуально або установкою границь системи координат у чисельному вигляді.

Масштабування зображення в обраному виміру кнопками можна виконувати двома способами. При першому способі після активізації відповідної кнопки курсор миші підводять до бажаної ділянки графіка і клацають на ньому лівою клавішею. При кожному натисканні масштаб буде збільшуватися, що приведе до відображення у вікні всі меншого і меншого фрагмента графіка. При другому способі фрагмент графіка, що хочуть збільшити, виділяють за допомогою курсору. Повернення до результатів попереднього масштабування виконується командою контекстуального меню *Zoom out*, а повернення до вихідного масштабу – кнопками *Autoscale* або *Restore saved axes settings*.

Для установки меж графіка по осі ординат у чисельному вигляді необхідно викликати контекстуальне меню щикликом правої кнопки миші в площині системи координат, вибрати з нього функцію *Axes Properties* і у вікні, що відкрилося, установити *параметри Y-min і Y-max*. У цьому ж вікні можна установити заголовок системи координат (параметр *Title*), замінивши коментар *%<SignalLabel>* бажаним текстом або вставивши цей текст перед коментарем.

Діапазон часу, що є спільним для всіх систем координат, можна змінити у вікні *Parameters*, що відкривається однойменною кнопкою й утримує 2 вкладки: *General*, *History* і *Style*.

За допомогою вкладки *General* (див. рис. 2.11) установлюються наступні параметри:

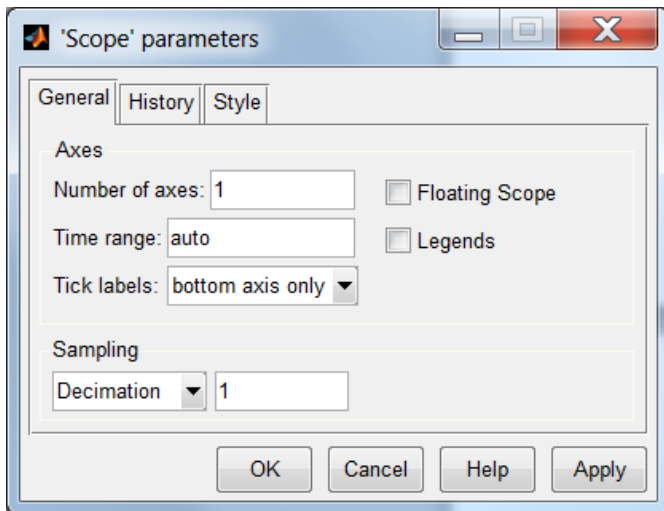


Рис. 2.11. Вкладка *General* вікна параметрів блоку *Scope*

значення часу виводяться тільки в самій нижній системі координат) і *none* (значення міток не виводяться, і графіки займають максимально можливу площу вікна);

- прапорці *Floating Scope* та *Legend*;
- параметри *Decimation* або *Sample Time*, що визначають дискретність відображення інформації (див. опис блока *To Workspace*).

За допомогою вкладки *History* (див. рис.2.12) назначаються параметри *Limit data point to last*, *Variable Name* і *Save format*. Два останніх параметри

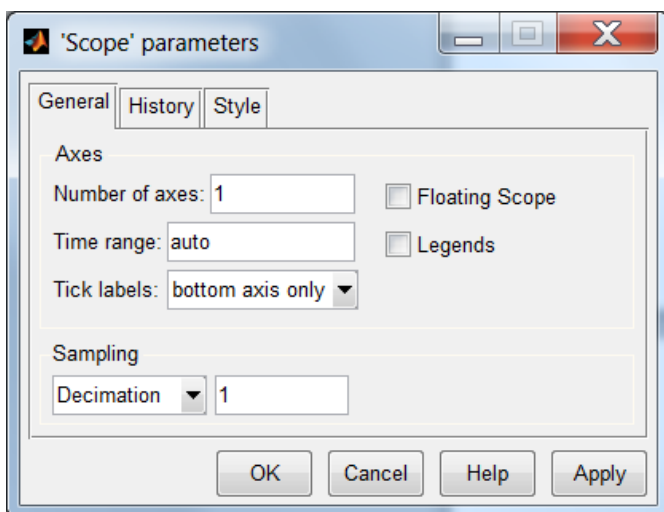


Рис. 2.12. Вкладка *History* вікна параметрів блоку *Scope*

- *Number of axes* – кількість систем координат, що визначає і кількість вхідних портів;
- *Time range* – діапазон часу;
- *Tick labels* – режим візуалізації чисельних значень міток координатної сітки, що може приймати значення *all* – значення часу виводяться в усіх підвікнах (системах координат), *bottom axis only* –

значення часу виводяться тільки в самій нижній системі координат) і *none* (значення міток не виводяться, і графіки займають максимально можливу площу вікна);

активізуються тільки при встановленому прапорці *Save data to workspace* (зберегти дані в оперативній пам'яті). При установці цього прапорця блок *Scope* починає виконувати, крім власних функцій, функції ланок *To Workspace* або *Out*. За замовчуванням для запису даних пропонується змінна з ім'ям *ScopeData*

у форматі *Structure with time*.

При виборі формату *Array* у перший стовбець матриці *ScopeData* записується час моделювання, а у інші – значення сигналів, приєднаних до осцилографа. Тому при побудові усіх графіків перехідних процесів в одному вікні в цьому випадку можна застосовувати оператор

```
plot (ScopeData (:, 1) , ScopeData (:, 2:end) )
```

Формат *Array* не можна використовувати для збереження інформації багатоканальним *Осцилографом*. Поле *signals* змінної *ScopeData* у цьому випадку є вектором структур, з кількістю елементів, рівним числу портів (каналів) *Осцилографа*. На додаток до полів *dimensions*, *values* і *label* кожен елемент змінної *signals* має ще поля *title* і *plotStyle*, які містять інформацію про заголовки графіків і стилі їхнього зображення.

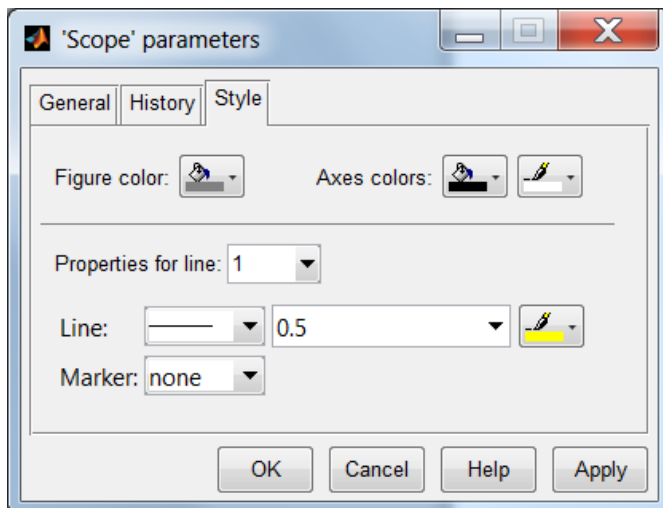


Рис. 2.13. Вкладка *Style* вікна параметрів блоку *Scope*

маркерів точок графіка.

Останні три параметри встановлюються окремо для кожного графіка, вибір якого здійснюється параметром *Parameters for line*:

Параметри *Осцилографа* можна редагувати в процесі моделювання.

Розмір і пропорції вікна *Scope* можна змінювати довільно.

Осцилограф, зображений на рис. 2.10, має 2 вхідних порти, на перший з яких надходить векторний сигнал, а на другий - скалярний. Скалярний сиг-

За допомогою вкладки *Style*

(рис. 2.13) назначаються такі параметри:

- *Figure Color* – колір графічної фігури;
- *Axes Color* – колір системи координат (колір фону і колір ліній градуїровки);
- *Line* – стиль зображення ліній, їх товщина та колір;
- *Marker* – стиль зображення маркерів точок графіка.

нал зображується завжди жовтим кольором, а для складових векторного сигналу використовуються за замовчанням повторювані циклічно 6 кольорів: *жовтий, малиновий, блакитний, червоний, зелений, синій*. Фон зображення за замовчанням *чорний*.

Блок *Floating Scope (Плаваючий Осцилограф)* можна взяти з бібліотеки або переключити звичайний осцилограф *Scope* у режим, що плаває, відповідною кнопкою або установкою відповідного прапорця.

Floating Scope не має входних портів. Підключення до нього реєструючих сигналів можна виконати двома способами. Перший спосіб полягає в активізації однієї із систем координат *Плаваючого Осцилографа* з наступним вибором однієї або кількох ліній зв'язку моделі, як це описано в розділі 2. Активізація системи координат виконується щигликом лівої кнопки миші в поле цієї системи, що приводить до автоматичного вмикання кнопки *Lock/Unlock axes selection* у стан *Unlock* і до облямівки активної системи координат синьою смугою. При другому способі вибір сигналів, що підключаються, здійснюється у вікні *Signal Selector*, що відкривається кнопкою 11. *Signal Selector*, завдяки наявності в ньому засобів навігації, дозволяє вибрати будь-які сигнали системи, включаючи внутрішні сигнали закритих підсистем.

На відміну від блоку *Scope*, плаваючий осцилограф не має буфера для збереження зареєстрованих даних. Тому для нього не можливі режими масштабування графіків (кнопки 3-6 не працюють).

Перевагою плаваючого осцилографа є економія оперативної пам'яті.

Блок *XY Graph (XY-графобудівник)* призначено для побудови фазових портретів, тобто для зображення одного з запам'ятованих сигналів у функції іншого (а не у функції часу). Його входними параметрами є координати меж графіка : *x-min*, *x-max*, *y-min* і *y-max*, а також параметр дискретизації *Sample time*.

Обидва входи блоку є скалярними. При моделюванні *Simulink* автоматично відкриває графічне вікно *MATLAB* і динамічно відображає в ньому задану користувачем графічну залежність.

Графобудівник створений на основі S -функції *sfunxy*. Для демонстрації роботи цього блоку можна з командного рядка завантажити демонстраційну модель *lorenzs*.

Реєстратор ***To File (У Файл)*** записує часи моделювання і вхідні сигнали в *mat-файл*, з ім'ям *File Name*, який має структуру матриці, що містить у першому рядку вектор часу, а в інших рядках відповідні йому вектори вхідних сигналів. Ім'я матриці задається параметром *Variable name*. Цей формат ідентичний формату блоку *From File* (див. формулу (2.11)), що забезпечує сумісність файлів. Для дискретизації сигналів так само, як у блоці *To Workspace*, використовуються параметри *Decimation* і/або *Sample time*.

При кожному старті процесу моделювання файл даних оновлюється.

Якщо якісь частини системи (наприклад, різні пристрої, що задають) працюють незалежно один від одного, то результати моделювання цих підсистем можна записати у файли і зчитувати ці данні з файлів при моделюванні частини системи, що залишилася. При багаторазовому моделюванні це дозволить скоротити час розрахунку перехідних процесів.

Цей блок також рекомендується використовувати в тому випадку, коли для запам'ятовування результатів моделювання не вистачає оперативної пам'яті або коли моделювання займає занадто багато часу. Отриманий файл результатів моделювання можна потім використовувати для побудови графіків за допомогою послідовно з'єднаних блоків *From File* і одного з пристроїв, що реєструють, наприклад, *Scope*.

Піктограма блоку відображає ім'я файлу, у який записуються матричні дані.

Матриця, записана у файл, стає доступною в середовищі *Matlab* після завантаження файлу командою *load FileName* у вигляді значення змінної, ім'я якої визначено в полі параметру *Variable name*.

Блоками ***Outport (Вихідний Порт)*** відзначають виходи моделей і підсистем.

У підсистемах вони необхідні для зв'язку підсистеми з моделлю більш високого рівня, а в моделях вищого рівня – для використання зовнішнього вихідного сигналу за допомогою опцій функції *sim* (при моделюванні з командного рядка) або за допомогою функції *Save to workspace* вікна *Simulation Parameters* (вкладка *Workspace I/O*) при встановленому прапорці *Output*, а також для інформування додатків, призначених для аналізу і синтезу систем автоматичного керування (наприклад, *Control Toolbox*) про можливі точки знімання вихідних сигналів.

Нумерація вихідних портів і їхнє відображення в піктограмі згорнутої підсистеми відбувається точно також, як і для вхідних портів *In*.

Скалярні і векторні сигнали, що надходять на вихідні порти можуть мати як дійсні, так і комплексні значення.

Параметри *Output when disabled* і *Initial output* мають значення тільки в підсистемах, виконуваних за умовою.

Блок ***Display*** (***Дисплей***) застосовується для виведення чисельних значень сигналів у заданому форматі, що вибирається за допомогою параметра *Format* і може приймати наступні значення:

- short* – 4 значущі цифри у форматі з фіксованою крапкою;
- long* – 14 значущих цифр у форматі з фіксованою крапкою;
- short-e* – 5 значущих цифр у мантисі чисел із плаваючою крапкою;
- long-e* – 16 значущих цифр у мантисі чисел із плаваючою крапкою.

Скалярні і векторні сигнали, що надходять до блоку, можуть мати як дійсні, так і комплексні значення.

Якщо розмір блоку *Display* малий для відображення всієї інформації, то в його правому нижньому куті з'являється чорний трикутник, що вказує, у якому напрямку варто розтягти піктограму блоку.

Так само, як і *Осцилограф*, *Дисплей* може працювати в плаваючому режимі. Для цього у вікні параметрів блоку необхідно установити прапор *Floating display*, а у вкладці *Advanced* вікна *Simulation parameters* змінити

стан *Optimisations-on* її *Signal storage reuse* (повторне використання запам'ятованого сигналу) з *on* на *off*.

Дискретизація сигналів виконується так само, як і в блоку *To Workspace*.

Блок ***Terminator (Заглушка)*** приєднується до виходів блоків, не зв'язаних з іншими блоками для запобігання висновку попереджувального повідомлення про наявність у моделі не приєднаних виходів.

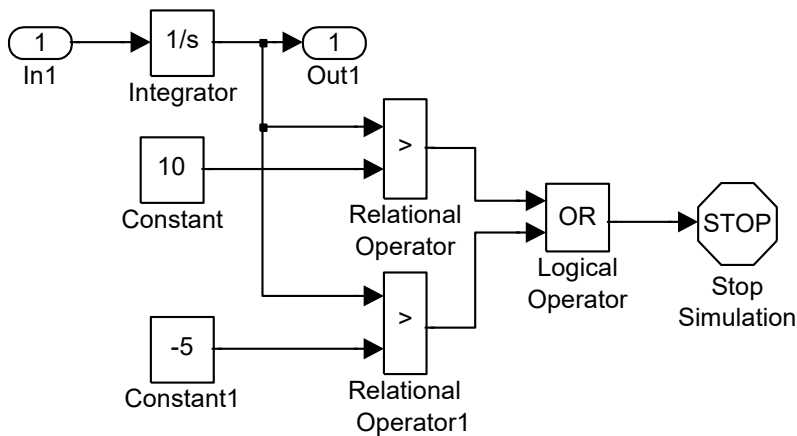


Рис. 2.11 – Модель із зупинкою моделювання за умовою

Блок ***Stop Simulation***

(Зупинка Моделювання)

зупиняє моделювання при ненульовому скалярному вхідному сигналі. У випадку векторного вхідного сигналу досить, щоб хоча б один з його компонентів став ненульовим. Оскільки в *MATLAB* результатом

логічних операцій і операцій порівняння є 1 (істина) або 0 (неправда), то вхідний сигнал блоку *Stop* зручно формувати за допомогою блоків *Relational Operator* і (або) *Logical Operator*. Наприклад, у приведеній на рис. 2.11 моделі процес моделювання перерветься, якщо вихідний сигнал інтегратора стане більш 10 або менше -5.

2.3 Завдання

1) Ознайомтеся з блоком *Step*. Сформууйте довільну (не періодичну) послідовність із 3-5 стрибкоподібних сигналів.

2) Ознайомтеся з блоком *Look-up Table*. Сформууйте довільну кусково-лінійну функцію часу із 4-6 ділянок.

3) Створити модель, яка формує синусоїду $s(t)$ з одиничною амплітудою та одиничним періодом блоком *Sine Wave*, прямокутний періодичний сигнал $p(t)$ з амплітудою 0.8, періодом 1.2с та відносною тривалістю імпульсу 70% блоком *Pulse Generator* і сигнал $k(t) = \sqrt{t}$ послідовним з'єднанням блоків *Clock* та *Sqrt*. Зафіксувати ці сигнали блоками бібліотеки *Sinks* у такі способи:

- трьома блоками *Scope*,
- одним блоком *Scope* з трьома системами координат,
- одним блоком *Scope* з однією системами координат,
- блоком *Floating Scope*,
- одним (спільним) та трьома (окремими для кожного сигналу) блоками *Out*;
- одним та трьома блоками *To Workspace*,
- блоком *XY Graph*,
- блоками *Display*;

використати одержану інформацію для побудови графіків $s(t)$, $p(t)$, $k(t)$ та $s(k)$.

4) Створити модель, що складається з блоків бібліотеки *Sources*, приєднаних через ключовий елемент *Multiport Switch* до блоків бібліотеки *Sinks*; ознайомитися з принципами роботи перелічених блоків, змінюючи їхні параметри.

5) Сформувані вхідні сигнали, подані на рис. 2.12, за даними табл. 2.1.

6) Сформувані вхідні сигнали за формулами, поданими у табл. 2.1. ($U = f(t)$).

2.4 Порядок виконання роботи

- 1) Ознайомтеся з блоком *Step*. Виконайте пункт 1 завдання.
- 2) Ознайомтеся з блоком *Look-up Table*. Виконайте пункт 2 завдання.
- 3) Ознайомтеся з основними способами реєстрації та візуалізації вихідних сигналів. Виконайте пункт 3 завдання.

Таблиця 2.1

№ вар	U_m	T	T_I	№ графіка	$U = f(t), t \in [0, 10]$
1	100	1	0,2	а, б, д	$U = 150 \sin(60t + \pi/10)$
2	100	1	0,4	а, в, є	$U = 125 \cos(50t - \pi/12)$
3	100	1	0,6	а, г, і	$U = 100 \sin(10t) e^{-t/4}$
4	150	2	0,5	а, д, з	$U = 100 e^{-t/4}$
5	150	2	0,8	ж, б, д	$U = 200 \sqrt{t}$
6	150	2	0,75	ж, б, є	$U = 100 \sqrt[3]{t}$
7	200	0,5	0,1	ж, в, і	$U = \frac{100}{t + 0.1}$
8	200	0,5	0,2	ж, г, д	$U = \frac{150t}{t + 1}$
9	200	0,5	0,15	ж, г, є	$U = 10t^2 + 50t + 4$
10	220	0,1	0,01	ж, г, і	$U = 100 \sin(\lg(t + 1))$
11	220	0,1	0,02	а, б, є	$U = 200 \cos(\ln(t + 1))$
12	220	0,1	0,04	а, б, і	$U = 150 \frac{1}{1 - 0.5 \sin t}$
13	250	10	1	а, г, д	$U = 100 \frac{t^2}{1 + t^3}$
14	250	10	2	а, г, є	$U = 200 \cos^2(t) \cos(2t)$
15	250	10	4	а, з, і	$U = 200 \sin^2(t) \sin(2t)$
16	75	5	1	а, з, є	$U = e^{\sin 2t} \sin t$
17	75	5	0,5	а, д, і	$U = 200 \sqrt[3]{t} \cdot \sin(3t)$
18	75	5	1,5	4, є, ж	$U = 100 \sqrt[2]{t} \cdot \cos(2t)$
19	110	2	0,1	а, г, і	$U = 5 \cos(2t) \cdot e^{-\frac{t}{2}}$
20	110	2	0,15	б, д, ж	$U = 3 \sin(4t) \cdot e^{-\frac{t}{4}}$

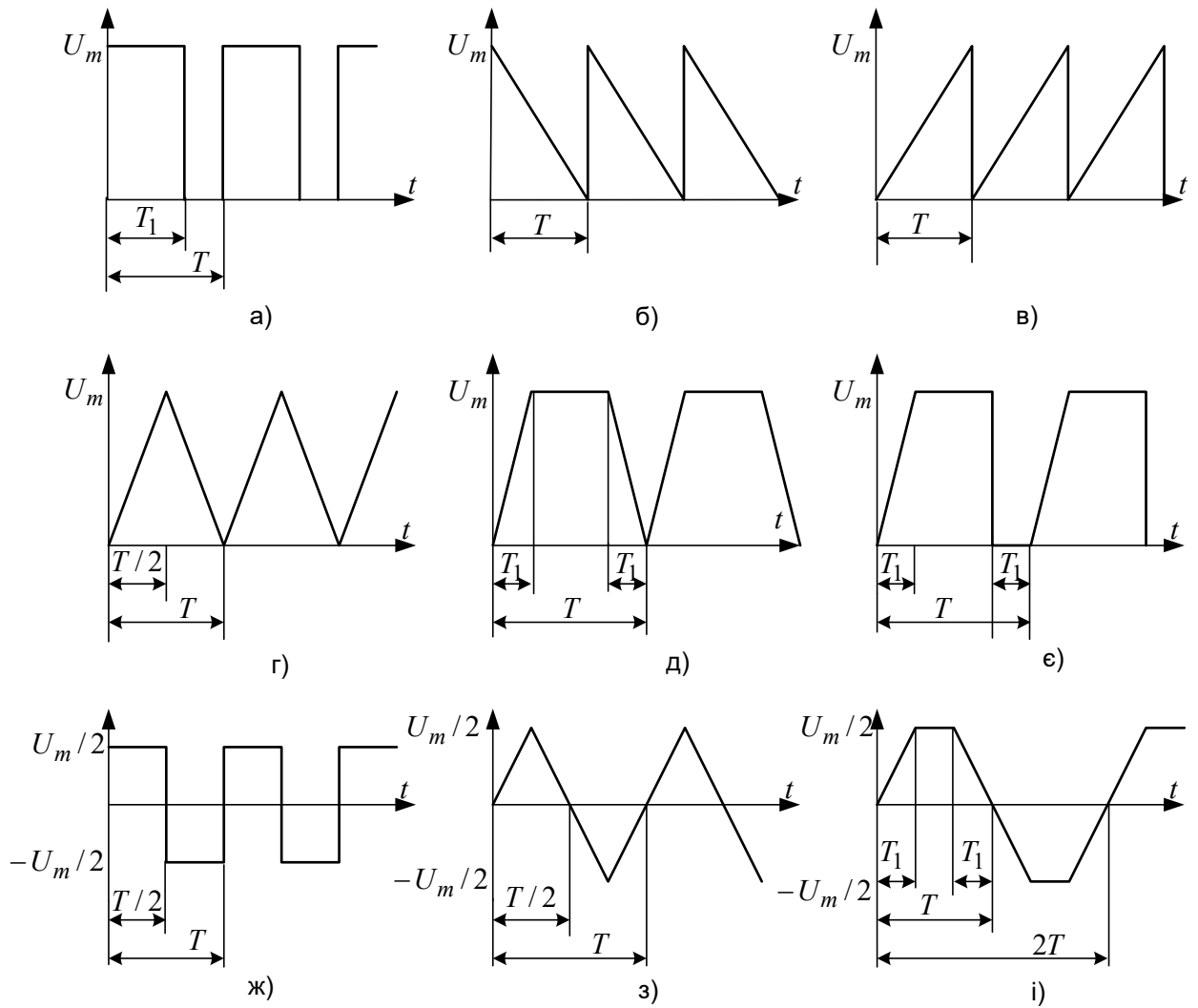


Рисунок 2.12 – Графіки вхідних сигналів до лабораторної роботи № 2

4) Ознайомтеся з блоками бібліотеки *Sources* та виконайте пункт 4 завдання.

5) Ознайомтеся з блоками *Pulse Generator* та *Repeating Sequence*, розгляньте їх побудову, зазирнувши під маску (команда *Look under Mask*) і виконайте пункт 5 завдання.

6) Ознайомтеся з блоками *Math Fn*, *Fn*, *Polynomial*, *Sqrt*, *Abs* та *Repeating Sequence* і виконайте пункт 6 завдання.

2.5 Методичні вказівки та рекомендації

1) Без зайвої потреби не використовуйте вікна моделей, бібліотек, осцилографів у повноекранному режимі.

2) Для швидкого з'єднання двох блоків виділіть щигликом миші вхідний блок, натисніть на клавішу <Ctrl> та виділіть мишею вихідний блок.

3) Для розмноження блоків та розгалуження ліній зв'язку використовуйте операцію *drag* (тягми) при натиснутій правій клавіші миші.

4) Деякі операції з блоками зручно робити через меню, що випадає при щиглику по блоку правою клавішею миші.

5) Слідкуйте за наочністю моделі, для чого притримуйтеся таких правил:

- намагайтеся використовувати мінімальну кількість зломів та перетинів ліній зв'язку;
- не робіть блоки занадто великими (модель повинна бути компактною);
- приховуйте імена блоків, призначення яких зрозуміло з їхніх піктограм;
- якщо це сприяє кращому порозумінню моделі, замінюйте імена стандартних блоків іншими іменами, іменуйте лінії зв'язку та доповнюйте моделі текстовими коментарями;
- при використанні різних кольорів ретельно продумайте, що саме ви бажаєте виділити тим чи іншим кольором;

6) Для реєстрації часу моделювання застосовуйте один з таких засобів:

- блок *Clock* приєднайте до блоку *To Workspace*, визначивши у полі *Variable name* цього блоку будь-яку змінну, наприклад, *t*.;

- у вкладці *Workspace I/O* вікна *Simulation Parameters* установіть прапорець *Time* функції *Save to workspace*.

7) При реєстрації даних блоками *To Workspace* і *Scope*, а також за допомогою функції *Save to workspace* вікна *Simulation Parameters* використовуйте формат *Array*. При великій кількості точок у векторі часу моделювання не забувайте збільшити або зробити безкінечним (*Inf*) параметр *Limit data point to last*.

8) Для об'єднання декількох скалярних сигналів в один векторний використовуйте блок *Mux* бібліотеки *Signals & Systems*. Це дасть змогу скоротити кількість блоків реєстрації.

2.5 Контрольні питання

- 1) Охарактеризуйте бібліотечні блоки реєстрації вихідних сигналів, перелічте їхні недоліки та переваги.
- 2) Опишіть структуру даних блоків *To Workspace* і *From Workspace*.
- 3) Поясніть призначення параметру *Decimation* блоків *Clock*, *To Workspace*, *To File* і *Scope*.
- 4) Поясніть призначення параметру *Interpolate data* блоку *From Workspace* і сенс можливих значень цього параметра.
- 5) Поясніть призначення параметру *Limit data points to last* блоків *Scope*.

Лабораторна робота №3

ЗНАЙОМСТВО З БІБЛІОТЕЧНИМИ НЕПЕРЕРВНИМИ ДИНАМІЧНИМИ ЛАНКАМИ ПРОГРАМИ *Simulink*

3.1 Теоретичні відомості

У більшості випадків на першому етапі досліджень роблять такі припущення, які дозволяють вважати САУ лінійними та стаціонарними. Такі системи в англomовній науково-технічній літературі називають *LTI-systems* – *Linear TimeInvariant systems*. Методи аналізу та синтезу таких системи є найпростішими та добре розвинутими.

Нехай довільна неперервна стаціонарна динамічна система n -го порядку має k входів та r виходів. Системи з декількома входами та декількома виходами називають **MIMO (Many Inputs Many Outputs)- systems**. Вони описуються у просторі стану системою лінійних диференціальних рівнянь

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (3.1)$$

з початковими умовами

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (3.2)$$

та лінійним матричним рівнянням виходу

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (3.3)$$

де

$$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T \quad \text{– вектор змінних стану;}$$

$$\mathbf{y} = [y_1 \quad y_2 \quad \dots \quad y_r]^T \quad \text{– вектор вихідних сигналів;}$$

$$\mathbf{u} = [u_1 \quad u_2 \quad \dots \quad u_k]^T \quad \text{– вектор вхідних сигналів;}$$

$$\mathbf{x}_0 = [x_{10} \quad x_{20} \quad \dots \quad x_{n0}]^T \quad \text{– вектор початкових умов (Initial conditions);}$$

$\mathbf{A}$ – матриця стану розміром $n \times n$;

$\mathbf{B}$ – матриця входу розміром $n \times k$;

$\mathbf{C}$ – матриця виходу розміром $r \times n$;

D – матриця обходу розміром $r \times k$, яка характеризує прямий (пропорційний) зв'язок між входом і виходом.

Для **SIMO (Single Input Many Outputs)-систем** ($k = 1$) матриці **B** і **D** вироджуються у вектори-стовпці:

$$\mathbf{B} = \mathbf{b} = [b_1 \quad b_2 \quad \dots \quad b_n]^T, \quad \mathbf{D} = \mathbf{d} = [d_1 \quad d_2 \quad \dots \quad d_n]^T.$$

Залежні змінні у таких ДР називають змінними стану. Їх кількість дорівнює кількості ДР і називається порядком системи.

Для **SISO (Single Input Single Output)-систем** ($k = 1, r = 1$) – матриця **B** залишається вектором-стовпцем, матриця **C** стає вектором-рядком розміром $1 \times n$, а матриця **D** вироджується у скаляр:

$$\mathbf{C} = \mathbf{c} = [c_1 \quad c_2 \quad \dots \quad c_n], \quad \mathbf{D} = d.$$

ДР (3.1) неоднорідні. Вони описують поведінку системи під дією зовнішніх вхідних сигналів. Якщо такі сигнали відсутні, то ДР стають однорідними:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t). \quad (3.4)$$

У сукупності з рівнянням виходу

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (3.5)$$

вони описують власні рухи системи під дією відхилення початкових умов від значень, що відповідають усталеному режиму.

Для переходу до операторної форми запису рівнянь (3.1), (3.2) в них треба замінити операцію диференціювання операцією множення на оператор Лапласа s :

$$s\mathbf{x}(s) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s), \quad (3.6)$$

$$\mathbf{y}(s) = \mathbf{C}\mathbf{x}(s) + \mathbf{D}\mathbf{u}(s). \quad (3.7)$$

Математичному опису у формі (3.7), (3.8) відповідає структурна схема, що зображена на рисунку 3.1.

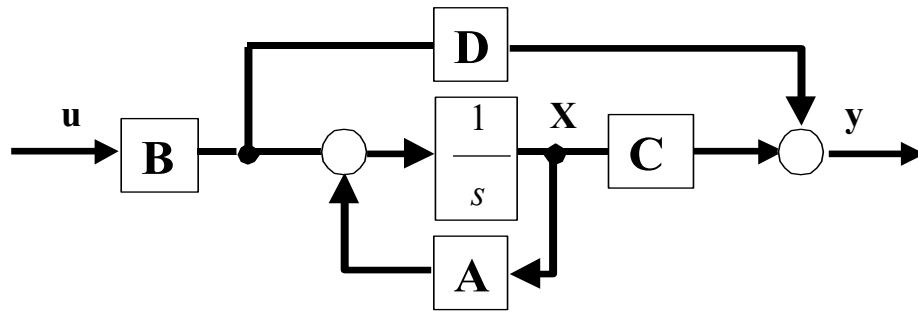


Рис. 3.1 – Структурна схема *LTI-MIMO*-системи у просторі стану

З рівнянь (3.6), (3.7) можна знайти матричні передавальні функції (ПФ) від вектора входу до вектору змінних стану розміром $n \times k$ та до вектора виходу розміром $r \times k$:

$$\mathbf{W}_x(s) = \frac{\mathbf{x}(s)}{\mathbf{u}(s)} = (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} = \frac{\text{Adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})} \mathbf{B}, \quad (3.8)$$

$$\mathbf{W}_y(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D} = \mathbf{C} \frac{\text{Adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})} \mathbf{B} + \mathbf{D}, \quad (3.9)$$

де $\mathbf{I}$ – одинична діагональна матриця розміром $n \times n$.

Знаменник передавальних функцій зовуть **характеристичним поліномом** (ХП) системи:

$$G(s) = \det(s\mathbf{I} - \mathbf{A}) = s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_1s + \alpha_0, \quad (3.10)$$

а рівняння

$$G(s) = 0$$

характеристичним рівнянням.

Корені характеристичного рівняння

$$\mathbf{s} = [s_1 \quad s_2 \quad \dots \quad s_n] \quad (3.11)$$

називають **полюсами системи** або **власними числами матриці $\mathbf{A}$** .

Чисельних кожної з скалярних ПФ, що є компонентами матричних ПФ (3.9) та (3.10), також уявляє собою степеневий поліном відносно змінної Лапласа, який називають **поліномом впливу**:

$$H(s) = \beta_m s^m + \beta_{m-1} s^{m-1} + \dots + \beta_1 s + \beta_0. \quad (3.12)$$

Його корені

$$\mathbf{s}_z = [s_{z1} \quad s_{z2} \quad \dots \quad s_{zm}] \quad (3.13)$$

звуть *нулями системи*.

Порядок поліному впливу пов'язаний з порядком характеристичного поліному умовою можливості фізичної реалізації

$$m \leq n. \quad (3.14)$$

Слід відзначити, що умова $m = n$ може бути виконана тільки для передавальних функцій від вхідного сигналу до вихідного, якщо вони мають між собою прямий зв'язок (без ланок затримки). Математичним виразом такого зв'язку є ненульове значення відповідного елементу матриці обходу $\mathbf{D}$.

Отже, скалярні передавальні функції можуть бути записані у поліноміальній формі, наприклад:

$$W_{ij}(s) = \frac{y_i(s)}{u_j(s)} = \frac{H_{ij}(s)}{G(s)} = \frac{\beta_m s^m + \beta_{m-1} s^{m-1} + \dots + \beta_1 s + \beta_0}{s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0}. \quad (3.15)$$

Як бачимо, ХП, визначений за формулою (3.10), і застосований у ПФ (3.15) виявляється нормованим за коефіцієнтом при старшому степені оператора Лапласа.

У класичній теорії автоматичного керування більш прийнято нормування поліномів у чисельнику та знаменнику передавальної функції за коефіцієнтами при вільному члені або, при його відсутності, за коефіцієнтами при найменших степенях оператора Лапласа, наприклад,

$$W_{ij}(s) = \frac{y_i(s)}{u_j(s)} = k \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + 1}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + 1} = k \frac{H_{0ij}(s)}{G_0(s)}. \quad (3.16)$$

Коефіцієнти ПФ (3.16) і (3.17) пов'язані між собою співвідношеннями:

$$a_n = \frac{1}{\alpha_0}, \quad a_i = \frac{\alpha_i}{\alpha_0}, \quad b_i = \frac{\beta_i}{\beta_0}, \quad k = \frac{\beta_0}{\alpha_0}. \quad (3.17)$$

Скалярній передавальній функції (3.16) відповідає ДР

$$\frac{d^n y_i(t)}{dt^n} + \alpha_{n-1} \frac{d^{n-1} y_i(t)}{dt^{n-1}} + \dots + \alpha_1 \frac{dy_i(t)}{dt} + \alpha_0 y_i(t) =$$

$$= \beta_m \frac{d^m u_j(t)}{dt^m} + \beta_{m-1} \frac{d^{m-1} u_j(t)}{dt^{m-1}} + \dots + \beta_1 \frac{du_j(t)}{dt} + \beta_0 u_j(t).$$

Використовуючи розкладення поліномів у чисельнику та знаменнику передавальної функції (3.16) або (3.17) на множники, отримуємо

$$W_{ij}(s) = K \frac{(s-s_{z1})(s-s_{z2})\dots(s-s_{zm})}{(s-s_1)(s-s_2)\dots(s-s_n)}, \quad (3.18)$$

де

$$K = \beta_m = k \frac{b_m}{a_n}. \quad (3.19)$$

3.2 Відомості про блоки бібліотеки *Continuous* програми *Simulink*

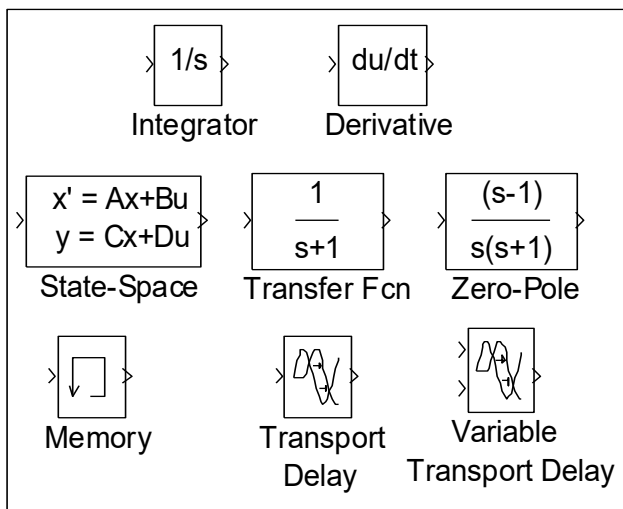


Рис. 3.2 – Бібліотека неперервних динамічних блоків (*Continuous*)

Бібліотека *Continuous*, представлена на рис. 3.2, містить блоки аналогового інтегрування (*Integrator*) і диференціювання (*Derivative*), лінійні неперервні динамічні ланки загального вигляду в різній формі (*Transfer Fcn*, *Zero-Pole* і *State-Space*) і ланки запізнювання (*Memory*, *Transport Delay* і *Variable Transport Delay*).

3.2.1 *Integrator* (Інтегратор)

У *Simulink 2.x* було 3 типи аналогових інтеграторів (*Integrator*, *Limited Integrator* і *Reset Integrator*). Починаючи з версії 3.0 усі три інтегратори об'єднані в одному, котрий може працювати в різних режимах за рахунок вибору відповідних установок у вікні параметрів. Зміни режиму роботи інтегратора відображаються на вигляді його піктограми.

На рис. 3.3 зображені піктограми блоку *Integrator* з різними установками режимів.

Найпростіший інтегратор (див. ланку *Integrator* на рис. 3.3) інтегрує вхідний сигнал u із заданою початковою умовою x_0 (*Initial condition*).

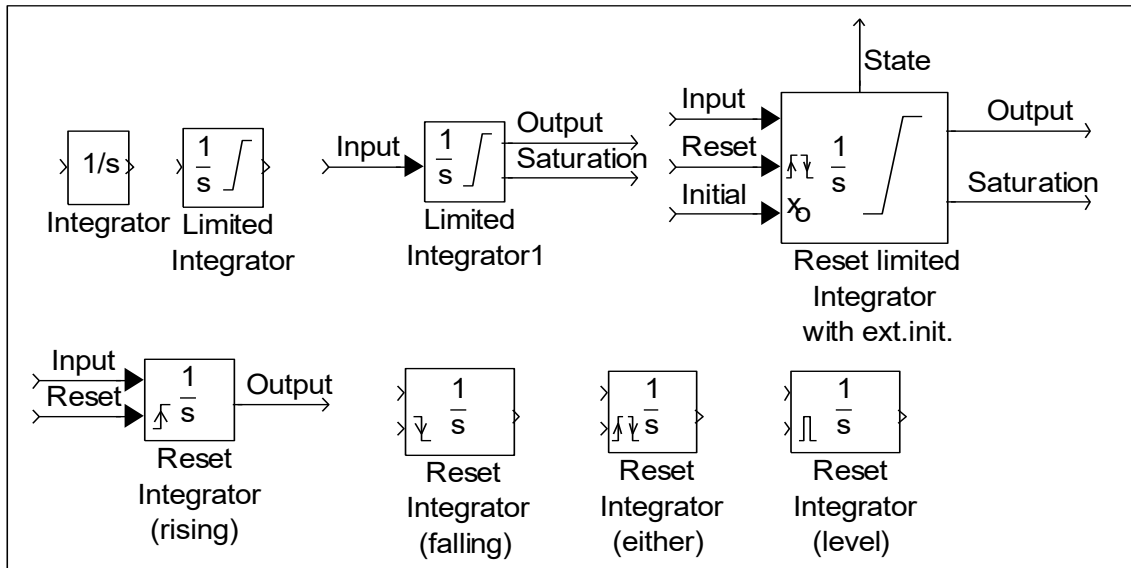


Рис. 3.3 – Піктограми інтеграторів, які працюють у різних режимах

Для того, щоб обмежити вихідний сигнал аналогового інтегратора (див. блок *Limited Integrator*), необхідно установити прапорець *Limit output* і ввести значення параметрів y_{max} (*Upper saturation limit*) і y_{min} (*Lower saturation limit*). Припустимим значенням цих параметрів є константа Inf (∞), що дозволяє обмежувати блок тільки знизу або тільки зверху.

Слід зазначити, що інтегратор з обмеженням не можна замінити послідовним з'єднанням інтегратора без обмеження і типової нелінійності «обмеження координат» (блок *Saturation* бібліотеки *Nonlinear*). Для правильного обмеження вихідного сигналу інтегратора при перетинанні ним однієї з меж необхідно перевести блок з режиму інтегрування в режим збереження інформації підключенням до його входу замість сигналу нульової константи, який інтегрується, як це показано на рис. 3.4.

Повернення в режим інтегрування відбувається при зміні знака вхідного сигналу.

Можна візуалізувати порт обмеження, установкою прапорця *Show saturation port* (див. блок *Limited Integrator1*).

Інтегратор може працювати зі скиданням вихідного сигналу в початкове значення (див. ланки *Reset Integrator*). Цим режимом керує меню, що випадає, опції *External reset*, яке пропонує наступні варіанти:

none – скидання не виконується;

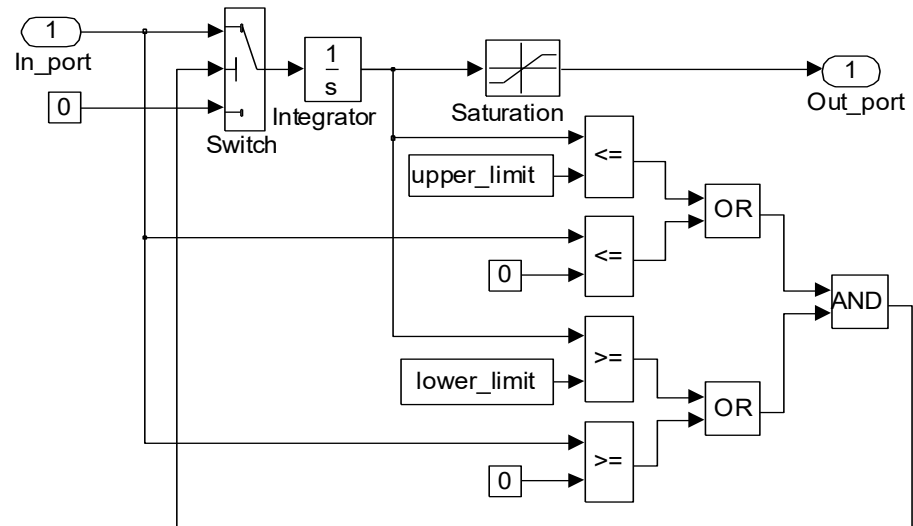


Рис. 3.4 – Модель інтегратора з обмеженням вихідного сигналу

rising – скидання виконується при перетинанні керуючим сигналом нульової лінії знизу нагору;

falling – скидання виконується при перетинанні керуючим сигналом нульової лінії зверху вниз;

either – скидання виконується при перетинанні керуючим сигналом “нуля” в обох напрямках;

level – при нульовому значенні керуючого сигналу виконується інтегрування, а при ненульовому – скидання та утримання інтегратора в початковому стані.

При виборі будь-якого варіанта, крім *none*, до блоку приєднується керуючий вхідний порт (*Reset port*).

Інтегратор може використовувати як внутрішні (*internal*), так і зовнішні (*external*) початкові умови. Вибір здійснюється за допомогою меню опції *Initial condition source*. При виборі режиму *external* блок одержує додатковий вхідний порт, до якого можна приєднати блок *IC* (Початкова умова) бібліотеки *Connections*. При виборі режиму *internal* діє початкова умова, що вста-

новлюється в полі параметра *Initial condition*. Задавати зовнішні початкові умови має сенс тільки для інтегратора зі скиданням (див. ланку *Reset limited integrator with ext. init.*).

Для організації скидання або зміни початкових умов у функції вихідного сигналу інтегратора необхідно, щоб уникнути утворення алгебраїчної петлі, замість вихідного порту використовувати порт стану, для чого потрібно візуалізувати його установкою прапорця *Show state port*. Сигнал стану інтегратора формується раніш, ніж вихідний сигнал, хоча і має з ним однакові значення. Приклад використання ланки *Integrator* у режимах *Reset* і *Limited* можна подивитися в демонстраційному прикладі *Bouncing ball* (модель *bounce*).

Друга область застосування порту стану – це передача сигналів з однієї підсистеми, виконуваної за умовою, в іншу. Для демонстрації цього приклада можна скористуватися моделлю *clutch*.

3.2.2 Derivative (Похідна)

Виконує чисельне диференціювання вхідного сигналу:

$$y(t_i) = \frac{\Delta u_i}{\Delta t_i} = \frac{u_i - u_{i-1}}{t_i - t_{i-1}} \approx \frac{du}{dt}. \quad (3.20)$$

Точність результату залежить від розміру кроку інтегрування. Менший розмір кроку дозволяє одержати більш точний результат.

Перед початком моделювання значення вхідного сигналу дорівнює нулю.

При диференціюванні дискретного сигналу з періодом переривання T_s вихідний сигнал являє собою послідовність імпульсів у моменти часу, які збігаються з моментами зміни вхідного сигналу:

$$y(t_i) = \begin{cases} \frac{u(kT_s) - u(kT_s - T_s)}{t_i - t_{i-1}} & \text{при } t_i = kT_s, \\ 0 & \text{при } t_i \neq kT_s. \end{cases} \quad (3.21)$$

У цьому випадку блок може бути описаний дискретною передавальною функцією

$$\frac{y(z)}{u(z)} = \frac{1 - z^{-1}}{\Delta t} = \frac{z - 1}{\Delta t \cdot z}. \quad (3.22)$$

3.2.3 *Transfer Fcn* (Передавальна Функція)

Блок *Transfer Fcn* реалізує передавальну функцію загального вигляду в поліноміальній формі:

$$W(s) = \frac{B_m(s)}{A_n(s)} = \frac{\sum_{i=0}^m b_{i+1} s^{m-i}}{\sum_{i=0}^n a_{i+1} s^{n-i}}. \quad (3.23)$$

Вхідними параметрами є вектори коефіцієнтів степенних поліномів чисельника (*Numerator*) і знаменника (*Denominator*), упорядковані за зменшенням степенів оператора Лапласа s . Вектор з $(n+1)$ елементів задає поліном степеня n . Порядок знаменника повинний бути більшим від порядку чисельника або дорівнювати йому. Блок має нульові початкові умови.

Коефіцієнти можуть бути введені трьома способами:

- 1) константами і скалярними змінними, об'єднаними квадратними дужками у вектор;
- 2) змінною без дужок;
- 3) змінною або виразом у круглих дужках.

При першому способі в піктограмі блоку відображаються поліноми з коефіцієнтами у вигляді введених констант і змінних при відповідних степенях оператора Лапласа s . Наприклад, введення вектора $[1 \ -2 \ T \ 4]$ відобразиться в блоці як

$$s^3 - 2s^2 + Ts + 4 ,$$

а введення вектора $[b0, b1, b2]$ – як

$$b0s^2 + b1s + b2 .$$

Змінні, що використані в цих векторах, повинні одержати значення в робочому середовищі *MATLAB* до початку процесу моделювання.

При другому способі поліном відображається у вигляді змінної(*s*). Наприклад, введення *num* відображається як *num(s)*, введення *An* – як *An(s)* та т.і.

При третьому способі *Simulink* шукає значення змінних, які входять до виразу, у робочому просторі *MATLAB*, обчислює значення виразу і відображає його так само, як і при першому способі введення. Наприклад, якщо поліном визначено у діалоговому вікні як $(G+B)$, а перемінні *G* і *B* визначено в *MATLAB* як $G=[4 \ 1 \ 1]$, $B=[0 \ 1 \ 0]$, то відображення полінома в блоці має вид:

$$4s^2 + 2s + 1.$$

Значення змінних повинні існувати в робочому середовищі в будь-який момент зображення блоку, інакше в піктограмі блоку відобразяться три знаки питання: ???.

Simulink дозволяє визначити передавальну функцію не тільки для систем з одним входом та одним виходом, але і для систем з одним входом та декількома виходами. Це досягається введенням чисельника передавальної функції як матриці коефіцієнтів, кількість рядків якої відповідає кількості виходів. Відображення коефіцієнтів чисельника в піктограмі блоку в цьому випадку не підтримується, так що чисельник представляється тільки в загальному вигляді, наприклад, *num(s)*, тобто матриця коефіцієнтів чисельника повинна задаватися тільки ім'ям перемінної без дужок. Якщо наступні за розглянутою ланкою блоки не сприймають векторних входних сигналів, то отриманий вихідний векторний сигнал необхідно пропустити через блок *Demux* (Демультіплексор) з бібліотеки *Conexions*.

3.2.4 Zero-Pole (Нулі-Полюси)

Являє собою ланку загального виду з передавальною функцією (3.18) та такими параметрами:

K — посилення (*Gain*);

s_{zi} — нулі (*Zeros*);

s_i — полюси (*Poles*).

Нулі і полюси можуть мати не тільки дійсні, але і комплексно-спряжені значення. Вони можуть уводитися тими ж трьома способами, що і вектори степеневих багаточленів блоку *Transfer Fcn*.

Розглянутий блок не підтримує режим *SIMO*, тобто не може бути використано, на відміну від ланки *Transfer Fcn*, для опису лінійної динамічної системи з декількома виходами.

3.2.5 State-Space (Простір Стану)

Забезпечує моделювання неперервної динамічної ланки загального виду за її математичному описом у просторі стану (3.1)-(3.3). Параметрами цього блоку є матриці коефіцієнтів **A**, **B**, **C**, **D** та вектор початкових умов (*Initial conditions*).

3.2.6 Memory (Пам'ять)

Здійснює затримку на один крок чисельного інтегрування. На виході утримується попереднє значення входу. Цей блок можна використовувати для розв'язки алгебраїчних контурів. Разом із блоком *Clock* його можна використовувати для визначення кроків інтегрування. У цьому блоці задається тільки один параметр: *Initial condition* – початкова умова.

3.2.7 Transport Delay (Чисте Запізнювання)

Передавальна функція ланки має вигляд:

$$\frac{y(p)}{u(p)} = e^{-s\tau}. \quad (3.24)$$

Вихідний сигнал відтворює вхідний сигнал із запізнюванням τ (*Time Delay*):

$$y(t) = \begin{cases} y_0 & \text{при } t \leq \tau \\ u(t - \tau) & \text{при } t > \tau \end{cases}, \quad (3.25)$$

де y_0 – початкове значення вхідного сигналу (*Initial input*).

Ланка функціонує за рахунок збереження необхідної кількості послідовних значень вхідного сигналу в круговому буфері, початковий розмір якого задається параметром *Initial Buffer Size*. Отже, при великих часах запізнюван-

ня і порівняно малому кроці інтегрування блок може використовувати велику кількість оперативної пам'яті.

3.2.8 *Variable Transport Delay (Змінне Запізнювання)*

Ця ланка має два входи, на другому з яких формується величина запізнювання вихідного сигналу стосовно першого вхідного. Параметрами, що вводяться, є *Maximum Delay*, *Input i Buffer Size*. Перший параметр використовується для обмеження запізнювання на заданому рівні і для розрахунку кількості точок у круговому буфері.

3.3 Відомості про лінійні арифметичні блоки бібліотеки *Math*

Описані в попередньому підрозділі блоки використовуються при моделюванні лінійних неперервних систем. Для зв'язку динамічних елементів між собою у складі таких систем застосовуються лінійні арифметичні блоки *Sum*, *Gain*, *Slider Gain* і *Matrix Gain* бібліотеки *Math* (рис. 3.3).

Блок *Sum* описано в підрозділі 3.1 (див. рис. 3.4).

Блок ***Gain (Підсилення)*** зі скалярним значенням основного параметру і скалярним входом є пропорційною ланкою, яка обчислює вираз

$$y = k \cdot u.$$

Піктограма блоку відображає скалярний коефіцієнт підсилення в тій же формі, у якій він визначений при введенні (змінна або константа). Якщо коефіцієнт заданий у вигляді змінної в круглих дужках, то усередині блоку відображається її значення. Якщо значення змінної або її ім'я, або вираз занадто великі для того, щоб відобразити їх у межах блоку, то піктограма відображає символи “-K-“. Щоб побачити вираз або значення коефіцієнта підсилення, необхідно збільшити розміри блоку.

У цьому блоці можна також встановити такі режими роботи з векторним або матричним коефіцієнтом підсилення:

- *Elementwise* ($K \cdot u$);
- *Matrix* ($K \cdot u$);
- *Matrix* ($u \cdot K$);

- *Matrix* ($K*u$), (u vector).

Відображення векторного коефіцієнта підсилення підтримується тільки в тому випадку, якщо він заданий змінною без дужок.

Блок ***Slider Gain (Повзункове Підсилення)*** виконує множення вхідного сигналу на коефіцієнт, що обирається в заданих межах *Low-High* повзунком віртуального реостату, розташованого у вікні вибору параметрів блоку.

3.4 Завдання

- 1) Вивчити властивості інтегратора у різних режимах його роботи.
- 2) Продемонструвати різні способи математичного опису лінійних динамічних систем у програмі *Simulink*.
- 3) Продемонструвати роботу блоків запізнення.

3.5 План виконання роботи

- 1) Зберіть модель, зображену на рис. 3.5, зафіксуйте результати моделювання та поясніть їх.
- 2) За допомогою блоків *Transfer Function*, *Zero-Pole* та *State Space* реалізуйте моделі, задані передавальними функціями у табл. 3.1. Впевніться у збігу результатів моделювання.
- 3) Подайте на вхід блоків запізнення синусоїдальний сигнал і зафіксуйте вхідний і вихідні сигнали. Поясніть отримані результати.

3.5 Методичні вказівки та рекомендації

- 1) Для визначення нулів та полюсів передавальної функції скористуйтеся функціями *roots* та *poly*.
- 2) Для математичного опису лінійної системи, заданої передавальною функцією, у просторі стану скористуйтеся деталізованою структурною схемою.
- 3) Для дослідження передавальних функцій скористайтеся структурною схемою на рис. 3.6.

4) Для дослідження блоків запізнення подавайте їм на входи синусоїдальні сигнали. Змінне запізнення організуйте за допомогою послідовного з'єднання блоків *Clock*, *Math Fn* та *Gain*.

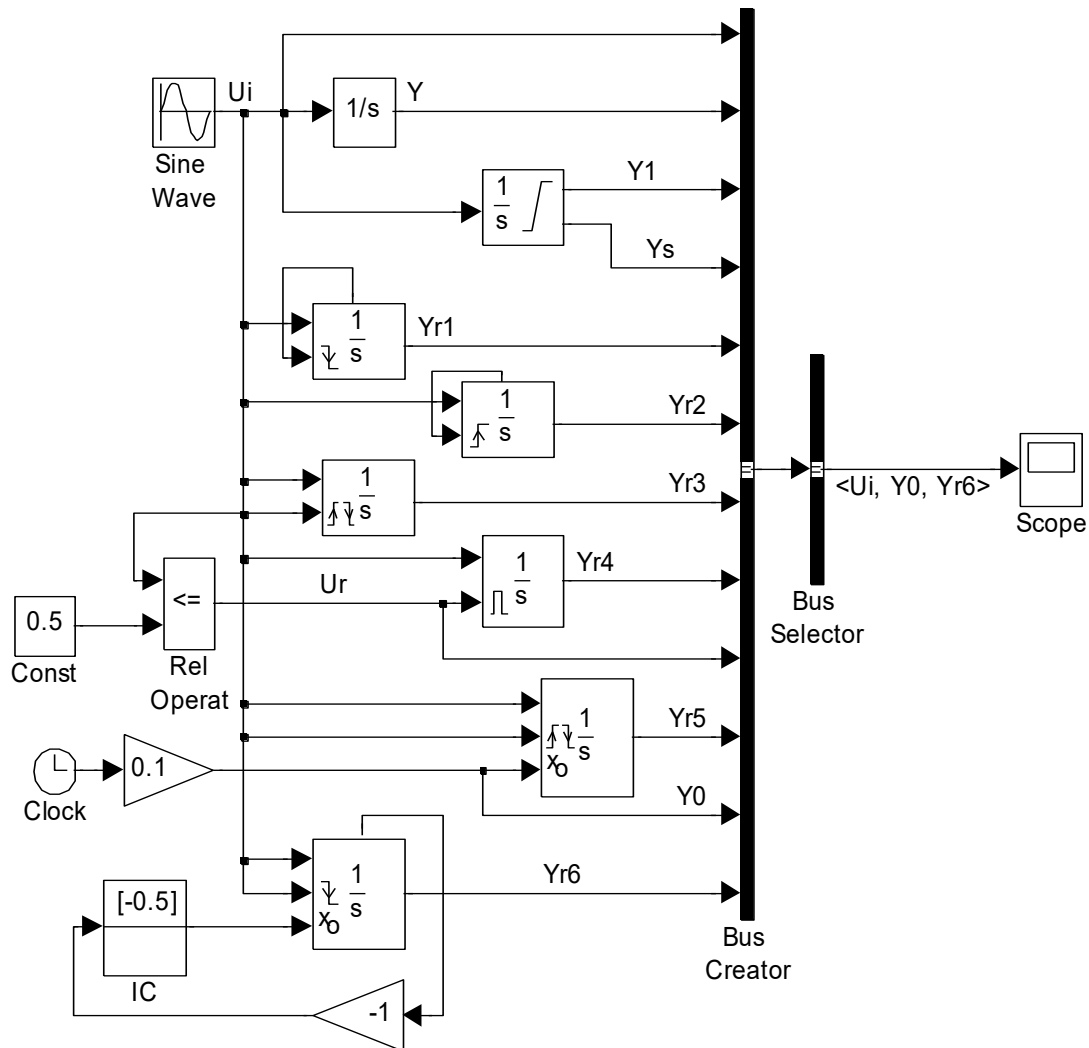


Рис. 3.5 – Модель для вивчення принципу роботи інтегратора в різних режимах

3.6 Контрольні питання

- 1) Які форми математичного опису лінійних неперервних систем Ви знаєте?
- 2) Що таке характеристичний поліном, характеристичне рівняння, нулі і полюси системи?
- 3) Як скласти математичний опис системи у просторі стану?
- 4) Як отримати передавальну функцію з диференційного рівняння?
- 5) У яких режимах може працювати блок *Integrator*?

Таблиця 3.1

№ вар.	Передавальна функція	Параметри
1	$W = \frac{k}{Tp}$	$k = 1, T = 2$
2		$k = 1.5, T = 3$
3		$k = 0.75, T = 4$
4	$W = \frac{k}{Tp + 1}$	$k = 1, T = 2$
5		$k = 1.5, T = 3$
6		$k = 0.75, T = 4$
7	$W = \frac{T_1 p}{T_2 p + 1}$	$T_1 = 2, T_2 = 3$
8		$T_1 = 1.5, T_2 = 4$
9		$T_1 = 5, T_2 = 2$
10	$W = \frac{T_1 p + 1}{T_2 p}$	$T_1 = 2, T_2 = 3$
11		$T_1 = 1.5, T_2 = 4$
12		$T_1 = 5, T_2 = 2$
13	$W = \frac{T_1 p + 1}{T_2 p + 1}$	$T_1 = 2, T_2 = 3$
14		$T_1 = 1.5, T_2 = 4$
15		$T_1 = 5, T_2 = 2$
16	$W = \frac{k}{T_1 p^2 + T_2 p + 1}$	$k = 1, T_1 = 2, T_2 = 3$
17		$k = 3, T_1 = 1.5, T_2 = 4$
18		$k = 2, T_1 = 5, T_2 = 2$
19	$W = k \frac{p}{(p - p_1)(p - p_2)}$	$k = 2.5, p_{1,2} = -3 \pm 2.3j$
20		$k = 3, p_{1,2} = -2.3 \pm 3j$
21		$k = 1.5, p_{1,2} = -2 \pm 1.8j$
22	$W(s) = \frac{(T_1 s + 1)(T_2 s + 1)}{T^2 s^2 + 2\xi T s + 1}$	$T_1 = 2, T_2 = 3, T = 5, \xi = 0.4$
23		$T_1 = 2, T_2 = 3, T = 5, \xi = 0.12$
24	$W(s) = \frac{(T_1 s + 1)(T_2 s + 1)}{T^2 s^2 + 1}$	$T_1 = 2, T_2 = 3, T = 5$
25		$T_1 = 5, T_2 = 3, T = 2$

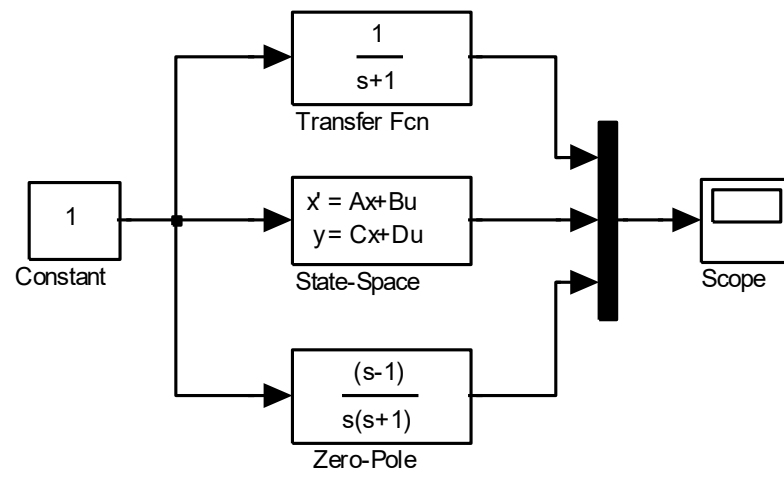


Рис. 3.6 – Модель для дослідження передавальних функцій

*Лабораторна робота №4***ЗНАЙОМСТВО З БІБЛІОТЕЧНИМИ НЕЛІНІЙНИМИ БЛОКАМИ
ПРОГРАМИ *Simulink*****4.1 Теоретичні відомості**

Нелінійні ланки, що найчастіше використовуються при моделюванні електромеханічних систем можна розподілити на такі групи:

- 1) блоки множення-ділення декількох сигналів;
- 2) функціональні перетворювачі, описувані нелінійними аналітичними виразами;
- 3) функціональні перетворювачі, задані таблицями вхідних та вихідних сигналів у деяких вузлових точках.

Серед блоків другої та третьої груп виділяють так звані типові нелінійності. Так називають блоки з досить простими кусочно-лінійними характеристиками вхід-вихід, що часто зустрічаються на практиці.

До них належать такі ідеалізовані блоки як *Обмеження координат*, *Сухо тертя*, *В'язке тертя*, *Зона нечутливості*, *Модуль*, *Люфт (Зазор)*, *Петля гістерезису*, *Реле*, *Компаратор* та деякі інші.

З іншого боку всі нелінійності можна поділити на однозначні та багатозначні (найчастіше двозначні). У багатозначних нелінійностей зв'язок між вхідним та вихідним сигналами визначається не тільки формою статичної характеристики, але й передісторією вхідного сигналу, наприклад, від того, зменшується вхідний сигнал чи наростає. Типовими двозначними не лінійностями є блоки *Петля гістерезису* та *Зазор в кінематичній передачі*.

При визначенні вихідних сигналів блоків четвертої групи в точках між вузловими, застосовують або апроксимацію, або інтерполювання.

При апроксимації найчастіше використовують метод найменших квадратів (МНК) або кусочно-лінійну апроксимацію. У якості апроксимуючих аналітичних функцій здебільш застосовують степеневі поліноми, комбінації експонент та розкладення періодичних характеристик в ряд Фур'є.

Серед методів інтерполяції для більшості технічних розрахунків достатню точність можна забезпечити локальним інтерполюванням степеневими поліномами першого-третього порядків. Для підвищення точності можна використовувати глобальну інтерполяцію або інтерполяцію кубічними сплайнами. Але застосування останніх методів може значно збільшити час розрахунку перехідних процесів.

4.2 Нелінійні блоки програми *Simulink*

Нелінійні ланки програми зосереджені в декількох бібліотеках.

Блоки першої, другої та частково третьої груп, описаних у попередньому підрозділі, знаходяться в бібліотеці *Math* (див. рис. 4.3). Крім описаних у підрозділі 4.1 блоків *Math Function* та *Trigonometric Function*, до них належать блоки *Product*, *Dot Product*, *Abs*, *Sign*, *MinMax*, *Rounding Function*.

Блок ***Product* (Множення-Ділення)** перемножує вхідні сигнали, кількість яких n визначається параметром *Number of inputs*:

$$y = u_1 \cdot u_2 \cdot \dots \cdot u_n. \quad (4.1)$$

При $n=1$ він накопичує добуток значень єдиного вхідного векторного сигналу:

$$y = \text{prod}(\mathbf{u}) = \prod_{i=1}^k u_i. \quad (4.2)$$

При цьому в піктограмі блоку символ “*” замінюється символом “П”.

Якщо замість кількості входів задати список, зіставлений з послідовністю знаків операцій множення “*” та ділення “/”, то блок буде виконувати над вхідними сигналами саме операції, задані списком.

Якщо вхідні сигнали є векторами однакового розміру, то операції множення / ділення можуть здійснюватись як матрично (*Matrix(*)*), так і поелементно (*Element-wise(*)*), що задається через меню параметру *Multiplication*.

Блок ***Dot Product* (Скалярний добуток)** обчислює скалярний добуток векторних вхідних сигналів $\mathbf{u}$ та $\mathbf{v}$ однакового розміру:

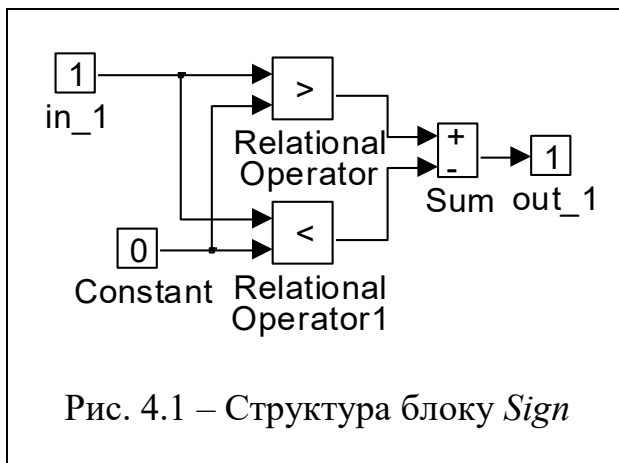
$$y = \text{dot}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} = \sum_{i=1}^n u_i v_i. \quad (4.3)$$

Складається з послідовно з'єднаних блоку *Product* із двома входами і блоку *Sum* з одним входом. Якщо вхідні сигнали – скаляри, то обчислюється їхній добуток.

Виходом ланки *Abs (Модуль)* є абсолютне значення входу: $y = |u|$.

Блок *Rounding Function (Округлення)* округляє значення вхідного сигналу одним з обраних способів: округлення до цілого з нестачею (*floor*), з надлишком (*ceil*), та за правилами арифметики (*round*), скорочення до цілого, тобто відкидання дробової частини (*fixed*).

Блок *MinMax (Мінімум / Максимум)* визначає мінімальний або максимальний (за вибором користувача) із вхідних сигналів, кількість яких задається параметром *Number of input ports*.



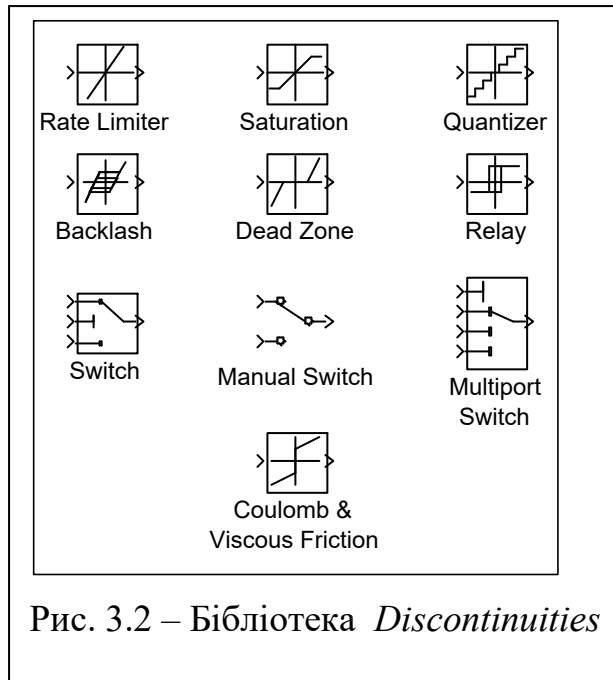
Блок *Sign (Знакова Функція, Ідеальне Реле)* формує вихідний сигнал, величина якого визначається знаком вхідного сигналу:

$$y = \begin{cases} 1 & \text{при } u > 0, \\ 0 & \text{при } u = 0, \\ -1 & \text{при } u < 0. \end{cases} \quad (4.4)$$

Цей замаскований блок має структурну схему рис. 4.1 і належить до типових нелінійностей. Якщо вхідним сигналом блоку *Sign* є швидкість (лінійна або кутова), а на виході встановлено ланку з коефіцієнтом, рівним амплітуді сили тертя (при поступальному руху) або моменту тертя (при обертальному руху), то одержують модель типової нелінійності *Сухе тертя*.

В бібліотеці *Discontinuities* (див. рис. 4.2) знаходяться найбільш поширені типові нелінійності: *Saturation (Обмеження координат)*, *Dead Zone (Зона нечутливості)*, *Coulomb & Viscous Friction (Сухе та в'язке тертя)*, *Quan-*

tizer (Квантизатор), Rate Limiter (Обмеження темпу), Backlash (Люфт), Relay (Реле з гістерезисом).



Піктограми перелічених блоків зображують статичні характеристики відповідних типових нелінійностей.

Блок ***Saturation*** (Обмеження Координат) являє собою пропорційну ланку з одиничним коефіцієнтом підсилення, вихідний сигнал якої обмежений зверху на рівні U (*Upper output limit*) і знизу – на рівні L (*Lower output limit*):

$$y = \begin{cases} u & \text{при } L \leq u \leq U, \\ U & \text{при } u > U, \\ L & \text{при } u < L. \end{cases} \quad (4.5)$$

Його властивості продемонстровані на рисунку 4.3.

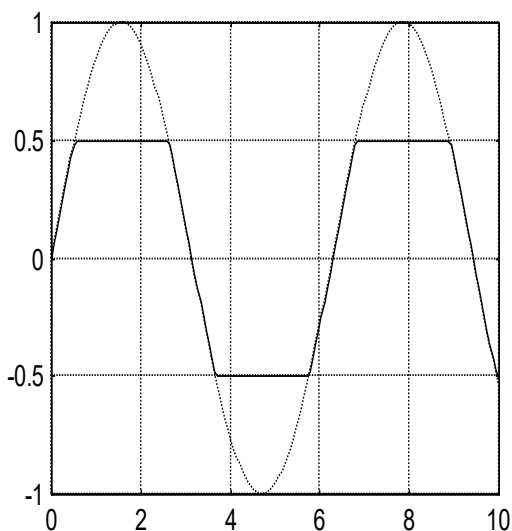


Рис. 4.3 – Перетворення синусоїди ланкою *Saturation*

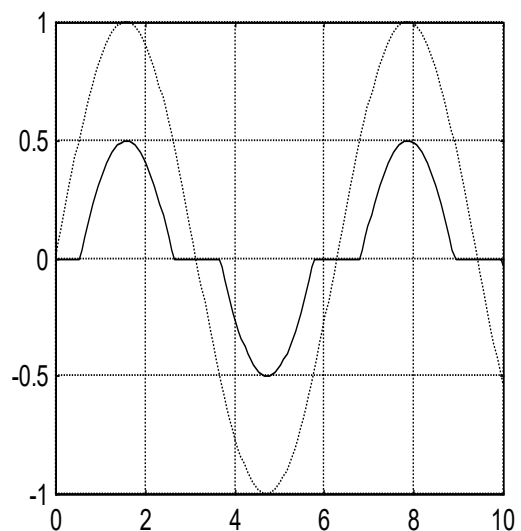


Рис. 4.4 – Перетворення синусоїди ланкою *Dead Zone*

Блок ***Dead Zone (Зона Нечутливості)*** має статичну характеристику, що описується формулою

$$y = \begin{cases} 0 & \text{при } L \leq u \leq R, \\ u + L & \text{при } u < L, \\ u - R & \text{при } u > R, \end{cases} \quad (4.6)$$

де L, R – ліва (*Start of dead zone*) і права (*End of dead zone*) границі зони нечутливості відповідно. Процес перетворення цієї ланкою синусоїдального сигналу відображено на рис. 4.4.

Блок ***Coulumbic & Viscous Friction (Сухе та В'язке Тертя)*** реалізує статичну характеристику, що описується рівнянням лінійну модель тертя:

$$y = (K \cdot |u| + y_0) \cdot \text{sign}(u), \quad (4.7)$$

де

K (*Gain*) – коефіцієнт передачі лінійної ділянки статичної характеристики;

y_0 (*Offset*) – абсолютне значення вихідного сигналу в нульовій точці.

Якщо вхідним сигналом є швидкість руху деякого механізму, то вихідний сигнал можна інтерпретувати як силу або момент тертя ідеалізованої лінійної моделі цього процесу.

При $K = 0$ тертя буде сухим, тобто незалежним від величини швидкості. При $K > 0$ тертя буде в'язким, тобто воно збільшуватиметься при підвищенні швидкості, що має місце, наприклад, при руху в рідинному або газовому середовищах. При $K < 0$ тертя зменшується при підвищенні швидкості за рахунок виникнення підіймальної сили, що знижує зчеплення між поверхнями, які спіткаються між собою в процесі руху.

На рисунку 4.5 наведені діаграми перетворення синусоїдального вхідного сигналу ланкою *Coulumbic Friction* при додатному (а) і від'ємному (б) значеннях коефіцієнта *Gain*.

Блок ***Quantizer (Квантизатор)*** квантизує вхідний сигнал за рівнем з заданою дискретою Δy (*Quantization interval*):

$$y = \text{round}(u / \Delta y). \quad (4.8)$$

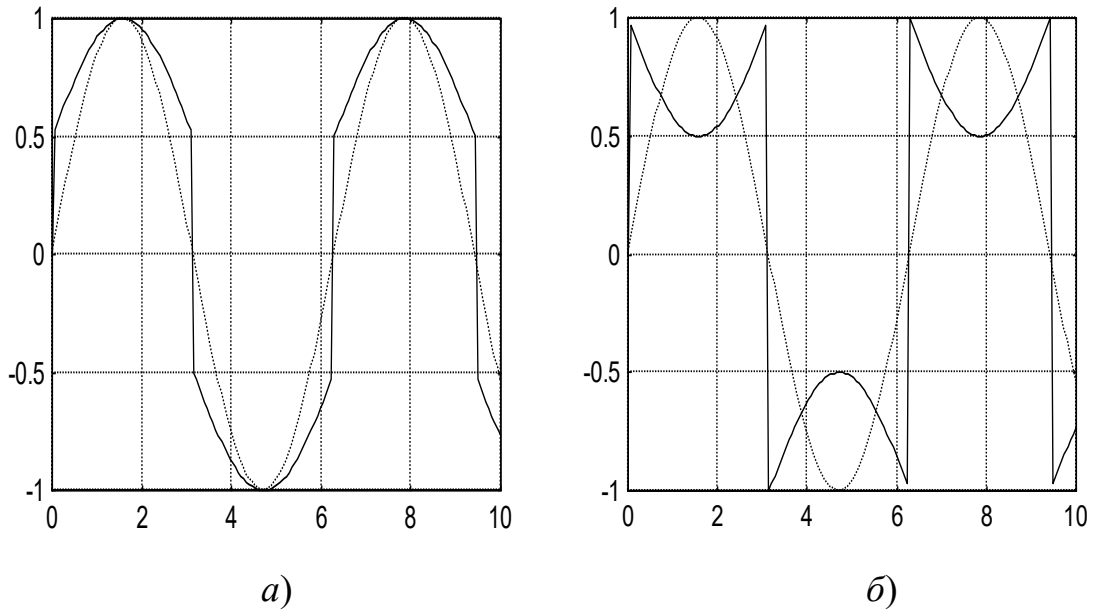


Рис. 4.5 – Перетворення синусоїди ланкою *Coulumbic & Viscous Friction*:

a) Offset = 0.5, Gain = 0.5; б) Offset = 1, Gain = -0.5

Блок **Rate Limiter (Обмеження Темпу)** відтворює вхідний сигнал $u(t)$ з обмеженням його першої похідної du/dt на рівні R (*Rising slew rate*) при збільшенні сигналу і F (*Falling slew rate*) – при його зменшенні відповідно до формули

$$y_i = \begin{cases} \Delta t \cdot \text{abs}(R) + y_{i-1} & \text{при } \left| \frac{du}{dt} \right| > R, \\ -\Delta t \cdot \text{abs}(F) + y_{i-1} & \text{при } \left| \frac{du}{dt} \right| < F, \\ u_i & \text{при } F \leq \left| \frac{du}{dt} \right| \leq R, \end{cases} \quad (4.9)$$

як це показано на рисунку 4.6.

Значення похідної обчислюється за формулою:

$$\frac{du}{dt} \approx \frac{\Delta u}{\Delta t} = \frac{u_i - u_{i-1}}{t_i - t_{i-1}} \quad (4.10)$$

Ланка **Backlash (Зазор, Люфт)** моделює зазор (люфт) у кінематичній передачі:

$$y_i = \begin{cases} u - \frac{\delta}{2} & \text{при } u - \frac{\delta}{2} > y_{i-1}, \\ u + \frac{\delta}{2} & \text{при } u + \frac{\delta}{2} < y_{i-1}, \\ y_{i-1} & \text{при } u - \frac{\delta}{2} \leq y_{i-1} \leq u + \frac{\delta}{2}, \end{cases} \quad (4.11)$$

де δ – величина зазору (*Deadband width*).

Вхідним сигналом тут є положення активної маси, а вихідним – пасивної.

Установка початкових значень вхідного (*Initial input value*) і вихідного (*Initial output value*) параметрів дозволяють цілком визначити початковий стан системи. Початкове значення виходу повинне бути в межах $1/2$ величини зазору; у противному випадку *Simulink* сигналізує про помилку.

Діаграма перетворення синусоїдального вхідного сигналу ланкою *Backlash* наведена на рисунку 4.7.

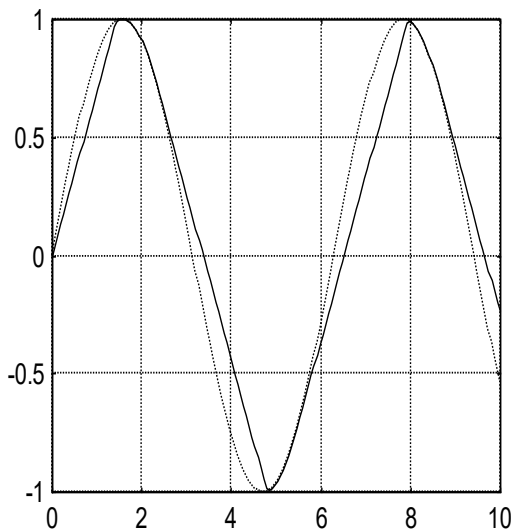


Рис. 4.6 – Перетворення синусоїди ланкою *Rate Limiter*

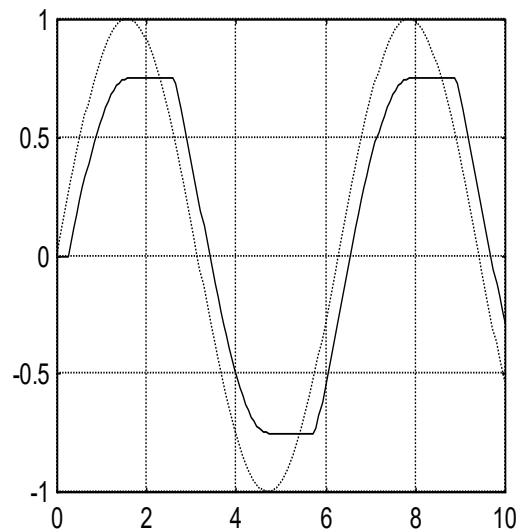


Рис. 4.7 – Перетворення синусоїди ланкою *Backlash* ($\delta=1$)

Блок ***Relay (Реле з Гістерезисом)*** дозволяє перемикає вихід між двома заданими значеннями *Output when on* (y_{on}) і *Output when off* (y_{off}). Коли реле включено, воно залишається в цьому стану доти, поки значення вхідного сигналу не упаде до заданого значення для вимикання реле *Input for off* (x_{off}).

Коли реле вимкнено, воно залишається в цьому стану доти, поки значення входу не перевищить задане значення для включення реле *Input for on* (x_{on}).

При $x_{on} > x_{off}$ моделюється реле з петлею гістерезиса, а при $x_{on} = x_{off}$ – ідеальне реле (див. рис. 4.8).

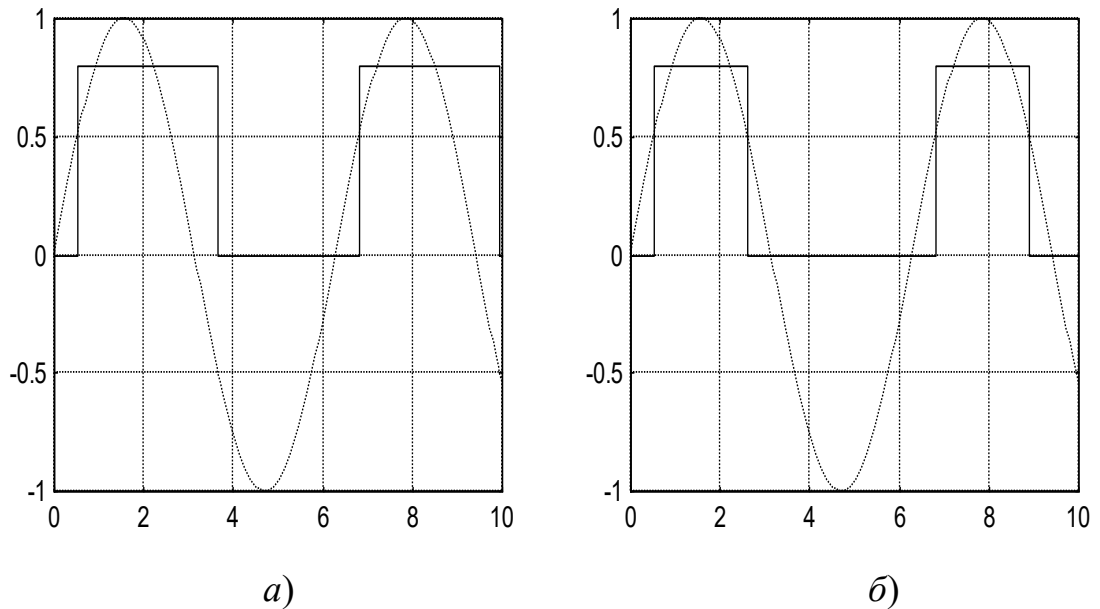


Рис. 4.8 – Перетворення синусоїди ланкою *Relay*:

a) $x_{on} = 0.5, x_{off} = -0.5$; б) $x_{on} = x_{off} = 0.5$

Кусочно-лінійну апроксимацію заданої табличної нелінійності виконує блок *Look-Up Table* (див. підрозділ 4.1). Для того, щоб статична характеристика такої нелінійної ланки була більш плавною можна замість блоку використати блок *MATLAB Fn*, звертаючись через нього до функції, що здійснює кусочно-кубічну інтерполяцію або інтерполяцію сплайнами: `interp1(X,Y,u,'cubic')` або `interp1(X,Y,u,'spline')` відповідно, де X, Y – вектори аргументів та значень табличної функції однакового розміру.

Для поліноміальної апроксимації табличної нелінійності методом найменших квадратів спочатку знаходять вектор коефіцієнтів степеневого полінома n -го порядку виконанням операції.

$$A = \text{polyfit}(X, Y, n)$$

а потім використовують визначені коефіцієнти у блоку *Polynomial*.

4.3 Завдання

- 1) Одержати перехідні процеси перетворення синусоїдального сигналу типовими нелінійностями та статичні характеристики цих блоків.
- 2) Реалізувати кусочно-лінійну апроксимацію, апроксимацію методом найменших квадратів та інтерполяцію кубічними поліномвми і кубічними сплайнами табличних нелінійностей, заданих в табл. 4.1.

4.4 Порядок виконання роботи

- 1) Зібрати модель, що складається із блоків типових нелінійностей, на вхід яких подається синусоїдальний сигнал, а виходи приєднуються до входів блоку *Multiport Switch*. Вихідний сигнал ключового елемента поєднується мультіплексором з вхідним синусоїдальним сигналом, і отриманий векторний сигнал приєднується до осцилографу та блоку *Out*. Підключити також у модель блок *XY Graph* для фіксації статичних характеристик типових нелінійностей.
- 2) Установити амплітуду і частоту синусоїди та обрати час моделювання рівним 2-м періодам синусоїди.
- 3) Шляхом керування блоком *Multiport Switch* у ручному режимі налаштувати по черзі кожен типову нелінійність, узгоджуючи її параметри з обраною амплітудою синусоїди у такий спосіб, щоб результати моделювання наочно демонстрували особливості кожного блоку.
- 4) Розробити програму, яка забезпечить по чергове дослідження усіх блоків і виведе в окремі вікна, поділені на 2 підвікна, графіки перехідних процесів вхідного і вихідного сигналів та статичні характеристики вхід-вихід досліджуваних нелінійних блоків.
- 5) Розробити модель, в якій нелінійна залежність вхід-вихід, задана таблично, реалізується 4-ма різними способами, оговореними у завдання. Вхідний сигнал змінюйте за лінійним законом у діапазоні зміни аргументів табличної функції. Усі сигнали виводьте у спільне вікно для їх порівняння.

Таблиця 4.1

№ вар.	Табличні функції									
1,2	x_t	-1	1	3	5	7	9	11	13	15
	y_t	8.71	109,8	124.4	122.5	112.1	96.6	80.2	63	57.9
3,4	x_t	2	3.2	4.4	6.2	7.8	9.5	5.9	11.5	12.7
	y_t	19.9	22	30	42.1	65	99.5	120	126.8	133.4
5,6	x_t	-3.5	-1.5	0.5	2.5	4.5	6.5	8.5	10.5	12.5
	y_t	0.45	-3.09	-4.01	-3.9	-3	-1.62	-0.18	0.99	1.72
7,8	x_t	1.25	2.59	4.4	6.54	8.5	11.5	13.5	14.5	15
	y_t	3.0	5.0	7.0	8.5	9.3	9.9	10.6	11.2	11.64
9,10	x_t	-2	0	2	4	6	8	10	12	14
	y_t	7.84	7.13	6.31	5.29	4.03	2.5	0.87	-0.68	-0.79
11,12	x_t	-1.5	1	2.7	5.5	6.5	8.3	9.6	11.2	12.75
	y_t	2.45	1.12	-1	-2.1	-2.3	-1.9	-1	2	3.5
13,14	x_t	0,67	1,5	2,5	3,5	5	6,5	10	12,4	14
	y_t	110	118,7	124,5	125,2	122,5	115,1	88,3	70	61,2
15,16	x_t	0,5	2,5	4,5	6,5	8,5	10,5	12,5	14,5	16,5
	y_t	23,7	20,1	27,8	45,3	79,2	115,4	132,9	141,1	147
17,18	x_t	-2,77	-0,5	1	2	3,5	7	10	11,5	12,5
	y_t	-1,5	-3,65	-4,03	-4,0	-3,54	-1,58	0,73	1,4	1,83
19,20	x_t	0,5	2,0	3,5	5,0	6,5	8,5	9,5	11,0	12,5
	y_t	1,23	0,92	0,78	0,68	0,6	0,53	0,49	0,47	0,45
21,22	xt	-1	1	3	5	7	9	11	13	15
	yt	1,02	2,57	5,51	7,52	8,69	9,38	9,79	10,35	11,64
23,24	xt	-3	0,5	1,5	2,5	4,3	6,2	7,7	9,0	11
	yt	9,4	7,52	6,75	5,8	3,6	0,53	-1,5	-2,94	-4,4
25	xt	-4	-2	0	2	4	6	8	10	12
	yt	3,1	2,66	1,74	0,35	-1,26	-2,28	-2,07	-0,54	2,53

4.5 Методичні вказівки та рекомендації

- 1) Організуйте виконання завдання в програмному режимі.
- 2) Для реалізації кусочно-лінійної апроксимації табличної нелінійності скористайтесь блоком *Look-Up Table*, для апроксимації методом найменших квадратів – блоком *Polynomial Fn*, а для кубічної інтерполяції та інтерполяції кубічними сплайнами – блоком *Matlab Fn*.
- 3) Вектор коефіцієнтів A степеневого полінома k -го порядку для апроксимації табличної нелінійності $Y(X)$ методом найменших квадратів знайдіть за допомогою функції `polyfit`:

$$A = \text{polyfit}(X, Y, k)$$

4.6 Контрольні питання

- 1) На які групи можна розподілити нелінійні блоки?
- 2) Що таке типові нелінійності?
- 3) В яких бібліотеках знаходяться блоки, що роблять систему нелінійною?
- 4) Які засоби моделювання табличних нелінійностей Ви знаєте?
- 5) Як реалізувати кусочно-лінійну апроксимацію табличної нелінійності?
- 6) Як здійснити апроксимацію табличної нелінійності степеневим поліномом методом найменших квадратів?
- 7) Як скористуватись функціями інтерполювання при моделюванні нелінійних функцій, заданих у вигляді таблиць?

Лабораторна робота №5

СТРУКТУРНЕ МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ КІЛ

5.1 Методика структурного математичного моделювання

У більшості випадків на першому етапі досліджень роблять такі припущення, які дозволяють вважати САУ лінійними та стаціонарними (*LTI-systems*). Методи аналізу та синтезу таких системи є найпростішими та добре розвинутими.

Для структурного моделювання на основі математичний опису треба утворити **структурну схему** досліджуваного об'єкту.

Неперервні стаціонарні динамічні системи (*LTI-systems* – *Linear TimeInvariante systems*) описуються сукупністю диференціальних та алгебраїчних лінійних та нелінійних рівнянь. Одним із універсальних способів розробки структурних схем таких об'єктів полягає у попередньому перетворенню диференціальних рівнянь довільного виду і порядку у систему диференціальних рівнянь першого порядку у нормальній формі Коші, у яких перші похідні від змінних стану виражені через вхідні, вихідні сигнали та змінні стану:

$$\begin{cases} \frac{dx_1(t)}{dt} = f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ \frac{dx_2(t)}{dt} = f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ \dots, \\ \frac{dx_n(t)}{dt} = f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \end{cases} \quad (5.1)$$

де

$$\begin{cases} y_1 = g_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k), \\ y_2 = g_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k), \\ \dots, \\ y_r = g_r(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k), \end{cases} \quad (5.2)$$

$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T$ – вектор змінних стану;

$\mathbf{y} = [y_1 \quad y_2 \quad \dots \quad y_r]^T$ – вектор вихідних сигналів;

$$\mathbf{u} = [u_1 \quad u_2 \quad \dots \quad u_k]^T \quad \text{— вектор вхідних сигналів;}$$

з початковими умовами

$$\mathbf{x}(t_0) = \mathbf{x}_0 = [x_{10} \quad x_{20} \quad \dots \quad x_{n0}]^T. \quad (5.3)$$

Рівняння (5.2) називають рівняннями виходу. Підстановкою (5.2) у (5.1) з диференціальних рівнянь виключаються вихідні сигнали і вони перетворюються у рівняння станів. Якщо функції у правих частинах рівнянь (5.1) і (5.2) лінійні то ці системи набувають вигляду (4.1) та (4.3).

Для складання структурної схеми диференціальні рівняння (5.1) треба записати в **операторній формі**. Для цього оператор диференціювання $\frac{d}{dt}$ замінюють оператором Лапласа s , і переходять від сигналів у просторі часу $x_i(t), u_i(t), y_i(t)$ до зображень цих сигналів у просторі змінної Лапласа $x_i(s), u_i(s), y_i(s)$:

$$\begin{cases} sx_1(s) = f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ sx_2(s) = f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r), \\ \dots, \\ sx_n(s) = f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_k, y_1, y_2, \dots, y_r). \end{cases} \quad (5.4)$$

У такий спосіб операція диференціювання у просторі часу замінюється операцією множення зображення сигналу на змінну Лапласа. Відповідно операції інтегрування відповідає операція ділення зображення сигналу на оператор Лапласа.

Структурну схему за математичним описом складають **методом неявних сигналів**, тобто перші похідні від змінних стану формують за рівняннями (5.1), не звертаючи уваги на те, що деякі з них є невідомими. Отримані похідні пропускають через інтегратори

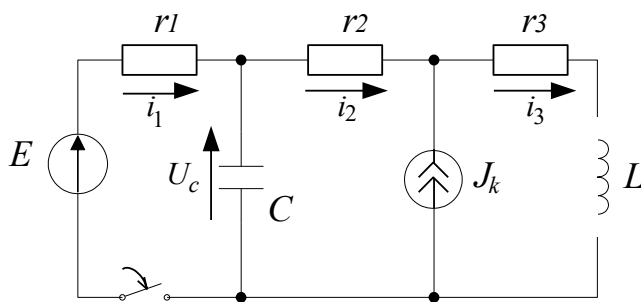
$$W_I(s) = \frac{1}{s}, \quad (5.5)$$

отримуючи на їхніх виходах зображення змінних стану. На останнє з відомих вхідних сигналів та змінних стану за рівняннями (5.2) формують вихідні сигнали.

Надалі діють у такій послідовності:

- структурну схему набирають із блоків Simulink, доповнюючи її блоками формування вхідних сигналів та реєстрації вихідних;
- виставляють параметри усіх блоків (краще задавати їх у вигляді змінних, а не констант);
- створюють, записують та виконують файл даних для ініціалізації моделі;
- розробляють план модельного експерименту і створюють програмні файли для його реалізації;
- налаштовують модель, обираючи метод розв'язання диференціальних рівнянь та його параметри;
- виконують власне симуляцію, фіксують, візуалізують та аналізують отримані результати.

5.2 Приклад структурного моделювання лінійного електричного кола



Як приклад, розглянемо розгалужену схему, зображену на рис. 5.1, для якої треба розрахувати перехідні процеси, тобто знайти $i_1(t)$, $i_2(t)$, $i_3(t)$, $U_c(t)$

Рис. 5.1. Досліджуване електричне коло

при замиканні ключа при таких параметрах: $E = 100$ В, $J_k = 5$ А, $C = 200$ мкФ, $L = 0,3$ Гн, $r_1 = 20$ Ом, $r_2 = 50$ Ом, $r_3 = 70$ Ом.

Математичний опис лінійних електричних кіл створюють, використовуючи закони Ома та Кірхгофа з урахуванням тієї обставини, що напруги і струми на резисторі індуктивності та конденсаторі пов'язані між собою співвідношеннями:

$$U_R(t) = RI_R(t), \quad U_L(t) = L \frac{dI_L(t)}{dt}, \quad I_C(t) = C \frac{dU_C(t)}{dt}. \quad (5.6)$$

Математичний опис схеми при замкненому стані ключа, згідно з законами Кірхгофа, має вигляд:

$$E = i_1 r_1 - U_c, \quad (5.7)$$

$$E = i_1 r_1 + i_2 r_2 + i_3 r_3 + L \frac{di_3}{dt}, \quad (5.8)$$

$$i_1 + C \frac{dU_c}{dt} = i_2, \quad (5.9)$$

$$i_2 + J_k = i_3. \quad (5.10)$$

Рівняння (5.8), (5.9) – диференційні; (5.7), (5.10) – алгебраїчні рівняння, які часто називають рівняннями зв'язку.

Перетворимо диференційні рівняння у нормальну форму Коші:

$$\begin{cases} \frac{dU_c}{dt} = (i_2 - i_1)/C, \\ \frac{di_3}{dt} = (E - i_1 r_1 - i_2 r_2 - i_3 r_3)/L. \end{cases} \quad (5.11)$$

Об'єднаємо змінні стану, вхідні та вихідні сигнали в одні відповідні вектори:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} U_c \\ i_3 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} E \\ J_k \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} U_c \\ i_3 \\ i_1 \\ i_2 \end{bmatrix} \quad (5.12)$$

Як бачимо, два перших вихідних сигналів співпадають зі змінними стану, а два останніх можна виразити через змінні стану з рівнянь зв'язку:

$$\begin{cases} i_1 = (E + U_c)/r_1, \\ i_2 = i_3 - J_k. \end{cases} \quad (5.13)$$

Запишемо ДР (5.11) в операторній формі:

$$\begin{cases} sU_c(s) = (i_2(s) - i_1(s))/C, \\ si_3(s) = (E(s) - i_1(s)r_1 - i_2(s)r_2 - i_3(s)r_3)/L. \end{cases} \quad (5.14)$$

Рівняння виходу (5.13) у зображеннях залишаються такими ж, як і у просторі часу.

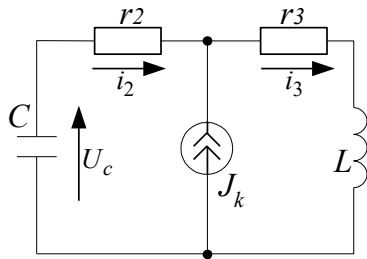


Рис. 6.2. Схема для визначення початкових умов

Розрахуємо початкові умови для заданої схеми, які збігаються з усталеними значеннями відповідних сигналів при розімкненому стані ключа. Така схема зображена на рис. 5.2.

Для складання структурної моделі лінійного неперервного динамічного об'єкта завжди можна обійтися тільки трьома типами блоків: інтеграторами, алгебраїчними суматорами та блоками множення на постійний коефіцієнт, які у *Simulink* називаються відповідно: *Integrator*, *Sum* та *Gain*. Ще декілька блоків знадобляться для запам'ятовування або для візуалізації результатів моделювання.

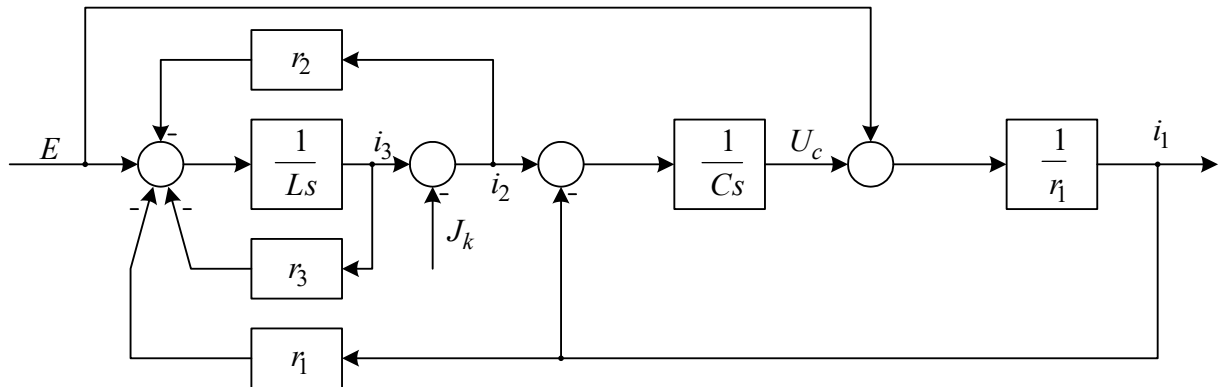


Рис. 5.3. Структурна схема досліджуваного електричного кола

Відповідна схема в блоках *Simulink* показана на рис. 5.4.

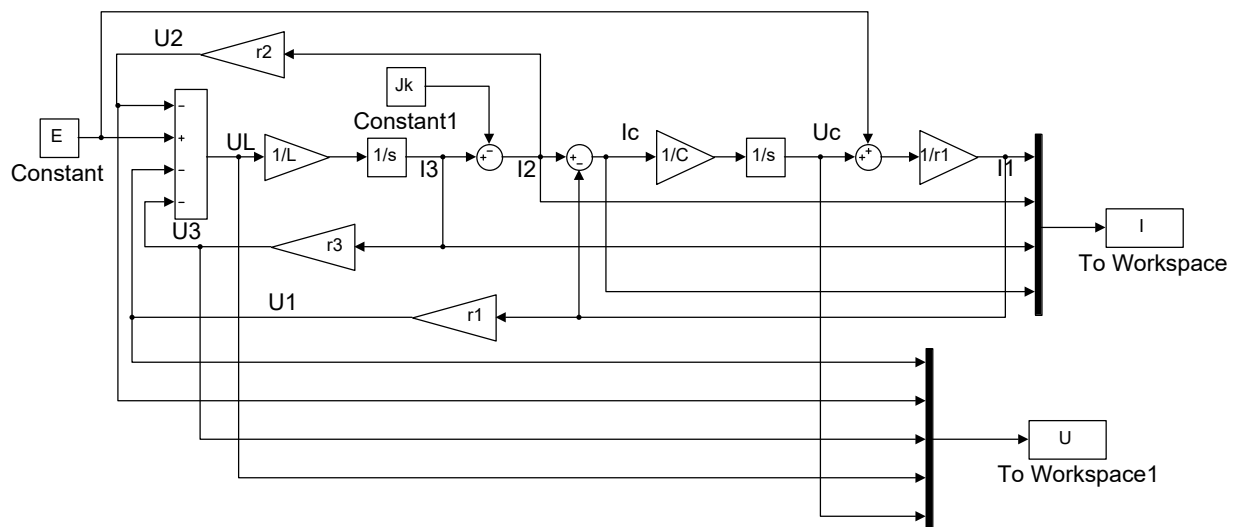


Рис. 5.4. Структурна *Simulink*-модель досліджуваного лінійного електричного кола

Результат роботи програми наведено на рис. 5.5.

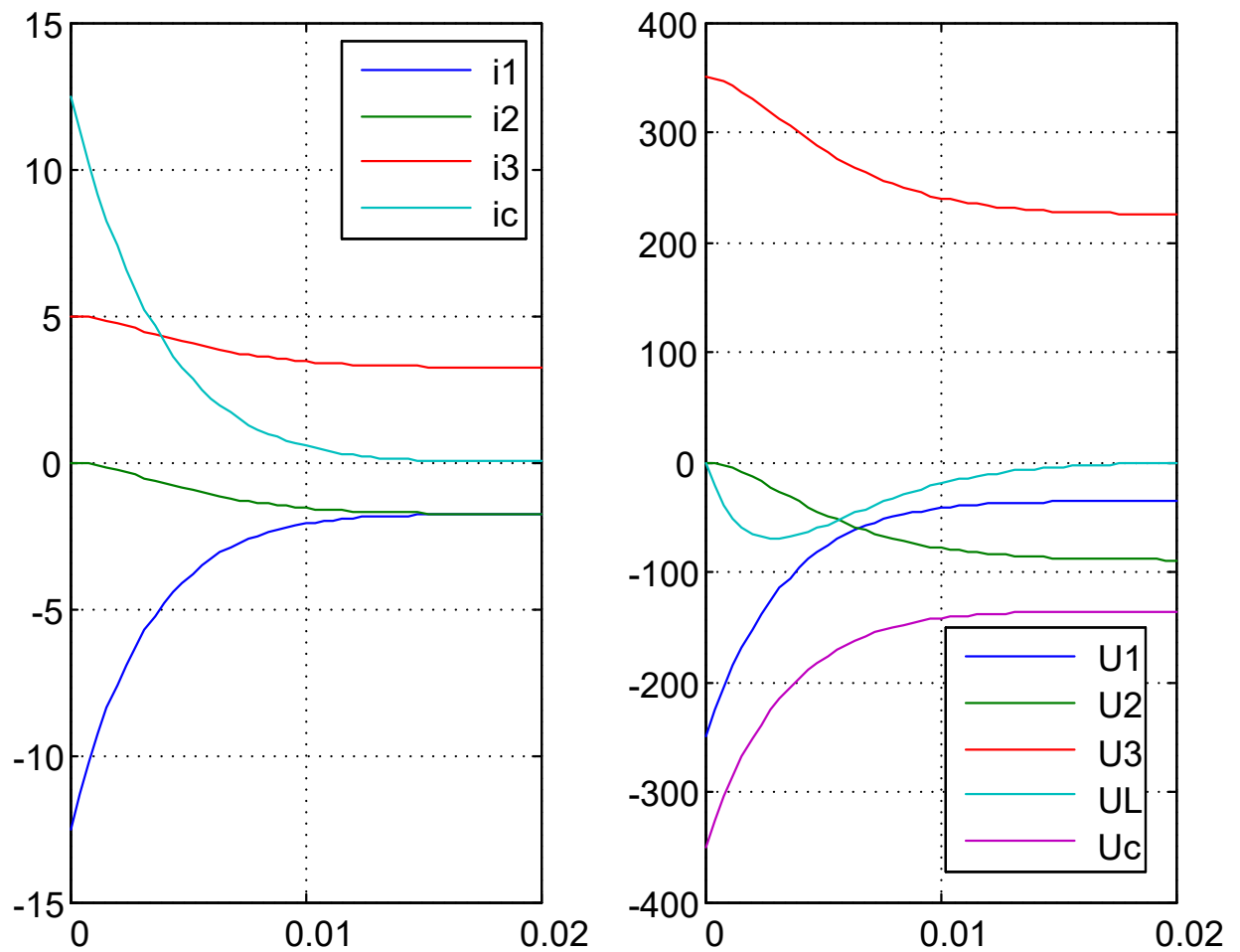


Рис. 7.5. Перехідні процеси у досліджуваному лінійному електричному колі

Якщо ЕРС та/або струми джерел змінюються у часі, то замість вхідних блоків *Const*, що генерують постійні сигнали E та J_k слід використати інші блоки бібліотеки *Sources*, наприклад, *Ramp*, *Sine Wave*, *Pulse Generator*, *Repeating Sequence*, тощо (див. лабораторну роботу 2).

7.3 Приклад структурного математичного моделювання нелінійного електричного кола

Якщо у досліджуваних динамічних об'єктах присутні елементи з нелінійними характеристиками «вхід-вихід», то диференційні рівняння, що їх описують, також стають нелінійними. Для цих випадків треба мати або аналітичний опис статичних характеристик нелінійних ланок, або таблиці від-

повідності аргументів та значень цих характеристик. За даними таких таблиць можна виконувати апроксимацію нелінійних залежностей методом найменших квадратів або інтерполювати їх на кожному кроці чисельного інтегрування.

У якості прикладу розглянемо електричну схему, подану на рис. 5.6, при $U = 80 \text{ В}$, $R = 4 \text{ Ом}$. Залежність магнітного потоку котушки індуктивності від струму намагнічування внаслідок ефекту насичення сталі є нелінійною. Графік цієї залежності має вигляд рис. 5.7.

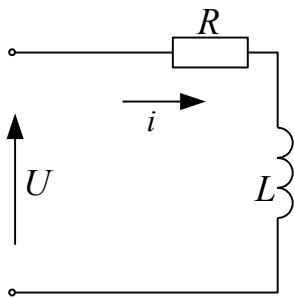


Рис. 5.6. Нелінійне електричне коло

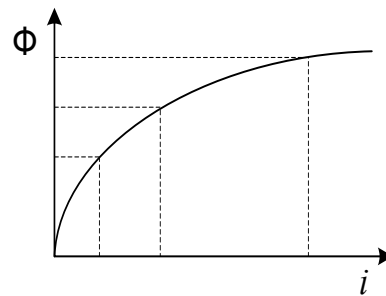


Рис. 5.7. Крива намагнічування індуктивності

Вона задана табличною функцією $\Phi_T(i_T)$, наприклад,

$$\begin{cases} i_T = [0 \ 0.02 \ 0.042 \ 0.067 \ 0.098 \ 0.141 \ 0.205 \ 0.3 \ 0.447 \ 0.669 \ 11.48] * 20, \\ \Phi_T = [0 : 0.1 : 1.1] * 3.2. \end{cases} \quad (5.16)$$

У такому разі індуктивність

$$L = \frac{d\Phi}{di} \quad (5.17)$$

буде змінною величиною.

Згідно із другим законом Кірхгофа напруга в електричній схемі визначається за формулою

$$U = iR + L \frac{di}{dt}. \quad (5.18)$$

Після підстановки (5.17) у (5.18) отримаємо

$$U = iR + \frac{d\Phi}{di} \cdot \frac{di}{dt} = iR + \frac{d\Phi}{dt},$$

або у нормальній формі Коші

$$\frac{d\Phi}{dt} = U - iR, \quad (5.19)$$

де

$$i = f(\Phi), \quad \Phi(0) = 0 \quad (5.20)$$

– нелінійне рівняння зв'язку та початкова умова відповідно.

Для відомої табличної залежності $\Phi_T = f(i_T)$ можна розрахувати $\Phi(i)$ для будь-якого значення i за допомогою апроксимації чи інтерполяції. При апроксимації кривих намагнічування звичайно використовують степеневі поліноми 3-го або 5-го порядку (обов'язково непарного, тому що функція $\Phi(i)$ центрально-симетрична).

У середовищі *Simulink* кусочно-лінійну апроксимацію здійснює блок *Look-Up-Table* з бібліотеки *Functions&Tables*. Параметрами цього блоку є вектор аргументів (*Vector of input values*) та вектор значень (*Table data*) табличної функції.

Для використання кубічної інтерполяції або інтерполяції сплайнами застосовують блок *MATLAB Fcn* з параметром *MATLAB function*: *interp1* (Ft, it, u, 'cubic') або *interp1* (Ft, it, u, 'spline') відповідно. Для використання апроксимації степеневим поліномом 5-го порядку застосовують блок *Polynomial* з параметром *Polynomial Coefficients*: $C = \text{polyfit}(it, Ft, 5)$.

Simulink-модель досліджуваної схеми зображена на рис. 5.8. Результати її симуляції наведено на рис. 5.9.

5.4 Завдання

1) Розрахувати перехідні процеси напруги на конденсаторі та струмів у гілках однієї з електричних схем, зображених на рис. 5.11, методом структурного математичного моделювання для трьох випадків:

- $U=U_m$ – джерело постійної напруги;
- $U=U_m \sin \omega t$ – джерело синусоїдальної напруги;
- періодична напруга (вигляд сигналу показано на рис. 5.12), $T=0.1$ с, $T1=0.07$ с.

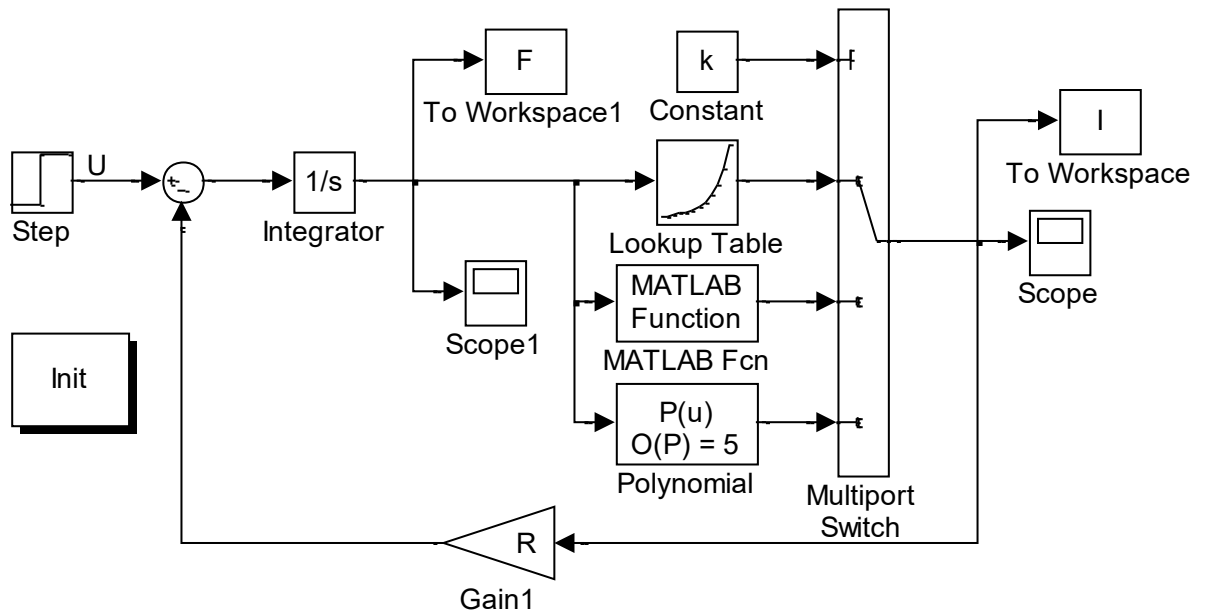


Рис. 5.8. Структурна *Simulink*-модель досліджуваного нелінійного електричного кола

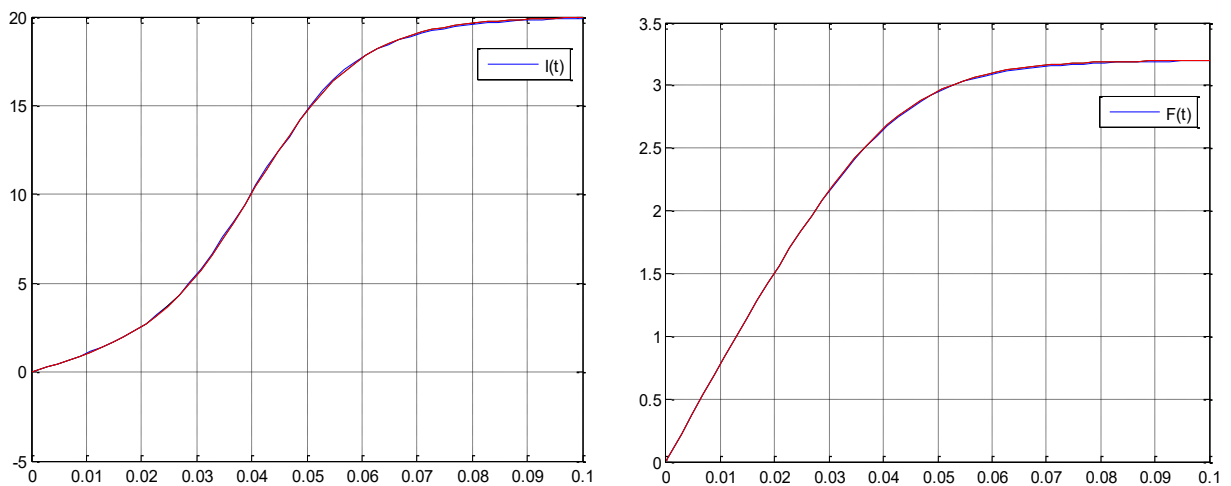


Рис. 5.9. Графіки перехідних процесів у досліджуваному нелінійному електричному колі

За результатами розрахунків побудувати графіки перехідних процесів.

Параметри схем надані у табл. 5.1.

2) Розрахувати перехідні процеси у схемах наданих на рис.5.13 методом структурного математичного моделювання. Параметри схем у табл. 5.2, характеристики нелінійних елементів L і R надані у табл.5.3. Побудувати графіки струмів у гілках електричної схеми у функції часу.

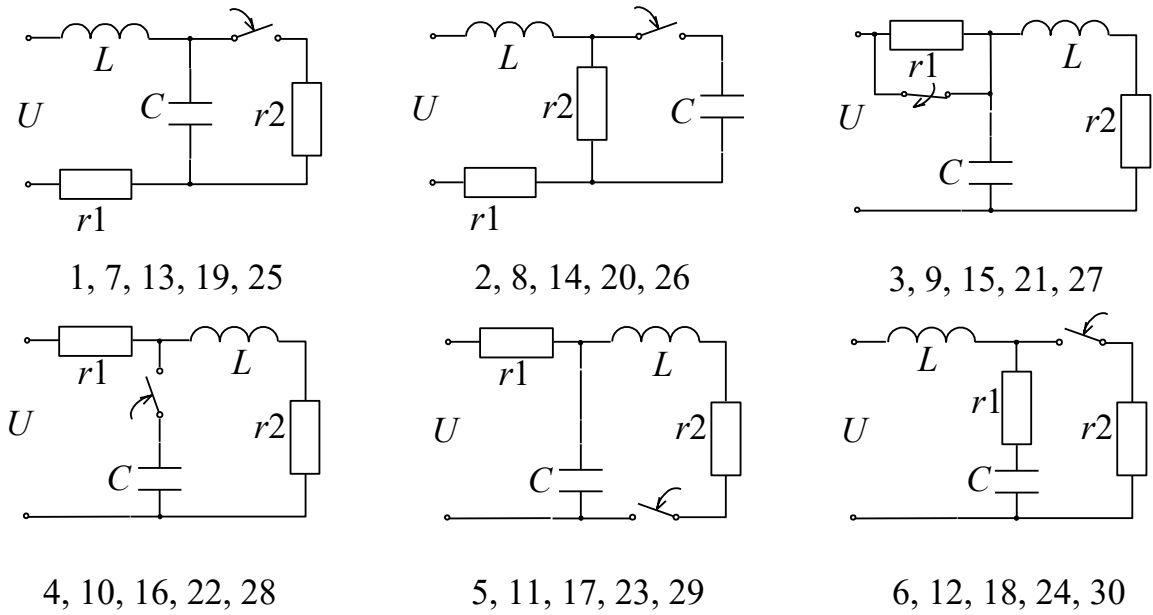


Рис. 5.11. Варіанти розгалужених лінійних електричних кіл

Таблиця 5.1

№ вар.	U_m , В	r_1 , Ом	r_2 , Ом	ω , с ⁻¹	L , Гн	C , мкФ
1-6	200	50	100	125	0,1	10
7-12	100	100	120	100	0,2	5
13-18	300	60	120	75	0,15	15
19-24	400	150	100	150	0,25	2
25-30	500	100	200	100	0,05	10

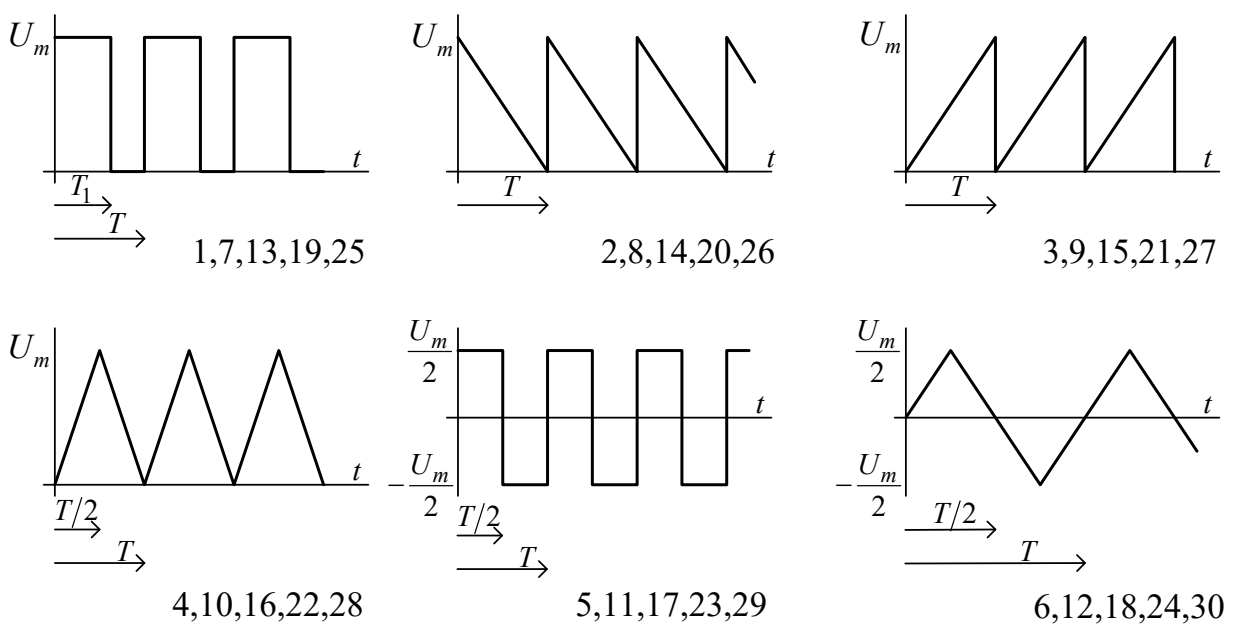


Рис. 5.12. Варіанти входних періодичних сигналів

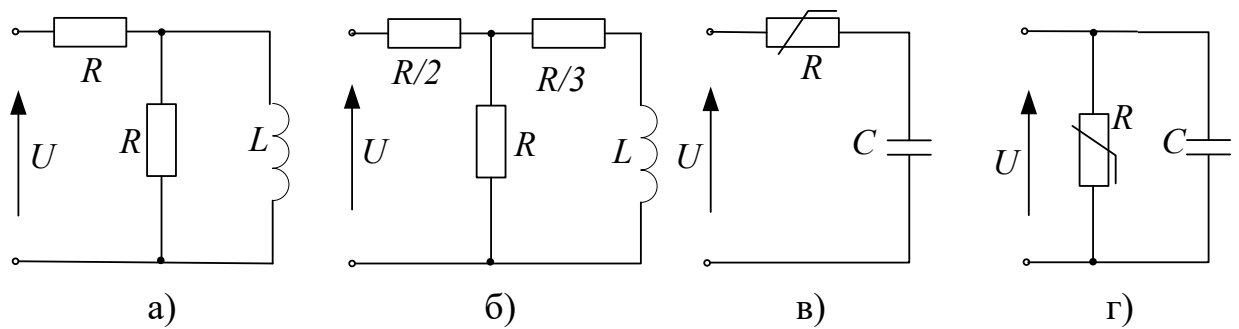


Рисунок 5.13

Таблиця 5.2

$U=E_m, \text{В}$	№ вар.	№ сх.	$R, \text{Ом}$	№ вар.	№ сх.	$C, \text{мкФ}$
200	1	а	21	7	в	150
220	2	а	23	8	в	170
240	3	а	25	9	в	190
250	4	б	22	10	г	100
270	5	б	24	11	г	120
300	6	б	26	12	г	140

Таблиця 5.3

$\Phi, \text{Вб}$	0	0.18	0.33	0.45	0.55	0.63	0.7	0.753	0.8	0.835	0.865
$i, \text{А}$	0	2	4	6	8	10	12	14	16	18	20
$U, \text{В}$	0	66.4	118	158	190	214	233	248	259	268	275
$i, \text{А}$	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

5.5 Порядок виконання роботи

1) Ознайомитися з методами складання математичного опису електричних кіл та розробки їх структурних моделей.

2) Виконати математичний опис досліджуваних електричних кіл, перетворити диференційні рівняння до нормальної форми Коші та записати їх в операторній формі.

3) За отриманими операторними рівняннями розробити структурні схеми.

4) Перейти від структурних схем до структурних моделей із застосуванням блоків *Simulink*-бібліотек, ввести їх у комп'ютер, встановити параметри блоків, обрати метод розв'язання ДР та його параметри.

- 5) Забезпечити програмно ініціалізацію моделей.
- 6) Виконати розрахунок перехідних процесів досліджуваних електричних кіл за допомогою створених моделей, візуалізувати та проаналізувати результати.

5.6 Методичні вказівки

- 1) Параметри моделей задавайте іменами змінних, а не константами.
- 2) Для попередньої візуалізації використовуйте блоки *Scope*, а для запам'ятовування чисельних результатів з метою подальшої побудови графіків перехідних процесів – вихідні порти *Out* та блоки *ToWorkspace*. Усі струми виводьте на один графік, а напруги – на інший.
- 3) Ініціалізацію моделей та виконайте за допомогою «кнопок», створених з блоків *Subsystem* (див. лабораторну роботу 5).
- 4) Час розрахунку перехідного процесу обирайте таким, щоб усталений режим був досягнутий, але не тривав дуже довго.
- 5) При моделюванні нелінійних електричних кіл апробуйте різні способи апроксимації нелінійних функціональних залежностей, заданих таблично.

5.7 Контрольні питання та завдання

- 1) Які закони теоретичної електротехніки використовують для математичного опису електричних кіл?
- 2) Запишіть рівняння напруги на індуктивному елементі та струму, що протікає через конденсатор.
- 3) Які способи апроксимації нелінійних табличних залежностей Ви знаєте? Як їх реалізувати блоками *Simulink*?
- 4) Як записати диференціальні рівняння у нормальній формі Коші?
- 5) Як створити структурну схему електричного кола з його математичного опису
- 4) Які методи розв'язання систем диференціальних рівнянь Ви знаєте?

*Лабораторна робота №6***МОДЕЛЮВАННЯ ДВИГУНА ПОСТІЙНОГО СТРУМУ З
КЕРУВАННЯМ У КОЛІ ЯКОРЯ****6.1 Математичний опис об'єкту моделювання**

Двигун постійного струму (ДПС) з незалежним збудженням широко застосовується у регульованих електроприводах, що працюють у напружених повторно-короткочасних режимах і потребують забезпечення високої якості перехідних процесів.

При складанні його математичного опису знехтуємо розмагнічуючою дією реакції якоря, падінням напруги на щітках та тертям, а індуктивність і активний опір якірного кола та сумарний момент інерції вважатимемо постійними величинами. Регулювання швидкості здійснюватимемо зміною напруги якоря при постійній напрузі збудження. При прийнятих припущеннях ДПС описується лінійними диференціальними та алгебраїчними рівняннями [17, 18, 20]:

$$U_{\text{я}}(t) - E_{\text{д}}(t) = \Delta U_{\text{я}}(t) = I_{\text{я}}(t)R_{\text{я}} + L_{\text{я}} \frac{dI_{\text{я}}(t)}{dt}, \quad (6.1)$$

$$M(t) = cI_{\text{я}}(t), \quad (6.2)$$

$$M(t) - M_{\text{с}}(t) = M_{\text{ж}}(t) = J \frac{d\omega(t)}{dt}, \quad (6.3)$$

$$E_{\text{д}}(t) = c\omega(t), \quad (6.4)$$

де

$U_{\text{я}}, I_{\text{я}}$ – напруга та струм якоря;

$\Delta U_{\text{я}}$ – падіння напруги у якірному колі;

$R_{\text{я}}, L_{\text{я}}$ – активний опір та індуктивність якоря;

$E_{\text{д}}, \omega$ – електрорушійна сила (ЕРС) та кутова швидкість двигуна;

$M, M_{\text{с}}, M_{\text{ж}},$ – електромагнітний момент двигуна, момент статичного опору та динамічний момент;

J – момент інерції;

$$c = k\Phi_{\text{зн}}; \quad (6.5)$$

$\Phi_{\text{зн}}$ – номінальний потік збудження двигуна;

$k = \frac{pN}{2\pi a}$ – конструктивний коефіцієнт двигуна;

p, a, N – кількість пар полюсів, паралельних гілок та активних провідників відповідно.

Рівняння (6.1) називають *рівнянням електромагнітної рівноваги якорного кола*, що складається за другим законом Кірхгофа, рівняння (6.3) – *рівнянням руху* двигуна, що складається за другим законом Кірхгофа для тіл обертального руху. Рівняння моменту (6.2) та ЕРС (6.4) свідчать про те, що при постійному потоку збудження електромагнітний момент двигуна є пропорційним струму якоря, а його ЕРС є пропорційною кутовій швидкості двигуна.

Для складання структурної схеми в цьому разі застосуємо дещо інший підхід, ніж у попередній лабораторній роботі, а саме, запишемо ДР в операторному вигляді без попереднього перетворення їх у нормальну форму Коші:

$$U_{\text{я}}(s) - E_{\text{д}}(s) = \Delta U_{\text{я}}(s) = I_{\text{я}}(s)R_{\text{я}} + L_{\text{я}}sI_{\text{я}}(s), \quad (6.6)$$

$$M(s) - M_{\text{с}}(s) = M_{\text{ж}}(s) = Js\omega(s). \quad (6.7)$$

З отриманих рівнянь визначаємо передавальні функції якорного кола і механічної частини двигуна як відношення зображень вихідного та вхідного сигналів:

$$W_{\text{я}}(s) = \frac{I_{\text{я}}(s)}{U_{\text{я}}(s) - E_{\text{д}}(s)} = \frac{I_{\text{я}}(s)}{\Delta U_{\text{я}}(s)} = \frac{1/R_{\text{я}}}{T_{\text{я}}s + 1}, \quad (6.8)$$

$$\text{де } T_{\text{я}} = L_{\text{я}} / R_{\text{я}} \quad (6.9)$$

– *електромагнітна стала часу якорного кола*.

$$W_{\text{м}}(s) = \frac{\omega(s)}{M(s) - M_{\text{с}}(s)} = \frac{\omega(s)}{M_{\text{ж}}(s)} = \frac{1}{Js}. \quad (6.10)$$

При дослідженні динамічних властивостей ДПС долучно ввести до розгляду, крім електромагнітної сталої часу ще й *електромеханічну сталу часу двигуна*

$$T_{\text{м}} = \frac{JR_{\text{я}}}{c^2}. \quad (6.11)$$

При її використанні передавальна функція механічної частини двигуна набуває вигляду

$$W_M(s) = \frac{R_{\text{я}}}{T_M c^2 s}. \quad (6.12)$$

Структурна схема ДПС, складена за рівняннями (6.8), (6.2), (6.12) та (6.4), зображена на рис. 6.1.

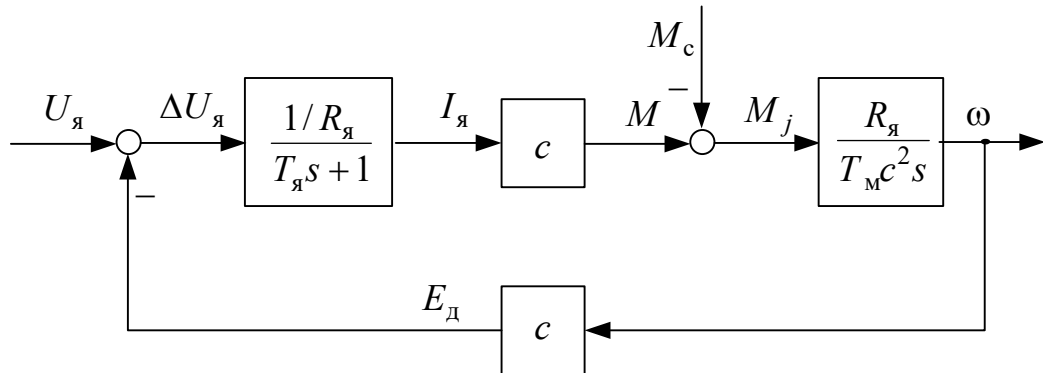


Рис. 6.1. Структурна схема двигуна постійного струму з нерегульованим незалежним збудженням

6.2 Завдання

Промодельуйте двигун постійного струму з незалежним збудженням (ТП-Д) при постійному потоку збудження з даними, які наведені в табл. 6.1, в таких режимах:

- при стрибкоподібній зміні напруги двигуна від 0 до номінального значення;
- при лінійній зміні напруги двигуна від 0 до номінального значення за час t_{p0} , що обирається з умов забезпечення бажаної величини динамічного моменту;
- при стрибкоподібному прикладанні активного номінального моменту статичного опору.
- при лінійній зміні напруги двигуна до номінального значення, стрибкоподібному накиданні і скиданні номінального навантаження на усталеній швидкості та лінійній зміні напруги якоря від номінального значення до нуля без навантаження.

Таблиця 6.1

№ варі- анту	Тип дви- гуна	$P_{\text{н}}$	$n_{\text{н}}$	$I_{\text{ян}}$	$R_{\text{я}}$	$2p$	$w_{\text{п}}$	R_3	$I_{\text{зн}}$	$\Phi_{\text{н}}$	J
		<i>кВт</i>	<i>об/хв</i>	<i>А</i>	<i>Ом</i>	—	<i>вит</i>	<i>Ом</i>	<i>А</i>	<i>мВб</i>	<i>кг·м²</i>
$U_{\text{ян}} = 440 \text{ В}, U_{\text{зн}} = 440 \text{ В}$											
1	П81	32	1500	83	0,250	4	2000	208,4	1,62	14,4	0,68
2	П82	25	1000	66	0,373	4	1525	137,6	2,46	19,8	0,78
3	П91	55	1500	143	0,143	4	1615	129,2	2,62	19,6	1,48
4	П101	42	750	107	0,206	4	1660	130,0	2,60	29,0	2,58
5	П81	19	1000	52	0,630	4	2000	208,4	1,62	14,4	0,68
6	П101	100	1500	256	0,051	4	1660	130,0	2,60	29,9	2,58
7	П102	125	1500	316	0,034	4	1450	101,5	3,33	38,3	3,00
8	П111	95	1000	240	0,067	4	1460	87,0	3,89	19,0	5,10
9	П111	160	1500	400	0,025	4	1460	87,0	3,89	20,3	5,10
10	П112	85	750	220	0,075	4	1310	82,0	4,13	25,3	5,57
11	П112	180	1500	450	0,023	4	1310	82,0	4,13	16,2	5,75
$U_{\text{ян}} = 220 \text{ В}, U_{\text{зн}} = 110 \text{ В}$											
12	П112	200	1500	1000	0,050	4	750	24,0	6,55	16,2	5,75
13	П52	14	3000	74	0,027	4	1200	94,8	1,79	7,2	0,10
14	П61	19	3000	100	0,057	4	1600	104,0	1,63	8,2	0,14
15	П72	25	1500	132	0,083	4	1100	67,0	2,34	12,5	0,40
16	П111	160	1500	809	0,064	4	850	28,0	5,61	18,6	5,10
17	П102	125	1500	632	0,079	4	840	32,5	4,84	13,0	3,00
18	П101	100	1500	508	0,013	4	950	37,8	4,16	13,7	2,58
19	П92	75	1500	381	0,014	4	830	31,8	4,94	20,1	1,75
20	П22	1	1500	5,9	4,170	2	4800	600,0	0,30	3,2	0,01
21	П42	7,4	3000	39,8	0,205	4	1350	136,0	1,24	5,1	0,05
22	П52	14	3000	74	0,027	4	1200	94,8	1,79	7,2	0,09

У таблиці позначено:

P_n , Φ_n , n_n – номінальні значення потужності, потоку збудження та кутової швидкості в обертах за хвилину;

$U_{ян}$, $U_{зн}$ – номінальні значення напруги якоря та напруги збудження;

$I_{ян}$, $I_{зн}$ – номінальні значення струму якоря двигуна та струму збудження.

6.3 Порядок виконання роботи

1) Складіть структурну математичну модель ДПС за його структурною схемою. Для вмикання і вимикання вхідних сигналів використовуйте ключові елементи з бібліотеки *Signal Routing*.

2) За даними табл. 6.1 розрахуйте параметри структурної моделі, часові параметри вхідних сигналів та орієнтовні тривалості перехідних процесів. Розрахунок виконайте за допомогою програмного файлу.

3) Виконайте дослідження двигуна у чотирьох заданих режимах при розрахованих параметрах. Потім перші два режими, прописані у завданні, промодельуйте спочатку для трьох значень електромагнітної сталої часу, вважаючи, що вона змінюється за рахунок зміни індуктивності, а потім для трьох значень електромеханічної сталої часу, вважаючи, що вона змінюється за рахунок моменту інерції. Третій режим промодельуйте при різних варіаціях активного опору якоря.

4) Проаналізуйте статичні та динамічні властивості ДПС та їх залежність від параметрів двигуна.

5) Створіть із структурної моделі двигуна незамасковану та замасковану підсистеми.

6.4 Методичні вказівки і рекомендації

1) При моделюванні фіксуйте напругу якоря, момент або струм якоря та кутову швидкість або ЕРС двигуна у відносних одиницях. У якості базових величин оберіть для напруги, моменту та струму якоря номінальні величини, а для швидкості – швидкість ідеального холостого ходу

$$\omega_0 = \frac{U_{\text{ян}}}{c}. \quad (6.13)$$

Усі графіки зображуйте в одній системі координат.

2) Перехід від кутової швидкості в обертах за хвилину (технічна система одиниць) до кутової швидкості у радіанах за секунду (міжнародна система одиниць СИ) здійснюйте за формулою

$$\omega_{\text{н}} = \frac{\pi n_{\text{н}}}{30}. \quad (6.14)$$

Після цього можна розрахувати номінальний момент

$$M_{\text{н}} = \frac{P_{\text{н}}}{\omega_{\text{н}}}, \quad (6.15)$$

коефіцієнт моменту та ЕРС

$$c = \frac{M_{\text{н}}}{I_{\text{ян}}},$$

швидкість ідеального холостого ходу

$$\omega_0 = \frac{U_{\text{ян}}}{c}, \quad (6.16)$$

статичну просадку швидкості при номінальному навантаженні

$$\Delta\omega_{\text{с}} = \frac{I_{\text{ян}} R_{\text{я}}}{c} \quad (6.17)$$

та струм короткого замикання

$$I_{\text{кз}} = \frac{U_{\text{ян}}}{R_{\text{я}}}. \quad (6.18)$$

2) Індуктивність якоря розрахуйте за емпіричною формулою Уманського-Лінвіля

$$L_{\text{я}} = k \frac{U_{\text{ян}}}{I_{\text{ян}} \omega_{\text{н}} p}, \quad (6.19)$$

де $k = 0.6$ для двигунів з компенсаційною обмоткою, $k = 0.25$ для двигунів без компенсаційної обмотки.

3) Орієнтовний час перехідного процесу для першого і третього режимів складає

$$t_{k1} = t_{k3} \approx 4T_{\text{м}} + T_{\text{я}};$$

для другого режиму

$$t_{k2} = t_{\text{p0}} + t_{k1},$$

де

$$t_{\text{p0}} = \frac{J\omega_0}{M_j}; \quad (6.20)$$

у четвертому режимі час накидання навантаження розрахуємо як

$$t_{\text{нн}} = t_{k2} + t_{k1},$$

час скидання навантаження – як

$$t_{\text{сн}} = t_{\text{нн}} + t_{k1},$$

час початку гальмування – як

$$t_{\text{пг}} = t_{\text{сн}} + t_{k1}$$

та загальний час перехідного процесу –

$$t_{k4} = t_{\text{пг}} + t_{k2}.$$

6.5 Контрольні питання і завдання

- 1) Як скласти структурну схему об'єкта моделювання за його математичним описом?
- 2) Поясніть, чому не можна здійснювати прямий пуск ДПС.
- 3) На яку фізичну величину впливає значення похідної від напруги якоря при розгоні та гальмуванні двигуна?
- 4) Як впливає електромагнітна стала часу якірного кола на максимальне значення струму при стрибкоподібній зміні $U_{\text{я}}$.
- 5) Як впливає електромеханічна стала часу привода на його швидкість?
- 6) Як впливає величина активного опору якоря на просадку його швидкості при накиді навантаження?

*Лабораторна робота №7***МОДЕЛЮВАННЯ ДВИГУНА ПОСТІЙНОГО СТРУМУ
З КЕРУВАННЯМ У КОЛІ ЯКОРЯ ТА КОЛІ ЗБУДЖЕННЯ****7.1 Математичний опис об'єкту моделювання**

У ДПС зі змінним потоком збудження рівняння електричної рівноваги та рівняння руху не відрізняються від відповідних рівнянь (7.1) та (7.2) при постійному потоці, але рівняння моменту і ЕРС набувають вигляду [18, 20]

$$M(t) = k \Phi_3(t) I_{\text{я}}(t), \quad (7.1)$$

$$E_{\text{д}}(t) = k \Phi_3(t) \omega(t). \quad (7.2)$$

До них ще додається рівняння електричної рівноваги кола збудження:

$$U_3(t) = i_3(t) R_3 + L_3 \frac{di_3(t)}{dt}. \quad (7.3)$$

де

U_3, i_3 – напруга та струм збудження;

R_3, L_3 – активний опір та індуктивність обмотки збудження.

При врахуванні так званого «ефекту насичення сталі», тобто нелінійності кривої намагнічування двигуна $\Phi_3(i_3)$, індуктивність обмотки збудження змінюється за законом

$$L_3 = k_3 \frac{d\Phi_3(t)}{di_3(t)}, \quad (7.4)$$

де

$$k_3 = 2p\xi w_n, \quad (7.5)$$

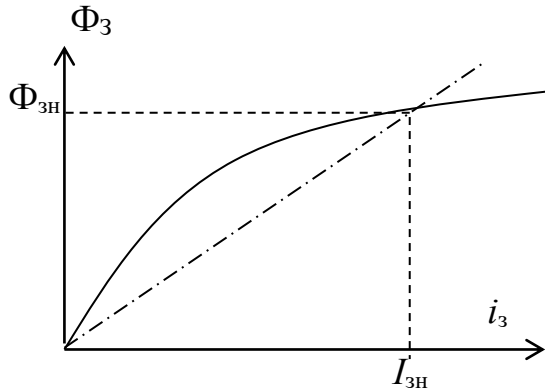
w_n – кількість витків обмотки збудження на полюс,

$\xi = 1 + (0.5 \div 0.7)(\sigma - 1)$ – коефіцієнт, що враховує вплив потоку розсіювання на індуктивність обмотки збудження і ту обставину, що частина потоку розсіювання зчеплена не з усіма витками обмотки,

$\sigma = (1.12 \div 1.18)$ – коефіцієнт розсіювання.

Після підстановки (7.4) у (7.3) отримуємо коефіцієнт

$$U_3(t) = i_3(t)R_3 + k_3 \frac{d\Phi_3(t)}{dt}. \quad (7.6)$$



Загальний вигляд кривої намагнічування подано на рис. 7.1. Зазвичай проєктувальники визначають її експериментально і подають у довідниках у вигляді таблиці.

Рис. 7.1. Крива намагнічування ДПС

Коло збудження характеризується *сталюю часу збудження*

$$T_3 = \frac{L_3(i_3, \Phi_3)}{R_3} = \frac{k_3}{R_3} \cdot \frac{d\Phi_3(t)}{di_3(t)} = \text{var}. \quad (7.7)$$

Усереднене значення сталої збудження визначають із лінеаризованої характеристики намагнічування за формулою

$$T_{3Н} = \frac{L_{3Н}}{R_3} = \frac{k_3}{R_3} \cdot \frac{\Phi_{3Н}}{i_{3Н}} = \frac{k_3 k_{\Phi Н}}{R_3} = \text{const}, \quad k_{\Phi Н} = \frac{\Phi_{3Н}}{i_{3Н}}. \quad (7.8)$$

Після ділення обох частин рівняння (7.6) на R_3 з урахуванням (7.8) отримуємо

$$\frac{U_3(t)}{R_3} = i_3(t) + \frac{T_{3Н}}{k_{\Phi Н}} \cdot \frac{d\Phi_3(t)}{dt}. \quad (7.9)$$

Позначимо

$$i_3^*(t) = \frac{U_3(t)}{R_3} \quad (7.10)$$

– завдання на зміну струму збудження.

Структурна схема ДПС, складена за рівняннями (6.1), (7.1), (6.3), (7.2) і (6.9) зі врахуванням позначення (6.10), зображена на рис. 7.2.

Треба відзначити, що, додатково до припущень, прийнятих у лабораторній роботі №7, модель рис.7.2 не враховує дію вихрових струмів, які виникають у масивних частинах магнітної системи потужних двигунів при зміні магнітного потоку.

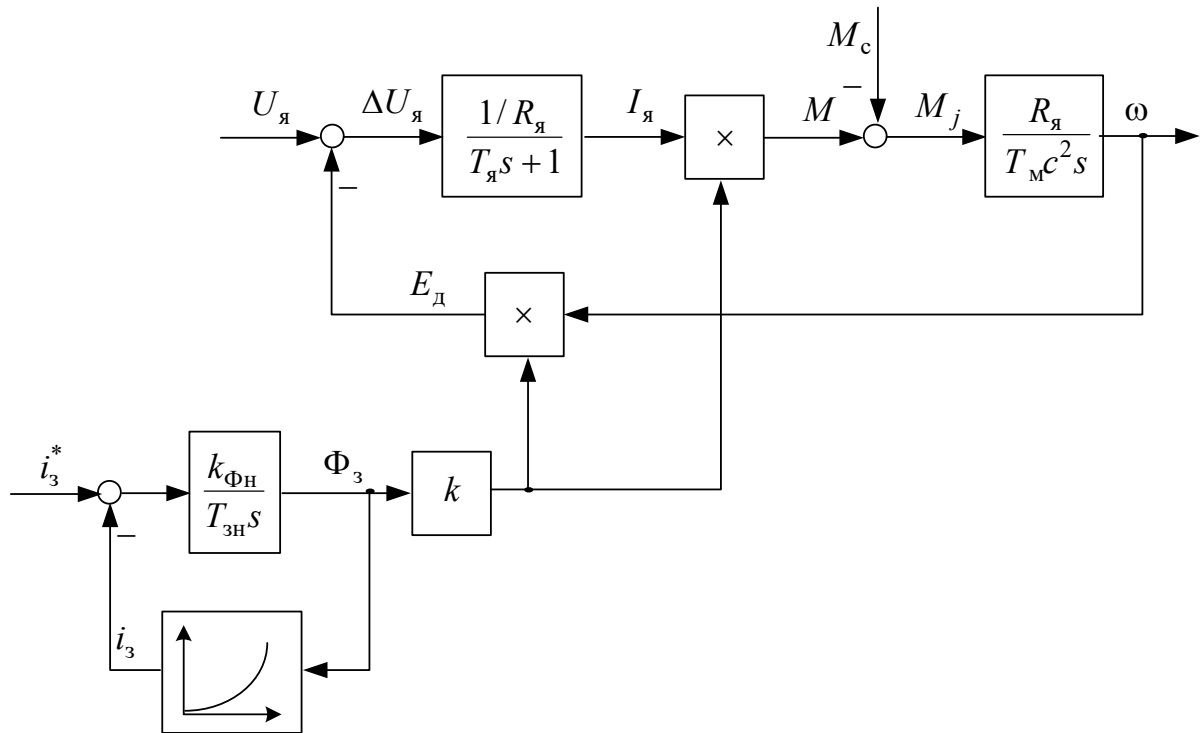


Рис. 7.2. Структурна схема двигуна постійного струму з керуванням у колах якоря і збудження

7.2 Завдання

Промодельуйте ДПС з керуванням напругою якоря та напругою збудження, змінюючи керовані напруги та момент статичного опору у такій послідовності:

- подайте стрибком на відповідний вхід моделі номінальну напругу збудження;
- після досягнення потоком збудження номінального значення змінійте лінійно напругу якоря від нуля до номінальної величини;
- у момент досягнення напругою якоря усталеного значення змінійте напругу збудження у такий спосіб, щоб завдання на ослаблення магнітного поля відбулося за законом

$$\Phi_3^*(t) = \frac{U_{я\text{н}}}{k \omega^*(t)}, \quad (7.11)$$

де

$$\omega^*(t) = \begin{cases} \omega_0 & \text{при } 0 \leq t \leq t_{\text{по}}, \\ \frac{\omega_0}{t_{\text{р0}}} t & \text{при } t_{\text{по}} \leq t \leq t_{\text{по}} + t_{\text{р0}}, \\ 2\omega_0 & \text{при } t \leq t_{\text{по}} + 2t_{\text{р0}}, \end{cases} \quad (7.12)$$

$t_{\text{по}}$ – час початку ослаблення поля;

- після досягнення двигуном швидкості $2\omega_0$ накиньте на нього стрибком номінальне навантаження.

Повторіть модельний експеримент при ослабленні магнітного поля стрибком.

Координати вузлових точок кривої намагнічування у відносних одиницях подані у табл. 7.1.

Таблиця 7.1

$\bar{I}_3$	0	0,02	0,042	0,067	0,1	0,14	0,205	0,3	0,447	0,67	1	1,48
$\bar{\Phi}_3$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	1,1

7.3 Порядок виконання роботи

- 1) Ознайомтесь з математичним описом двигуна і його структурною схемою.
- 2) Складіть структурну математичну модель ДПС за його структурною схемою.
- 3) Перерахуйте дані табл. 7.1 з відносних одиниць у абсолютні.
- 4) За даними табл. 6.1 і перерахованої табл. 9.1 розрахуйте параметри структурної моделі, часові параметри вхідних сигналів та орієнтовну тривалість перехідних процесів. Розрахунок виконайте за допомогою програмного файлу.
- 5) Розробіть структурні моделі вузлів для формування вхідних сигналів.
- 6) Виконайте дослідження двигуна за завданням.
- 7) Проаналізуйте статичні та динамічні властивості ДПС з ослабленим потоком збудження, порівняйте їх з результатами попередньої роботи.

7.4 Методичні вказівки і рекомендації

2) При моделюванні фіксуйте напругу якоря, електромагнітний момент, струм якоря, кутову швидкість, ЕРС двигуна, потік збудження та струм збудження у відносних одиницях, як у попередній лабораторній роботі. Усі графіки зображуйте в одній системі координат.

3) Час намагнічування двигуна визначте за приблизною формулою

$$t_3 \approx 4T_{3H};$$

час наростання напруги якоря від 0 до номінального значення та – за формулою (5.16), а час початку ослаблення поля за формулою

$$t_{по} = t_3 + t_{p0}.$$

Інші часові інтервали обирайте у процесі виконання модельного експерименту.

4) Сигнали $U_{я}(t)$ та $\omega_3^*(t)$ утворюйте блоками *Lookup Table*, для формування сигналу $\Phi_3^*(t)$ за формулою (6.11) використайте блок *Constant* зі значенням $U_{яH}/k$ і блок ділення, сигнал $\Phi_3^*(t)$ перетворіть у сигнал $i_3^*(t)$ за допомогою нелінійного функціонального перетворення $i_3^*(\Phi_3^*)$ блоком *Lookup Table*.

5) Для формування нелінійної функціональної залежності $i_{\mu}^* = f(\Phi^*)$ спробуйте використати кусково-лінійну інтерполяцію, кубічну інтерполяцію та апроксимацією степеневим поліномом 5-го порядку. Порівняйте результати. Обґрунтуйте обраний варіант.

7.5 Контрольні питання і завдання

1) Запишіть рівняння, на підставі яких побудована структурна схема рис. 7.2.

2) Як визначається індуктивність обмотки збудження при врахуванні нелінійності характеристики намагнічування?

3) Поясніть збільшення струму якоря та просадки швидкості при осла-

бленні поля двигуна.

- 4) Обґрунтуйте зменшення швидкодії у зоні ослаблення поля.
- 5) Як змінюється електромеханічна стала часу двигуна при регулюванні потоку збудження? На що це впливає?
- 6) Як змінюється електромагнітна стала часу збудження при регулюванні потоку?
- 7) Чому потік збудження прагнуть змінювати за формулою (6.11)?
- 8) Поясніть різницю в перехідних процесах при розгоні приводу за рахунок зміни напруги якоря і напруги збудження.
- 9) Чим відрізняються процеси накиду навантаження при номінальному й ослабленому потоках збудження.

*Лабораторна робота №8***СТВОРЕННЯ ПІДСИСТЕМ ТА ЇХ МАСКУВАННЯ****8.1 Теоретичні відомості****8.1.1 Створення підсистем**

Підсистеми використовують для зменшення кількості блоків в моделі, що сприяє її компактності, та для об'єднання фрагментів моделі, що виконують деяку спільну функцію, в єдине ціле.

Створити підсистему можна двома способами.

Перший з них полягає у тому, що в вікно моделі вставляється блок, який при подвійному щиглику мишею по його піктограмі відчиняє власне вікно, в якому збирається модель підсистеми. До входів підсистеми приєднують входні (*In*), а до виходів – вихідні (*Out*) порти.

Згідно з другим способом вже існуючий фрагмент моделі виділяється та об'єднується в підсистему командою *Edit* → *Create Subsystem* (^G). В цьому разі входні та вихідні порти додаються до фрагменту, що утворює підсистему, автоматично.

Блок *Subsystem* може бути відображеним на екрані в режимах *Hide Port Labels* (імена портів сховані) та *Show Port Labels* (з зображенням імен портів), які обираються користувачем через меню *Format*. Якщо підсистема має декілька входів та виходів, то стиль *Show Port Labels* є кращим, бо він дозволяє безпомилково з'єднувати блок *Subsystem* з іншими блоками, не звіряючись для перевірки з моделлю підсистеми.

Створену підсистему можна редагувати, зокрема в ній можна змінити власне ім'я та імена портів, що підвищує наочність нового макроблоку. Для наочності імена портів бажано називати іменами входних та вихідних змінних моделі. Створену підсистему також бажано перейменувати у відповідності з назвою об'єкту, структуру якого вона відображує.

На рис. 8.1 показана згорнута підсистема двигуна постійного струму з видимими та схованими іменами портів та варіант розгорнутої підсистеми.

Розгрупувати створену другим способом підсистему можна тільки відміною утворення командою *Edit* → *Undo Create Subsystem* (^Z). Треба мати на увазі, що ця команда відмінює тільки графічні зміни. Зокрема вона не може повернути змінене ім'я або змінене значення якого-небудь параметра.

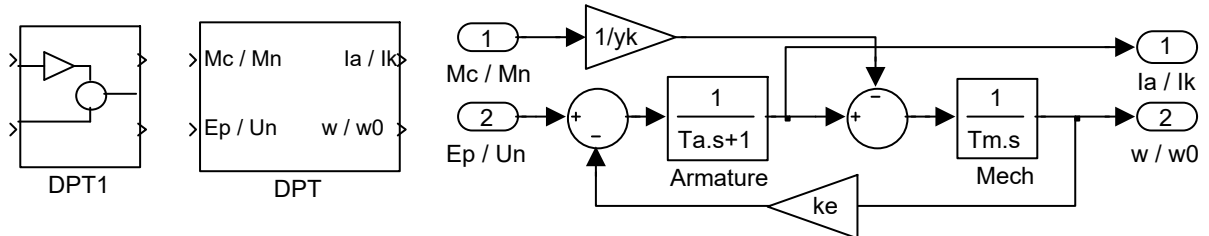


Рис. 8.1 – Підсистема двигуна постійного струму

8.1.2 Маскування підсистем

Будь-яку підсистему можна замаскувати.

Існують 2 рівня маскування.

Перший рівень полягає тільки у заміні іконки блоку. При такому способі подвійний щиглик по іконці як і для незамаскованої підсистеми відчиняє вікно з розгорнутою моделлю підсистеми.

При використанні другого рівня маскування підсистема стає схожою на звичайний блок, тобто при подвійному щиглику на його піктограмі відкривається не розгорнута модель підсистеми, а вікно введення параметрів *Block Parameters*. Для того, щоб побачити розгорнуту модель треба „заирнути під маску” виконанням команди *Edit* → *Look Under Mask* (^U).

Маскування підсистеми починають командою *Mask Subsystem* (^M), на яку система реагує відкриттям вікна маскування *Mask Editor*, що має 3 вкладки (панелі): *Icon*, *Initialization* та *Documentation*, які показані на рис. 8.2.

В поле *Mask type* необхідно ввести тип блока. Пізніше цей тип з поміткою (*Mask*) будет відображатися у вікні введення параметрів замаскованого блока.

В поле *Block description* панелі *Documentation* можна ввести короткий коментар про призначення блока та про його параметри, який буде відобра-

жений у вікні *Block Parameters*, а в поле *Block Help* – більш докладний опис блока, який буде виводитися на екран в *html*-форматі через при натисканні клавіші *Help* в однойменному вікні.

Для того, щоб параметри підсистеми можна було б визначати в одному вікні, не користуючись розширеною моделлю, у вкладці *Initialization* треба після натискання клавіші *Add* визначати в полі *Prompt* запрошення на введення параметра, а в полі *Variable* – ім'я змінної, яка отримає значення після введення його у відповідь на запрошення. У вікні *Block Parameters* запрошення будуть розташовані у порядку, зворотному до порядку їх визначення у вікні *Mask Editor*. Цей порядок можна змінити клавішами *Up* і *Down*. Зайві параметри можна знищити клавішею *Delete*.

Для кожного параметра спосіб, яким буде обиратися або вводиться значення параметру, визначається вибором у полі *Control type*. У ньому пропонуються наступні засоби:

Edit – введення константи, змінної або виразу;

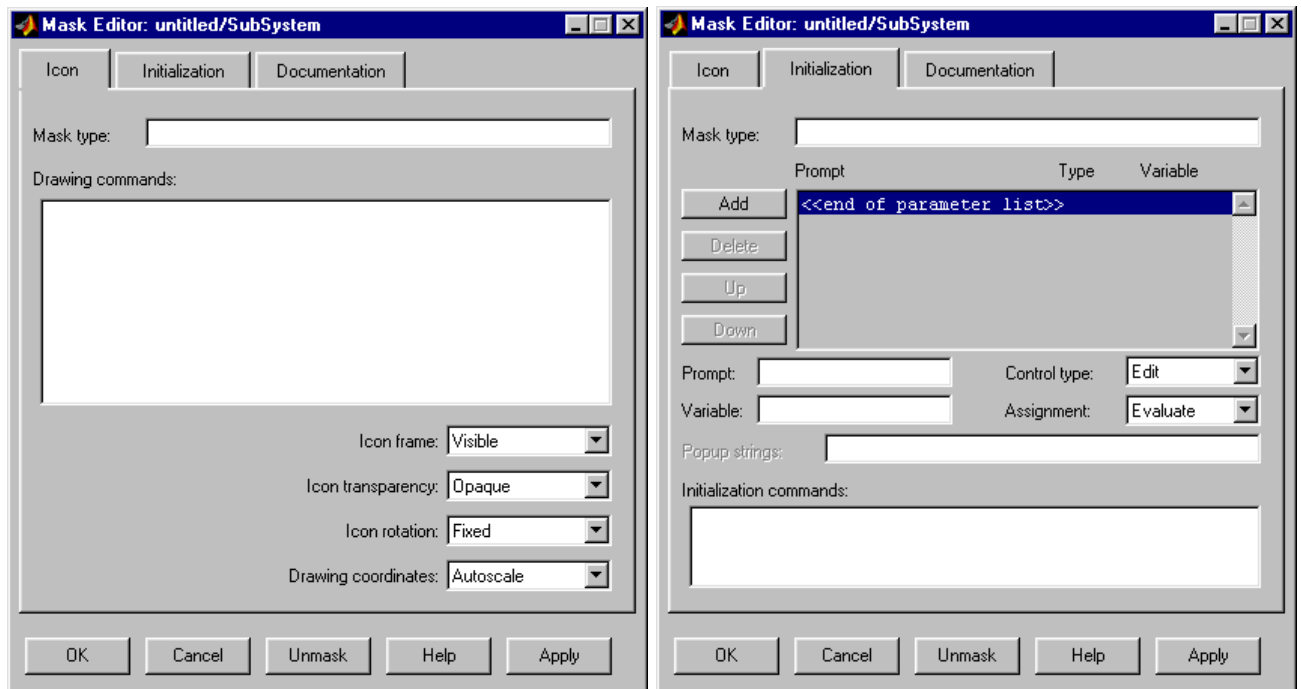
Checkbox – вибір одного з двох альтернативних варіантів *on* або *off*, перший з яких діє при встановленому „прапорці”, тобто при поміченому „галочкою” полі параметру, а другий – при невстановленому „прапорці”, тобто при відсутності помітки в полі параметру;

Popur – вибір варіанта з випадаючого меню.

При використанні способу *Popur* треба у полі *Popur strings* ввести варіанти строк (без апострофів) для вибору, відокремлюючи їх одну від одної символом „|”.

Значення обраного або введенного параметра може використовуватись як у чисельній формі, так і у форматі рядка символів. Тип операції присвоєння визначається станом поля *Assigned: Evaluate* (обчислити значення) або *Literal* (інтерпретувати значення як рядок символів).

Спосіб обчислення значення параметра в режимі *Evaluate* пояснюється таблицею 8.1.



a)

б)



в)

Рис. 8.2 – Вкладки вікна маскування підсистем

Таблиця 8.1

<i>Control type</i>	<i>Numerical value</i>
---------------------	------------------------

<i>Edit</i>	Введена константа, обчислене значення введеної змінної або введеного виразу
<i>Checkbox</i>	1 – при встановленому „прапорці” (стан <i>on</i>) 0 – при відсутності „прапорця” (стан <i>off</i>)
<i>Popup</i>	Ціле число, що відповідає порядковому номеру (зверху вниз) обраного варіанту

Панель *Icon* призначена для зміни піктограми (іконки) підсистеми та її властивостей.

Зовнішнім видом іконки керують команди, занесені в поле *Drawing commands*. Принципово у піктограмі блоку може бути виведеним будь-який текст, передавальна функція, графік статичної або динамічної характеристики та не дуже складне графічне зображення.

При визначенні запрошень (*Prompt*) кириличні шрифти краще не використовувати, бо це може привести до непередбачених результатів. В інших текстових полях для більшої надійності також краще користуватися латиницею.

Для виведення текстів можна використовувати один з наступних операторів:

```
disp ('text')      або   disp (StringVariableName)
text (x, y, 'text')  або   text (x, y, StringVariableName)
text (x, y, 'text', 'HorizontalAligment', HorAl, 'VerticalAligment', VertAl)
fprintf ('text')    або   fprintf ('format', VariableName)
port_label (port_type, port_number, label)
```

Для розподілу тексту на рядки в операторах `disp` і `text` можна використовувати комбінацію символів `\n`, наприклад, `disp ('Mask\nSubsystem')`.

В операторах `text` параметри `x,y` уявляють собою координати точки в піктограмі блоку, до якої прив'язується текст. Для явного визначення

системи координат, що використовується при зображенні іконки можна скористатися оператором

```
plot (xlb, ylb, xrt, yrt)
```

де `xlb`, `ylb` і `xrt`, `yrt` – координати лівого нижнього та правого верхнього кутів графічного вікна. Використання операторів `xlim`, `ylim` та `axes []` з цією ж метою у полі *Drawing commands* не припустимо.

Стиль прив'язки визначається параметрами `HorAl` (вирівнювання по горизонталі) та `VertAl` (вирівнювання по вертикалі), які можуть приймати такі значення: `'center'`, `'left'`, `'right'`, і `'middle'`, `'base'`, `'bottom'`, `'cap'`, `'top'` відповідно. За замовчанням діють опції `'left'` та `'middle'`. Результат використання перелічених опцій пояснюється на рис. 8.3.

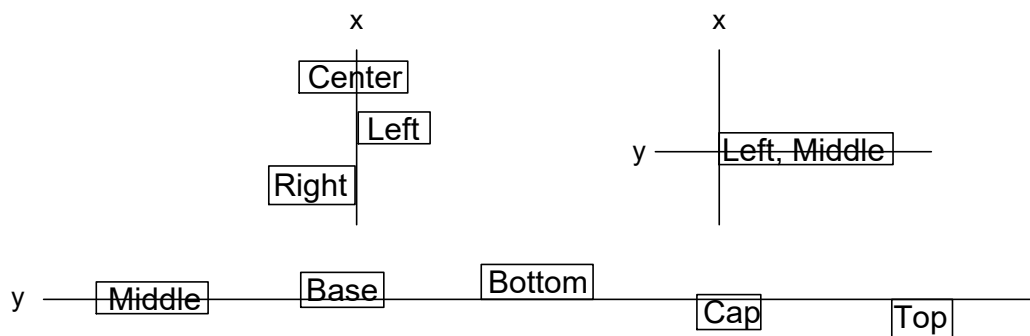


Рис. 8.3. Демонстрація дії опцій вирівнювання

При застосуванні команди `port_label` параметр `port_type` може приймати значення `'input'` або `'output'`, а параметр `label` – будь-якого рядка символів, наприклад, в результаті виконання команди

```
port_label ('output', 2, 'a')
```

другий вихідний порт згорнутої підсистеми буде помічено символом `a`, незалежно від імені цього порта у розгорнутій підсистемі.

Для зображення на піктограмі блока його передавальної функції можна застосовувати одну із наступних команд:

```
dpoly (num, den) або dpoly (num, den, char)
```

```
droots (z, p, k) або droots (z, p, k, char)
```

Команда **dpoly** виводить передавальну функцію в поліноміальній формі, а **droots** – у вигляді розкладення на нулі-полюси (див. опис блоків *Transfer Function* і *Zero-Pole-Gain* в підрозділах 4.2.3 і 4.2.4). Необов'язковий символний параметр **char** має за замовчанням значення 's'. Для виведення дискретних ПФ його можна змінити на 'z' або 'z-'. (див. опис блоків *Discrete Transfer Function* і *Discrete Filter* в підрозділі 8.2.3).

Параметри команд **dpoly** і **droots** звичайно визначаються через параметри підсистеми у полі *Initialization commands* вкладки *Initialization*. Для прикладу на рис. 8.4 показана у розгорнутому та згорнутому виглядах підсистема *Aperiodic link with initial value* (Аперіодична ланка з початковими умовами). На рис. 8.4, б іконка замаскованої підсистеми сформована командою **dpoly**, а на рис 8.4, в – командою **droots**, для чого у полі *Initialization commands* записані оператори присвоювання

$\text{num} = [k]; \text{den} = [T \ 1];$

$z = []; p = -1/T; K = k/T;$

відповідно.

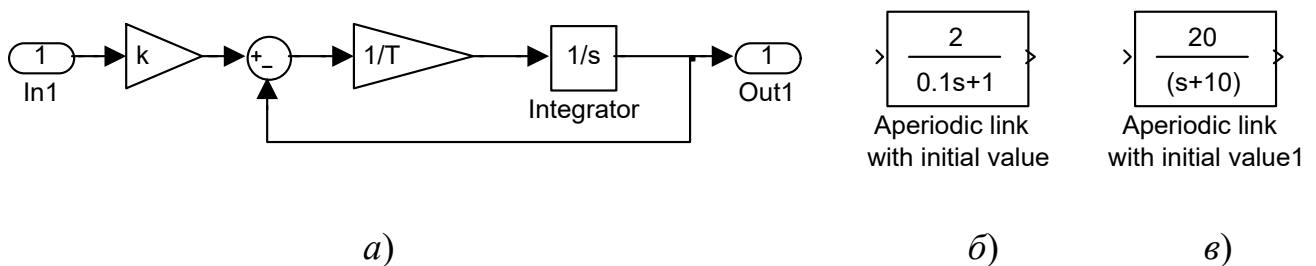


Рис. 8.4. Розгорнута (а) та згорнуті (б, в) підсистеми *Aperiodic link with initial value*

Як бачимо, параметри піктограм, сформованих у такий спосіб виводяться тільки у вигляді констант. Для того, щоб результати операцій присвоювання не відображувались у командному вікні, після кожного з операторів треба поставити точку з крапкою (;).

Команди ініціалізації виконуються в таких випадках: при завантаженні моделі; при старті моделювання; при повороті блоку; в усіх випадках, коли

треба заново зобразити іконку блоку, а команди зображення використовують результати команд ініціалізації.

Для виведення на піктограмі графічних зображень, заданих координатами вузлових точок, можна скористуватись командами

`plot (y)   plot (x, y)   plot (x1, y1, x2, y2, ..., xN, yN)`

де x, y – вектори абсцис та ординат табличної функції, або

`plot (xlb, ylb, xrt, yrt, y)   plot (xlb, ylb, xrt, yrt, x, y)`

`plot (xlb, ylb, xrt, yrt, x1, y1, x2, y2, ..., xN, yN)`

Якщо аргументи x табличної функції не задані, то по осі абсцис відкладаються індекси вектора y . Остання група операторів `plot` явно визначає систему координат.

Іконки, які уявляють собою досить прості графічні зображення, що складаються з обмеженої кількості відрізків прямих, можна створити за допомогою утиліти `iconedit`. При виклику її з командного рядка *MATLAB* на екрані по черзі з'являються 2 запити: *Name of block diagram (Ім'я моделі)* і *Name of block (Ім'я підсистеми)*. Після відповіді на ці запити на екрані відчиняється графічне вікно, у якому можна намалювати бажане зображення, фіксуючи за допомогою миші вузлові точки рисунку та використовуючи наступні клавішні команди: *d=delete* – знищити останню з зафіксованих точок, *n=new pt* – почати нову лінію і *q=quit* – завершити створення іконки.

Для зображення зафарбованих плоских замкнених фігур використовують команду

`patch (x, y, [r g b])`

де x, y – вектори абсцис та ординат точок замкненої фігури, `[r g b]` – її колір, визначений інтенсивністю трьох кольорів: червоного, зеленого та синього у вигляді констант зі значеннями в діапазоні від 0 до 1. За замовчанням (при відсутності останнього параметру) фігура зафарбовується кольором „заднього фону” (*Background color*), тобто тим кольором, яким зображені контури іконки. Наприклад, оператор

```
patch ([ -1 0 1], [0 1 0], [0 1 1])
```

створює піктограму у вигляді трикутника, зафарбованого бірюзовим кольором, зображену (у чорно-білому варіанті) на рис. 8.5.

Для зображення на іконці малюнків, створених іншими програмними продуктами, треба скористуватися командою

```
image (imread ('FileName.Extension'))
```

де *FileName* – ім'я графічного файла, а *Extension* – його розширення.

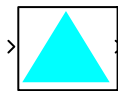


Рис. 8.5. Замаскована підсистема, іконка якої створена командою *patch*

MATLAB підтримує такі типи графічних файлів: **.bmp* (*Bitmap*), **.hdf* (*Hierarchical Data Format*), **.jpg* (*Joint Photographic Experts Group*), **.pcx* (*Paintbrush*), **.png* (*Portable Network Graphics*), **.tif* (*Tagged Image File Format*), **.xwd* (*X Window Dump*).

Властивостями піктограми керують опції, імена яких розташовані нижче поля *Drawing commands* вкладки *Icon*, а значення обираються за допомогою *popup-menu*. Інформація про ці опції наведена в табл. 8.2.

Прикладом блоку з невидимою рамкою є блок *Manual Switch*.

Якщо зробити іконку прозорою, то крізь неї можуть бути видні, наприклад, імена портів.

Ширину та висоту блока в пікселях при використанні стилю *Pixel* при визначенні системи координат можна розрахувати у полі *Drawing commands* за допомогою операторів

```
pos = get_param (gcb, 'Position');
```

```
width = pos(3) – pos(1);
```

```
hight = pos(4) – pos(2);
```

Опція	Значення	Сенс
<i>Icon Frame</i> (Рамка Іконки)	<i>Visible</i>	Рамку видно
	<i>Invisible</i>	Рамку не видно
<i>Icon Transparency</i> (Прозорість Іконки)	<i>Opaque</i>	Іконка непрозора
	<i>Transparent</i>	Іконка прозора
<i>Icon Rotation</i> (Обертання Іконки)	<i>Fixed</i>	При повороті блоку іконка не обертається
	<i>Rotates</i>	При повороті блоку іконка обертається
<i>Drawing Coordinates</i> (Система Координат)	<i>Autoscale</i>	$x_{min} = \min(\mathbf{X}), \quad x_{max} = \max(\mathbf{X}),$ $y_{min} = \min(\mathbf{Y}), \quad y_{max} = \max(\mathbf{Y})$
	<i>Normalized</i>	$x_{min} = 0, \quad x_{max} = 1, \quad y_{min} = 0, \quad y_{max} = 1$
	<i>Pixel</i>	$x_{min} = 0, \quad x_{max} = \text{block width},$ $y_{min} = 0, \quad y_{max} = \text{block hight}$

Відмінити маскування можна клавішею *Unmask*. При виконанні команди *Edit* → *Mask Subsystem* (^M) вікно маскування відновлює свої попередні установки.

8.2 Створення кнопок, що керують процесом виконання модельного експерименту

За допомогою блоку *Subsystem* можна створити „кнопки”, що керують процесом виконання модельного експерименту, тобто такі відокремлені блоки (без вхідних та вихідних портів), подвійний щиглик мишею на іконці яких дозволяє виконувати бажані оператори, функції або команди пакета *MATLAB*.

Звичайно користувачі-початківці створюють моделі, параметри яких задають константами. В такому разі для досліджень впливу зміни якого-небудь параметру на властивості досліджуваного об’єкта доводиться відкривати вікна визначення параметрів усіх блоків, до яких входить варійований параметр, вручну змінювати значення цього параметру, виконувати моделювання через *Simulink*-меню і запам’ятовувати його результати. Таку послідовність дій багатократно повторюють, а потім обробляють результати модель-

ного експерименту. Викладену методику не можна назвати раціональною при наявності розвинутої алгоритмічної мови і можливості виконувати моделювання в програмному режимі.

Більш ефективно дотримуватися такої методики:

- параметри моделі задавати іменами змінних;
- для ініціалізації моделі створити файл даних, в якому задаються або обчислюються значення цих змінних;
- для виконання файлу даних, не виходячи з вікна *Simulink*-моделі, створити „кнопку” з умовною назвою *Init*;
- при налагодженні моделі виконувати моделювання досліджуваної системи після процесу ініціалізації через *Simulink*-меню, фіксуючи результати блоками *Scope*;
- після того, як ви впевнитесь у працездатності моделі, треба розробити план модельного експерименту і *MATLAB*-програму або декілька програм, що його реалізують;
- для виконання розроблених програм, не виходячи з вікна *Simulink*-моделі, створити „кнопки” з умовною назвою *Run-Plot*;
- наприкінці можна створити кнопку, активізація якої виводить в окремому вікні інформацію про модель та рекомендації щодо її застосування.

Модель, параметри якої задані іменами змінних, а не числами є більш наочною. При програмному визначенні параметрів зменшується час коригування параметрів та можливість помилок, особливо тоді, коли один і той же параметр, як, наприклад, період дискретності, треба визначити в декількох блоках.

Програма модельного експерименту може отримувати у своєму складі цикли, в яких змінюються варійовані параметри або параметри, що керують ключовими елементами моделі. В цих циклах можна здійснювати розрахунок перехідних процесів, частотних характеристик, тощо, та обробляти отримані результати, зокрема, виводити їх у вигляді графіків.

Створення „кнопок” допомагає пов’язати модель з програмами, які її ініціюють та досліджують.

Цей процес можна виконати у такому порядку:

- 1) відкрити бібліотеку *Subsystems* і перетягти з неї у вікно досліджуваної моделі блок *Subsystem*;
- 2) відкрити блок *Subsystem*, знищити його вхідний та вихідний порти і лінію зв’язку поміж ними;
- 3) активізувати контекстуальне меню щигликом правої кнопки миші по піктограмі блоку та обрати в ньому функцію *Block Properties*, що приведе до відчинення відповідного вікна;
- 4) у рядку *Open function* відчиненого вікна набрати ім’я файлу або послідовність *MATLAB*-команд, які ви бажаєте виконати при натисканні створеної кнопки, після чого натиснути клавішу *OK*;
- 5) виділити блок і виконати з ним послідовно операції *Hide Name* і *Show Drop Shadow* функції *Format Simulink*-меню;
- 6) замаскувати підсистему ($\wedge M$) виведенням в її іконці назви „кнопки”, наприклад, `disp ('Init')`, остання команда набирається в полі *Drawing commands* вікна маскування;
- 7) Зафарбувати „кнопку” яким-небудь (бажано світлим) кольором, наприклад, за допомогою операцій *Format* $\rightarrow$ *Background Color* $\rightarrow$ *Yellow*.

Власне кажучи, необхідними є тільки перша і третя операції. Усі останні дії спрямовані на те, щоб придати „кнопці” презентабельний вигляд (див. рис. 8.6).

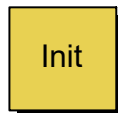


Рис. 8.6. Вигляд „кнопки” *Init*, створеної за наведеною методикою

Для створення інформаційної „кнопки” після виконання пункту 2, тобто після знищення портів, треба у звільненому вікні набрати текст, який пояснює призначення моделі, або якусь іншу інформацію. Пункти 3 і 4 викону-

вати не треба, а в пункті 6, тобто при маскуванні підсистеми, треба вивести в іконці „кнопки” відповідний текст, наприклад, 'Info' або '?'.

8.3 Завдання

Варіанти завдань до перших чотирьох пунктів завдання відображені у табл. 8.3.

Таблиця 8.3

№ варіантів	Задавальний пристрій (п. 1)	Регулятор з обмеженням (п. 2)	Фільтр (п. 3)			Нелінійні блоки (п. 4)
		Тип	Тип	Маск.блок	Іконка	Спосіб
1, 7, 13, 19	I-ЗІ	Дискретний ПІ	Баттерворта	<i>Transfer Function</i>	Перехідна функція	<i>MATLAB Function</i>
2, 8, 14, 20	I ² -ЗІ	Аналоговий ІФ	Бесселя	<i>Zero-Pole-Gain</i>	АЧХ	Розгорнута структура
3, 9 15, 21	ПД-ЗІ	Дискретний ПІ	Баттерворта	<i>Zero-Pole-Gain</i>	Перехідна функція	<i>MATLAB Function</i>
4, 10, 16, 22	ЗП	Аналоговий ІФ	Бесселя	<i>Transfer Function</i>	АЧХ	Розгорнута структура
5, 11 17, 23	I-ЗІ	Аналоговий ІФ	Баттерворта	<i>Zero-Pole-Gain</i>	АЧХ	<i>MATLAB Function</i>
6, 12, 18, 24	ЗП	Дискретний ПІ	Бесселя	<i>Transfer Function</i>	Перехідна функція	Розгорнута структура

1) Створити незамасковану підсистему задатчика інтенсивності (ЗІ) або задатчика положення (ЗП) у форматі *Show Port Label*.

2) Створити замасковану підсистему регулятора з обмеженням вихідного сигналу. В іконці відобразити передавальну функцію регулятора та графічне зображення характеристики вхід-вихід блоку *Saturation*.

3) Створити замасковану підсистему фільтра на основі блоків *Transfer Function* або *Zero-Pole-Gain*. В іконці відобразити перехідну функцію або амплітудно-частотну характеристику (АЧХ).

4) Створити замасковані підсистеми „Обмеження координат” та „Зона нечутливості”, у яких рівні обмеження та межі зони задаються не параметрами, а вхідними сигналами.

5) Створити інформаційні „кнопки” та „кнопки” для керування моделями, розробленими при виконанні попередніх лабораторних робіт за вказівкою викладача. Перевірте працездатність створених підсистем та кнопок.

8.4 Методичні вказівки та рекомендації

1) Для створення задавальних пристроїв скористуйтеся моделями, поданими на рис. 8.7

На моделях позначені такі параметри:

W_z – завдання на усталену швидкість;

F_{iz} – завдання на усталене положення;

R_o , e_o , W_o – обмеження на ривок, прискорення та швидкість відповідно;

t_r – час наростання завдання на швидкість до заданого значення або до рівня обмеження;

t_n – час наростання прискорення до рівня обмеження.

Для перевірки працездатності моделей скористайтесь такими параметрами: $W_z=1$; $F_{iz}=1.2$; $t_r=100$; $t_n=20$; $W_o=1$; $e_o=W_o/t_r$; $R_o=e_o/t_n$; $T_v=4$; $k_2=b_2/T_v$; $k_1=b_1-k_2$; $b_2=50$; $b_1=10$.

Для моделювання ЗП використовуйте метод *ode5* з кроком 0.1, для моделювання ЗІ – метод *ode23s*.

2) Для створення аналогових регуляторів з передавальними функціями

$$W_{III}(s) = \frac{T_1 s + 1}{T_2 s} = \frac{T_1}{T_2} + \frac{1}{T_2 s} = \frac{T_1}{T_2} \left(1 + \frac{1}{T_1 s} \right),$$

$$W_{III}(s) = k \frac{T_1 s + 1}{T_2 s + 1}.$$

та обмеженням вихідного сигналу скористайтесь деталізованими моделями, наведеними на рис. 10.8.

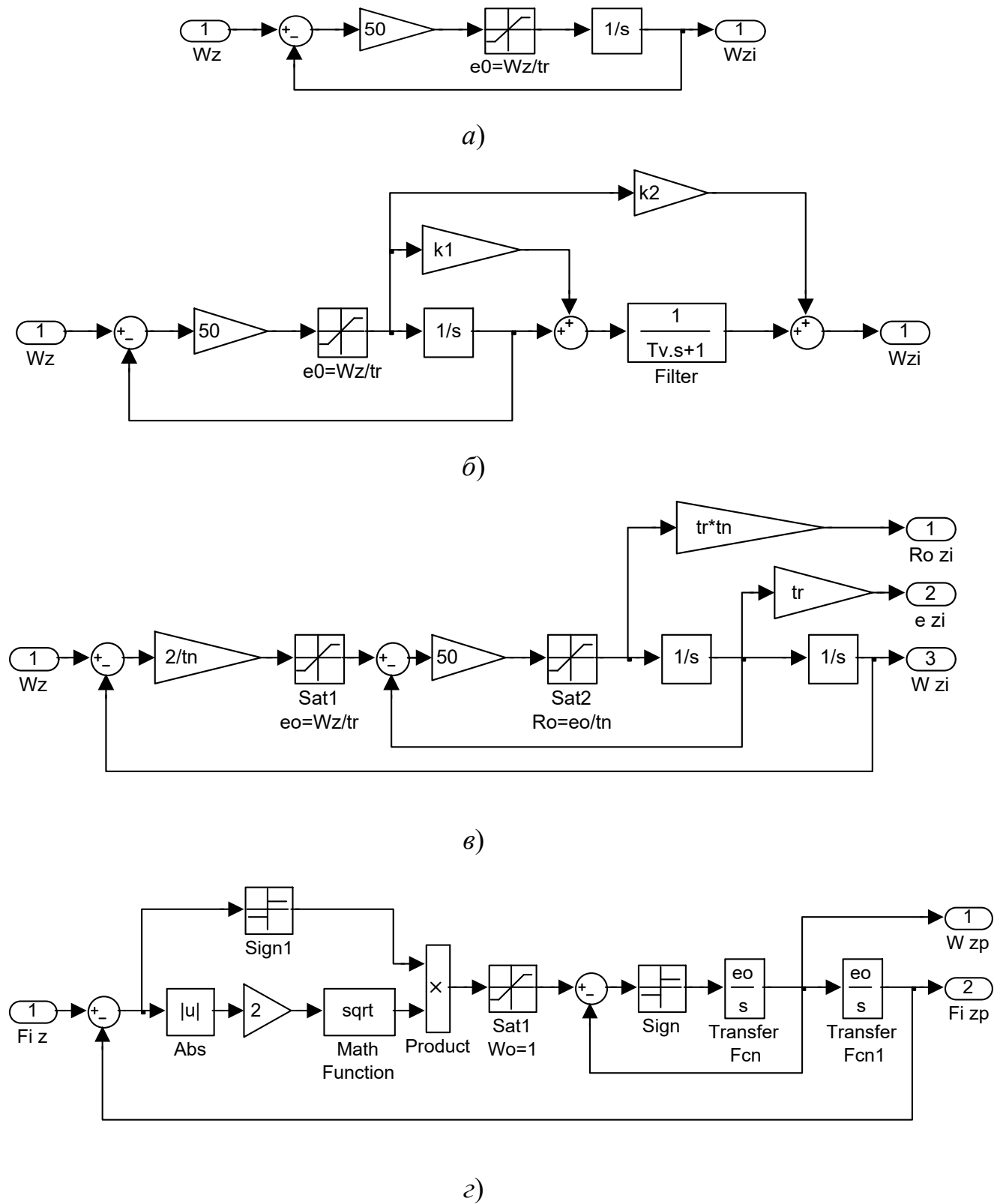


Рис. 10.7. Моделі задавальних пристроїв:

а) I-3I, б) ИД-3I, в) I²-3I, г) 3П

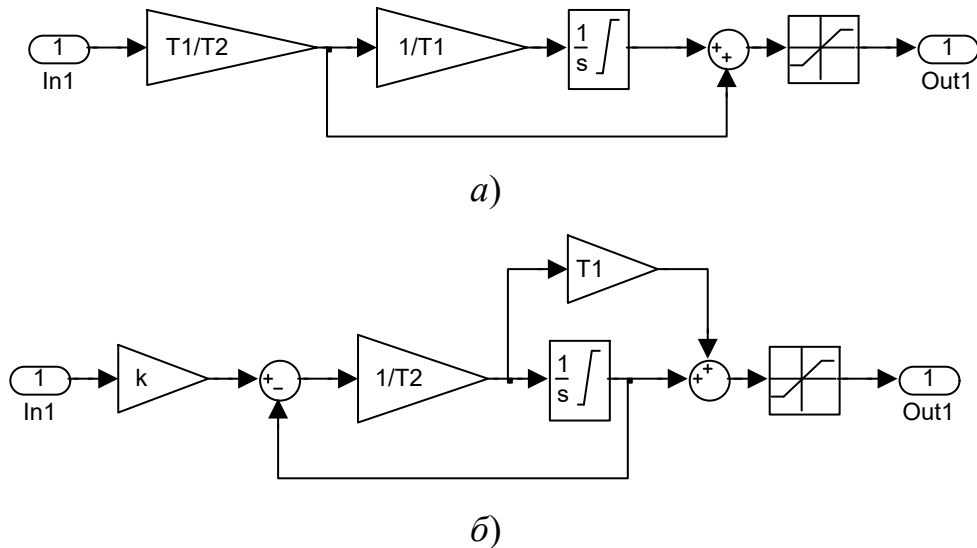


Рис. 8.8. Моделі регуляторів з обмеженням вихідного сигналу:

а) ПІ; б) ІФ

Рівні обмеження інтеграторів та блоків *Saturation* повинні співпадати з рівнем обмеження регулятора.

При перевірці працездатності та адекватності розроблених регуляторів скористайтесь такими параметрами: $T_1 = 2$; $T_2 = 4$; $k = 1$; $y_{\max} = 0.75$; $y_{\min} = -0.75$. На входах моделей за допомогою блока *Step* сформууйте сигнал, який при $t = 5$ змінює своє значення з 1 на -1.

3) У перелічених в табл. 8.4 фільтрів параметрами є порядок фільтра n та частота пропускання w_0 . Створити фільтри можна за допомогою маскування блоків *Transfer Function* або *Zero-Pole-Gain*, сформувавши параметри цих блоків і масиви даних для зображення іконки у полі *Initialization commands*.

Фільтр Баттерворта можна створити послідовністю команд

```
[num, den] = butter (n, w0, 's');
```

```
[z, p, K] = tf2zp (num, den);
```

а фільтр Бесселя –

```
[z, pn, K] = besslap (n);
```

```
p = pn * w0;
```

```
[num, den] = zp2tf ([z, p, K]);
```

Для побудови перехідних функцій треба задати вектор часу

```
t = linspace (0, 20/w0, 21);
```

та розрахувати реакцію блоку на одиничний стрибок оператором

```
y = step (num, den, t);
```

Для зображення АЧХ треба задати вектор частот

```
w = linspace (0, w0*5);
```

та розрахувати вектор амплітуд:

```
A = bode (num, den, w);
```

Вивід графіків у вікно іконки блоку здійснюється однією з команд

```
plot (t, y)
```

```
plot (w, A),
```

що набираються у полі *Drawing commands* вкладки *Icon*.

4) При структурному способі створення підсистеми „Змінне обмеження координат” скористуйтеся блоком *MaxMin* з двома входами.

При структурному способі створення підсистеми „Змінна зона нечутливості” обміркуйте, у який спосіб можна отримати цю типову нелінійність із нелінійності „Обмеження координат”.

Для перевірки адекватності створених підсистем скористайтесь моделлю, зображеною на рис. 8.9.

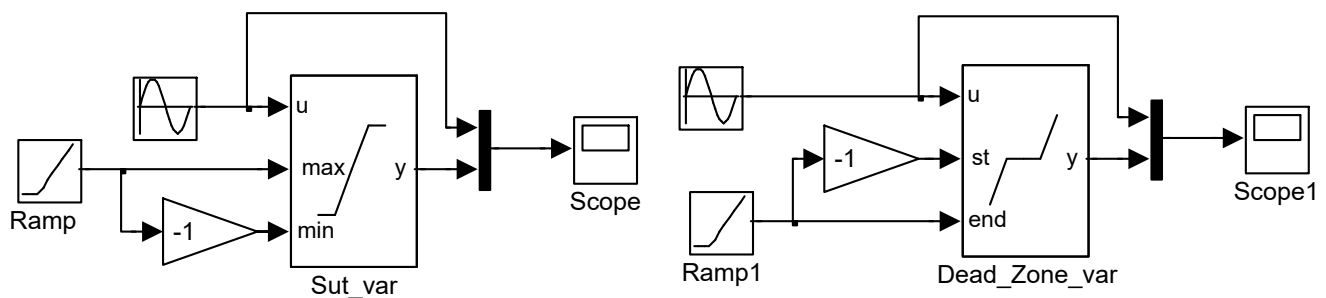


Рис. 8.9. Модель для перевірки адекватності підсистем „Змінне обмеження координат” та „Змінна зона нечутливості”

Спробуйте створити для підсистем, розроблених при виконанні четвертого пункту завдання, іконки, подібні до показаних на рис. 8.9.

8.5 Контрольні питання

- 1) Які типи підсистем Ви знаєте?
- 2) Чим відрізняється замаскована підсистема від незамаскованої?
- 3) Як замаскувати підсистему?
- 4) Як змінити піктограму блоку?
- 5) Що можна відображати в іконках?
- 6) Як організувати введення параметрів замаскованої підсистеми?
- 7) Як визначити „допомогу” для користування замаскованою підсистемою?
- 8) Як створити „кнопку”, пов’язану з моделлю?
- 9) Для чого доцільно використовувати при моделюванні віртуальні „кнопки”?

*Лабораторна робота №9***ЕКВІВАЛЕНТНІ ПЕРЕТВОРЕННЯ ЛІНІЙНИХ СТРУКТУРНИХ СХЕМ
У СЕРЕДОВИЩІ MATLAB****9.1 Теоретичні відомості**

При дослідженні лінійних динамічних систем одним із джерел інформації про їхні статичні та динамічні властивості є передавальні функції. Щоб отримати передавальні функції із структурних схем їх послідовно спрощують методом еквівалентних перетворень.

Перелічимо основні *правила еквівалентних перетворень*.

- 1) При послідовному з'єднанні декількох ланок $W_i(s)$, де $i = 1, 2, \dots, k$, еквівалентна передавальна ланка визначається як добуток складових:

$$W(s) = \prod_{i=1}^k W_i(s). \quad (9.1)$$

- 2) При паралельному з'єднанні декількох ланок $W_i(s)$, де $i = 1, 2, \dots, k$, еквівалентна передавальна ланка визначається як сума складових

$$W(s) = \sum_{i=1}^k W_i(s). \quad (9.2)$$

- 3) Передавальна функція замкненого контуру визначається за формулою

$$W(s) = \frac{W_{\Pi}(s)}{1 \pm W_{\Pi}(s)W_3(s)}, \quad (9.3)$$

де $W_{\Pi}(s)$, $W_3(s)$ – передавальні функції прямої частини контуру та зворотного зв'язку відповідно. Знак «+» відповідає від'ємному зв'язку, а знак «-» – додатному.

- 4) При перенесенні алгебраїчного суматора зі входу деякої ланки $W_{\Pi}(s)$ на її вихід вхідні сигнали суматора слід помножити на $W_{\Pi}(s)$; при перенесенні суматора у зворотному напрямку його вхідні сигнали треба розділити на $W_{\Pi}(s)$.

9.2 Еквівалентні перетворення лінійних динамічних систем в середовищі *MATLAB* у символьному вигляді

Як бачимо, еквівалентні структурні перетворення здійснюються за допомогою звичайних арифметичних операцій. Тому для їх виконання у середовищі пакету *MATLAB* достатньо тільки описати потрібні змінні як символьні за допомогою оператора

```
syms var1 var2 ... varN,
```

виконати потрібні арифметичні дії над цими змінними, скористатися функціями додатку *Extended Symbolic Toolbox*, що спрощують отримані вирази, та вивести їх на екран за допомогою функції

```
pretty(expr)
```

Для перетворення виразів найчастіше використовують такі функції:

- `simple(expr)` – намагається за допомогою різноманітних алгебраїчних формул скоротити вираз;
- `expand(expr)` – розкриває дужки;
- `collect(expr, var)` – збирає подібні члени за вказаною змінною;
- `subs(expr, old, new)` – виконує в `expr` підстановку: `old` замінює на `new`;
- `numden(expr)` – виділяє із заданого виразу чисельник та знаменник.

Для прикладу наведемо програму, що визначає передавальні функції

$$\text{двигуна постійного струму } W_{U\omega}(s) = \frac{\omega(s)}{U_{\text{я}}(s)}, \quad W_{UI}(s) = \frac{I_{\text{я}}(s)}{U_{\text{я}}(s)}, \quad W_{MM}(s) = \frac{M(s)}{M_c(s)}$$

$$\text{та } W_{M\omega}(s) = \frac{\omega(s)}{M_c(s)} \text{ за структурною схемою рис. 7.1.}$$

```
clc
syms Ta Tm Ra c s
Wa=1/Ra/(Ta*s+1)
Wm=Ra/c^2/Tm/s

W_UE=Wa*c^2*Wm/(1+Wa*c^2*Wm);
W_UE=simple(W_UE);
W_UE=expand(W_UE);
disp(blanks(1)'), disp('W_UE='), pretty(W_UE)
```

```

W_Uw=W_UE/c;
disp(blanks(1)'), disp('W_Uw='), pretty(W_Uw)

W_UM=W_Uw/Wm;
disp(blanks(1)'), disp('W_UM='), pretty(W_UM)
W_Mw=-W_Uw/Wa/c
disp(blanks(1)'), disp('W_Mw='), pretty(W_Mw)

W_MM=Wm*c*Wa*c/(1+Wa*c^2*Wm);
W_MM=simple(W_MM);
W_MM=expand(W_MM);
disp(blanks(1)'), disp('W_MM='), pretty(W_MM)

```

При виконанні наведеної програми отримуємо такий результат:

Wa =

$$1/(Ra*(Ta*s + 1))$$

Wm =

$$Ra/(Tm*c^2*s)$$

W_UE=

$$\frac{1}{Ta Tm s^2 + Tm s + 1}$$

W_Uw=

$$\frac{1}{c (Ta Tm s^2 + Tm s + 1)}$$

W_UM=

$$\frac{Tm c s}{Ra (Ta Tm s^2 + Tm s + 1)}$$

W_Mw=

$$\frac{Ra (Ta s + 1)}{c (Ta Tm s^2 + Tm s + 1)}$$

$$W_{MM} = \frac{1}{T_a T_m s^2 + T_m s + 1}$$

9.3 Знаходження результуючих передавальних функцій лінійних динамічних систем у чисельному вигляді

Для автоматизованого визначення математичного опису лінійної стаціонарної системи у чисельному вигляді можна скористатися такою методикою:

- 1) розробити структурну *Simulink*-модель досліджуваного об'єкта, у якій на входах і виходах треба обов'язково установити вхідні і вихідні порти (блоки *In* та *Out*) та зберегти її під умовним ім'ям *ModelName*;
- 2) ініціалізувати модель та визначити чисельні значення специфічних параметрів математичного опису її у просторі станів за допомогою команди

```
[A, B, C, D]=linmod (ModelName)
```

- 3) створити так званий *ss*-об'єкт (*ss* походить від *State Space*) оператором
`ObjName_ss=ss (A, B, C, D)`
- 4) перетворити *ss*-об'єкт у матрицю *tf*-об'єктів (*tf* походить від *Transfer Function*) оператором

```
ObjName_tf=tf (ObjName_ss)
```

Елементи отриманої матриці мають першим індексом номер виходу *out*, а другим – номер входу *in*. Отже звернутися до конкретної передавальної функції можна за її ім'ям та списком індексів у круглих дужках: `ObjName_tf (out, in)`.

Зі скалярної передавальної функції можна виділити чисельник і знаменник за допомогою оператора

```
[num, den]=tfdata (ObjName_tf (out, in))
```

Для прикладу на рис. 9.1 наведена структурна модель ДПС з керуванням у колі якоря, розглянута у лабораторній роботі №7.

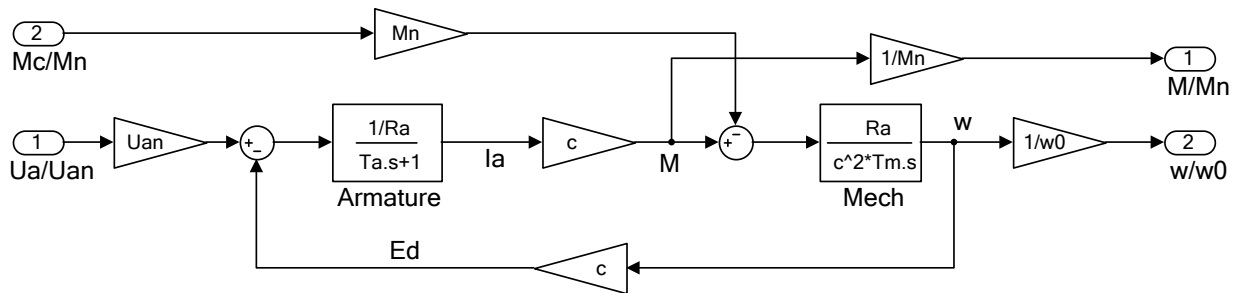


Рис. 9.1. Структурна модель ДПС з постійним збудженням

Якщо її записати у файл *dpt_SI* та виконати програму

```
clc, close all
[A,B,C,D]=linmod('dpt_SI');
dps=ss(A,B,C,D)
dps_tf=tf(dps)
Wdps_wU=dps_tf(2,1)
[num,den]=tfdata(Wdps_wU)
```

отримаємо такий результат:

```
dps =
a =
      x1      x2
x1 -24.69  -3.91
x2 262.6    0
b =
      u1      u2
x1  440      0
x2   0 -220.7
c =
      x1      x2
y1  1.19      0
y2   0 0.008885
d =
      u1 u2
y1 0 0
y2 0 0
```

Continuous-time state-space model.

dps_tf =

From input 1 to output...

$$\begin{aligned} & \text{-- 523.6 s} \\ 1: & \frac{\quad}{s^2 + 24.69 s + 1027} \\ & \quad 1027 \\ 2: & \frac{\quad}{s^2 + 24.69 s + 1027} \end{aligned}$$

From input 2 to output...

$$\begin{aligned} & \quad 1027 \\ 1: & \frac{\quad}{s^2 + 24.69 s + 1027} \\ & \quad -1.961 s - 48.41 \\ 2: & \frac{\quad}{s^2 + 24.69 s + 1027} \end{aligned}$$

Continuous-time transfer function.

Wdps_wU =

$$\frac{1027}{s^2 + 24.69 s + 1027}$$

Continuous-time transfer function.

num =

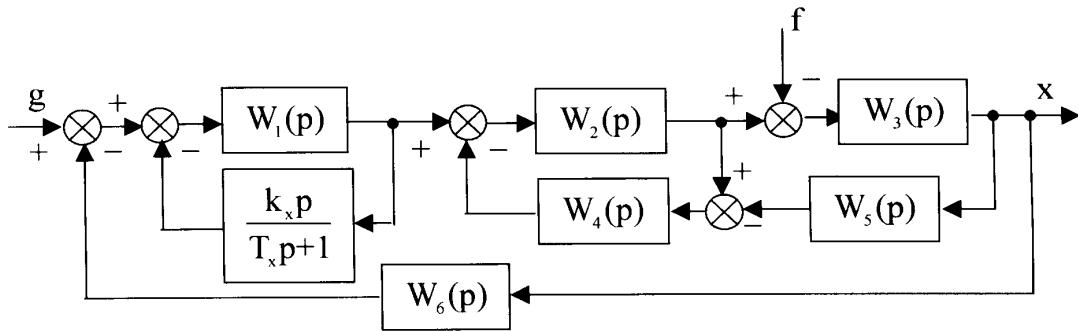
$$1.0e+03 * \begin{bmatrix} 0 & 0 & 1.0266 \end{bmatrix}$$

den =

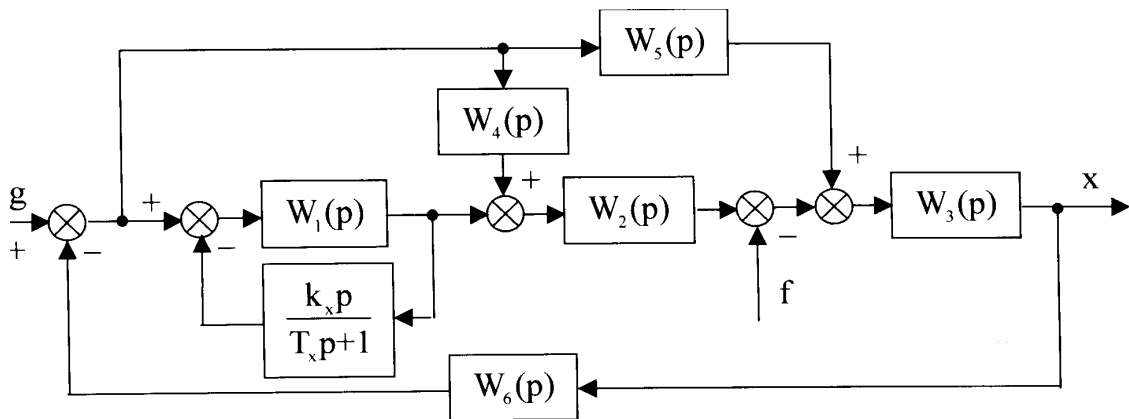
$$1.0e+03 * \begin{bmatrix} 0.0010 & 0.0247 & 1.0266 \end{bmatrix}$$

9.4 Завдання

Знайдіть передавальні функції за структурними схемами, наданими на рис. 9.2, з передавальними функціями, наведеними у табл. 9.1, та їх параметрами, наведеними у табл. 9.2, символьним методом. Перевірте отримані результати чисельним методом.



Варіанти 31-40



Варіанти 41-50

Рис. 9.2. Структурні схеми для дослідження

Таблиця 9.1

Номери варіантів	$W_1(p)$	$W_2(p)$	$W_3(p)$	$W_4(p)$	$W_5(p)$	$W_6(p)$
1-10	$\frac{K_1}{T_1 p + 1}$	$\frac{K_2}{T_2 p + 1}$	$\frac{K_3}{T_3 p}$	K_4	K_5	K_6
11-20	K_1	$\frac{K_2}{T_1^2 p^2 + T_2 p + 1}$	$\frac{K_3}{T_3 p}$	K_4	K_5	K_6
21-30	$\frac{K_1}{(T_1 p + 1)(T_2 p + 1)}$	K_2	$\frac{K_3}{T_3 p}$	K_4	K_5	K_6
31-40	$\frac{K_1}{T_1 p + 1}$	K_2	$\frac{K_3}{T_2 p + 1}$	$\frac{K_4 p}{T_3 p + 1}$	K_5	K_6
41-50	K_1	$\frac{K_2}{T_1 p + 1}$	$\frac{K_3}{T_2^2 p^2 + T_3 p + 1}$	K_4	K_5	K_6

Таблиця 9.2

Номери варіантів	T_1	T_2	T_3	K_1	K_2	K_3	K_4	K_5	K_6
1;11;21;31;41	0.20	0.10	0.05	100	0.5	1.0	0.05	0.20	0.10
2;12;22;32;42	0.10	0.05	0.10	90	0.5	4.0	0.10	0.10	0.02
3;13;23;33;43	0.30	0.20	0.25	80	1.0	5.0	0.02	0.30	0.05
4;14;24;34;44	0.15	0.05	0.20	70	0.3	1.0	0.05	0.40	0.10
5;15;25;35;45	0.50	0.20	0.25	60	1.0	4.0	0.04	0.50	0.05
6;16;26;36;46	0.25	0.10	0.20	50	0.8	1.0	0.30	0.60	0.06
7;17;27;37;47	0.10	0.05	0.04	50	0.5	2.0	0.10	0.10	0.07
8;18;28;38;48	0.40	0.10	0.25	40	0.5	0.5	0.30	0.20	0.25
9;19;29;39;49	0.30	0.10	0.25	80	0.5	2.0	0.02	0.25	0.10
10;20;30;40;50	0.20	0.50	0.10	20	0.8	1.0	0.1	0.30	0.10

9.5 Порядок виконання роботи

1) Розробіть послідовність еквівалентних перетворень заданої структурної схеми, що дозволить Вам одержати передавальні функції за керуючою та збурюючою діями.

2) Складіть програму з використанням символьних змінних для визначення передавальних функцій досліджуваної системи.

3) Введіть структурну модель досліджуваної системи ініціалізуйте її, визначте її математичний опис у просторі станів і знайдіть з нього передавальні функції системи у чисельному вигляді.

4) Порівняйте між собою передавальні функції, знайдені символьним та чисельним методами. У разі наявності розбіжностей знайдіть та виправте помилки.

9.6 Методичні вказівки і рекомендації

1) Еквівалентні перетворення у символьному вигляді виконуйте поступово, результати кожного перетворення виводьте на екран, добиваючись відсутності багатоповерхових дробів, приведення подібних членів, скорочення

однакових виразів у чисельнику і знаменнику та взагалі наочної і зручної форми їх подання.

2) У структурній схемі входи і виходи позначте вхідними та вихідними портами.

3) Визначте передавальні функції досліджуваної системи за методикою, викладеною у пункті 9.3.

4) Для порівняння передавальних функцій у символьній та чисельній формах після визначення символьних передавальних функцій присвойте параметрам системи чисельні значення і виконайте послідовно необхідну кількість функцій `eval(expr)`. Наприклад, виконання в кінці розглянутої у 9.2 програми послідовності операторів

```
Ta=0.04; Tm=0.1;
W_MM=eval(W_MM);
disp(blanks(1)'), disp('W_MMc='), pretty(W_MM)
W_MMc=factor(W_MM);
disp(blanks(1)'), disp('W_MMc='), pretty(W_MMc)
```

дає результат

```
W_MMc=
      1
-----
      2
 s  s
--- + -- + 1
250 10
```

```
W_MMc=
      250
-----
      2
 s + 25 s + 250
```

9.7 Контрольні питання

- 1) Які правила еквівалентних структурних перетворень Ви знаєте?
- 2) Як описуються символьні змінні?
- 3) Які функції еквівалентних перетворень символьних виразів Ви знаєте? Які з них Ви використали у своїй програмі?

*Лабораторна робота №10***АНАЛІЗ ЛІНІЙНИХ ДИНАМІЧНИХ СИСТЕМ В СЕРЕДОВИЩІ
ПАКЕТА *MATLAB*****10.1 Теоретичні відомості**

Аналіз та синтез лінійних стаціонарних динамічних систем у середовищі пакета *MATLAB* виконують функції додатку ***Control Toolbox*** [12-16], інформацію про які можна отримати за допомогою команди `help control`.

Крім виконання аналізу систем у програмному режимі, можна скористатися інтерактивною оболонкою ***LTI Viewer***, яка активізується командою `ltiview`.

Для використання обох методів треба перш за все створити у середовищі *MATLAB* так званий *LTI*-об'єкт в одному з форматів математичного опису, розглянутих у лабораторній роботі №3 (підрозділ 3.1), та у роботі №5. Аналіз лінійних динамічних систем виконують як безпосередньо, аналізуючи якість перехідних процесів, так і опосередковано, використовуючи методи частотного аналізу та аналіз розташування нулів-полісів.

10.1.1 Характеристика показників якості перехідних процесів

Вигляд перехідного процесу (ПП) залежить від характеру дії, що його викликає.

Реакцію системи на одиничну функцію Хевісайда (стрибкоподібний сигнал одиничної амплітуди)

$$1(t) = \begin{cases} 0 & \text{при } t < 0, \\ 1 & \text{при } t \geq 0 \end{cases} \quad (10.1)$$

називають ***перехідною функцією*** $h(t)$ (***step response***), а реакцію на δ -функцію Дірака (імпульс нескінченної амплітуди та одиничної площини)

$$\delta(t) = \frac{d1(t)}{dt} = \begin{cases} 0 & \text{при } t < 0 \text{ \& } t > 0, \\ \infty & \text{при } t = 0 \end{cases} \quad (10.2)$$

– *ваговою функцією* $w(t)$ (*impulse response*).

Із (10.2) та визначення перехідної та вагової функцій випливає, що

$$w(t) = \frac{dh(t)}{dt} \quad (10.3)$$

Серед вхідних сигналів, що найчастіше використовуються при дослі-

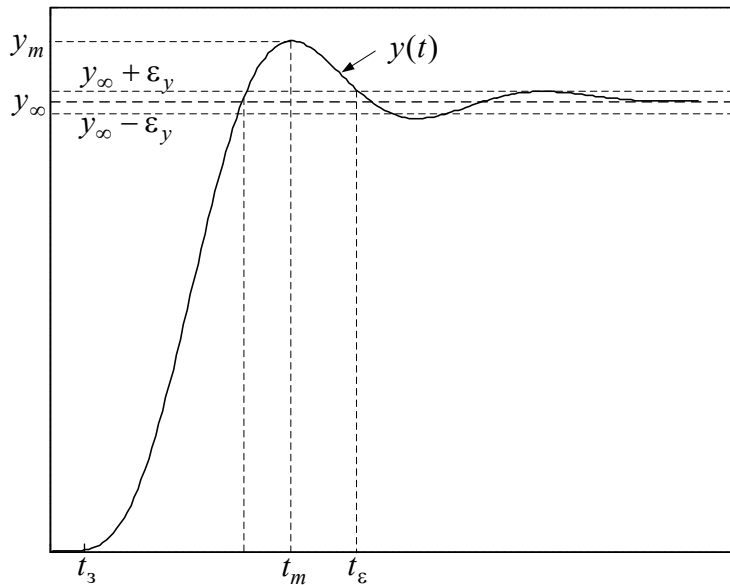


Рис.10.1. Параметри перехідного процесу

дженні САУ слід ще назвати лінійний і синусоїдальний сигнали та так званий “білий шум”.

Для оцінки якості перехідних процесів використовують такі поняття як *перерегулювання, коливальність, демпфірування, швидкодія*.

Загальний вигляд перехідного процесу $y(t)$ по-

дано на рис. 10.1. На рисунку позначені

Швидкодія процесу визначається декількома параметрами. До них належать:

- усталене y_∞ і максимальне y_m значення перехідної функції
- час першого узгодження сигналу $y(t)$ зі сталим значенням t_{1y} ;
- час досягнення перехідною $y(t)$ максимального значення t_m ;
- час установа реакції t_ϵ , який характеризує час входження перехідної функції в зону $y_\infty \pm \epsilon_y$, де $\epsilon_y = (0.01 \div 0.05)y_\infty$;
- час запізнювання реакції t_3 .

При проектуванні систем найчастіше намагаються забезпечити перехідні процеси з невеликим перерегулюванням (5-10%), невеликою коливальністю та максимально можливою швидкістю. Ще однією вимогою, що вступає

у супереч з швидкодією є перешкодозахищеність, тобто система повинна пропускати та досить швидко відпрацьовувати корисні сигнали та затримувати і послабляти сигнали перешкод. Системи повинні також мати малу чутливість до зміни параметрів. Цю якість систем звать робастністю.

Поширення *Control System Toolbox* дозволяє розрахувати та побудувати перехідні та імпульсні перехідні функції, реакцію системи на ненульові початкові умови і виконати моделювання лінійних систем при довільних вхідних сигналах. Перелік *MATLAB*-функцій, пов'язаних з розрахунком та графічним відображенням перехідних процесів у лінійних стаціонарних системах, наведено у табл. 10.1.

Таблиця 10.1

<code>initial</code>	Реакція системи на ненульові початкові умови
<code>step</code>	Перехідна функція
<code>impulse</code>	Вагова функція
<code>lsim</code>	Реакція системи на заданий вхідний вплив

Функції `initial`, `step`, `impulse` та `lsim` мають такі загальні властивості:

- при зверненні до них з лівосторонніми (вихідними аргументами) вони розраховують значення відповідних сигналів без побудови графіків, а при зверненні до них без вихідних аргументів – будують відповідні графіки без збереження у пам'яті результатів розрахунку;
- вихідними аргументами функцій можуть бути $[y, t, x]$, $[y, t]$ та y , де y – масив вихідних координат, t – вектор-стовпець моментів часу, x – масив змінних стану;
- параметри y та x є тривимірними; для масиву y перший індекс визначає момент часу, другий – номер виходу, а третій – номер входу;
- параметр t може бути як присутнім, так і відсутнім у списку вхідних (правосторонніх) аргументів;
- параметр t у списку вхідних аргументів може бути заданим скаляром

t_end або рівномірно розподіленим вектором $t0:dt:t_end$, де t_end – час закінчення розрахунку, $t0$ – час початку розрахунку (за замовченням $t0=0$), dt – шаг зміни часу (за замовченням він назначається автоматично на підставі аналізу параметрів моделі); при відсутності параметра t у списку вхідних аргументів всі його складові назначаються відповідною функцією системи *MATLAB* автоматично;

- для будь-якої багатомірної системи кожний графік від деякого входу до деякого виходу будується у власній системі координат власного підвікна загальної графічної фігури.

Наприклад, якщо для створеного у попередній лабораторній роботі об'єкту `dps` виконати команду

```
step(dps), grid on,
```

то отримаємо 4 перехідні функції, зображені на рис. 10.2

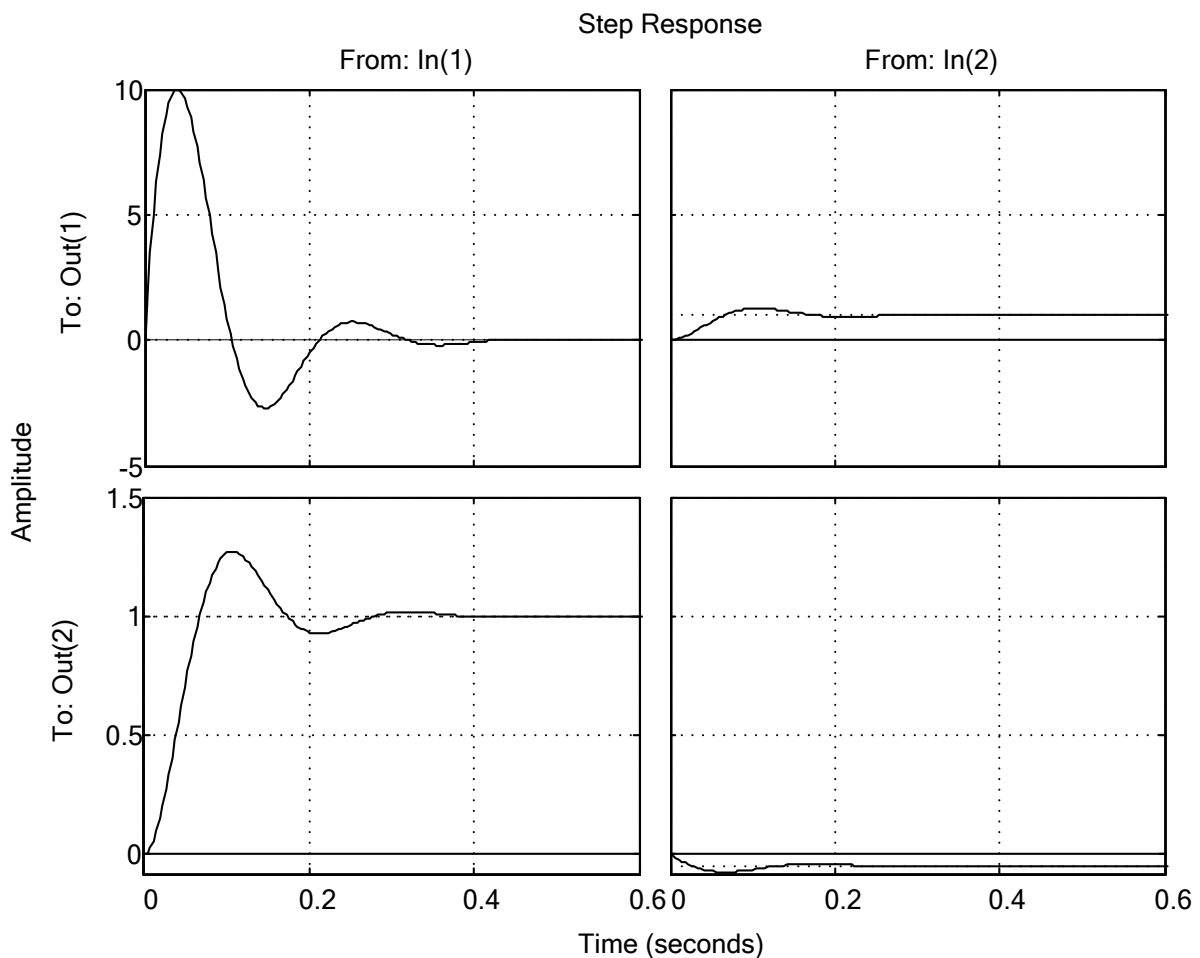


Рис. 10.2. Перехідні функції ДПС, отримані командою `step` без лівосторонніх аргументів

Аналогічні графіки можна отримати і за допомогою оболонки *LTI Viewer*.

Якщо спочатку розрахувати перехідні функції, а потім побудувати їх командою `plot`, то можна гнучкіше впливати на вигляд графіків. Наприклад, в результаті виконання команд

```
dps_tf=tf(dps)
Wdps_wU=dps_tf(2,1)
[y,t]=step(dps,0.5);
plot(t,y(:, :, 1)), legend('M/Mn (U/Un) ', 'w/w0 (U/Un) ')
grid on
figure
plot(t,y(:, :, 2)), legend('M/Mn (Mc/Mn) ', 'w/w0 (Mc/Mn) ')
grid on
```

отримуємо

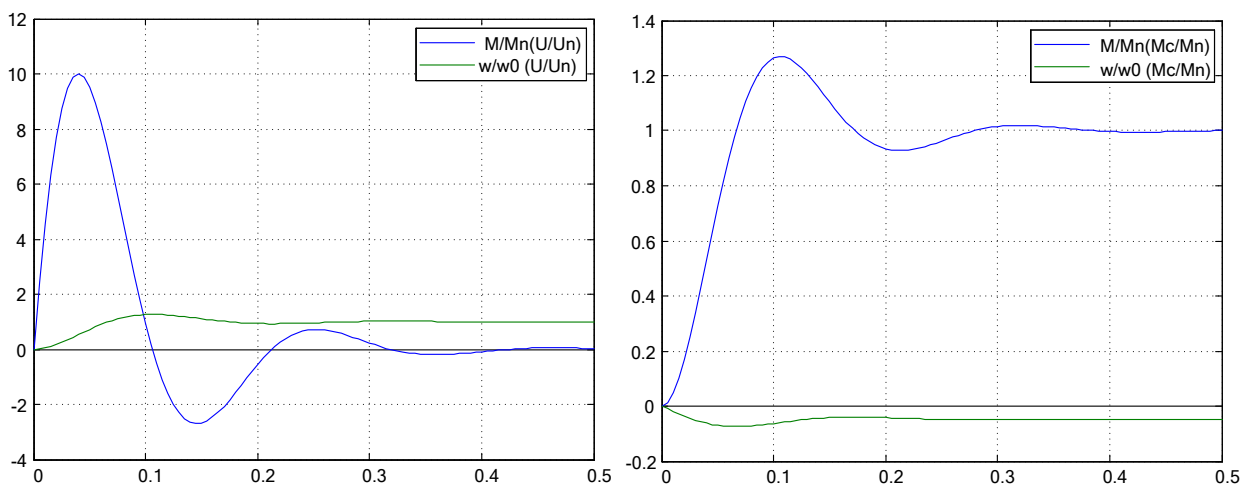


Рис. 10.3. Перехідні функції ДПС, отримані командами `plot` із застосуванням даних команди `step` з лівосторонніми аргументами

Зверніть увагу на формування у наведеній програмі матриць вихідних сигналів, що відображають реакцію системи на перший $y(:, :, 1)$ та другий $y(:, :, 1)$ вхідний сигнали.

Аналогічно можна отримати і вагові функції досліджуваної системи.

Команду `initial` можна застосовувати тільки для *ss*-об'єктів.

Для використання команди `lsim` треба до вхідних параметрів додати ще узгоджений з вектором часу t вектор вхідного сигналу u . Одним із способів формування цього сигналу є використання функції `gensig`.

10.1.2 Частотний аналіз

Важливу роль в аналізі динамічних систем грають характеристики, що пов'язані з описом систем у частотній області. Це логарифмічні частотні характеристики (діаграми Боде), амплітудно-частотні та фазово-частотні характеристики, амплітудно-фазові характеристики (годографи Найквіста) та інші.

Частотну передавальну функцію $W(j\omega)$ LTI-моделі можна отримати підстановкою $s = j\omega$ у звичайну передаточну функцію системи. При фіксованому значенні ω частотна характеристика є комплексним числом, і може бути представлена в вигляді:

$$W(j\omega) = W(s)|_{s=j\omega} = U(\omega) + jV(\omega) = W(j\omega)A(\omega)e^{j\varphi(\omega)}, \quad (11.4)$$

де

$$U(\omega) = \operatorname{Re}(W(j\omega)), \quad V(\omega) = \operatorname{Im}(W(j\omega)) \quad (11.5)$$

– **дійсна (ДЧХ) та уявна (МЧХ) частотні характеристики;**

$$A(\omega) = |W(j\omega)| \quad (11.6)$$

– **амплітудно-частотна характеристика (АЧХ);**

$$\varphi(\omega) = \arg(W(j\omega)) = \operatorname{arctg} \frac{V(\omega)}{U(\omega)} \quad (11.7)$$

– **фазо-частотна характеристика (ФЧХ).**

Амплітудно-частотна характеристика показує, як пропускає система сигнал різноманітної частоти. Оцінка пропускання робиться за відношенням амплітуд вихідної та вхідної величин.

Фазово-частотна характеристика показує фазові зсуви, що вносить система на різноманітних частотах.

Геометричне місце точок вектора $W(j\omega)$ на комплексній площині, тобто залежність $V(\omega)(U(\omega))$, побудовану у декартових координатах, або $A(\omega)(\varphi(\omega))$, побудовану у полярних координатах, при зміні частоти ω від 0 до ∞ , називається **амплітудно-фазовою характеристикою (АФХ) або частотним годографом Найквіста.**

Амплітудно-фазова характеристика дає можливість наочно представи-

ти для кожної частоти вхідного впливу відношення амплітуд вихідної та вхідної величин і зсув фаз між ними.

Широке практичне значення має **логарифмічна амплітудно-частотна характеристика (ЛАЧХ)**

$$L = 20 \lg A(\omega), \quad (11.8)$$

що відповідає вимірюванню амплітуди у децибелах (дБ). Її будують у логарифмічному масштабі як функцію від $\lg \omega$.

У парі з ЛАЧХ аналізують **логарифмічну фазо-частотну характеристику (ЛФЧХ)** $\varphi(\lg \omega)$. **Логарифмічних частотні характеристики (ЛЧХ)** називають **діаграмами Бode (Bode diagram)**.

За допомогою ЛЧХ розімкнених систем можуть бути визначені такі властивості замкнених систем, як частота зрізу, резонансні частоти, коефіцієнт передачі за постійним сигналом, запаси стійкості за амплітудою і фазою. Вони ж дають можливість приблизно оцінити швидкодію і перерегулювання замкненої системи.

Перелік найбільш важливих функцій для дослідження частотних характеристик *LTI*-моделей наведено у табл. 10.2.

Таблиця 10.2

bode	Логарифмічні амплітудні і фазові частотні характеристики (діаграми Бode)
margin	Обчислення запасів стійкості за фазою і модулем
norm	Обчислення максимального значення модуля частотної характеристики і частоти, що йому відповідає
nyquist	Частотний годограф Найквіста

Функції `bode` та `nyquist` мають такі загальні властивості:

- при зверненні до них з лівосторонніми (вихідними) аргументами вони розраховують значення відповідних сигналів без побудови графіків, а при зверненні до них без вихідних аргументів – будують відповідні графіки без збереження у пам'яті результатів розрахунку;

- розміри масивів `mag` (амплітуда) та `phase` (фаза) дорівнюють “кількість виходів” $\times$ “кількість входів” $\times$ “довжина вектору частот”;
- параметр `w` (частота) може бути як присутнім, так і відсутнім у списку вхідних (правосторонніх) аргументів;
- параметр `w` у списку вхідних аргументів може бути заданим вектором рівномірно розподіленим у логарифмічному масштабі на такий спосіб: `w = logspace(w0, w_end)` або `{w0, w_end}`; за замовченням вінзначається автоматично на підставі аналізу параметрів моделі);
- параметри досліджуваної системи можуть бути задані у початку списку вхідних аргументів у вигляді *LTI*-об’єкту будь-якого підкласу `sys`;

Вихідними аргументами функції `bode` можуть бути `[mag, phase, w]`, `[mag, phase]`, та `mag`, а вихідними аргументами функції `nyquist` – параметри `[Re, Im, w]`, `[Re, Im]`, та `Re`, де `mag` – масив амплітуд частотних характеристик у абсолютних одиницях (не в дБ), `phase` – масив фаз частотних характеристик у градусах, `w` – вектор-стовпець частот у рад/с, `Re` – дійсні складові годографу Найквіста, `Im` – уявні складові годографу Найквіста;

Функція `margin` обчислює запаси стійкості за фазою і модулем, а також відповідні частоти для *LTI*-моделей одномірних розімкнених систем. Значення запасів стійкості вказують, на скільки частотна характеристика розімкненої системи віддалена від критичної точки $(-1 + j0)$.

Запас стійкості за модулем – це значення амплітудної частотної характеристики на частоті, де фазова частотна характеристика має значення 180 град.

Запас стійкості за фазою – це відстань фазової частотної характеристики від значення -180 градусів на частоті зрізу.

Частота зрізу – це частота, де значення амплітудної частотної характеристики дорівнює 0 дБ, тобто точка перетину ЛАЧХ з віссю частоти.

Функція `margin`, яку можна застосовувати тільки для *SISO*-об’єктів, без лівосторонніх аргументів будує логарифмічні частотні характеристики

розімкненої системи з вказівкою запасів її стійкості. Вихідними аргументами її можуть бути $[G_m, P_m, W_{cg}, W_{cp}]$, $[G_m, P_m, W_{cg}]$, $[G_m, P_m]$ та G_m , де G_m – запас стійкості за амплітудою, P_m – запас стійкості за фазою, W_{cg} – частота, що відповідає точці визначення запасу за амплітудою, тобто частота перетину ЛФЧХ рівня -180° , W_{cp} – частота, що відповідає точці визначення запасу за фазою, тобто частота зрізу. Якщо існують декілька точок перетину амплітудною характеристикою осі частот і фазовою рівня -180° , то повертаються найменші значення відповідних запасів стійкості.

Наприклад, при виконанні команд

```
bode(dps), grid on
figure, margin(dps(2,1)), grid on
figure, nyquist(dps(2,1)), grid on
```

отримуємо діаграми Боде рис. 10.4, рис. 10.5 та ліаграму Найквіста рис. 10.6

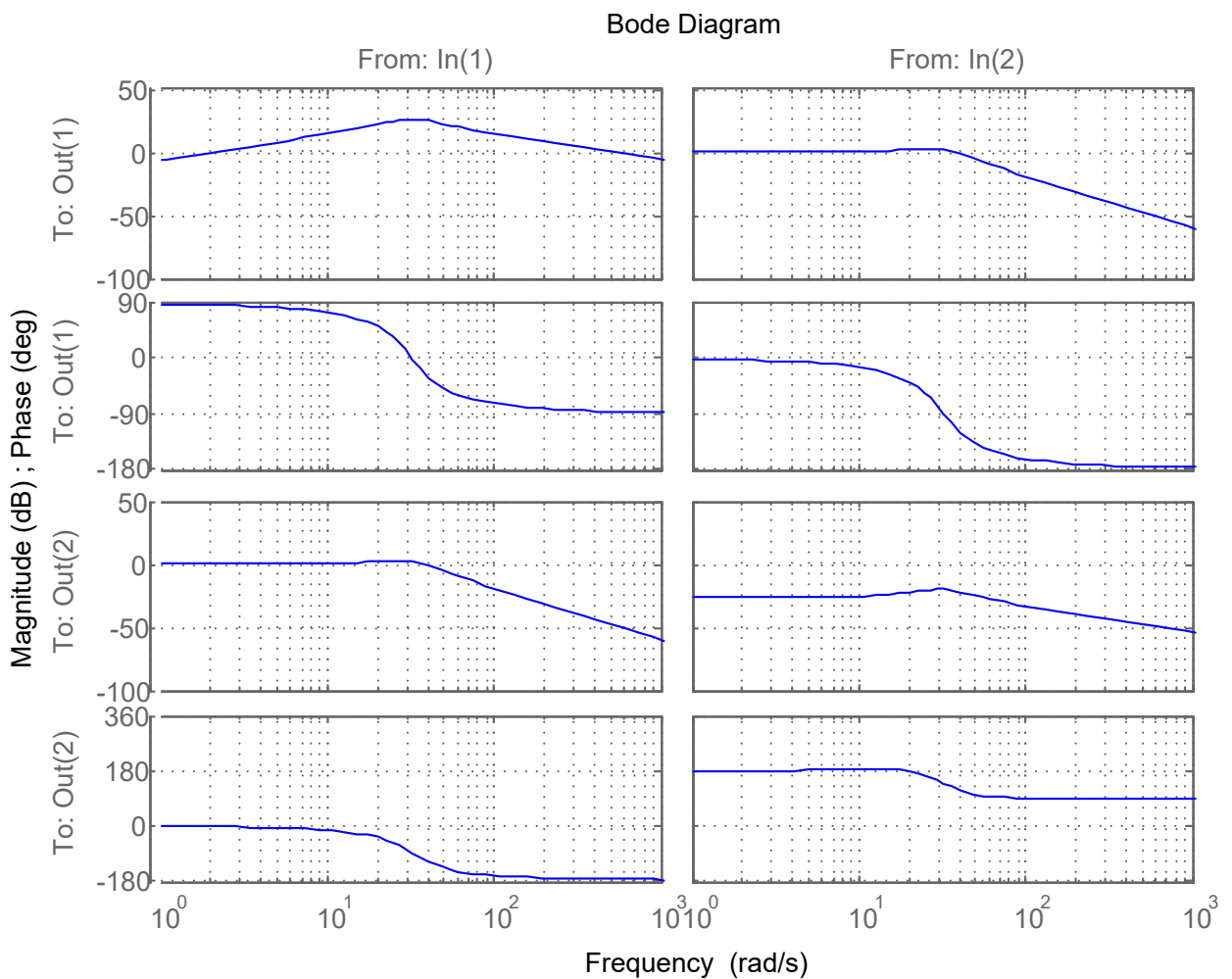


Рис. 10.4. Діаграми Боде для двигуна постійного струму

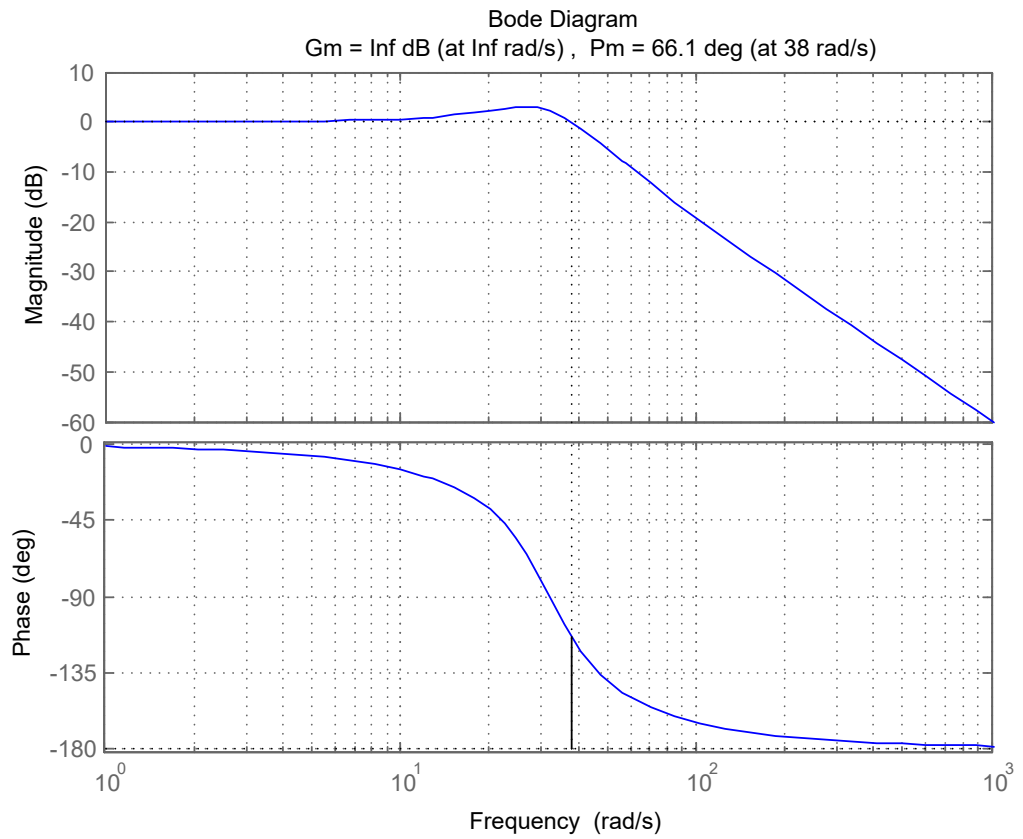


Рис. 10.5. Діаграми Бode для ДПС із зазначенням запасів стійкості, побудована за передавальною функцією $\omega(s)/U_{\text{я}}(s)$

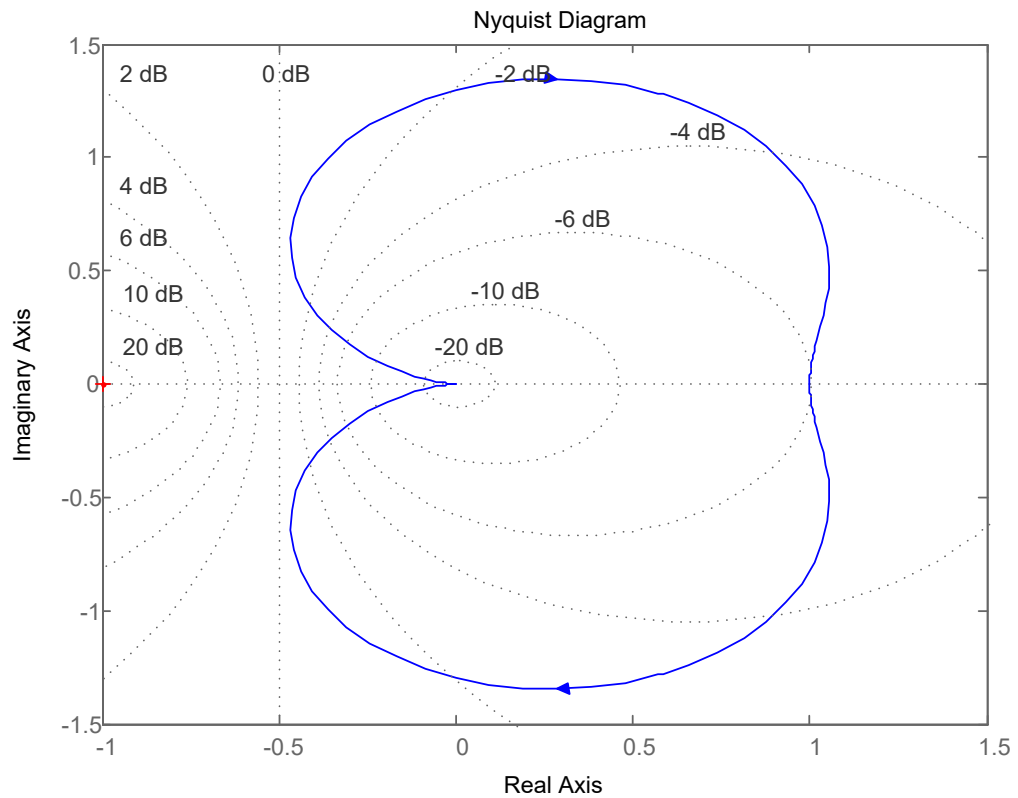


Рис. 10.6. Діаграма Найквіста для ДПС, побудована за передавальною функцією $\omega(s)/U_{\text{я}}(s)$

Для побудови інших частотних характеристик треба спочатку розрахувати необхідні сигнали, наприклад, при виконанні наступного фрагменту

```
[A,fi,w]=bode(dps(2,1),{0.1,250});
A=A(:);
figure, plot(w,A), grid on
[U,V,w]=nyquist(dps(2,1),{0.1,250});
U=U(:);
hold on, plot(w,U, '--')
legend('АЧХ', 'ДЧХ')
```

отримаємо графіки рис. 10.7.

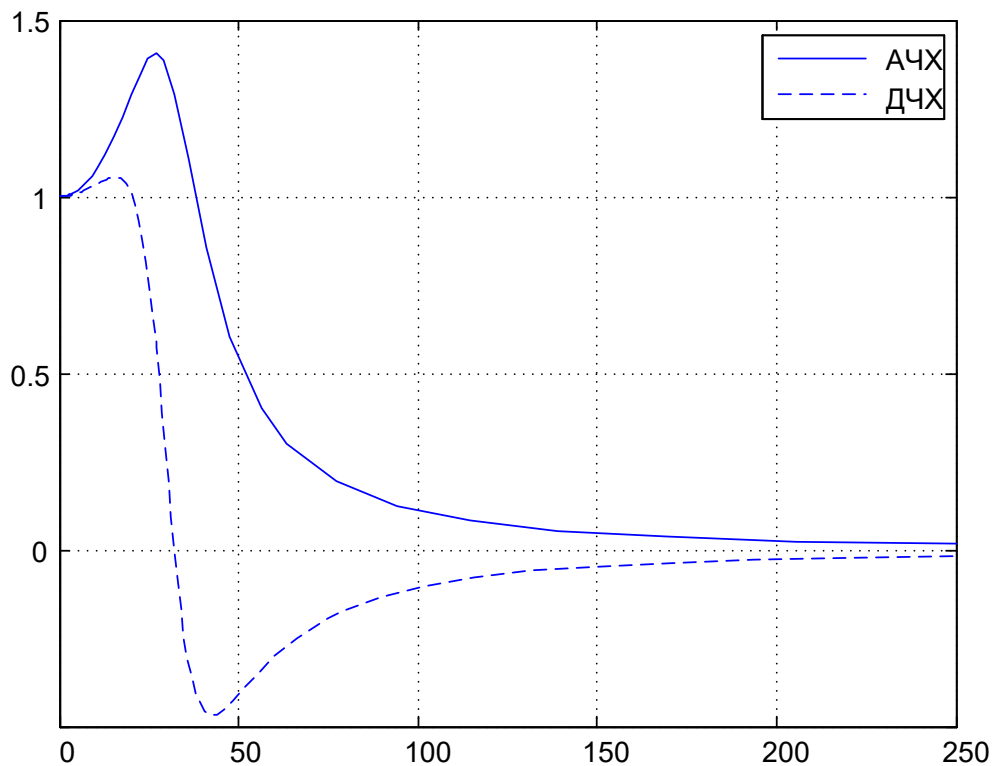


Рис. 10.7. АЧХ та ДЧХ ДПС, побудовані за передавальною функцією $\omega(s)/U_{\text{я}}(s)$

10.1.3 Аналіз лінійних динамічних систем за розподіленням полюсів та нулів

Знання розташування на комплексній площині полюсів і нулів передаточної функції дозволяє з достатньою точністю відповісти на запитання про характер перехідного процесу в системі.

Кількість нулів та полюсів визначається порядком поліномів у чисельнику та знаменнику передавальної функції. Якщо усі коефіцієнти цих поліномів є дійсними числами, то кожний з комплексних коренів відповідних ал-

гебраїчних рівнянь обов'язково має спряжену пару.

Що стосується полюсів, то перш за все треба знати, що у стійкої системи усі вони розташовуються зліва від уявної осі комплексної площини, тобто мають від'ємні дійсні частини. По-друге, треба розуміти, що найбільше на якість перехідного процесу впливає так звана **домінуюча пара комплексно-спряжених полюсів**, що розташована найближче до уявної осі.

Розглянемо одну пару комплексно-спряжених полюсів, розташування яких на комплексній площині зображено на рис. 10.8:

$$p_{12} = -\alpha \pm j\beta = \Omega e^{\pm j\psi} = \Omega e^{\pm j(\pi-\varphi)}, \quad (10.9)$$

де

α, β – відстані полюсів від уявної та дійсної осей;

$\Omega = \sqrt{\alpha^2 + \beta^2} = \sqrt{p_1 p_2}$ – амплітуда полюсів, або їх власна частота, Гц;

$\varphi = \arctg(\beta / \alpha)$.

Знайдемо характеристичний поліном, що утворює така пара полюсів;

$$G(s) = (s - p_1)(s - p_2) = s^2 - (p_1 + p_2)s + p_1 p_2 = s^2 + 2\alpha s + \Omega^2. \quad (10.10)$$

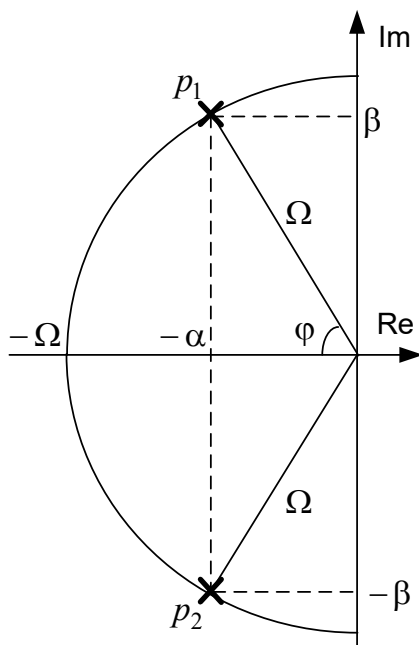


Рис. 10.8. Параметри полюсів

Прирівнявши його до характерис-

тичного полінома коливальної ланки

$$s^2 + 2\alpha s + \Omega^2 = s^2 + 2\xi\Omega s + \Omega^2,$$

отримуємо вираз для коефіцієнта демпфі-
рування ξ , що впливає на коливальність
перехідного процесу:

$$\xi = \frac{\alpha}{\Omega} = \cos\varphi. \quad (10.11)$$

При $\xi=1$ (полюс, розташований на
дійсній осі, $\varphi=0$) перехідний процес
протікає без перерегулювання. При $\xi<1$
($\varphi>0$) перехідні процеси мають колива-
льний характер, при чому перерегулю-

вання, та коливальність зростають зі зменшенням коефіцієнта демпфірування і при $\xi=0$ (полюси розташовані на уявній осі, $\varphi=90^\circ$), перерегулювання сягає 100% і система здійснює незагасаючі коливання, знаходячись на межі стійкості. Тому кут φ домінуючої пари полюсів не повинен перевищувати $(60-70)^\circ$.

Частота полюсів визначає швидкодію коливальної ланки. Відповідно швидкодію лінійної динамічної системи довільного порядку n визначає величина його середньгеометричного кореня (СГК)

$$\Omega_0 = \sqrt{p_1 p_2 \dots p_n} = \sqrt{\Omega_1 \Omega_2 \dots \Omega_n}. \quad (10.12)$$

Мале значення СГК обмежує швидкодію системи, а занадто велике сприяє підсиленню перешкод.

Щодо розташування нулів на комплексній площині треба враховувати, що розташування нулів на уявній осі зменшує і перерегулювання, і швидкодію системи, а розташування їх на дійсній осі підвищує ці параметри. При куті $(40-45)^\circ$ часто вдається підвищити швидкодію без збільшення перерегулювання.

У табл. 11.3 наведено перелік функцій додатку *Control System Toolbox*, які дозволяють виконати обчислення, сортування і графічне відображення полюсів і нулів *LTI*-моделі на комплексній площині.

Таблиця 10.3

pole	Обчислення полюсів
tzero	Обчислення нулів
pzmap	Розрахунок або відображення полюсів та нулів на комплексній площині
sgrid	Нанесення сітки постійних ліній коефіцієнта демпфірування та частоти на площині s
dcgain	Обчислення коефіцієнта передачі
damp	Обчислення власних частот та коефіцієнтів демпфірування

Результат дії цих функцій продемонструємо на прикладі ДПС:

```

k=dcgain(dps)
Z=tzero(dps)
P=pole(dps)
damp(dps)
figure, pzmap(dps), sgrid
figure, pzmap(dps), sgrid, axis equal

```

k =

0 1.0000

1.0000 -0.0472

Z =

Empty matrix: 0-by-1

P =

-12.3462 +29.5662i

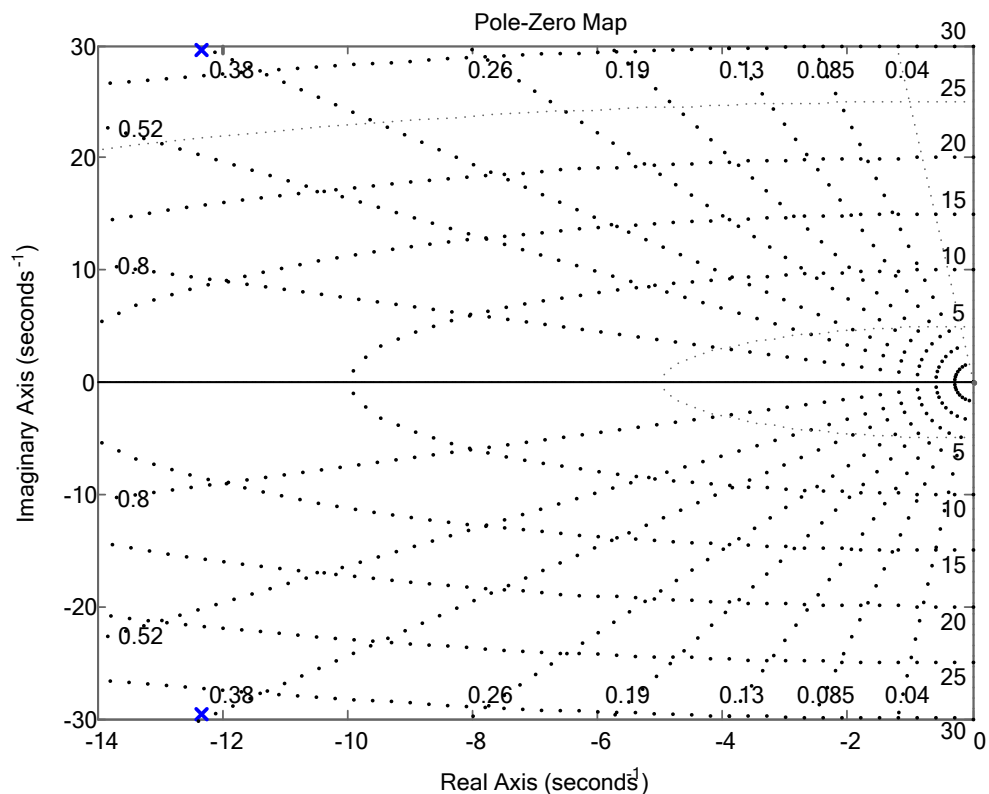
-12.3462 -29.5662i

Eigenvalue Damping Frequency

-1.23e+01 + 2.96e+01i 3.85e-01 3.20e+01

-1.23e+01 - 2.96e+01i 3.85e-01 3.20e+01

(Frequencies expressed in rad/seconds)



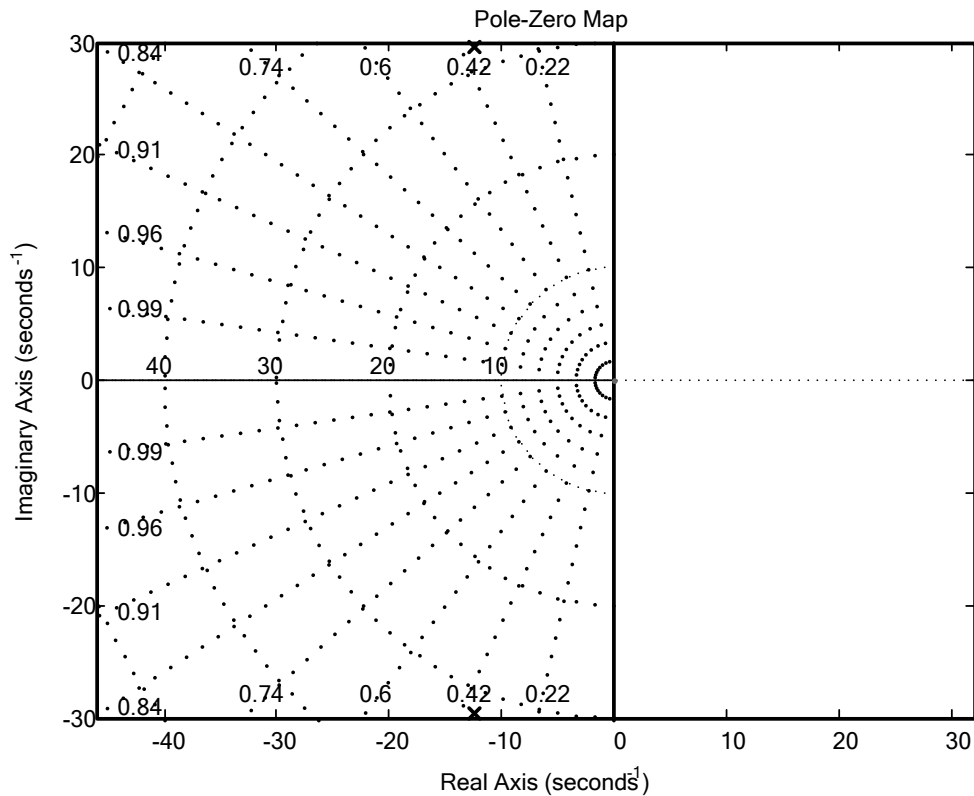


Рис. 10.8. Карти розташування нулів-полісів ДПС

10.2 Завдання

Для структурних схем, заданих у попередній лабораторній роботі,

1) побудувати

- перехідні та вагові функції за задавальним та збурюючим сигналами,
- діаграми Бодє розімкненої системи,
- діаграму Найквіста за передавальною функцією за задавальною дією,
- АЧХ та ДЧХ,
- карту розташування нулів та полюсів;

2) розрахувати

- коефіцієнти підсилення,
- нулі та полюси,
- параметри полюсів;

3) визначити

- показники якості перехідних процесів,
- параметри характерних точок діаграм Бодє,

- сталість системи за розташуванням нулів-полісів, за діаграмами Боде та за діаграмою Найквіста.

10.3 Порядок виконання роботи

- 1) Розробіть структурні моделі досліджуваних систем.
- 2) За розробленими структурними схемами створіть лінійні динамічні об'єкти досліджуваних систем.
- 3) Апробуйте дію основних функцій аналізу в режимі командного інтерпретатора.
- 4) Розробіть програму для виконання завдань лабораторної роботи, виконайте її та проаналізуйте результати.
- 5) Ознайомтеся з оболонкою LTI Viewer. Повторіть виконання завдань в інтерактивному режимі. Знайдіть шляхи подання результатів у найкращому вигляді.

10.4 Методичні рекомендації

- 1) Скористайтесь результатами визначення математичного опису системи з попередньої лабораторної роботи.
- 2) При виконанні завдань використовуйте наведені приклади.

10.5 Контрольні запитання та завдання

- 1) Як можна подивитись перелік функцій, що виконують аналіз лінійних динамічних систем?
- 2) На які групи можна поділити функції аналізу?
- 3) Який правосторонній аргумент присутній в усіх функціях аналізу?
- 4) Перелічіть сигнали, реакції яких зазвичай досліджують при аналізі динамічних систем.
- 5) Які показники якості перехідних процесів Ви знаєте? Як їх визначити програмно?
- 6) Які частотні характеристики Ви знаєте? Як їх можна побудувати?
- 7) Які параметри полісів Ви знаєте? Нащо вони впливають?

*Лабораторна робота №11***ВІРТУАЛЬНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ КІЛ З
ВИКОРИСТАННЯМ БЛОКІВ БІБЛІОТЕК *SimPowerSystem*****11.1 Основні відомості про бібліотеку *SimPowerSystem*****та правила її застосування**

Програма *Simulink* поряд з власними блоками, розглянутими у роботах 1-8 може використовувати блоки бібліотеки *SimPowerSystems (SPS)*, призначеної для віртуального фізичного моделювання електротехнічних, електромагнітних, електронних та електромеханічних пристроїв та схем.

В останніх версіях *Simulink* розширення *SimPowerSystems* входить до складу додатку віртуального фізичного моделювання *SimScape* поряд з фундаментальною бібліотекою процесів *Foundation Library (Electrical, Hydraulic, Magnetic, Mechanical, Pneumatic, Thermal)* та бібліотеками *SimElectronics, SimMechanics, SimHydraulics* та *SimDriveline*.

На відміну від *Simulink*-блоків, *SPS*-блоки подано у вигляді позначень відповідних елементів на принципових електричних схемах. З'єднуючись між собою, ці блоки утворюють електричні кола. Математичний опис окремих елементів приховано від користувача, завдяки чому створюється ілюзія віртуального фізичного моделювання. Насправді ж кожному з блоків *SimPowerSystems* поставлено у відповідність неперервні та дискретні *Simulink*-моделі, які можна побачити після завантаження файлів *powerlib_models*, *powerlib_extras*, *powerlib_meascontrol*. Шлях до цих файлів у версії *R2013a* має вигляд: *ProgramFiles\MATLAB\R2013a\toolbox\physmod\powersys\powersys*.

Отже, не зважаючи на ілюзію фізичного моделювання, користувач повинен чітко уявляти, що *SPS*-блоки створені на основі математичного опису, складеного розробниками пакету із певними припущеннями, які необхідно знати для адекватного застосування *SPS*-моделей в процесі досліджень. Для цього треба користуватись довідковою інформацією за допомогою функції

help. Якщо цієї інформації не вистачає, можна проаналізувати відповідні *Simulink*-моделі.

SPS-блоки мають такі особливості [7]:

- їх входи та виходи, на відміну від *Simulink*-блоків (*S*-блоків), не вказують напрямок передачі сигналу, бо вони фактично є еквівалентами електричних контактів;
- лінії зв'язку між *SPS*-блоками є моделями ідеальних (без опору) електричних проводів, по яким струм може протікати в обох напрямках;
- *SPS*- та *S*-блоки не можуть з'єднуватися один з іншим безпосередньо; сигнал від *S*-елементів можна передати до *SPS*-елементів тільки через керовані джерела енергії (Controlled Voltage/Current Source) *SPS*-бібліотеки *Electrical Sources*, а навпаки – через блоки бібліотек засобів вимірювання (*Measurements*);
- в моделі, яка отримує в собі *SPS*-блоки, має бути присутнім хоча б один з вимірювальних *SPS*-приборів, що пов'язано з особливостями перетворення *SPS*-моделі в еквівалентну розрахункову *S*-модель;
- в *SPS*-модель необхідно встановлювати блок *powergui*.

Зміст бібліотеки *SimPowerSystems* показано на рис. 11.1.

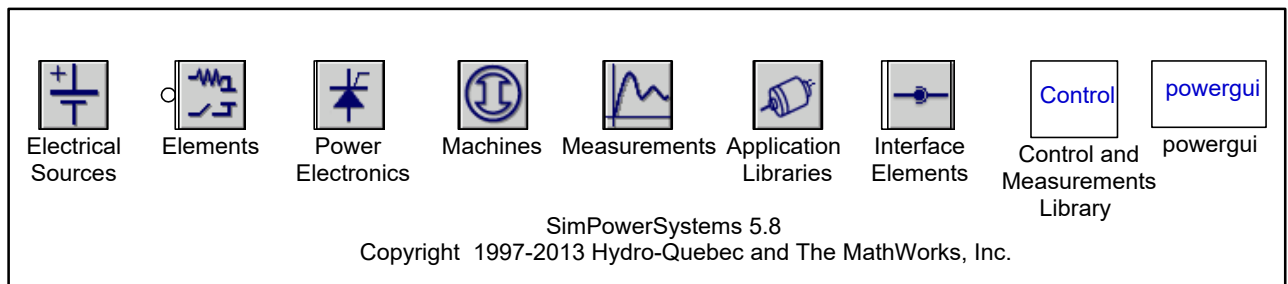


Рис. 11.1 Розділи бібліотеки *SimPowerSystems*

Вона отримує такі розділи:

- *Electrical Sources* – джерела електричної енергії;
- *Elements* – пасивні електротехнічні елементи;
- *Power Electronics* – пристрої силової електроніки, керовані напівпровідникові ключі;

- *Machines* – моделі електричних машин;
- *Measurements* – вимірювальні прилади;
- *Application Libraries* – прикладні бібліотеки;
- *Interface Elements* – інтерфейсні елементи;
- *Control and Measurements Library* – бібліотека керування та вимірювання;
- *PowerGUI* – графічний інтерфейс користувача.

Перелік блоків *SimPowerSystems*, необхідний для моделювання та аналізу властивостей розгалужених лінійних електричних кіл постійного і змінного струмів, надано на рис 11.2. Вони знаходяться у різних розділах бібліотеки *SPS*.

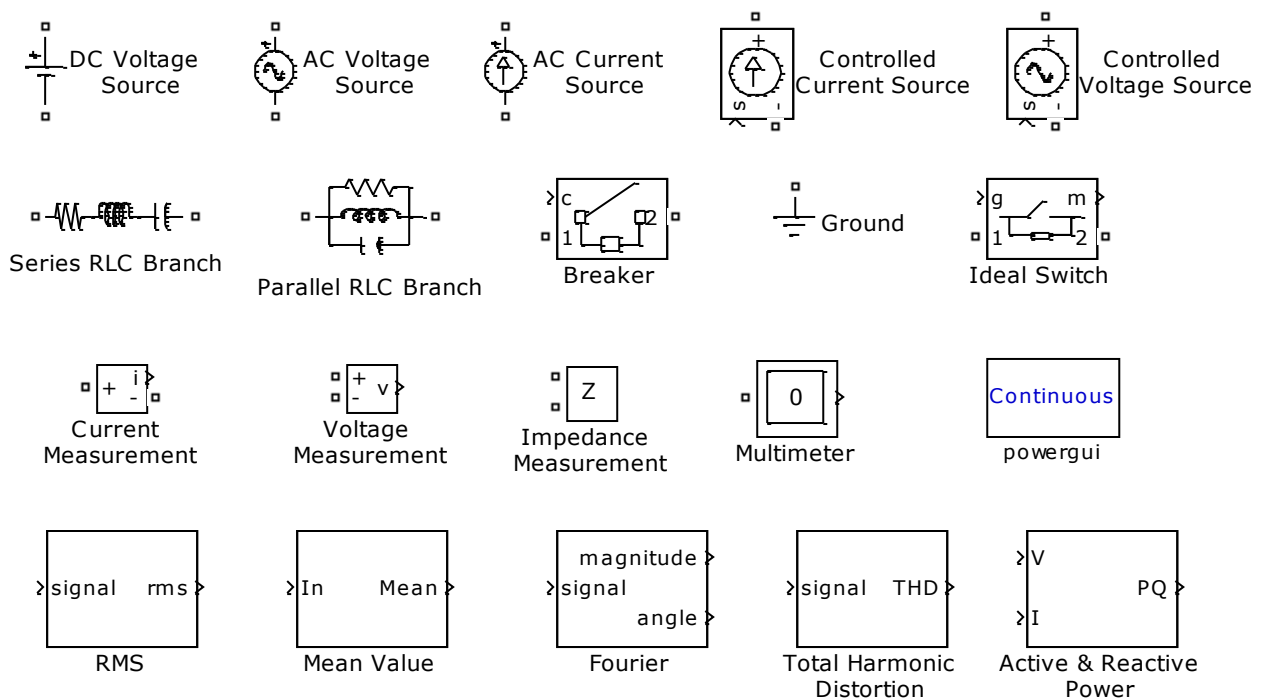


Рис. 11.2. *SPS*-блоки для віртуального фізичного моделювання простих електричних кіл

Подані блоки мають порти двох типів: *SPS*-порти, позначені дрібними квадратами та *S*-порти, позначені кінцівками стрілок, що допомагає запобігти помилок при з'єднанні блоків.

Параметри блоків можна розділити на **основні**, які пов'язані безпосередньо з фізичною сутністю блоків і без встановлення яких робота блоків

принципово не можлива, і **допоміжні**, які в багатьох випадках можна залишати такими, що встановлені за замовченням.

До допоміжних параметрів відносяться:

1) *Measurements* – здійснює вибір сигналів, які стають доступними для вимірювання блоком *Multimeter*. У меню, що випадає, можна обрати такі значення цього параметру:

- *None* – немає сигналів, доступних для вимірювання;
- *Voltage* – (для джерел напруги);
- *Current* – можна виміряти струм (для джерел струму);
- *Branch voltage* – можна виміряти напругу *RLC*-гілки або будь-якого сполучення резистора, котушки індуктивності та конденсатора (для блоків *RLC Branch* та *RLC Load*);
- *Branch current* – можна виміряти струм *RLC*-гілки;
- *Branch voltage and current* – можна виміряти струм і напругу *RLC*-гілки.

2) *Set the initial inductor current* у сполученні з параметром *Inductor initial current (A)*, дозволяють встановити початкове значення струму індуктивного елемента (для блоків *RLC Branch* та *RLC Load*).

3) *Set the initial capacitor voltage* у сполученні з параметром *Capacitor initial voltage (V)*, дозволяють встановити початкове значення напруги конденсатора (для блоків *RLC Branch* та *RLC Load*).

4) *Sample Time* – період дискретизації; встановлюється тільки при моделюванні дискретних процесів; при моделюванні неперервних процесів його залишають нульовим (0).

При створенні віртуальних фізичних моделей електричних кіл дотримуйтесь таких правил:

- 1) Не забувайте встановити блок *powergui*.
- 2) Враховуйте, що в *SPS*-моделях, на відміну від *S*-моделей, спочатку розраховується усталений режим.
- 3) Для вимірювання напруг та струмів треба, як мінімум, або підключити у схему амперметри та вольтметри, або скористатися мультиметром. До ви-

хідних *S*-портів вимірювальних *SPS*-блоків приєднайте якісь із блоків *Simulink*-бібліотеки *Sinks*. Що здійснюють реєстрацію та/або візуалізацію результатів.

4) Розгалуження електричного кола не можна виконати з точок, що позначають *SPS*-порти та з точок між *SPS*-портами одного блоку. Тому, якщо Ви хочете, наприклад, виміряти напруги на кожному з елементів послідовної *RLC*-гілки, то треба включити послідовно 3 блоки *RLC Branch*, один з яких зробити чисто активним, другий – чисто індуктивним, а третій – чисто ємнісним.

5) Для формування напруг та струмів складної форми використовуйте будь-які *S*-блоки, сформований сигнал подавайте на входні *S*-порти контрольованих *SPS*-джерел електричної енергії *Controlled Voltage/Current Source* з *SPS*-бібліотеки *Electrical Sources*.

6) При виборі методу та параметрів чисельного інтегрування диференціальних рівнянь, що здійснюється при виконанні команди *Mogel Configurations Parameters* меню *Simulink*, ретельно аналізуйте можливий характер та тривалість перехідних процесів. При недостатній кількості розрахованих точок для якісної візуалізації переходьте від методу *ode45*(вкладка *Solver*), що пропонується за замовченням до методів *ode23s*, *ode15s*, *ode23tb* або при використанні методу *ode45* встановлюйте значення параметра *Refine Factor* (вкладка *Data Import/Export, Refine* – підвищувати якість) більшим за 1 (максимальне значення дорівнює 10), при використанні якого щільність точок збільшується в установлене число разів методом інтерполювання. У колах змінного струму для якісної візуалізації іноді достатньо встановити максимальний шаг інтегрування (*Max Step Size*) таким, щоб на періоді синусоїди розраховувалась достатня кількість точок, наприклад, $h_{\max} = 1/(50 \div 100)f$.

7) Для більшої наочності доцільно більшість блоків перейменувати у відповідності з досліджуваною електричною схемою, а імена вимірювальних блоків приховати (*Diagram*→*Format*→*Hide Block Name*), якщо їх призначення зрозуміло із піктограми блоку.

11.2 Характеристика основних блоків для моделювання кіл постійного струму та однофазних кіл змінного струму

Розглянемо призначення та основні параметри блоків, наведених на рис. 11.2.

11.2.1 Джерела електричної енергії

1) **DC Voltage Source** – ідеальне джерело постійної напруги з основним параметром *Amplitude (V)* – величина постійної напруги U_c (В).

2) **AC Voltage Source** – ідеальне джерело синусоїдальної напруги (з нульовим активним опором) з параметрами

- *Peak Amplitude (A)* – амплітуда напруги U_m (А),
- *Phase (deg)* – початкова фаза φ° (град.),
- *Frequency (Hz)* – частота f (Гц), що формує напругу

$$U(t) = U_m \sin\left(2\pi ft + \frac{\varphi^\circ}{180^\circ} \pi\right). \quad (11.1)$$

3) **AC Current Source** – ідеальне джерело синусоїдального струму (з нульовою провідністю) з параметрами

- *Peak Amplitude (A)* – амплітуда струму I_m (А),
- *Phase (deg)* – початкова фаза φ° (град.),
- *Frequency (Hz)* – частота f (Гц), що формує струм

$$I(t) = I_m \sin\left(2\pi ft + \frac{\varphi^\circ}{180^\circ} \pi\right). \quad (11.2)$$

Оскільки джерело постійного струму у бібліотеках *SPS* відсутнє, то замість його можна використати блок *AC Voltage Source* з параметрами:

$$I_m = I_c, \quad f = 0, \quad \varphi^\circ = 90^\circ.$$

4) **Controlled Voltage Source, Controlled Current Source** – керовані джерела напруги і струму. Використовуються для створення джерел, які виробляють струм або напругу, що відрізняються від постійної величини або від синусоїдальних сигналів (1.1), (1.2). Для цього треба сформувати бажані сиг-

нали в *Simulink* і подати їх на *S*-входи керованих джерел електроенергії, які перетворюють *S*-сигнали у *SPS*-напруги або *SPS*-струми, що живлять віртуальні електричні кола через *SPS*-контакти, позначені як «+» та «-».

При використанні керованих джерел за прямим призначенням немає сенсу виконувати ініціалізацію контрольованих джерел.

Окрім *S*-блоків, включених до складу базових *Simulink*-бібліотек, до *SPS*-бібліотеки також залучено деякі *S*-блоки, які можуть стати у пригоді при створенні та при аналізі віртуальних електротехнічних схем. Зокрема, у розділі *Control and Measurements Library* знаходиться група блоків *Pulse & Signal Generators*, які можна застосовувати при формуванні нестандартних джерел напруги. Частина з них показана на рис. 11.3.

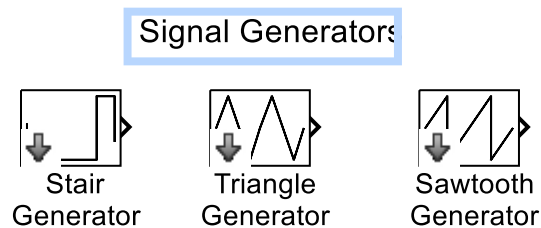


Рис. 11.3. Блоки із групи *Pulse & Signal Generators*

Блок *Stair Generator* (Ступінчастий генератор) формує послідовність ступінчатих сигналів на основі блоку *Look-Up Table*, як це показано на рис. 11.4. Його параметрами є координати точок, в яких відбувається ступінчата зміна вихідного сигналу, а саме: вектор часу – *Time (s)* та вектор амплітуд – *Amplitude*. Його зручно використовувати для керування станом блоку *Ideal Switsh*.

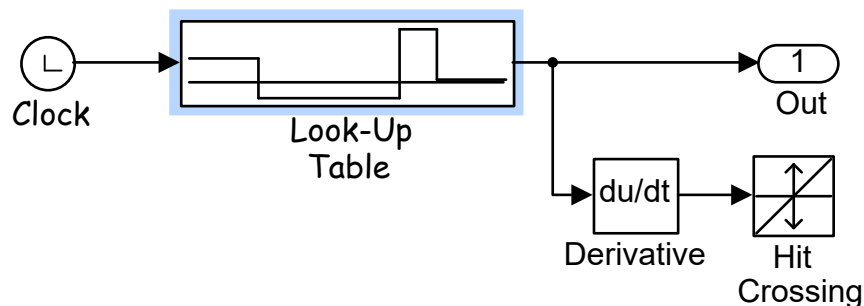


Рис. 11.4. Структура блоку *Stair Generator*

Блок *Triangle Generator* (Трикутний Генератор) формує трикутний, а блок *Sawtooth Generator* пилоподібний періодичні сигнали одиничної амплітуди з заданою частотою *Frequency (Hz)* та фазою *Phase (degree)*. Структура блоку *Sawtooth Generator* показано на рис. 1.5.

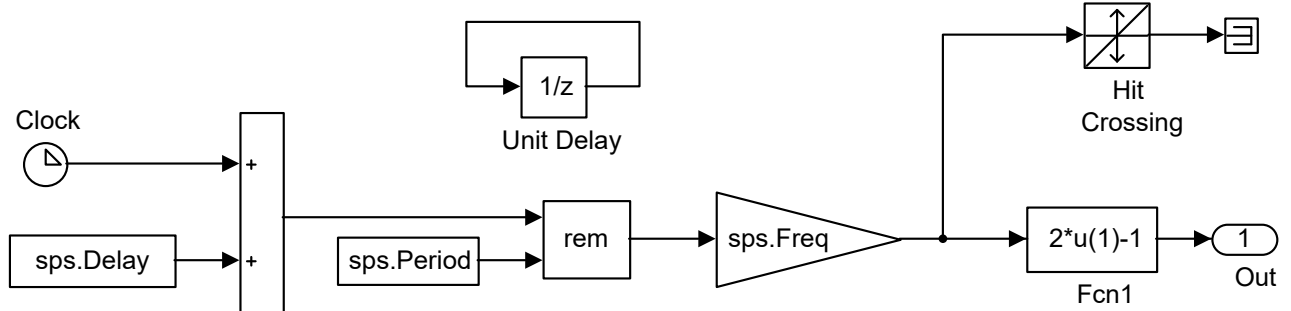


Рисунок 11.5 – Структура блоку *Sawtooth Generator*

11.2.2 Електротехнічні елементи

1) ***Series RLC Branch*** – послідовне *RLC*-з'єднання з такими основними параметрами:

- *Branch type* – визначає комплектність з'єднання, обирається із випадваючого меню і може приймати одне із значень: *RLC, RL, RC, LC, R, L, C*;
- *Resistance R (Ohms)* – активний опір, Ом;
- *Inductance L (H)* – індуктивність, Гн;
- *Capacitance C (F)* – ємність, Ф.

Вікно введення параметрів цього блоку відображене на рис. 11.6.

2) ***Parallel RLC Branch*** – паралельне *RLC*-з'єднання з такими ж основними параметрами, як і блок *Series RLC Branch*.

11.2.3 Основні вимірювальні прилади

1) ***Voltage Measurement*** – ідеальний вольтметр.

Перетворює різницю потенціалів точок, приєднаних до входів «+» і «-», в еквівалентний *S*-сигнал на виході «v», який можна перетворювати, фіксувати або візуалізувати будь-яким блоком *Simulink*-бібліотек.

2) ***Current Measurement*** – ідеальний амперметр.

Перетворює *SPS*-сигнал електричного струму, що протікає у гілці, в яку

включено прибор через контакти «+» і «-», в еквівалентний *Simulink*-сигнал на *S*-виході «i», який можна перетворювати, фіксувати або візуалізувати будь-яким блоком *Simulink*-бібліотек.

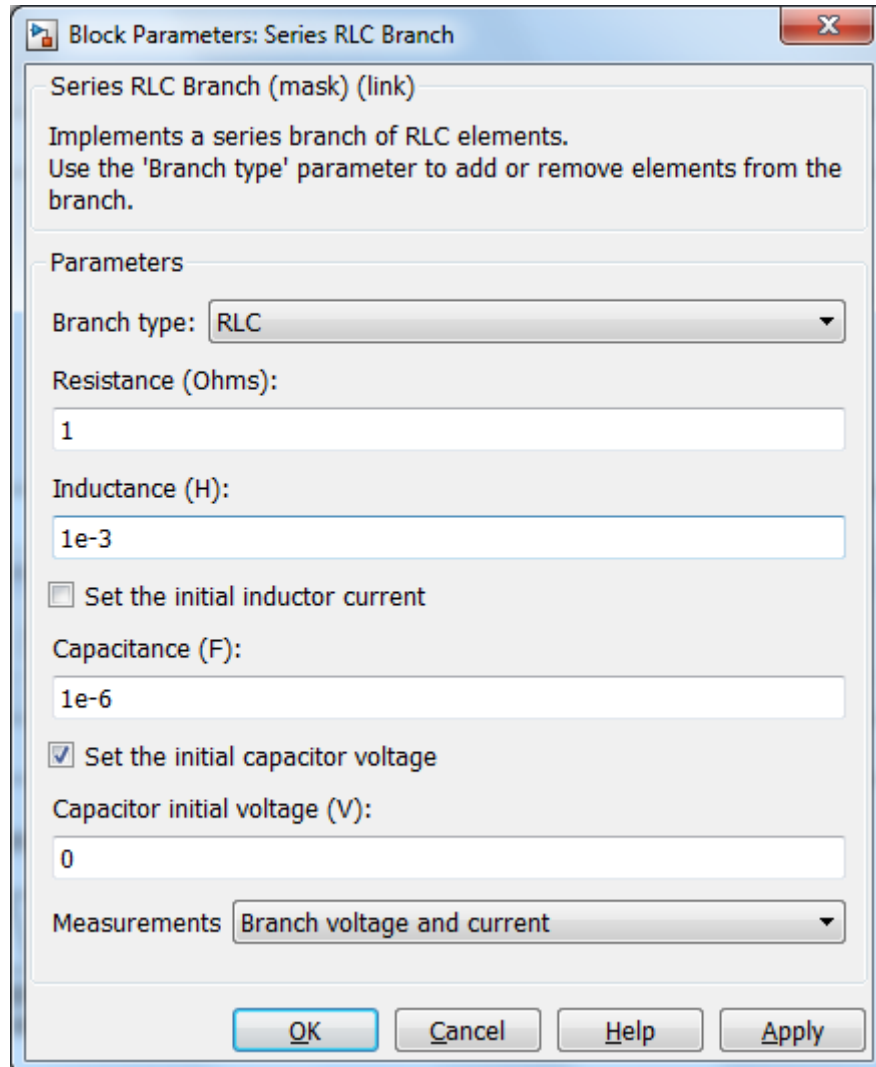
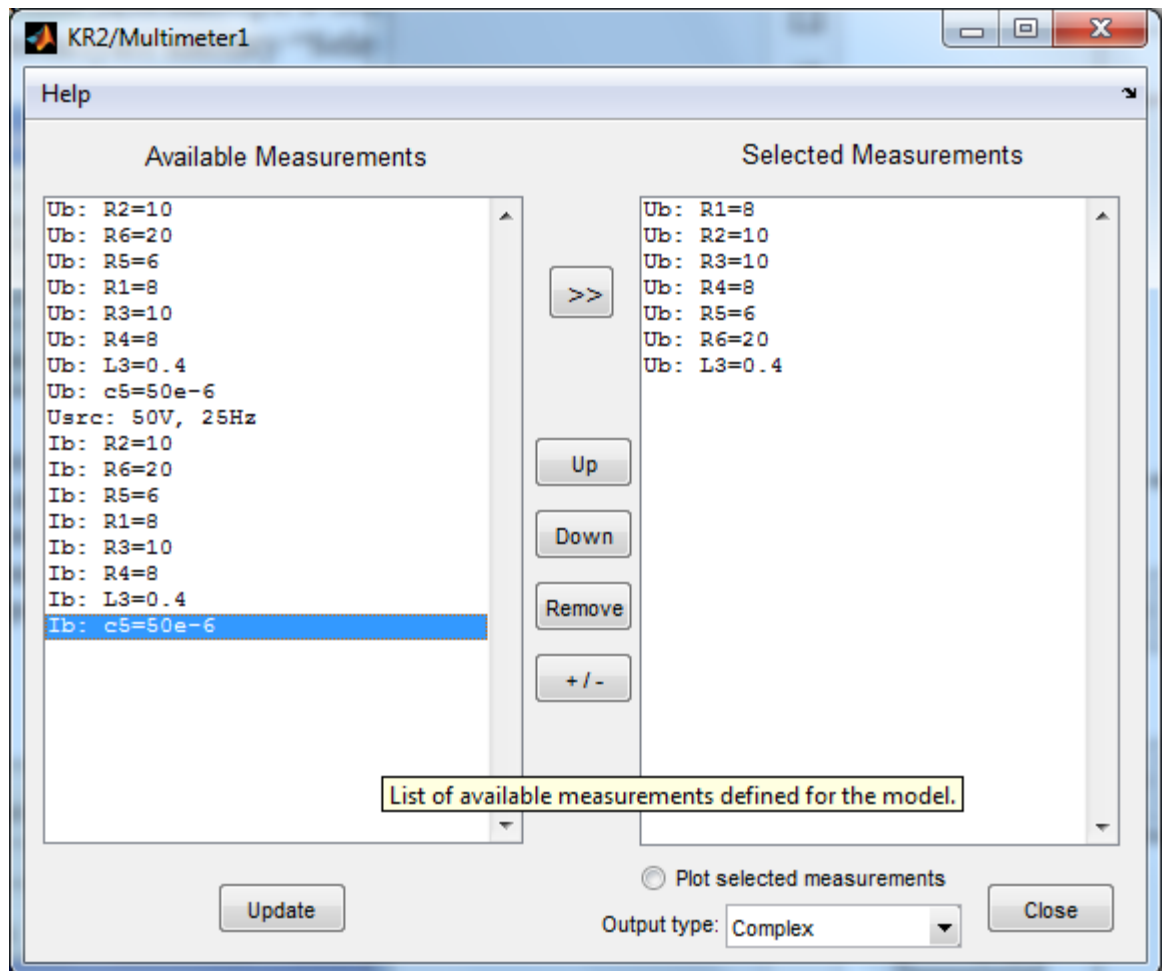


Рис. 11.6. Вікно параметрів блоку *Series RLC Branch*

3) *Multimeter* – багатоканальний вимірювач струмів та напруг.

Сигнали, доступні для вимірювання цим прибором повинні бути обрані у блоках елементів схеми за допомогою допоміжного параметра *Measurement*, як це описано у пункті 11.2.

Якщо такі сигнали існують, то після подвійного щиклика по піктограмі (іконці) блока відчиняється вікно, зображене на рис. 11.7.

Рис. 11.7. Вікно налаштування блоку *Multimetr*

У колонці *Available Measurements* відображається список величин, доступних для вимірювання, а у колонці *Selected Measurements* – список величин, обраних для вимірювання.

Для вибору колонці *Available* виділяють мишкою потрібний сигнал і кнопкою *>>* дублюють його у колонку *Selected Measurements*.

Видалити сигнал з колонки *Selected Measurements* можна кнопкою *Remove*. Змінити порядок сигналів у колонці *Selected Measurements* можна кнопками *Up* і *Down*. Кнопкою *+/-* можна міняти полярність обраних сигналів. На піктограмі блока автоматично відображується кількість обраних сигналів.

Обновити список доступних сигналів після внесення змін у віртуальну модель можна кнопкою *Update*.

Якщо встановити прапорець у полі параметру *Plot selected measurements*, то після розрахунку перехідних процесів відкриється вікно з графіками

обраних сигналів. Отже, у самому простому випадку у блоці можна не використовувати а ні вхідні, а ні вихідні порти.

4) **Impedance Measurement** – вимірювач повного (комплексного) опору (імпедансу) ділянки електричного кола у заданому діапазоні частот.

Відображення амплітудної та фазової частотних характеристик імпедансу відбувається за допомогою блоку *Powergui* у вікні, що відчиняється кнопкою *Impedance vs Frequency Measurements* (див. рис. 11.8). У цьому вікні можна скоригувати діапазон частот, нанести координатні сітки, обрати логарифмічний або лінійний масштаби зображення частотних характеристик, запам'ятати результати розрахунків.

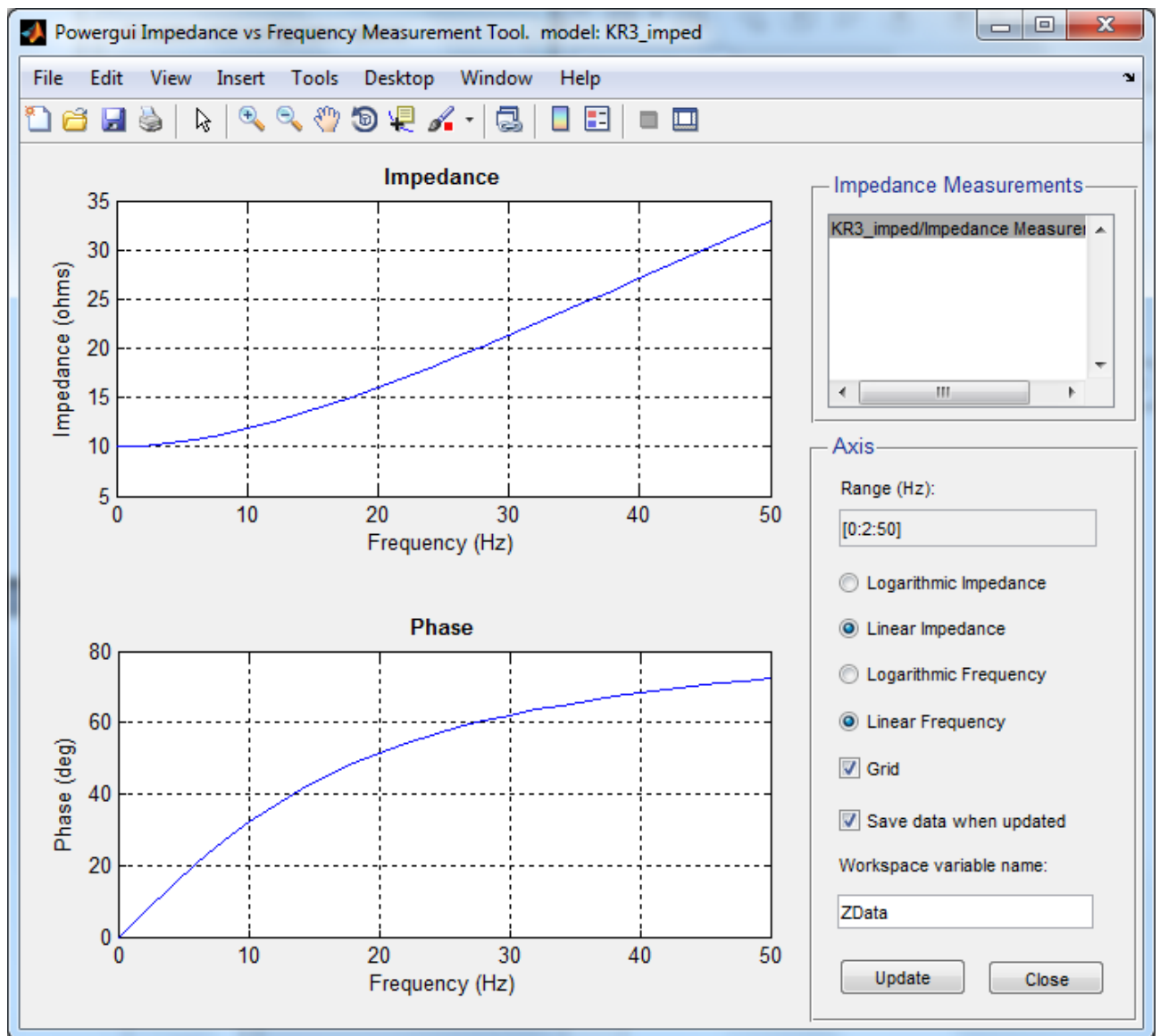


Рис.11.8. Вікно відображення частотних характеристик комплексного опору ділянки електричного кола

При використанні вимірювача повного опору слід мати на увазі, що цей блок виконано на основі джерела струму. Тому його не можна включати послідовно з індуктивними елементами. Для усунення цього обмеження **блок *Impedance Measurement* при наявності у вимірюваній ділянці котушок індуктивності слід шунтувати резистором із достатньо великим опором.** Величину опору треба обирати такою, щоб властивості схеми суттєво не змінювались.

Для вимірювання комплексних опорів пасивних елементів необхідно вилучити зі схем джерела енергії.

11.2.4 Ключові елементи

1) **Breaker** – перемикач, що здійснює комутацію між контактами 1 і 2, який може керуватися як у функції зовнішнього сигналу, що подається на керований вхід "c" (*External control mode*), так і за допомогою внутрішнього інтервального таймера (*Internal control mode*) (див. рис. 11.9).

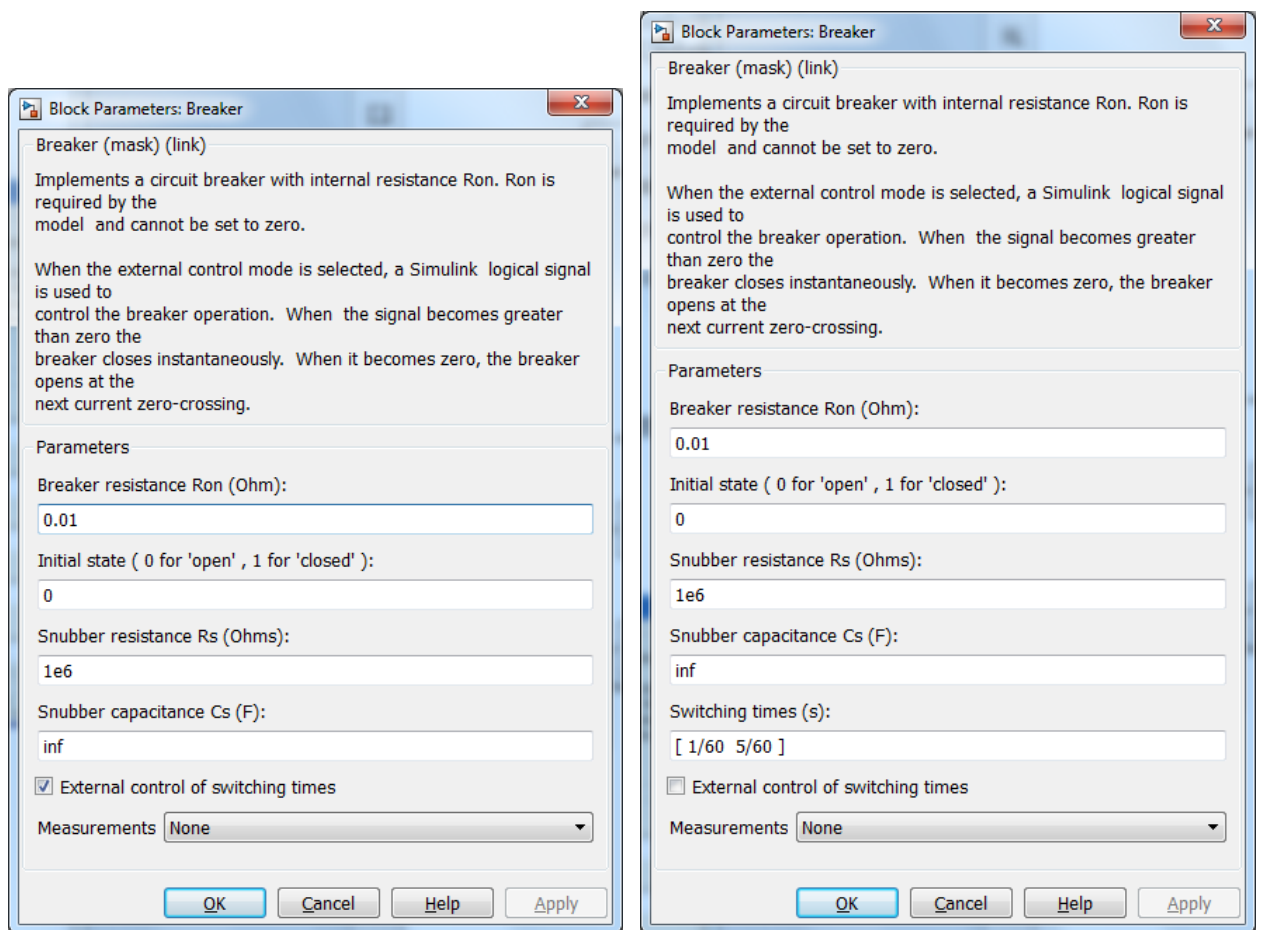


Рис. 11.9. Вікна параметрів блоку *Breaker*

Початковий стан ключа задається параметром *Initial State* (0 for 'open', 1 for 'close'). **Параметр Breaker resistance R_{on} (Ohm)** визначає внутрішній опір ключа у замкненому стані, який **не можна встановлювати нульовим**. У розімкненому стані ключ має опір $R_{off} = \infty$.

У модель «брейкера» включено послідовну R_s - C_s -гілку (*Snubber*), що здійснює функцію дугогасіння при розмиканні ключа, з параметрами *Snubber resistance R_s (Ohm)* та *Snubber capacitance C_s (F)*. **За замовчанням** параметр C_s має значення $inf(\infty)$, тобто «снабер» є чисто резистивним. **Встановлення $C_s = 0$ рівнозначно розмиканню «снабера»**. При цьому змінюється «іконка» блоку (див. рис. 11.10). «Снабер» необхідно застосовувати, якщо послідовно до «брейкера» приєднані котушка індуктивності або джерело струму.

У режимі *External control mode* на іконці ключа з'являється S -вхід "с", що дозволяє здійснити керування станом ключа зовнішнім логічним сигналом: 1 подає команду на замикання ключа, а 0 – на розмикання. При відключенні режиму *External control mode* вхідний порт зникає, а у вікні параметрів з'являється поле *Switching times (s)*, в якому треба задати вектор моментів часу, в які стан ключа змінюється на протилежний.

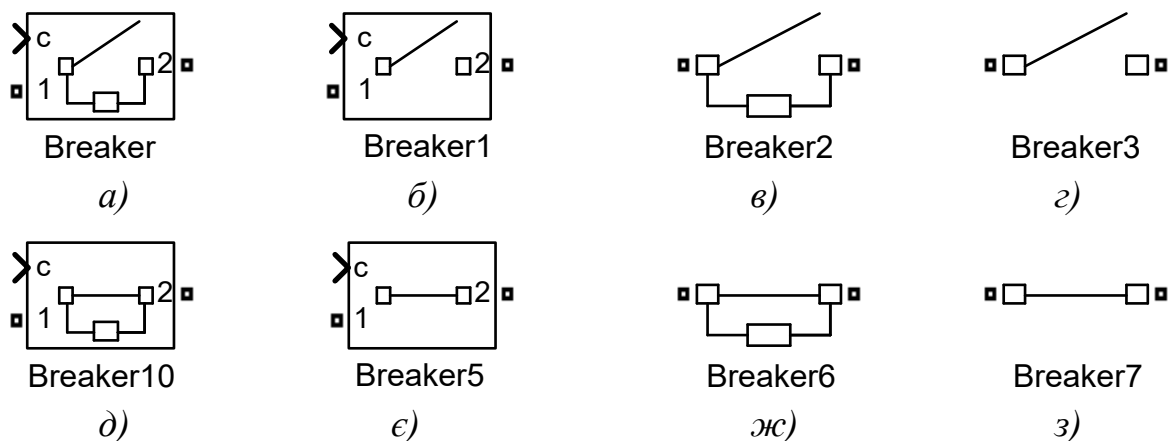


Рис. 11.10. Піктограми блоку *Breaker*:

Initial State = 0 – а, б, в, г; *Initial State* = 1 – д, е, ж, з;

External control mode – а, б, д, е; *Internal control mode* – в, г, ж, з;

$C_s = inf$ – а, в, д, ж; $C_s = 0$ – б, г, е, з

Особливістю блоку *Breaker* є те, що він *розмикає електричне коло після подачі сигналу на розмикання тільки тоді, коли струм, що протікає через ключ, стає нульовим*. Отже, *Breaker* може замкнути, але не може розімкнути коло постійного струму. Його призначення – комутація у колах змінного струму.

При наявності у віртуальній моделі блоку *Breaker* краще за все використовувати для розрахунку перехідних процесів метод *ode23tb*.

2) *Ideal Switch* – ідеальний ключ, керований *Simulink*-сигналом, що подається на вхідний *S*-порт *gate (g)*: ключ замикається при додатному керуючому сигналі, що перевищує 1, а розмикається при нульовому. Вихідний вимірювальний *S*-порт *m* у включеному стані виводить струм та напругу ключа.

Вікно параметрів цього блоку показано на рис. 11.11.

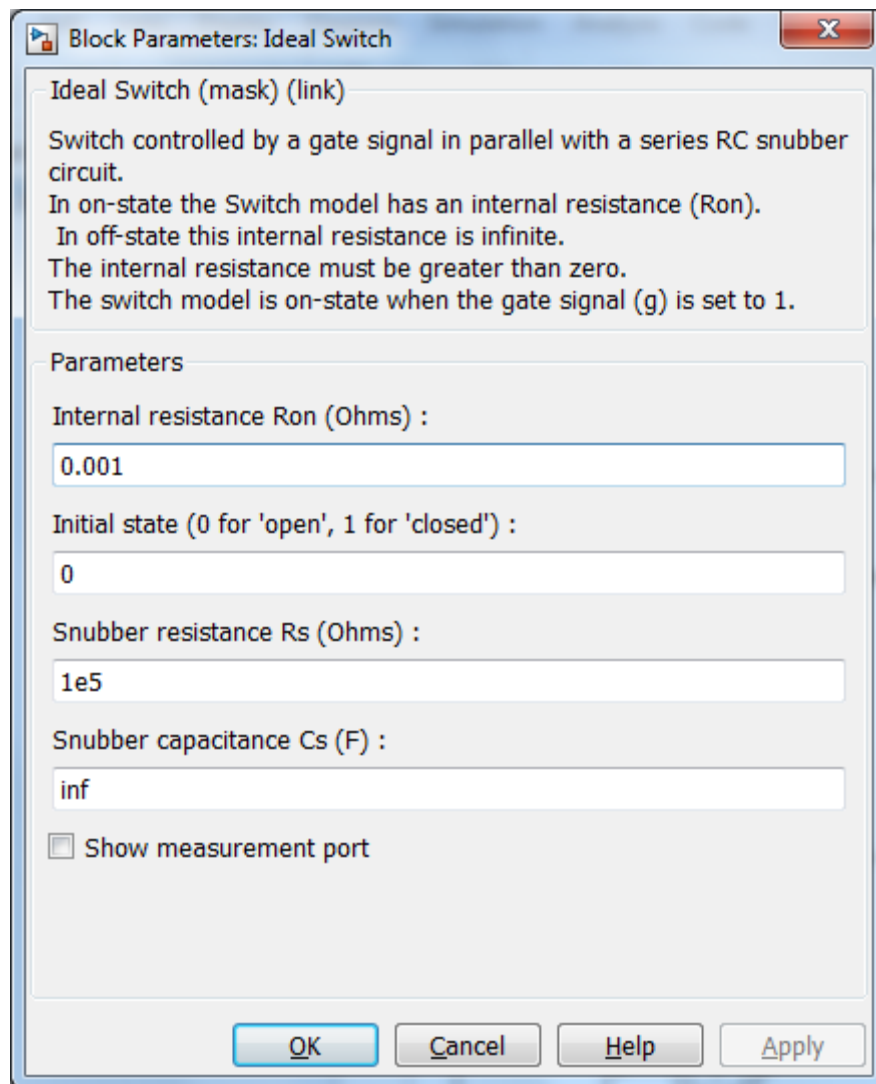


Рис. 11.11. Вікно параметрів блоку *Ideal Switch*

Як бачимо, він має багато спільного із блоком *Breaker*. Це і початковий стан ключа, який задається параметром *Initial State* (0 for 'open', 1 for 'close'), і наявність резистивного «снабера», який можна вимкнути, встановленням $Cs = 0$, що відображається на вигляді піктограми блок (див. рис. 1.9), і наявність внутрішнього опору ключа *Internal resistance Ron (Ohm)*.

Але, на відміну від блоку *Breaker*, блок *Ideal Switch* розмикає електричне коло одразу після того, як на порт приходить нульовий сигнал. Тому **цей блок може повноцінно застосовуватися для комутації у колах постійного струму.**

Менш суттєвими відмінностями є

- відсутність внутрішнього таймера для керування стану ключа
- відсутність внутрішнього таймера для керування стану ключа
- наявність вимірювального порту, який можна робити видимим і невидимим за допомогою параметра *Show measurement port*;
- на порядок менші опори *Ron* (0.001 Ом за замовченням замість 0.01 Ом у «брейкера») і *Rs* (10 кОм у порівнянні зі 100 кОм у «брейкера»);
- не змінює піктограму при замкненому початковому стані (див. рис. 11.9).

11.2.5 Графічний інтерфейс користувача *Powergui*

Powergui (Powerlib Graphical User Interface) призначений для аналізу віртуальних фізичних моделей електричних кіл і систем.

Блок обов'язково треба розміщати у вікні моделі, у якій присутній хоча б один з блоків бібліотек *SimPowerSystem* (у версіях MATLAB 7 і вище поміщається у вікно моделі автоматично). Блок повинен бути тільки один, і його не можна перейменувати. Вікно завдання параметрів цього блоку подано на рис. 11.12.

Він дозволяє вирішувати такі задачі:

1) ***Simulation type*** – вибір методу розрахунку динамічних та статичних режимі, зазвичай у вигляді вікна вибору однієї з опцій: *Continuous* (неперервні), *Discretize electrical model* (дискретні, треба вказати період дискретності

Sample Time), *Phasor simulation* (метод узагальнених векторів, треба вказати основну частоту).

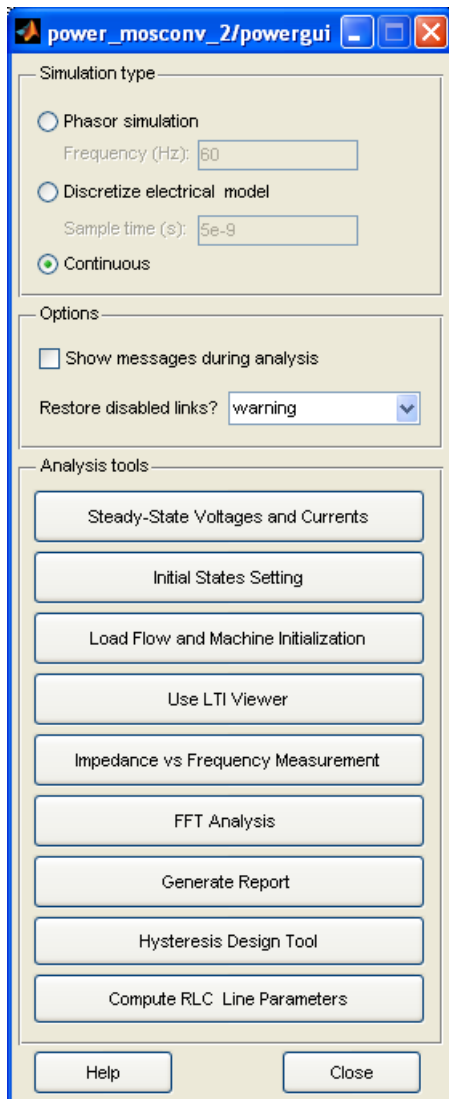


Рис. 11.12. Вікно параметрів блоку *Powergui*

2) *Steady-State Voltages and Currents* – відображення усталених значень перемінних (вікно *steady state values*).

Вікно визначення параметрів цієї операції зображено на рис. 11.13. До них належать:

- *Units* (одиниці) – *peak values* (амплітудні значення) / *RMS values* (діючі значення);
- *Frequency* (частота в Гц);
- *Display* – прапорцями слід задати одну/один або декілька змінних стану (*States*), блоків вимірювання (*Measurements*) та джерел (*Sources*);
- *Format* (формат виведення чисел) – з фіксованою точкою або плаваючою;
- *Ordering* – порядок виводу (*Value then name*), *Name then value*

Оновлення інформації досягається натисканням на кнопку *Update steady state values*.

3) *Initial States setting* – редагування або установка початкового стану моделі (завдання початкових умов змінним стану), дозволяє виконати ініціалізацію моделі, визначаючи початок моделювання з усталеного режиму або з нульових початкових умов. Вікно визначення початкових умов подано на рис. 11.14.

4) *Load Flow and Machine Initialization* – ініціалізація трифазних кіл (*Three-Phase Dynamic Load*) та електричних машин (*Machines*). Вікно для виконання цієї операції зображено на рис. 11.15. Більш детально ознайомитися з можливостями цієї операції можна у [Черних].

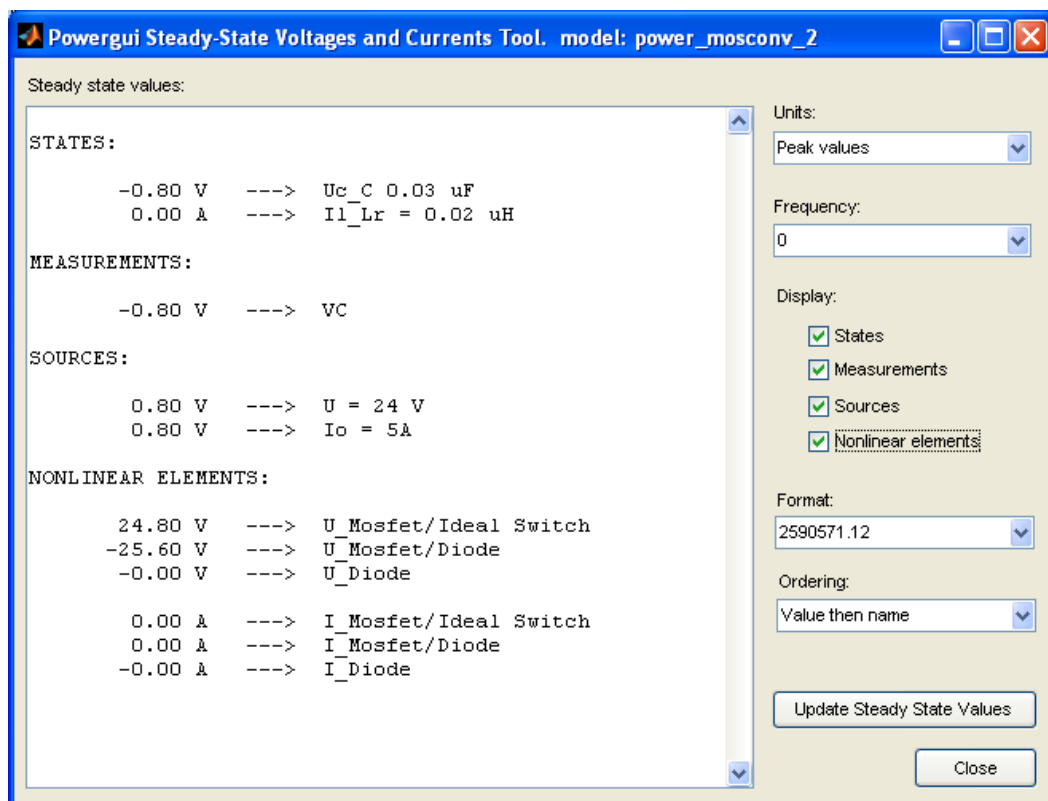
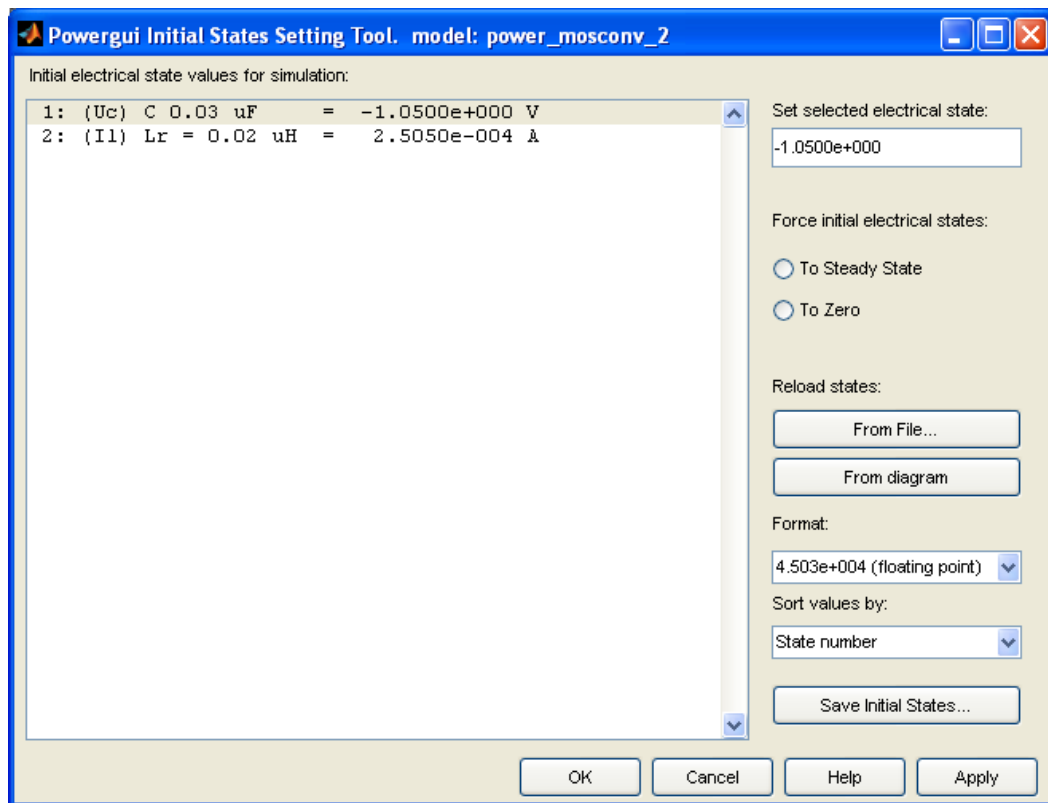


Рис. 11.13. Вікно відображення усталених значень

Інші функції інтерфейсу користувача будуть розглянуті пізніше у інших дисциплінах або при необхідності самостійно.

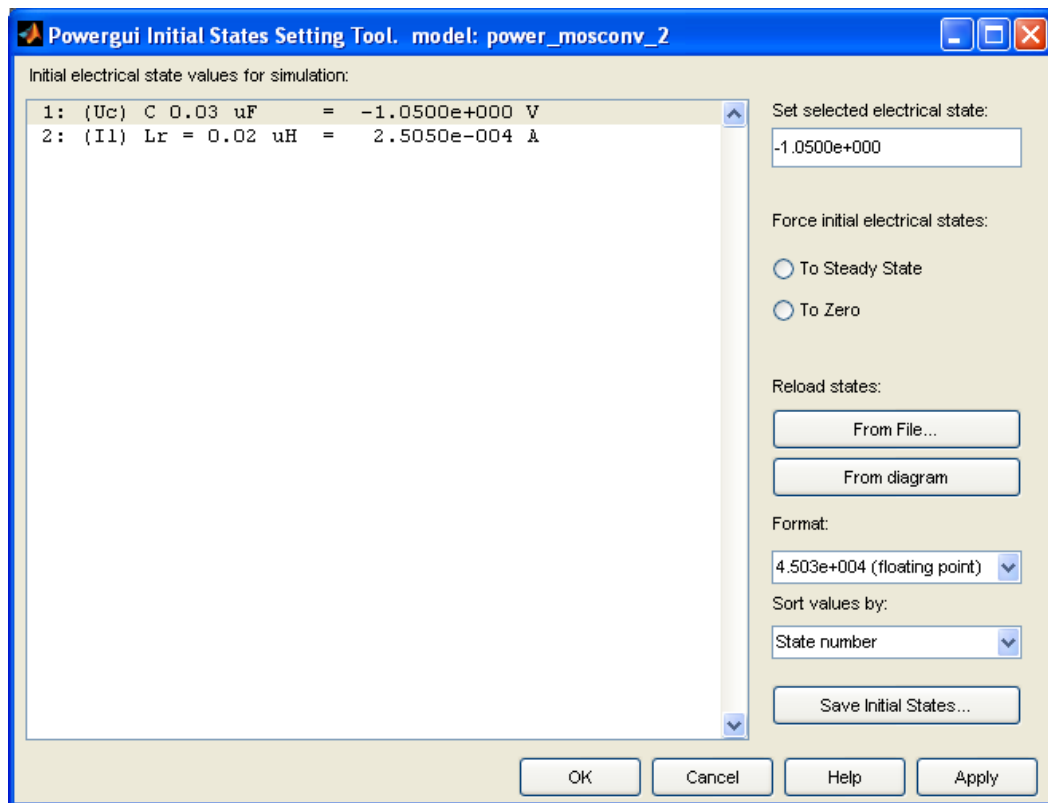


Рис. 11.14. Вікно відображення усталених значень

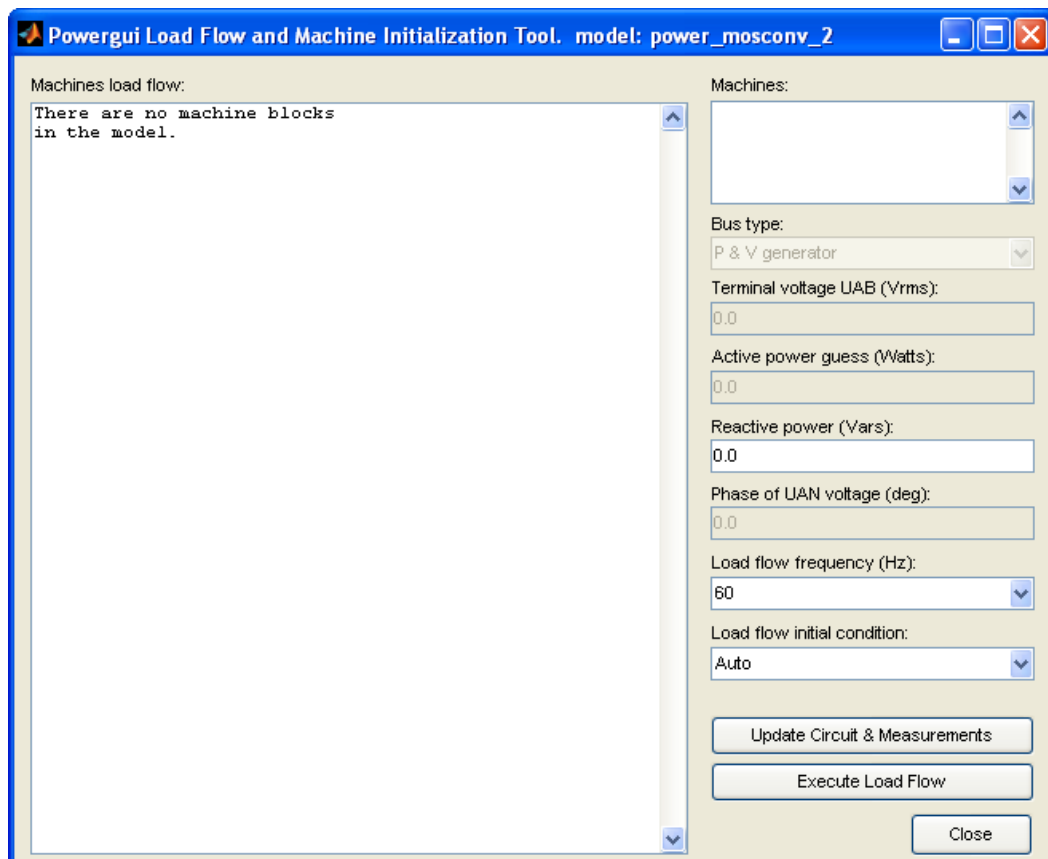


Рис. 11.15. Вікно ініціалізації динамічних трифазних електричних кіл та електричних машин

11.3 Приклад віртуального фізичного моделювання лінійного розгалуженого електричного кола

Перевагою віртуального фізичного моделювання є те, що для його застосування не потрібно складати математичний опис досліджуваної системи. Ця функція покладається на програмне забезпечення. Задачею користувача в даному разі є складання принципової електричної схеми у блоках віртуальної бібліотеки *SimPowerSystem*, що потребує деякого досвіду. Для прикладу на рис. 11.16 подана *SPS*-модель електричного кола, зображеного на рис. 6.1.

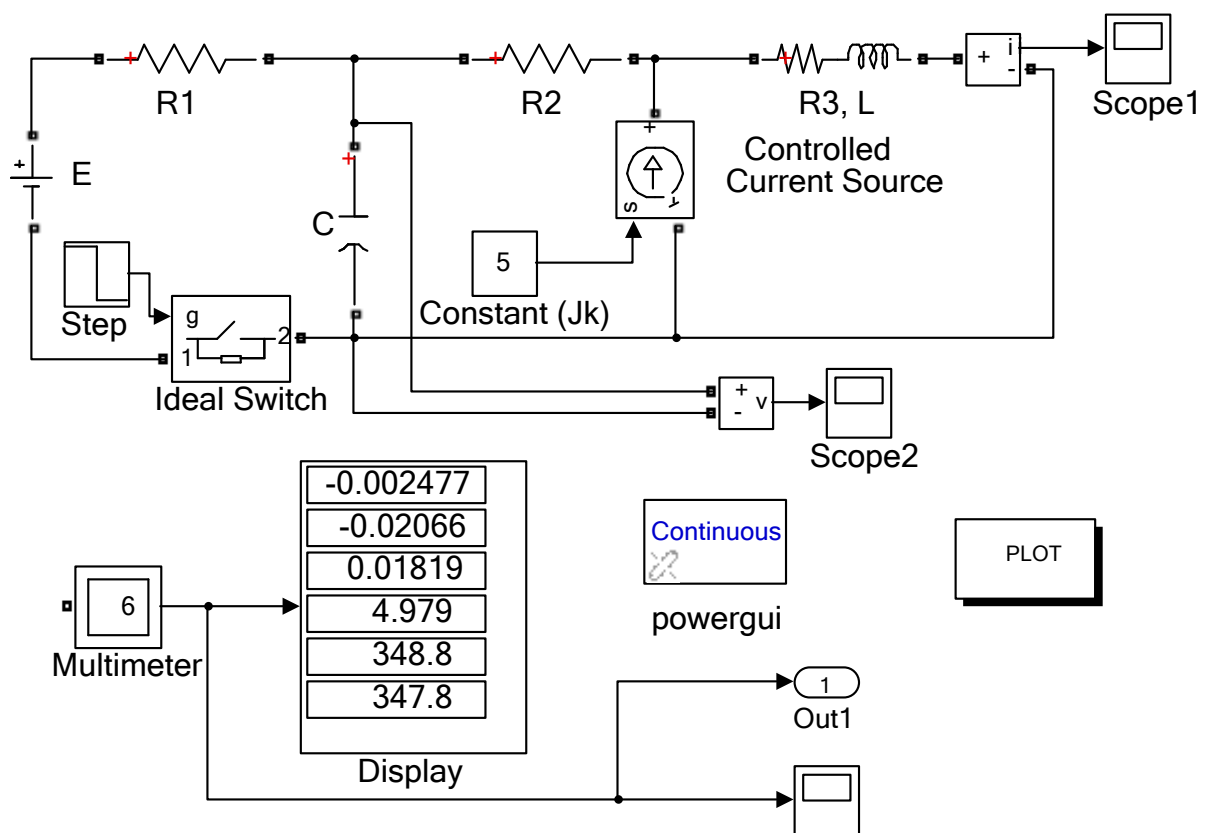


Рис. 11.16. *SPS*-модель лінійного електричного кола, зображеного на рис. 6.1

Simulink-блоки у моделі рис. 11.16 використані тільки для фіксації результатів, формування вихідного сигналу джерела струму та керування ключовим елементом *Ideal Switch*. На рисунку продемонстровано можливість вимірювання сигналів різними інструментами.

Графіки перехідних процесів, отриманих за результатами симуляції досліджуваної моделі показані на рис. 11.17.

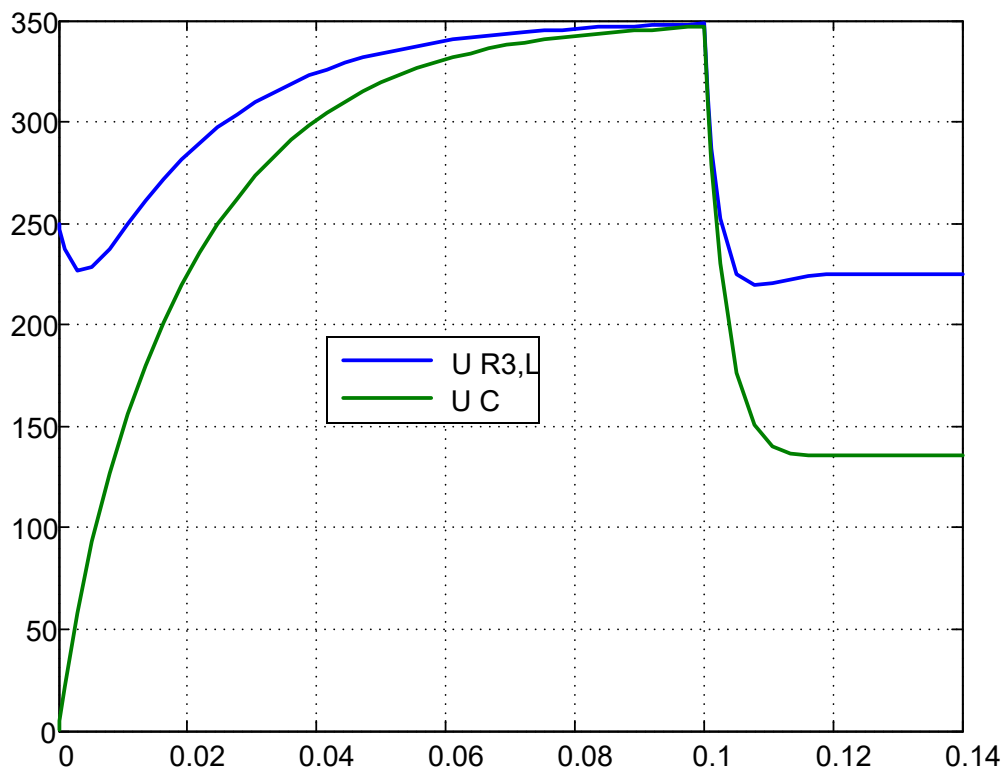
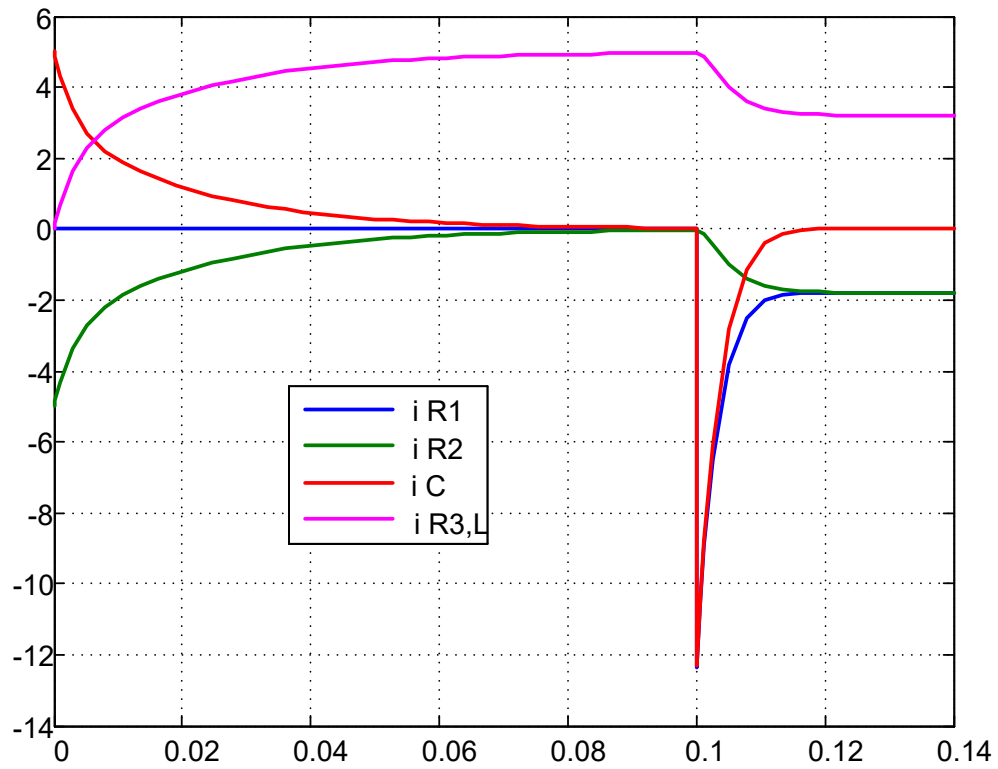


Рис. 11.17. Результати симуляції моделі рис. 11.16

Графіки на рис. 11.16 відрізняються від графіків на рис. 6.5 тим, що перші з них отримані при розмиканні ключа та його замиканні, а другі – тільки при замиканні.

11.2 Завдання

Виконати завдання 1 з лабораторної роботи №6 методом віртуального фізичного моделювання.

11.3 Порядок виконання роботи

- 1) Ознайомитися з блоками бібліотек *SimPowerSystem*.
- 2) Зібрати SPS-модель досліджуваного електричного кола, виставити її параметри.
- 3) Скласти програму для її ініціалізації.
- 4) Виконати необхідні віртуальні експерименти, побудувати графіки перехідних процесів та проаналізувати їх.
- 5) Порівняти результати моделювання з результатами, отриманими в лабораторній роботі №6.

11.4 Методичні вказівки

- 1) Параметри моделей задавайте іменами змінних, а не константами.
- 2) Для попередньої візуалізації використовуйте блоки *Scope*, а для запам'ятовування чисельних результатів з метою подальшої побудови графіків перехідних процесів – вихідні порти *Out* та блоки *ToWorkspace*. Усі струми виводьте на один графік, а напруги – на інший.
- 3) Ініціалізацію моделей та побудову графіків виконайте за допомогою «кнопок», створених з блоків *Subsystem* (див. лабораторну роботу 5).
- 4) Час розрахунку перехідного процесу обирайте таким, щоб усталений режим був досягнутий, але не тривав дуже довго.
- 5) Апробуйте різні способи вимірювання сигналів.

11.5 Контрольні питання та завдання

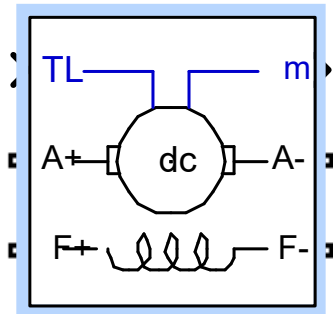
- 1) Чим відрізняються блоки бібліотек *SimPowerSystem* від *Simulink*-блоків?
- 2) Які порти мають SPS-блоки?
- 3) Як поєднати між собою SPS-блоки і блоки *Simulink*?

- 4) Для чого застосовують блок *Powergui*?
- 5) Як працюватиме *SPS*-модель електричного кола без ключових елементів, якщо їй не задати нульові початкові умови?
- 6) Порівняйте між собою переваги та недоліки віртуального фізичного та структурного математичного моделювання.

Лабораторна робота №12

ВІРТУАЛЬНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ ДВИГУНА ПОСТІЙНОГО СТРУМУ З НЕЗАЛЕЖНИМ ЗБУДЖЕННЯМ

12.1 SPS- модель двигуна постійного струму



DC Machine

Рис. 12.1. SPS-блок

DC Machine

На рис. 12.1 подано зображення блок *DC Machine* – машина постійного струму, який знаходиться в розділі *Machines* бібліотеки *SimPowerSystems*. Як бачимо, воно складається з якоря та обмотки збудження. Отже, на основі цього блоку можна створити машину постійного струму з послідовним, паралельним та незалежним збудженням.

Блок має «електричні» порти *A+*, *A-* та *F+*, *F-*, що відповідають затискачам якоря та обмотки збудження відповідно. Крім «електричних» портів, блок *DC Machine* має один механічний вхід та векторний «інформаційний» вихід *m* (від *measurements*), який передбачає подальшу обробку або фіксацію сигналів засобами базових блоків *Simulink*.

На рис. 12.2 та 12.3 зображені вкладки вікна введення параметрів цього блоку.

Вкладка *Parameters* (рис. 12.2) діалогового вікна установки параметрів блока дозволяє встановити значення таких параметрів ДПС:

- *Armature resistance* R_a (ohms) – активний опір якоря R_a (Ом);
- *Armature inductance* L_a (H) – індуктивність якоря L_a (Гн);
- *Field resistance* R_f (ohms) – активний опір обмотки збудження R_f (Ом);
- *Field inductance* L_f (H) – індуктивність обмотки збудження L_f (Гн);
- *Field-Armature mutual inductance* L_{af} (H) – взаємна індуктивність, L_{af} (Гн);
- *Inertia* J (kg.m²) – момент інерції двигуна, J (кг · м²);

- *Viscous friction coefficient* B_m (N.m.s) – коефіцієнт в'язкого тертя $k_{f\omega}$ (Нм/(рад/с));
- *Coulomb friction torque* T_f (N.m) – момент сухого тертя M_T (Нм);
- *Initial speed* (rad/s) – початкова кутова швидкість.

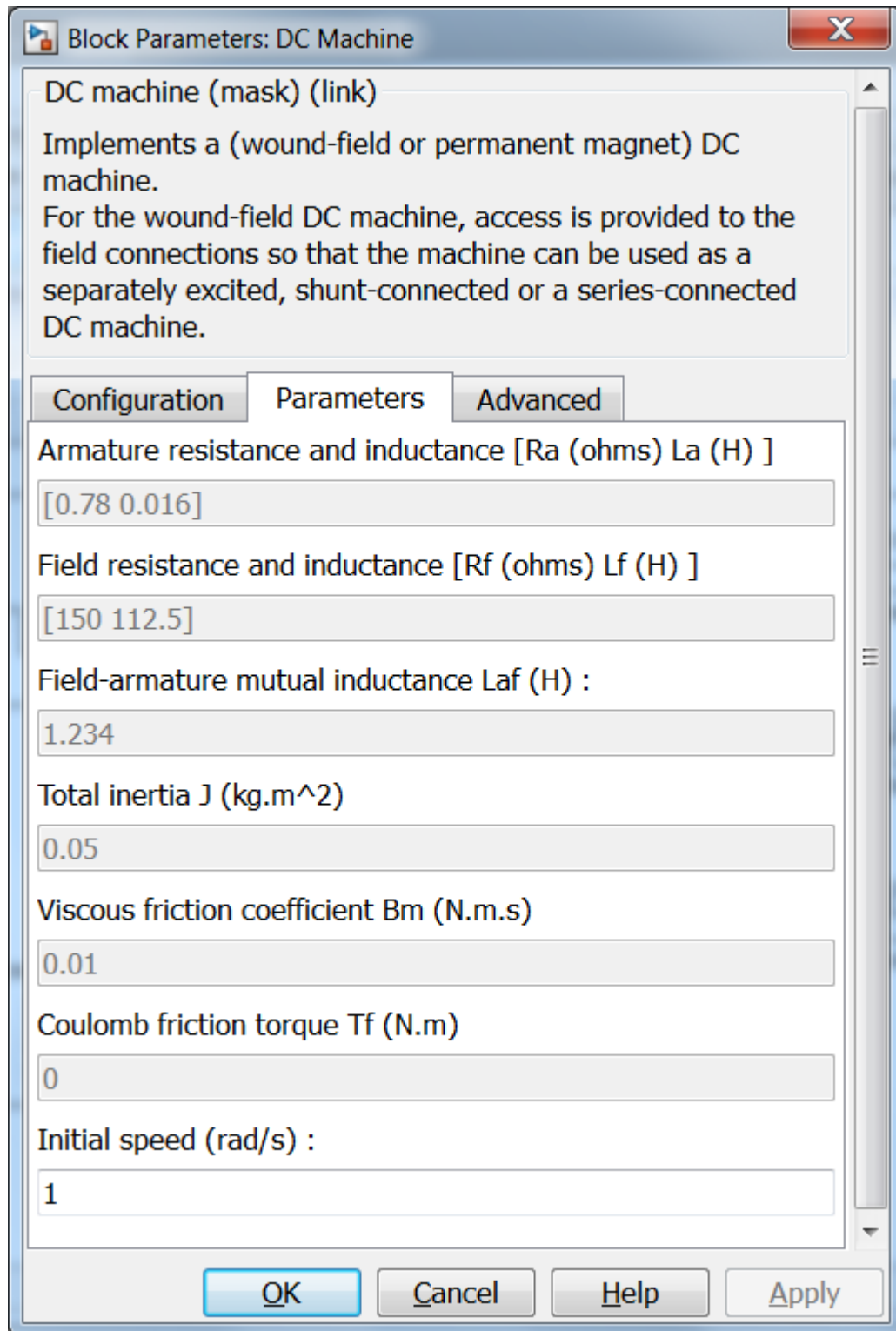


Рис. 12.2 – Вкладка *Parameters* вікна встановлення параметрів блоку *DC Machine*

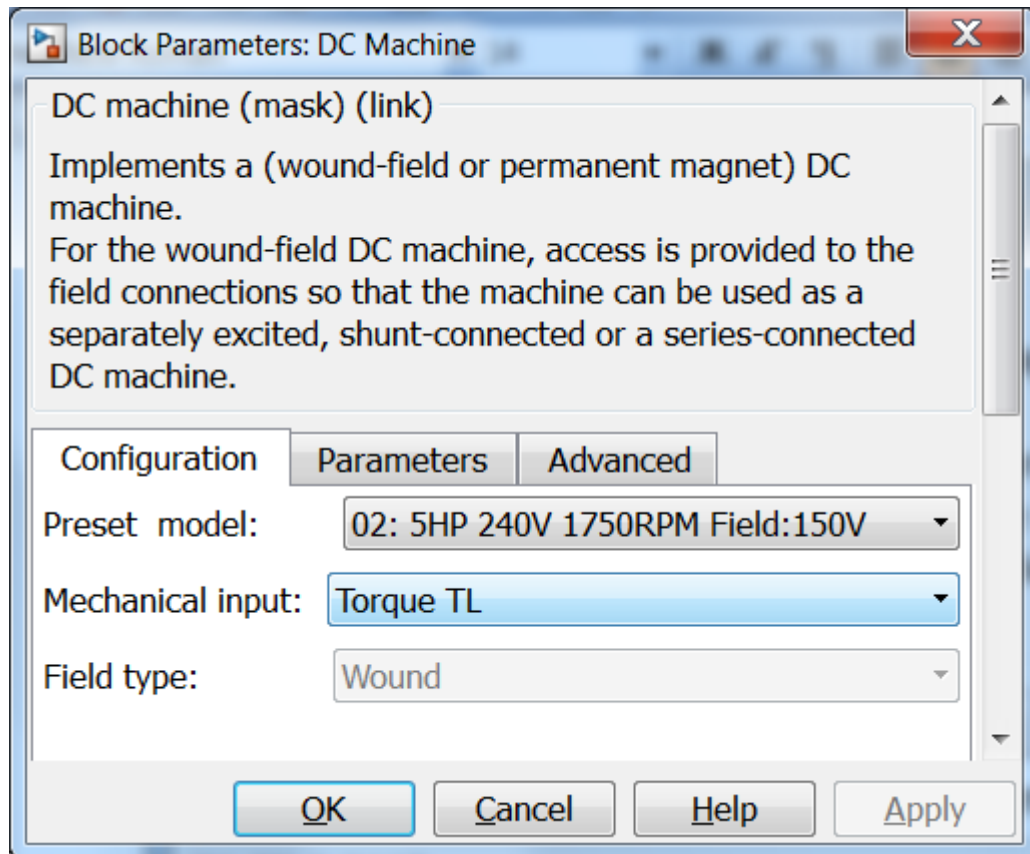


Рис. 12.3 – Вкладка *Configuration* вікна встановлення параметрів блоку *DC Machine*

Параметри ДПС та можливість їх встановлення або коригування залежать від стану опції *Preset Model* (попередня ініціалізація моделі даними деякого двигуна) вкладки *Configuration* (рис. 12.3).

У початковому стані ця функція має значення *No*, параметри вкладки *Parameters* мають певні значення, які після переміщення блоку у вікно моделі можна змінювати. Це зручно робити, якщо Ви обрали зі стороннього джерела якийсь двигун та, крім тих даних, що вводяться у вікно вкладки *Parameters* знаєте потужність двигуна та його номінальні дані, які необхідні для моделювання.

Якщо у Вас немає даних двигуна для дослідження, то Ви можете його обрати з випадаючого меню опції *Preset Model*, в якому для кожного з двигунів наводиться його потужність у кінських силах **HP** ($1\text{HP} = 746\text{ Вт}$), напруга якоря та напруга збудження (*Field:*) у вольтах (**V**) і номінальна швидкість в обертах за хвилину (**RPM** – *Revolutions Per Minute*).

Вибір певного двигуна, підтверджений натисканням кнопки *Apply*, призводить до автоматичного встановлення відповідних параметрів у вкладці *Parameters*, які тепер не можуть бути скориговані користувачем.

Для того, щоб зробити дані попередньо обраного двигуна доступними до коригування, треба після вибору цього двигуна віртуальною кнопкою *Apply* знову встановити функцію *Preset Model* у значення *No*.

Якщо порівняти параметри структурної моделі ДПС, розглянутої в лабораторній роботі №7 з параметрами *SPS*-блоку *DC Machine*, то можна побачити, що у структурній моделі існує параметр c , відсутній в *SPS*-моделі, але не існує параметру L_{af} . Зв'язок між цими параметрами встановлюється формулою

$$c = L_{af} i_{fn} = L_{af} \frac{U_{fn}}{R_f} = L_{af} \frac{U_{3H}}{R_3}. \quad (12.1)$$

Модель для дослідження ДПС з незалежним збудженням, у якій використано розглянути вище блок *DC Machine* показано на рис. 12.4.

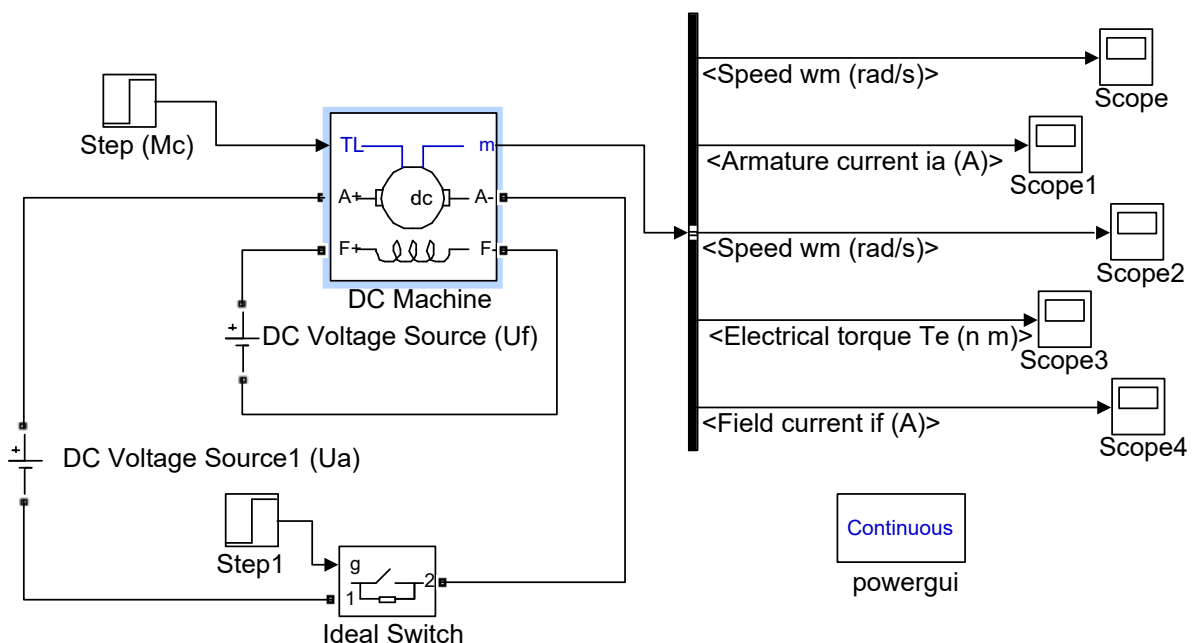


Рис. 12.4 Віртуальна фізична модель двигуна постійного струму з незалежним збудженням

12.2 Завдання

Виконати завдання 1 з лабораторної роботи №7 методом віртуального фізичного моделювання.

12.3 Порядок виконання роботи

1) Ознайомитися з параметрами *SPS*-блоку *DC Machine* бібліотек *SimPowerSystem*.

2) Зібрати *SPS*-модель досліджуваного електродвигуна постійного струму з незалежним збудженням.

3) За даними двигуна розрахувати необхідні параметри і встановити їх.

4) Розрахувати електромагнітні сталі часу якірного і кола збудження та електромеханічну сталу часу двигуна.

5) Виконати такі віртуальні експерименти:

- збудження двигуна;
- розгін збудженого двигуна при замиканні якірного кола з нерегульованим джерелом ЕРС;
- розгін збудженого двигуна при замиканні якірного кола з регульованим джерелом ЕРС, що змінює свою напругу за лінійним законом від 0 до номінального значення;
- послідовне виконання таких операцій: збудження двигуна, розгін за лінійним законом до швидкості ідеального холостого ходу, стрибкоподібне накидання та скидання номінального навантаження, гальмування двигуна за лінійним законом;
- спробуйте послідовно виконати такі операції: збудження двигуна, розгін за лінійним законом до швидкості ідеального холостого ходу, стрибкоподібне накидання та скидання номінального навантаження, зменшення напруги збудження за гіперболічним законом у два рази, стрибкоподібне накидання та скидання номінального навантаження.

6) Порівняти результати моделювання з результатами, отриманими в лабораторній роботі №6.

12.4 Методичні вказівки

1) Переключення режимів здійснюйте блоком *Breaker* або *Ideal Switch*. Уважно слідкуйте за послідовністю розмикання та замикання ключових елементів.

2) При налаштуванні часових параметрів користуйтеся результатами виконання лабораторних робіт №6 і №7.

3) Для дослідження процесу збудження двигуна за допомогою *Initial State Setting* блоку *powergui* (див. рис. 11.2) установіть в моделі початкові умови (*Force initial electrical states*) у стан *To Zero* (див. рис. 11.4) та прослідкуйте, щоб ключовий елемент у якірному колі був розімкненим до закінчення процесу збудження.

4) Для дослідження процесу розгону збудженого двигуна (без візуалізації процесу збудження) установіть в моделі початкові умови (*Force initial electrical states*) у положення *To States* (рис. 11.4) та прослідкуйте, щоб ключовий елемент у якірному колі замикався практично одразу, але при $t = 0$ був розімкненим.

5) При багатократному замиканні та розмиканні ключових елементів застосовуйте блок *Step* з векторними параметрами, на виході якого установіть суматор з одним входом.

6) Для формування напруги якоря за лінійним законом скористайтесь послідовно з'єднаними *Simulink*-блоками *Lookup Table* і *Clock* та *SPS*-блоком *Controlled Voltage Source*.

7) Для контролю усталених значень сигналів у абсолютних одиницях використовуйте блоки , а перехідні процеси виводьте у відносних одиницях.

8) Ініціалізацію моделей та побудову графіків виконайте за допомогою «кнопок», створених з блоків *Subsystem* (див. лабораторну роботу 5).

12.5 Контрольні питання та завдання

1) Які параметри ДПС треба знати, щоб скористатися блоком *DC Machine*?

- 2) Як визначити параметр L_{af} ?
- 3) Як скористатися блоком *Powergui* для встановлення початкових умов?
- 4) Чи передбачає блок *DC Machine* врахування ефекту насичення сталі?

Лабораторна робота №13

ВІРТУАЛЬНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ КОРОТКОЗАМКНЕНОГО АСИНХРОННОГО ДВИГУНА

13.1 SPS-блоки моделей асинхронної машини

В розділі *Machines* бібліотеки *SimPowerSystems* представлені два блоки трифазної асинхронної машини: *Asynchronous Machine SI Units* (асинхронна машина в абсолютних одиницях системи СІ) та *Asynchronous Machine pu Units* (асинхронна машина у відносних одиницях).

Блоки мають порти А, В, С та а, b, c, що відповідають «електричним» затискачам статора та ротора відповідно. Крім «електричних» портів, моделі мають один механічний вхід та векторний «інформаційний» вихід *m* (від *measurements*), який передбачає подальшу обробку або фіксацію сигналів засобами базових блоків *Simulink*.

Зовнішній вигляд блоків *Asynchronous Machine* (рис. 13.1) визначається типом ротора та типом механічного входу, які встановлюються за допомогою меню параметрів *Rotor type* і *Mechanical input* вкладки *Configuration* (рис. 13.2а).

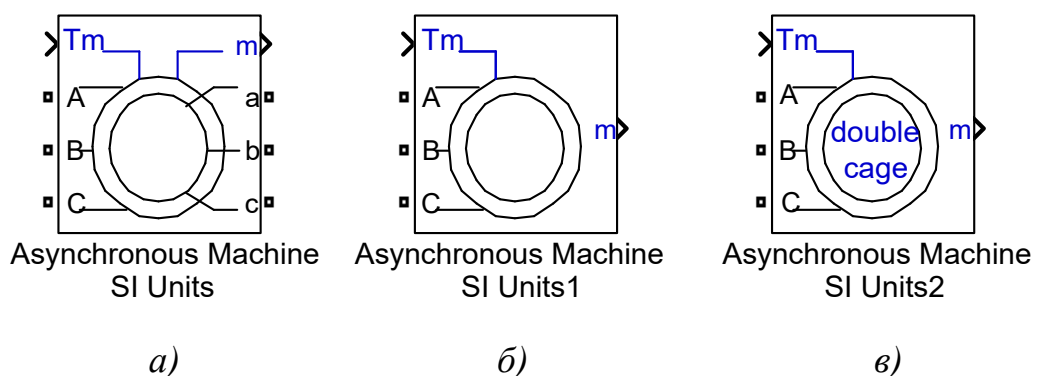
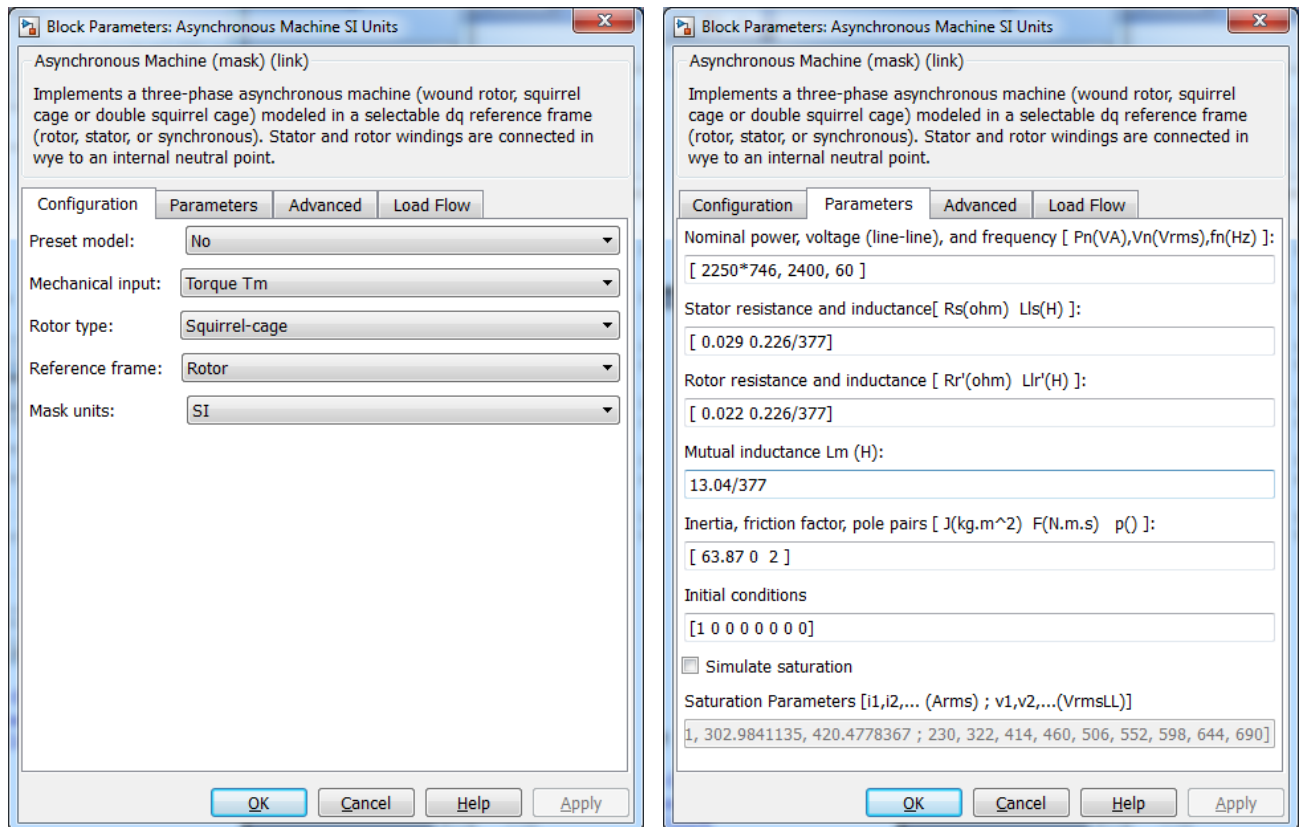


Рис. 13.1. Блоки *Asynchronous Machine* з різними типами ротора (*Rotor Type*):

а – з фазним ротором (*Wound*),

б – з білячою кліткою на роторі (*Squirrel-cage*),

в – з двома білячими клітками на роторі (*Double squirrel-cage*)



а)

б)

Рис. 13.2 – Вкладки діалогового вікна встановлення параметрів блоку *Asynchronous Machine SI Units*

Вкладка *Parameters* (рис. 13.2б) діалогового вікна установки параметрів блоку дозволяє встановити значення таких параметрів асинхронного двигуна(АД):

- *Nominal power* P_n (VA) – номінальна потужність P_n (ВА);
- *Voltage (line-line)* V_n (Vrms) – діюче, тобто ефективне, або середньоквадратичне (*rms – root mean square*) значення номінальної лінійної напруги статора U_{len} (В);
- *Frequency* f_n (Hz) – номінальна частота напруги живлення f_n (Гц);
- *Stator resistance* R_s (Ohm) – активний опір фази статора, R_s (Ом);
- *Stator inductance* L_{ls} – індуктивність розсіяння фази статора, $L_{s\sigma}$ (Гн);
- *Rotor resistance* R_r' (Ohm) – активний опір фази ротора, приведений до статора R_r (Ом);

- *Rotor inductance* $L_{lr'}$ – індуктивність розсіяння фази ротора, приведена до статора $L_{r\sigma}$ (Гн);
- *Mutual inductance* L_m (H) – головна взаємна індуктивність, L_m (Гн);
- *Inertia* J (kg.m²) – момент інерції ротора, J (кг · м²);
- *Friction factor* F (N.m.s) – коефіцієнт в'язкого тертя k_{fv} (Н · м/(рад/с));
- *Pole pairs* p () – кількість пар полюсів Z_p ;
- *Initial conditions* – початкові умови за координатами: ковзання, кутове положення ротора, амплітуди та фазові кути струмів фаз статора та (якщо треба) ротора;
- *Simulate saturation* – при наявності в полі цього параметру прапорця буде враховуватись ефект насичення магнітного кола за табличними даними кривої намагнічування $U_s = f(I_s)$, що вводяться у вигляді дворядкової матриці параметру *Saturation Parameters*; перший стовбець цієї матриці отримує координати тієї точки кривої намагнічування, з якої починає проявлятися ефект насичення, тобто він повинен бути ненульовим.

Параметри АД та можливість їх встановлення або коригування залежать від стану опції *Preset Model* (попередня ініціалізація моделі даними деякого двигуна) вкладки *Configuration* (рис. 13.2a).

У початковому стані ця функція має значення *No*, параметри вкладки *Parameters* мають певні значення, які після переміщення блоку у вікно моделі можна змінювати.

В меню функції *Preset Model* наводиться перелік, з якого можна обрати конкретний двигун за його потужністю у кінських силах HP (1HP = 746 Вт), діючим значенням лінійної напруги статора у V_{rms} , номінальною частотою в Hz та номінальною швидкістю в RPM (*Revolutions Per Minute* = об/хв). Вибір певного двигуна, підтверджений натисканням кнопки *Apply*, призводить до автоматичного встановлення відповідних параметрів у вкладці *Parameters*, які тепер не можуть бути скориговані користувачем.

Слід зазначити, що до параметрів попередньо обраних двигунів не вхо-

дять координати кривої намагнічування, тобто, ці координати задані тільки для одного двигуна, параметри якого встановлюються за замовченням (початковий стан моделі).

Для того, щоб зробити дані попередньо обраного двигуна доступними до коригування, треба після вибору цього двигуна віртуальною кнопкою *Apply* знову встановити функцію *Preset Model* у значення *No*. Підкреслимо, що можливість ініціалізації моделі через встановлення певного набору параметрів АД передбачена тільки для двигунів з однією білячою кліткою на роторі.

Параметри вкладок *Reference frame* будуть пояснені у подальших дисциплінах, а параметри вкладок *Advanced* та *Load Flow* (рис. 5.3а) зазвичай можна не змінювати.

На рис. 13.3 зображена *SPS*-модель, призначена для дослідження прямого пуску АД [19].

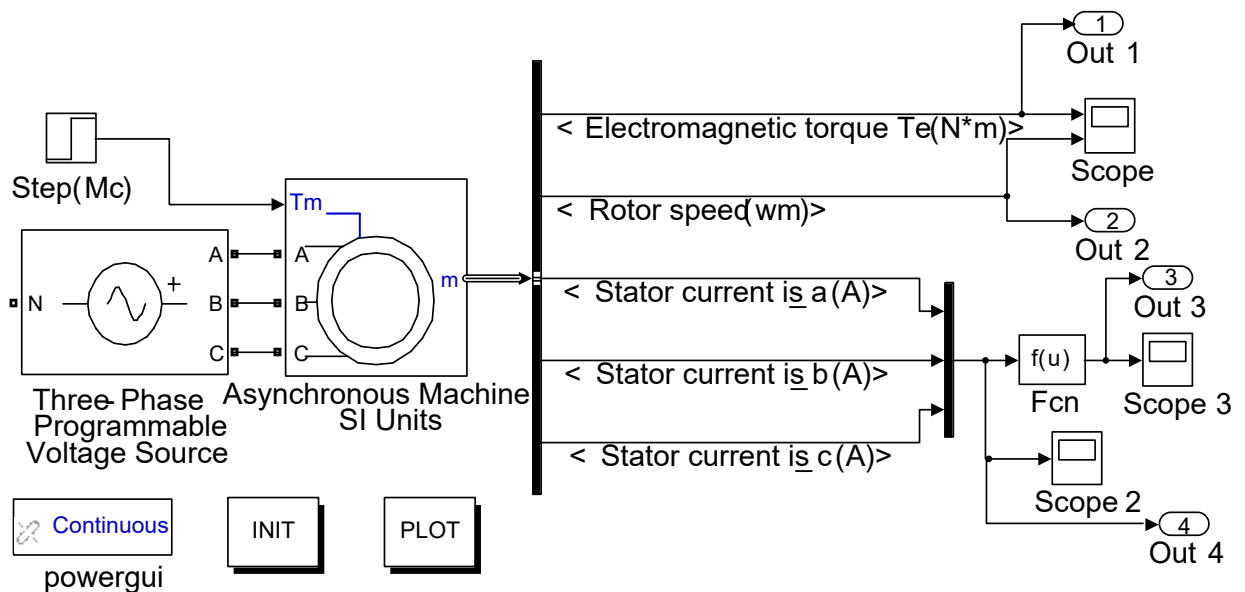


Рис. 13.3. *SPS*-модель прямого пуску асинхронного двигуна

У наведеній моделі вибір сигналів для реєстрації здійснюється з загального вектору вихідних *Simulink*-сигналів (порт *m*) *SPS*-блока *Asynchronous Machine SI Units* *S*-блоком *Bus Selector* бібліотеки *Signal Routing*. Вікно блоку *Bus Selector* показане на рис. 13.4.

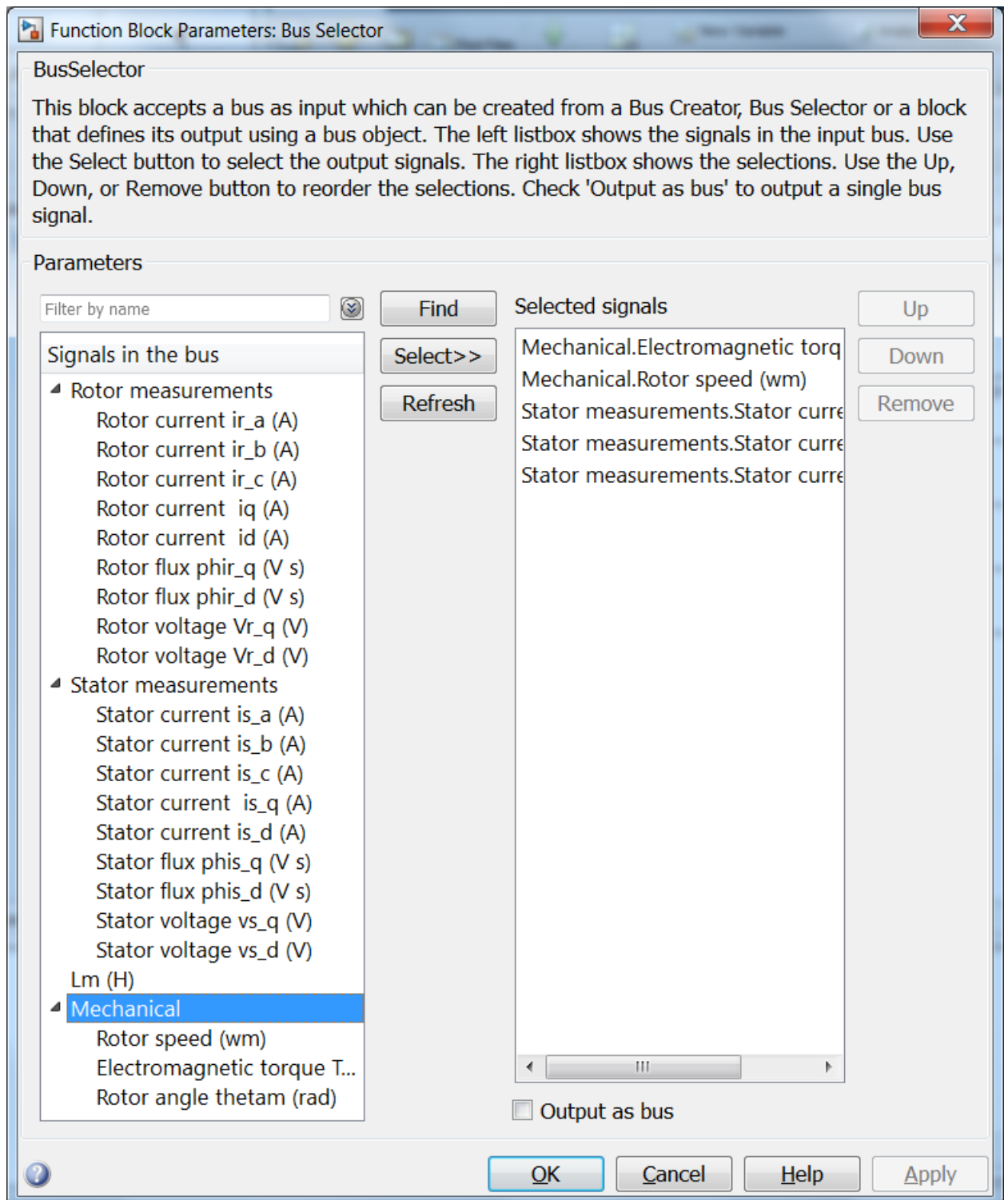


Рис. 13.4 – Діалогове вікно блоку *Bus Selector*, підключеного до вихідного інформаційного порту двигуна

Обчислення діючого (ефективного) значення струму статора виконує блок *Fcn* за формулою:

$$I_{se} = \sqrt{(I_A^2 + I_B^2 + I_C^2)/3}. \quad (13.1)$$

Фазні напруги формуються джерелом *Tree-Phase Programmable Voltage Source*, вікно параметрів якого відображено на рис. 13.5.

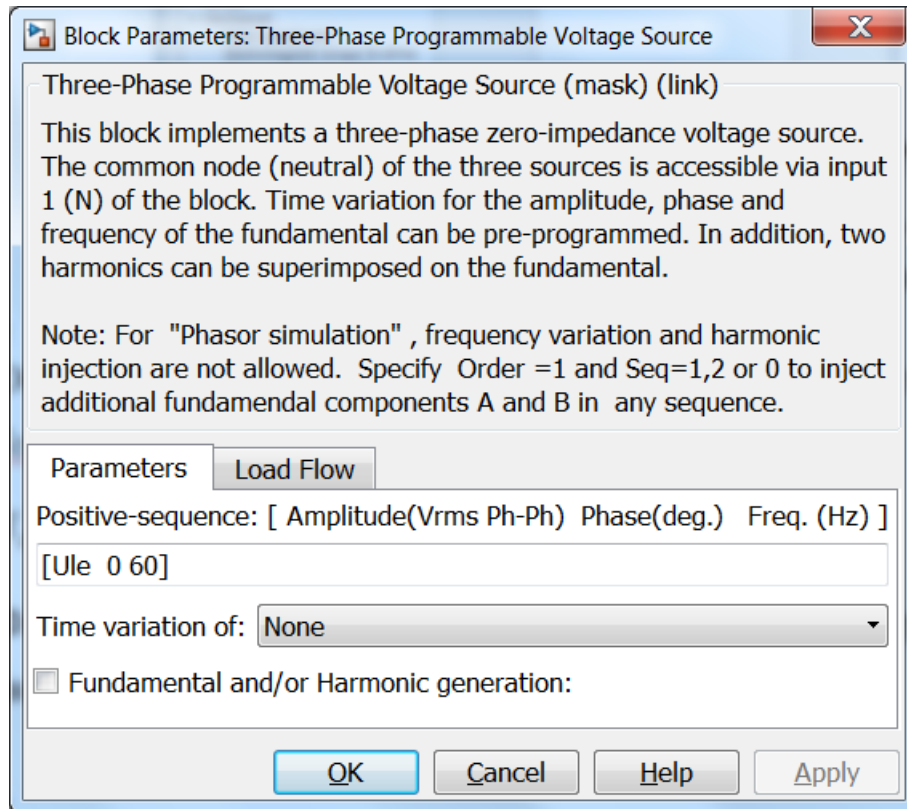


Рис. 13.5 – Діалогове вікно блоку *Tree-Phase Programmable Voltage Source*

Замість одного джерела *Tree-Phase Programmable Voltage Source*, можна використати три блока *AC Voltage Source* (див. лабораторну роботу №11, пункт 11.2.1), з'єднані зіркою.

При визначенні параметрів джерел необхідно звернути увагу на те, що в параметрах двигуна задано діюче значення лінійної напруги [Voltage (line-line) (Vrms)], а блоки *AC Voltage Source* потребують амплітудних значень фазних напруг [Peak amplitude (V)], а також, що SPS-джерела *AC Voltage Source* використовують частоту не в рад/с, а в герцах, а фазовий зсув синусоїдальних сигналів – не в радіанах, а в градусах [Phase (deg)].

Перехідні процеси, отримані за результатами симуляції моделі рис. 13.3 подані на рис. 13.6. За допомогою цієї ж моделі можна побудувати і статичні характеристики двигуна, якщо змінювати момент статичного опору за лінійним законом настільки повільно, щоб практично не призводити до збудження електромагнітного перехідного процесу. Завершення сеансу моделювання у

цьому експерименті можна здійснити блоком *Stop Simulation*, який здійснює зупинку моделювання тоді, коли швидкість АД після перекидання явно перейде в область від'ємних значень, що досягається застосуванням блоку *Relational Operator*.

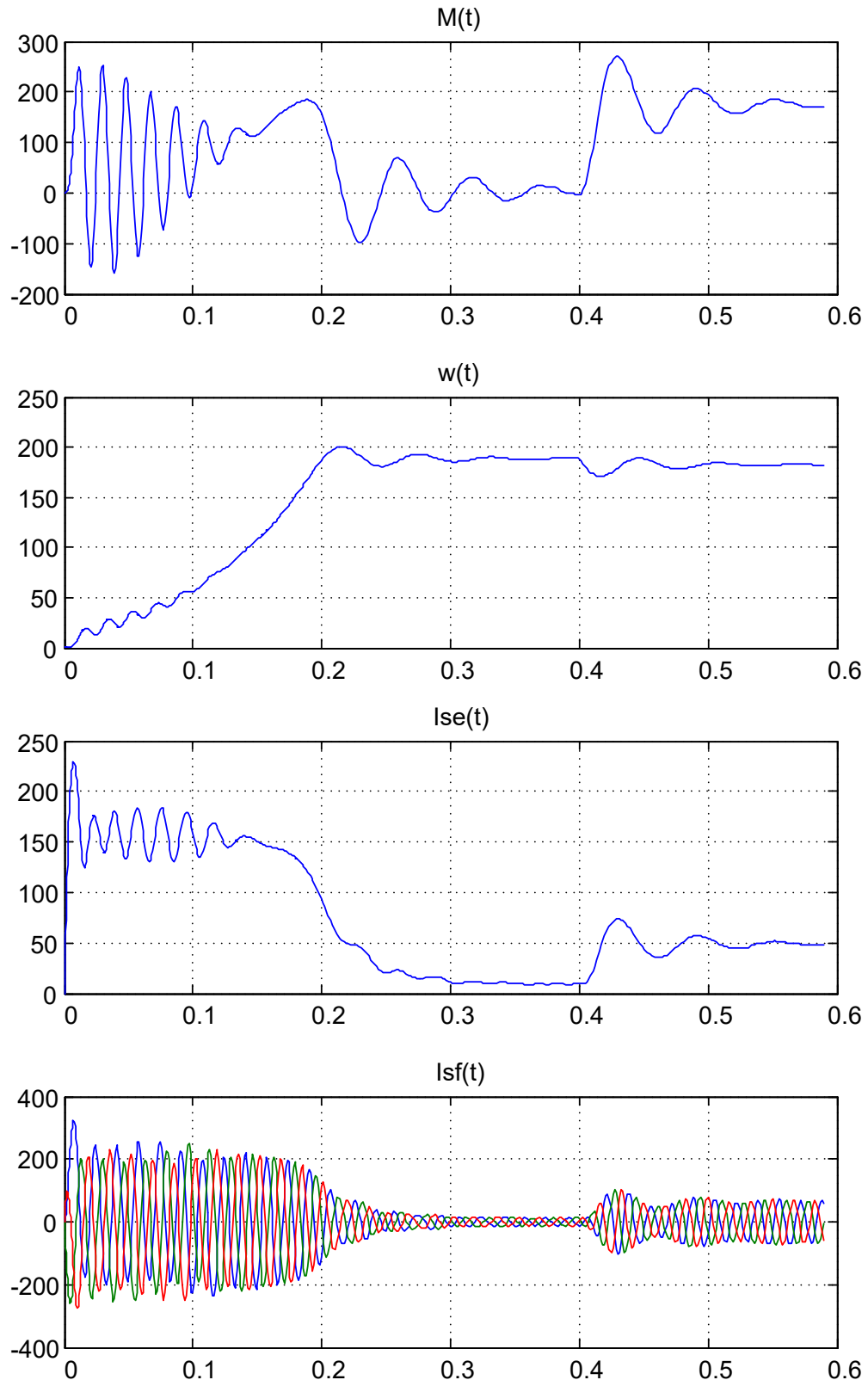


Рис. 13.6 – Перехідні процеси при прямому пуску асинхронного двигуна

13.2 Завдання

Виконати дослідження процесу прямого пуску асинхронного двигуна методом віртуального фізичного моделювання за даними табл. 13.1, наведеними для АД з короткозамкненим ротором серії МТК, 380 В, 50 Гц.

Таблиця 13.1

№ вар.	Тип	P_n кВт	n_n об/хв	X_m Гн	Статор		Ротор		
					R_1 Ом	$X_{1\sigma}$ Ом	R_2 Ом	$X_{2\sigma}$ Ом	J кг*м ²
1	МТК011-6	1,4	870	54,0	5,98	3,93	8,41	3,80	0,02
2	МТК012-6	2,2	875	36,4	3,6	2,58	5,7	2,63	0,028
3	МТК-11-6	2,2	883	48,4	3,67	2,54	5,02	2,88	0,04
4	МТК-12-6	3,5	875	35,2	2,09	1,61	3,36	1,92	0,063
5	МТК111-6	3,5	870	29,8	2,16	2,03	3,33	1,46	0,046
6	МТК-21-6	5,0	910	24,5	1,11	1,07	1,86	1,5	0,098
7	МТК112-6	5,0	890	22,9	1,32	1,39	2,34	1,02	0,065
8	МТК-22-6	7,5	905	17,6	0,69	0,74	1,33	1,07	0,138
9	МТК-31-8	7,5	682	17,3	0,79	0,90	1,30	0,79	0,25
10	МТК-31-6	11,0	920	13,5	0,42	0,47	0,82	0,71	0,25
11	МТК-41-8	11,0	685	11,0	0,43	0,53	0,84	0,66	0,445
12	МТК-42-8	16,0	685	8,8	0,27	0,36	0,60	0,47	0,65
13	МТК-51-8	22,0	692	6,9	0,18	0,30	0,38	0,39	1,05
14	МТК-52-8	28,0	695	6,0	0,14	0,23	0,30	0,30	1,38

13.3 Порядок виконання роботи

- 1) Ознайомитися з параметрами *SPS*-блоку *Asynchronous Machine SI Units* бібліотек *SimPowerSystem*.
- 2) Зібрати *SPS*-модель асинхронного двигуна для його дослідження у режимі прямого пуску та накиду навантаження.
- 3) За даними двигуна розрахувати необхідні параметри і встановити їх.

5) Виконати такі віртуальні експерименти:

- прямий пуск двигуна від джерела, що виробляє трифазну синусоїдальну напругу, параметри якої відповідають номінальній напрузі та номінальній частоті статора; після досягнення двигуном синхронної швидкості здійснити стрибкоподібний накид навантаження;
- поступово підвищуючи момент статичного опору на валу двигуна, знайдіть його критичний момент та критичне ковзання;
- виконайте попередні 2 експерименти зі зменшеною на 50% частотою джерела при номінальній амплітуді та зі зменшеною на 20% амплітудою напруги джерела при номінальній частоті;
- здійсніть пуск двигуна у протилежному напрямку.

6) Спробуйте побудувати статичну механічну та статичну швидкісні характеристики двигуна.

7) Проаналізуйте результати досліджень. Сформулюйте висновки.

13.4 Методичні вказівки

1) Виведіть на екран додатково до графіків, зображених на рис. 13.6 струми ротора та потокозчеплення ротора і статора.

2) Для зміни напрямку обертання двигуна поміняйте місцями 2 будь-які фази джерела. На ходу це можна зробити блоком *Selector*.

3) При побудові статичних характеристик відкиньте з векторів часу та вихідних величин значення, що передують зміні моменту статичного опору за допомогою логічної функції *find* та використання пустої матриці ([]).

4) Ініціалізацію моделей та побудову графіків виконайте за допомогою «кнопок», створених з блоків *Subsystem* (див. лабораторну роботу 5).

13.5 Контрольні питання та завдання

1) Які параметри АД треба знати, щоб скористатися блоком *Asynchronous Machine SI Units*?

2) Як впливає на усталену швидкість зміна частоти напруги статора?

3) Як впливає на час розгону двигуна, просадку швидкості при накиді навантаження та критичний момент зміна амплітуди напруги статора?

4) Який із струмів підтримує АД у намагніченому стані: струм статора чи струм ротора?

Лабораторна робота №14

**СТРУКТУРНЕ МАТЕМАТИЧНЕ ТА ВІРТУАЛЬНЕ ФІЗИЧНЕ
МОДЕЛЮВАННЯ ДВОМАСОВИХ МЕХАНІЧНИХ СИСТЕМ З
ВИКОРИСТАННЯМ БЛОКІВ БІБЛІОТЕКИ *SimMechanics***

**14.1. Математичний опис жорстких механічних об'єктів
з одним ступенем свободи**

Двомасові механічні системи належать до механічних об'єктів з одним ступенем свободи, що здійснюють поступальні або обертальні рухи.

Механічну інертність об'єктів поступального руху характеризує маса (m , кг), а основними параметрами його є **переміщення** (s , м), **лінійна швидкість** (v , м/с), **лінійне прискорення** (a , м/с²) та **ривок** (ρ , м/с³), що пов'язані одне з одним диференціальними рівняннями

$$v(t) = \frac{ds(t)}{dt}, \quad a(t) = \frac{dv(t)}{dt} = \frac{d^2s(t)}{dt^2}, \quad \rho(t) = \frac{da(t)}{dt} = \frac{d^2v(t)}{dt^2} = \frac{d^3s(t)}{dt^3}. \quad (14.1)$$

Механічну інертність об'єктів обертального руху характеризує **момент інерції** (J , кг·м²), а основними параметрами його є **кутове переміщення** (φ , рад), **кутова швидкість** (ω , рад/с), **кутове прискорення** (ε , рад/с²) та кутовий ривок (ρ_t , рад/с³), що пов'язані одне з одним диференціальними рівняннями

$$\omega(t) = \frac{d\varphi(t)}{dt}, \quad \varepsilon(t) = \frac{d\omega(t)}{dt} = \frac{d^2\varphi(t)}{dt^2}, \quad \rho_t(t) = \frac{d\varepsilon(t)}{dt} = \frac{d^2\omega(t)}{dt^2} = \frac{d^3\varphi(t)}{dt^3}. \quad (14.2)$$

Найбільш просто математично описуються динамічні системи із зосередженими параметрами, зокрема з зосередженими масами.

Рівняння руху одномасових механічних об'єктів з постійною механічною інерційністю визначається **другим законом Ньютона**:

$$\sum_{i=1}^n F_i(t) = ma(t) = m \frac{dv(t)}{dt} = m \frac{d^2s(t)}{dt^2}, \quad (14.3)$$

$$\sum_{i=1}^n M_i(t) = J\varepsilon(t) = J \frac{d\omega(t)}{dt} = J \frac{d^2\varphi(t)}{dt^2}, \quad (14.4)$$

де $\sum_{i=1}^n F_i(t)$ – алгебраїчна сума сил, що діють на тіло по осі його поступального руху; $\sum_{i=1}^n M_i(t)$ – алгебраїчна сума моментів, що діють на тіло у площині його обертання.

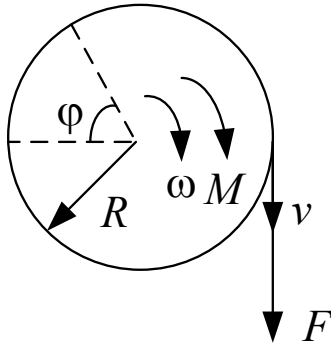


Рис. 14.1. Обертальні та поступальні параметри суцільного циліндру

Взаємозв'язок між лінійними та обертальними параметрами суцільного циліндра або диска радіусом R , що обертається навколо своєї осі (див. рис. 14.1) визначаються рівняннями

$$M = FR, \quad \omega = \frac{v}{R}, \quad \varphi = \frac{s}{R}, \quad J = \frac{mR^2}{2}. \quad (14.5)$$

При визначенні моментів інерції тіл, що обертаються, їх форму зазвичай спрощують. Наприклад, ротор (або якір) електричного двигуна уявляють тонкостінним порожнистим циліндром, а пристрої намотки – товстостінними порожнистими циліндрами, для яких відповідно

$$J = mR^2, \quad J = \frac{m(R_1^2 + R_2^2)}{2}. \quad (14.6)$$

14.2. Математичний опис пружно-в'язких механічних об'єктів з одним ступенем свободи

Механізми або їх робочі органи приводяться до руху двигунами. Рух від двигуна до механізму передається різноманітними пристроями: валами, шківками, муфтами, зубчатими колесами, черв'ячними передачами, кулачковими механізмами, тощо. Такі пристрої мають пружні властивості і характеризуються *коефіцієнтами жорсткості (коефіцієнтами пружності) c_{ij} та коефіцієнтами в'язкого (внутрішнього) тертя d_{ij}* .

Лінійна пружна деформація виникає при розтягуванні (стисненні) пружин та пружинно подібних елементів, наприклад, довгих канатів. При

обертальному русі може виникнути **пружна деформація скручування** довгих та тонких валів.

При пружній деформації у деформованому тілі з'являється пружне зусилля або пружний момент, пропорційний величині деформації, тобто різниці між переміщеннями протилежних кінцівок передачі:

$$F_{\Pi} = c_{ij} \cdot \Delta s = c_{ij} (s_i - s_j) \quad M_{\Pi} = c_{ij} \cdot \Delta \varphi = c_{ij} (\varphi_i - \varphi_j). \quad (14.7)$$

Пружній деформації перешкоджають момент або зусилля внутрішнього в'язкого тертя, зумовлені силами міжмолекулярної взаємодії деформованого матеріалу і пропорційне різниці між швидкостями протилежних кінцівок передачі:

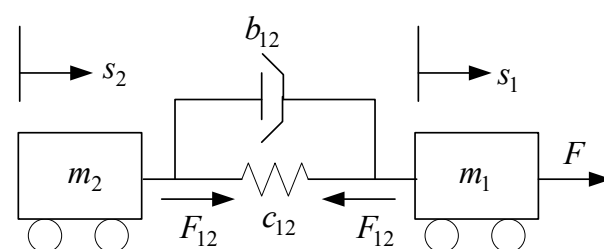
$$F_{\text{в}} = d_{ij} \cdot \Delta v = d_{ij} (v_i - v_j) \quad M_{\text{в}} = d_{ij} \cdot \Delta \omega = d_{ij} (\omega_i - \omega_j). \quad (14.8)$$

При взаємодії величин (4.7) та (4.8) утворюється пружно-в'язке зусилля або пружно-в'язкий момент

$$F_{ij} = F_{\Pi} + F_{\text{в}} \quad M_{ij} = M_{\Pi} + M_{\text{в}}. \quad (14.9)$$

Для прикладу розглянемо кінематичну схему двомасової пружно-в'язкої механічної системи поступального руху, зображену на рис. 14.2.

Її математичний опис без урахування сухого тертя має вигляд



$$\begin{cases} F - F_{12} = m_1 \frac{dv_1}{dt}, \\ F_{12} = m_2 \frac{dv_2}{dt}, \\ \frac{ds_1}{dt} = v_1, \\ \frac{ds_2}{dt} = v_2, \\ F_{\Pi} = c_{12} (s_1 - s_2), \\ F_{\text{в}} = d_{12} (v_1 - v_2), \\ F_{12} = F_{\Pi} + F_{\text{в}}. \end{cases} \quad (14.10)$$

Рис. 14.2. Кінематична схема двомасової механічної системи поступального руху

Структурна *Simulink*-модель, складена за рівняннями (14.10), зображена на рис. 14.3.

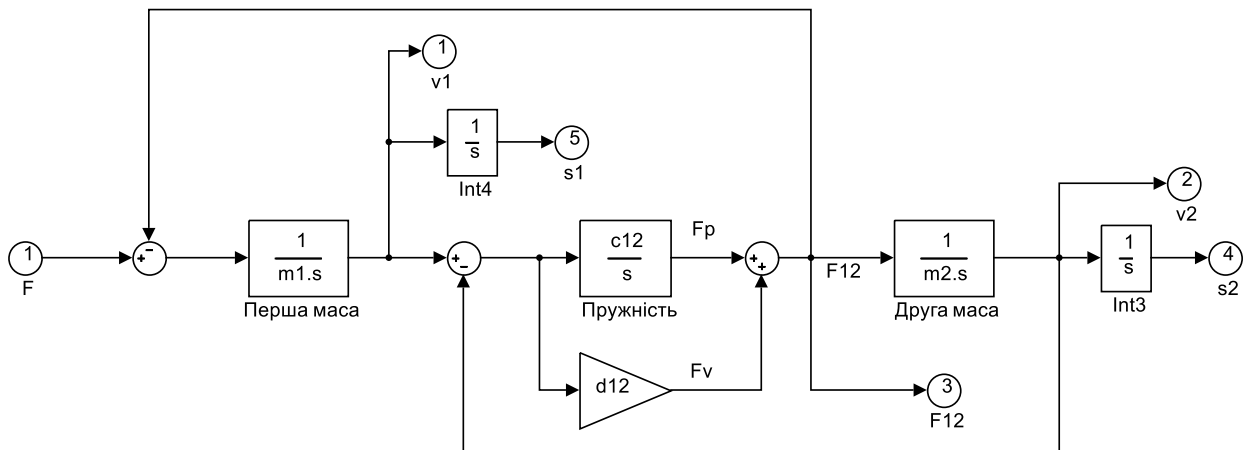


Рис. 14.3. Структурна *Simulink*-модель двомасової механічної системи поступального руху

Кінематична схема двомасової механічної системи обертального руху зображена на рис. 14.4.

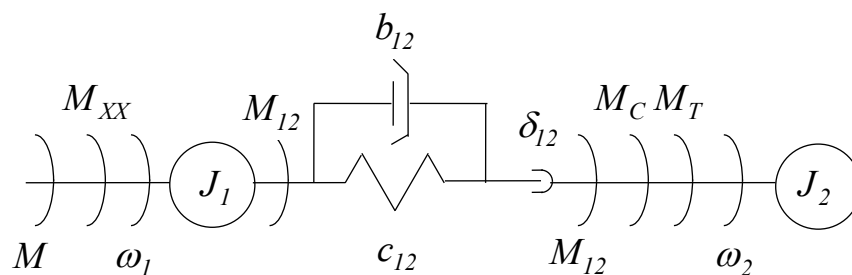


Рис. 14.4. Спрощена кінематична схема двомасової електромеханічної системи

На ній позначено:

- M_{xx} , M_{l2} , M_C , M_T – електромагнітний момент двигуна, момент холостого ходу двигуна, момент скручування валу, момент статичного опору від навантаження і момент сухого тертя в механізмі відповідно;
- ω_1 , ω_2 – кутові швидкості двигуна і механізму;
- c_{l2} , b_{l2} – коефіцієнти жорсткості і в'язкого тертя;
- δ_{l2} – люфт (зазор) у кінематичній передачі;
- J_1 , J_2 – моменти інерції двигуна і механізму.

Математичний опис такої системи без врахування зазору в кінематичній передачі та сухого тертя має вигляд [20]:

$$\left\{ \begin{array}{l} M - M_{12} = J_1 \frac{d\omega_1}{dt}, \\ M_{12} - M_c = J_2 \frac{d\omega_2}{dt}, \\ \frac{d\phi_1}{dt} = \omega_1, \\ \frac{d\phi_2}{dt} = \omega_2, \\ M_{\Pi} = c_{12}(\phi_1 - \phi_2), \\ M_B = b_{12}(\omega_1 - \omega_2), \\ M_{12} = M_{\Pi} + M_B. \end{array} \right. \quad (14.11)$$

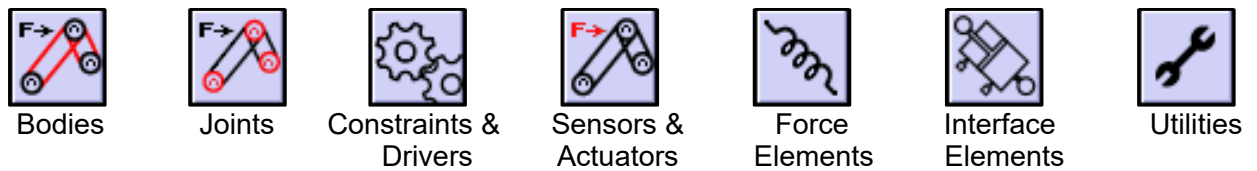
14.3 Віртуальне фізичне моделювання механічних об'єктів у середовищі *MATLAB-Simulink-SimMechanics*

Для дослідження кінематики та динаміки механізмів різного рівня складності ефективно використовувати моделювання за допомогою бібліотеки *SimMechanics* [20]. Віртуальна фізична модель подається у вигляді блочної діаграми. Блоки пакета є моделями механічних пристроїв, розташування яких у просторі та відносно одне від одного може змінюватися у відповідності до законів механіки. Моделі *SimMechanics* зображують фізичну структуру механізмів, геометричні та кінематичні відношення їх компонентів. *SimMechanics* автоматично перетворює ці блочні зображення в еквівалентну математичну модель.

SimMechanics оперує не з сигналами, а з механічними зусиллями. В силу третього закону Ньютона, зв'язки між входами і виходами блоків не можна розглядати як односпрямовані. Ці зв'язки слугують для передачі силових дій, якими обмінюються частини механізмів або механізми між собою. Тому вони не позначаються стрілками.

SimMechanics підтримує засоби анімації для демонстрації роботи механізмів у динаміці. Анімація застосовує інструменти *Microsoft Audio Video Interleave® (AVI)*, мають розширення *.avi, підтримує засоби *OpenGL* та може використовувати віртуальні вимірювальні засоби.

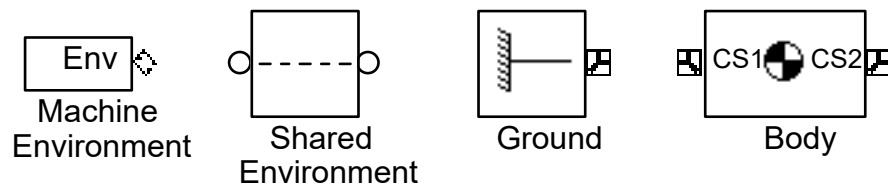
Розділи бібліотеки показані на рис. 14.5.



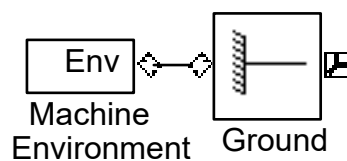
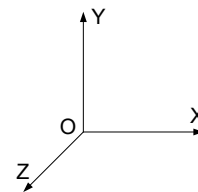
SimMechanics First Generation

Рис. 14.5. Бібліотека *SimMechanics*

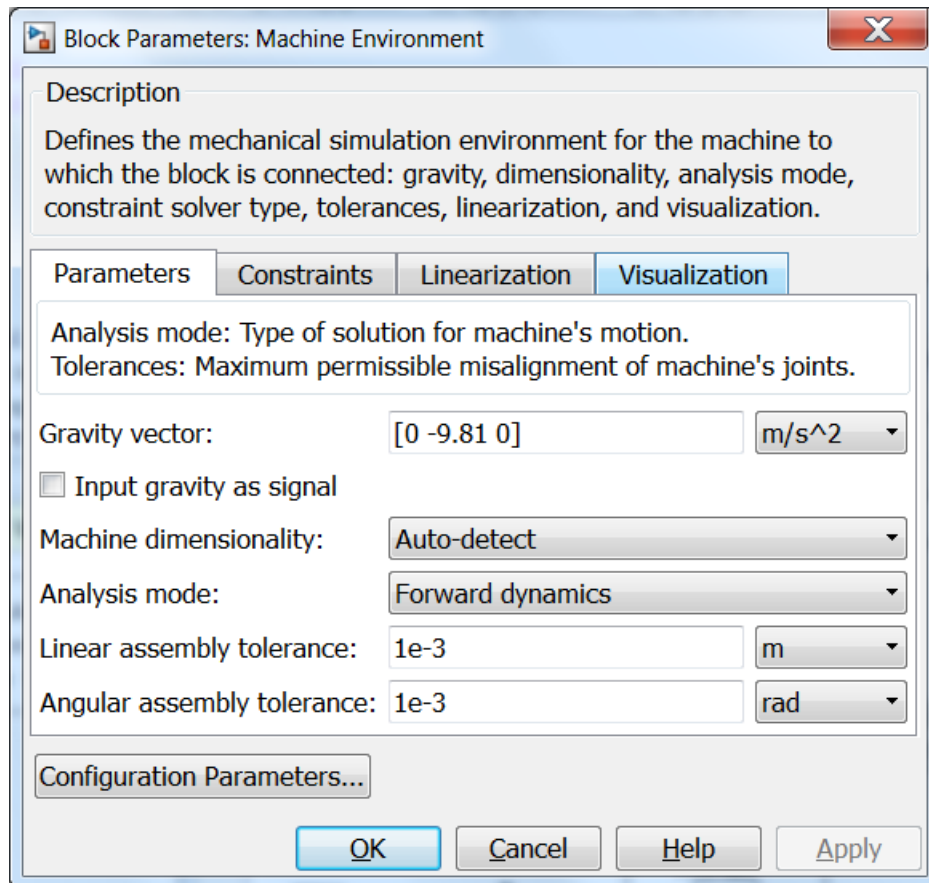
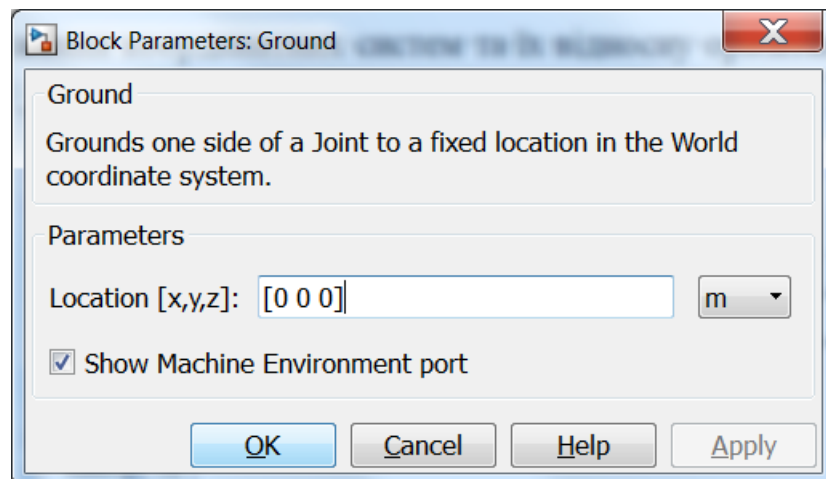
Вміст розділу *Bodies* відображено на рис. 14.6.

Рис. 14.6. Блоки розділу *Bodies*

Комбінація блоків *Machine Environment* і *Ground* задає гравітаційні сили, прикладені до центру координат та їх орієнтацію в заданій системі відліку (див. рис. 14.7). *За замовчанням у SimMechanics прийнята Ейлерова правобічна система координат (Euler X-Y-Z)*, зображена на рис. 14.8. Тому вектор сили тяжіння (*Gravity vector*) у блоці *Machine Environment* (рис. 14.9) має вигляд $[0 \ 9.81 \ 0]$.

Рис. 14.7. Комбінація блоків *Machine Environment* і *Ground*Рис. 14.8. Система координат, прийнята у *SimMechanics*

Блок *Ground* уявляє собою нерухому основу, жорстко пов'язану з абсолютною світовою системою координат *World*. До кожного блоку *Ground* приєднують свій блок *Machine Environment*. Щоб з'єднати ці блоки треба у вікні налаштування останнього (рис. 14.10) встановити прапорець у полі *Show Machine Environment Port*.

Рис. 14.9. Вікно налаштування блоку *Machine Environment*Рис. 14.10. Вікно налаштування блоку *Machine Environment*

При дослідженні механічних об'єктів скористаємося такою термінологією [9]:

- **механічна ланка** – сукупність деталей, з'єднаних у такий спосіб, що їх взаємне розташування не змінюється у процесі руху;

- **кінематична пара** – з'єднання двох механічних ланок, що стикаються, але можуть змінювати своє відносне положення у просторі;
- **нижчі кінематичні пари** – це такі пари, в яких ланки торкаються одна одної по поверхням концевих розмірів або по площині. До них відносяться поступальна та оберտальна пари першого, другого і третього класів, а також гвинтова пара.
- **вищі кінематичні пари** – це пари, в яких контакт елементів відбувається по лінії або у точці.

Кінематичні пари відрізняються одна від одної кількістю ступенів свободи, які можуть бути як поступальними, так і обертальними.

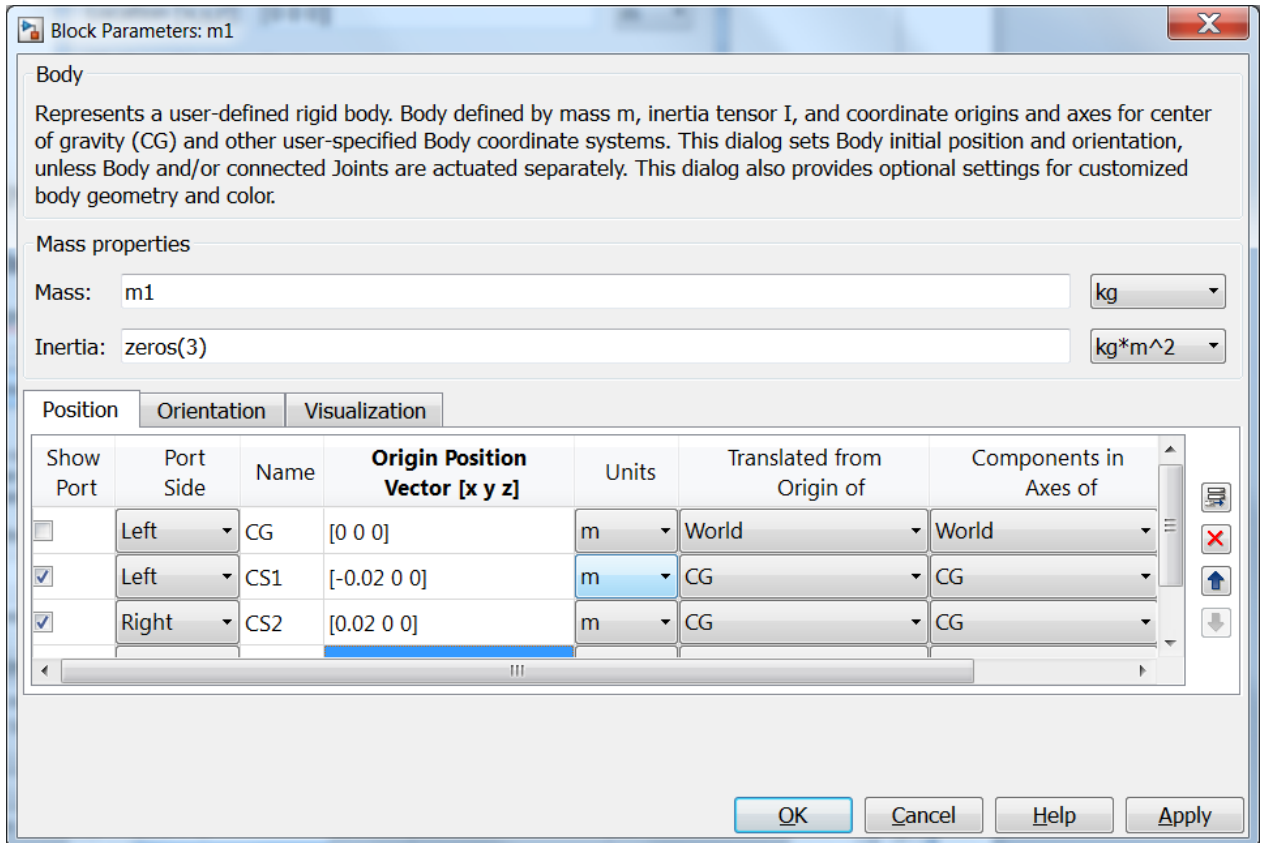
Механічні ланки відображаються блоками *Body*, а зв'язки між ними (з'єднання, суглоби, кінематичні пари) – блоками розділу *Joints*.

Вікно налаштування блок *Body* показано на рис. 14.11.

Його структура дозволяє повністю визначити фізичні параметри матеріальних тіл: масу, тензор інерції, геометричний центр мас, положення координатних систем та їх відносну орієнтацію.

Тензор інерції тіла відносно його центра мас (*Inertia*) уявляє собою матрицю 3×3 , яка для симетричних тіл є діагональною $\mathbf{J} = \text{diag}([J_{ix} \ J_{iy} \ J_{iz}])$. На головній діагоналі розташовані моменти інерції тіла відносно осей власної локальної системи координат, пов'язаної з його центром мас. Якщо тіло моделюється як точкова маса, то усі складові тензору інерції є нульовими: $\mathbf{J} = \text{zeros}(3,3)$.

Декартові координати характерних точок тіла (*Origin Position Vector*), серед яких обов'язково повинна бути точка центру тяжіння (CG) і довільна кількість інших точок (CS1, CS2, ..., CSN), призначених або для з'єднання їх з іншими блоками або для підвищення наочності візуалізації. У першому випадку у полі *Show Port* треба встановити прапорець. Для кожної характерної точки можна обрати місце її розташування у вигляді порту *Port Side* (*Left* або *Right*). «Кнопки» з правого боку вікна на рис. 14.11 дозволяють додавати, видаляти та міняти черговість цих точок у списку.

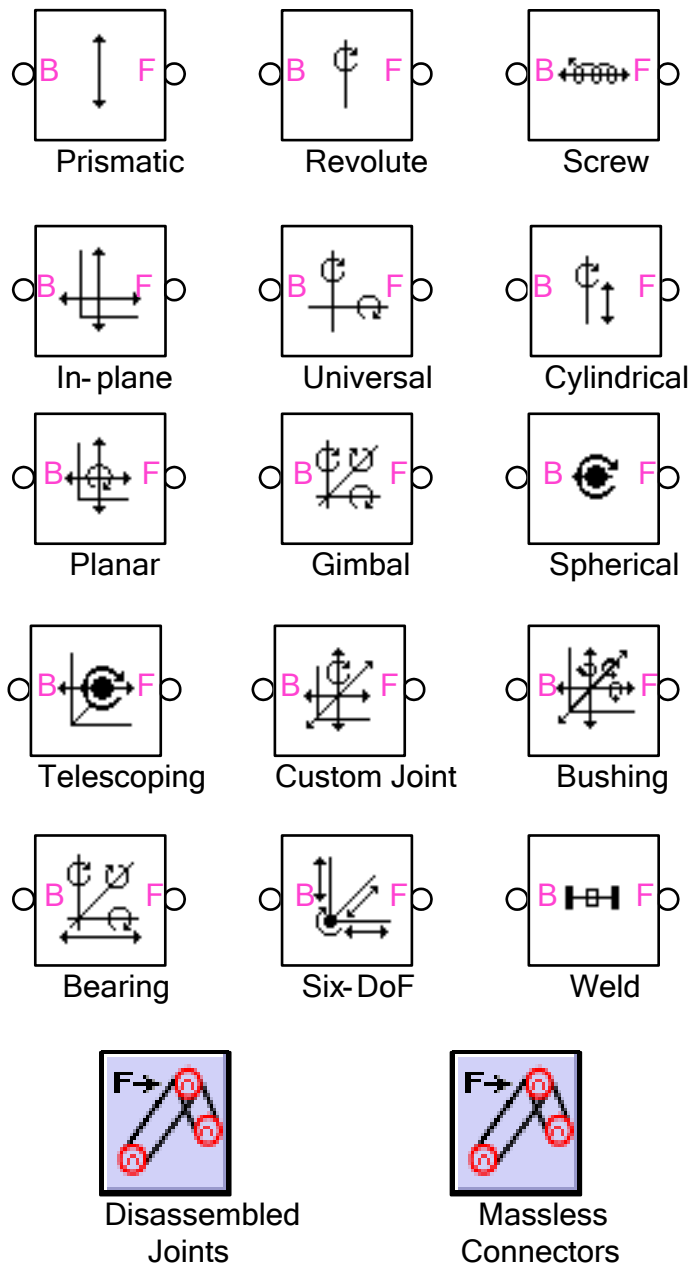
Рис. 14.11. Вікно налаштування блоку *Body*

Координати тіл і суглобу, що складають кінематичну пару, повинні бути узгодженими, тобто позиції поєднаних через суглоб тіл матимуть збігатися.

Блоки бібліотеки *Joints* показано на рис. 14.12. Розглянемо деякі з них.

Жодного ступеня свободи немає пара ***Weld*** (**нерухоме з'єднання, стойка, сварка**).

До пар з **одним ступенем свободи** належать: ***Prismatic*** (**кінематична пара поступального руху**) – забезпечує переміщення впродовж однієї прямої, ***Revolute*** (**поворотний важіль, обертальна пара, циліндричний шарнір**) – забезпечує обертальний рух навколо однієї з осей тримірного простору, тобто в одній площині (наприклад, математичний маятник), ***Screw*** (**гвинтова пара, черв'як, шнек**) – перетворює обертальний рух у поступальний впродовж однієї прямої.

Рис. 14.12. Блоки розділу *Joints* (Суглоби)

пальний рух у просторі, **Gimbal** (**шарнір, кардан**) – забезпечує обертальний рух з можливістю повороту осі обертання, **Spherical** (**сферичний шарнір, шарова опора**) – забезпечує обертальний рух навколо однієї точки з можливістю повороту площини обертання.

При складанні моделі тіла і суглоби повинні чергуватися, тобто будь-який суглоб розташовується завжди між двома тілами, попереднє з яких, що називається *Base Body*, приєднується до порту *B*, а наступне (*Following Body*) – до порту *F* суглобу.

До пар з **двома ступенями свободи** належать: **In-plane** (**планарна, плоска пара**) забезпечує поступальний рух в одній площині, **Cylindrical** (**циліндрична пара**) – забезпечує обертальний рух навколо однієї з осей тримірного простору та поступальний рух впродовж цієї ж осі, **Universal** (**універсальна обертальна пара**) – забезпечує обертальний рух з можливістю повороту площини обертання наприклад, маятник, що може змінювати площину коливань.

До пар з **трьома ступенями свободи** належать: **Planar** (**пара поступального руху**) – забезпечує посту-

Для прикладу на рис. 14.13 подано вікно налаштування циліндричного суглобу.

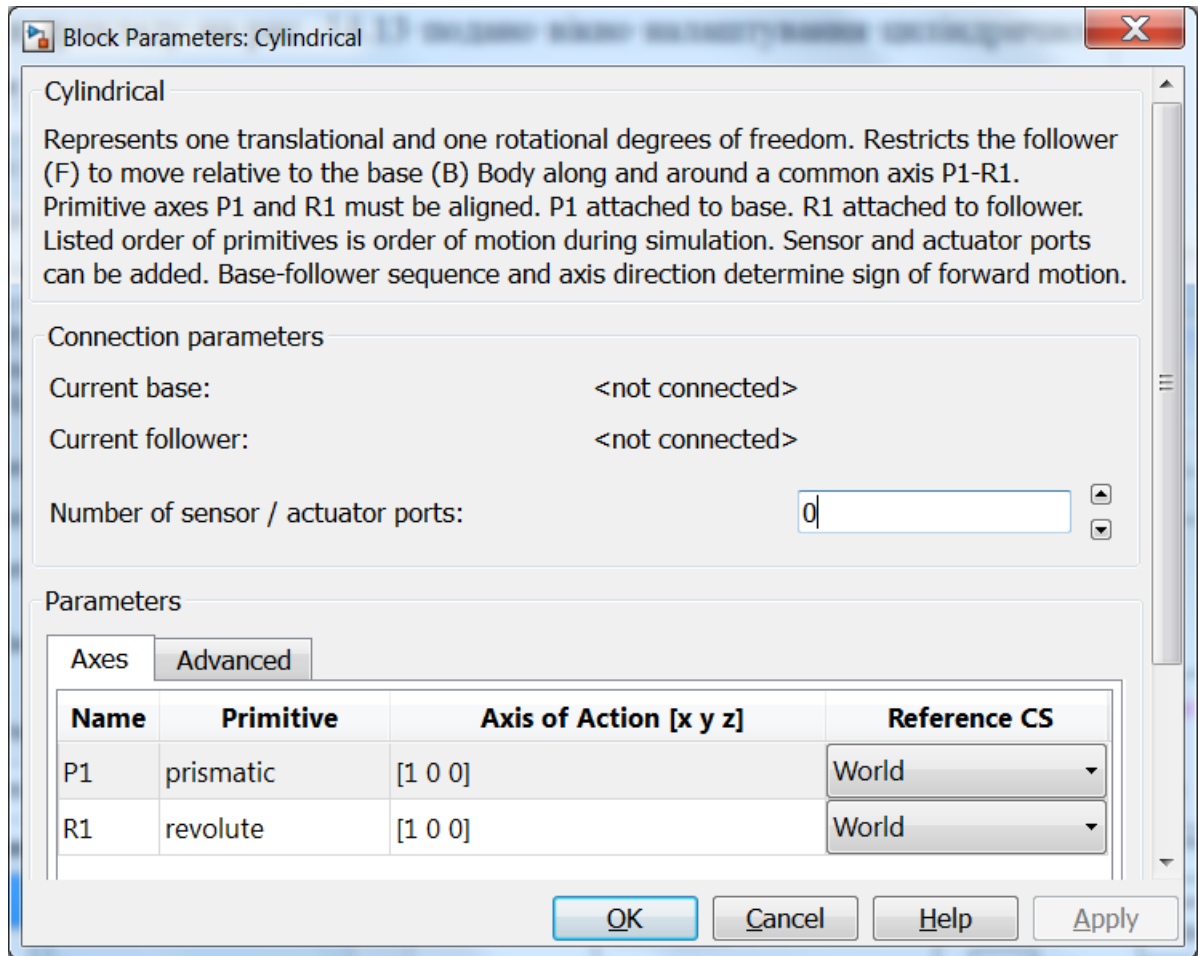


Рис.14.13. Вікно налаштування суглобу *Cylindrical*

Для моделювання пружно в'язких з'єднань між механічними ланками використовують блоки розділу *Force Elements* (див. рис. 14.14).

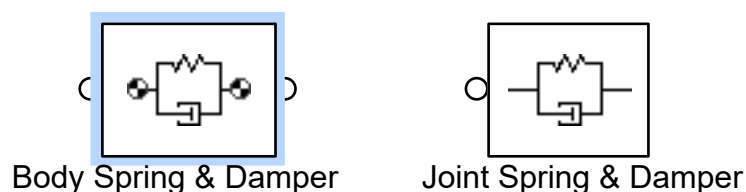
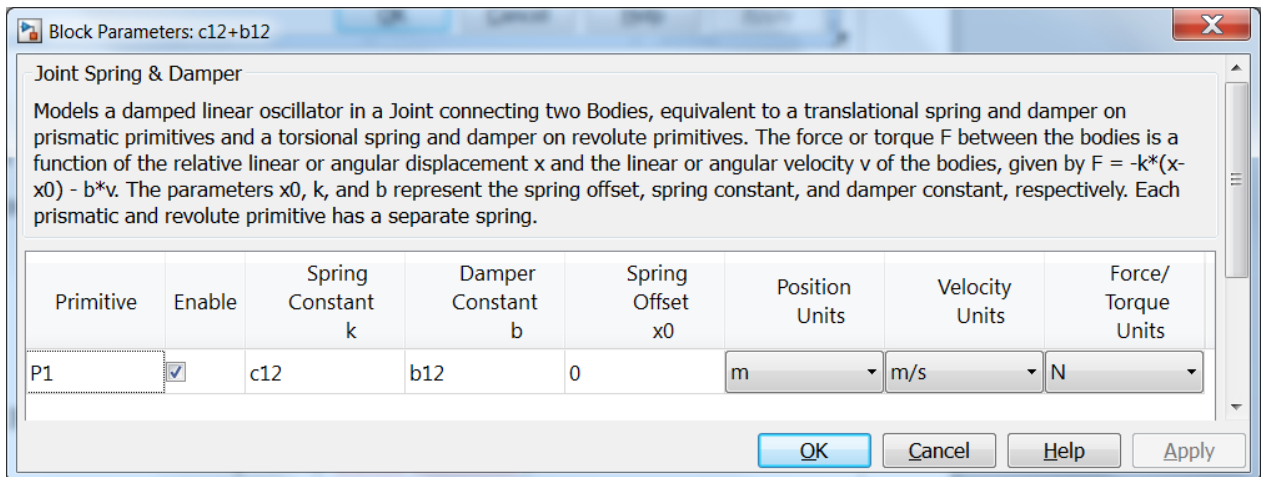
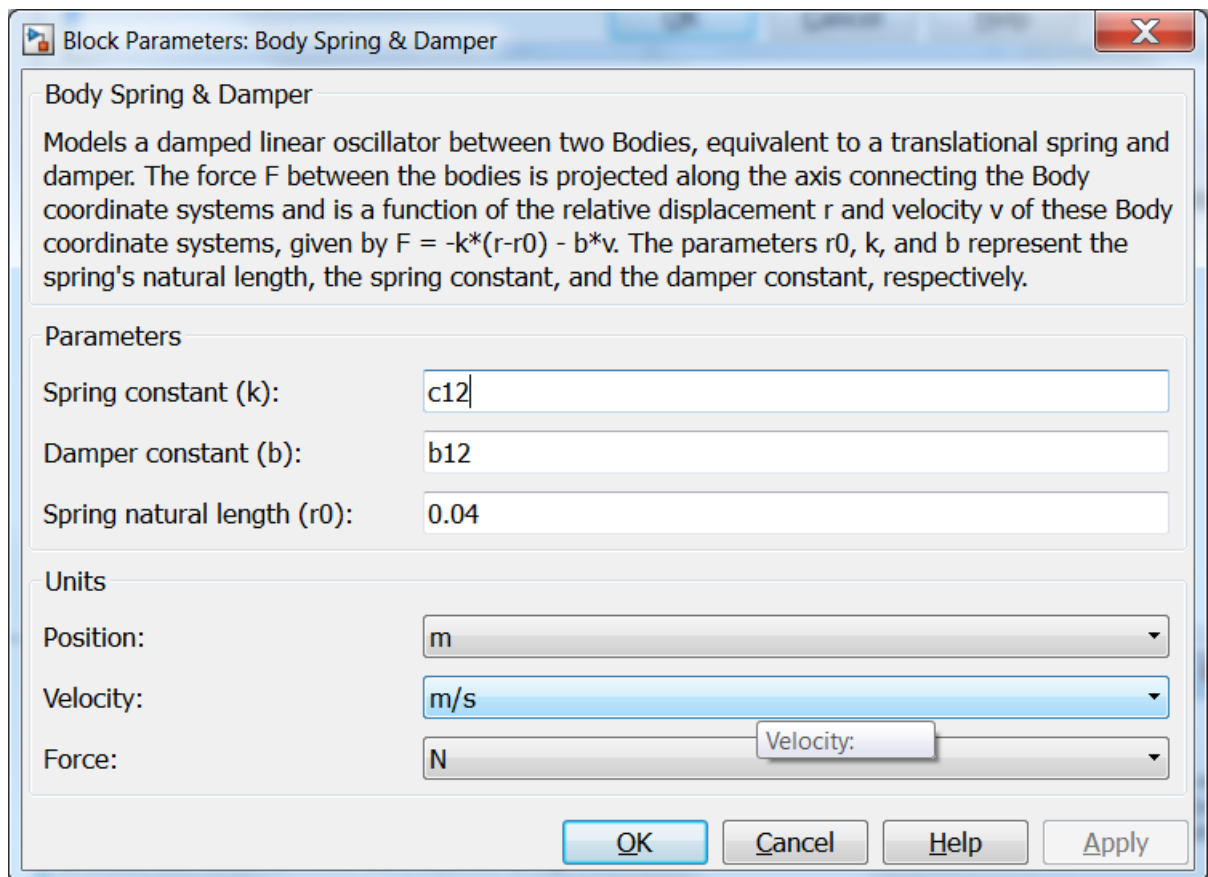


Рис. 14.14. Блоки для моделювання пружно-в'язких зв'язків розділу *Force Elements*

Вікна параметрів цих блоків подані на рис. 14.15 та 14.16.

Рис.14.15. Вікно налаштування блоку *Joint Spring & Damper*Рис.14.16. Вікно налаштування блоку *Body Spring & Damper*

Перший з цих блоків має тільки один порт, що приєднується до суглобу, а другий має два порти, які з'єднують між собою два тіла, вмикаючись паралельно з суглобом даної кінематичної пари. Цей блок може збуджувати один або декілька примітивів суглобу, що визначається установкою прапор-

ців у стовпчику *Enable* (розблокувати). Пружно-в'язке зусилля або пружно-в'язкий момент у першому випадку прикладається до частин суглобу, а у другому – до обох тіл у заданих точках з протилежними знаками.

Параметри *Spring offset* та *Spring naturel length* визначають відстань між точками взаємодії тіл, при якій сила пружності дорівнює 0.

У розділі *Sensors & Actuators* (рис. 14.17) розташовані датчики для вимірювання механічних сигналів та приводні механізми, що дають можливість прикласти до механічних ланок або з'єднань між ними зусилля або моменти.

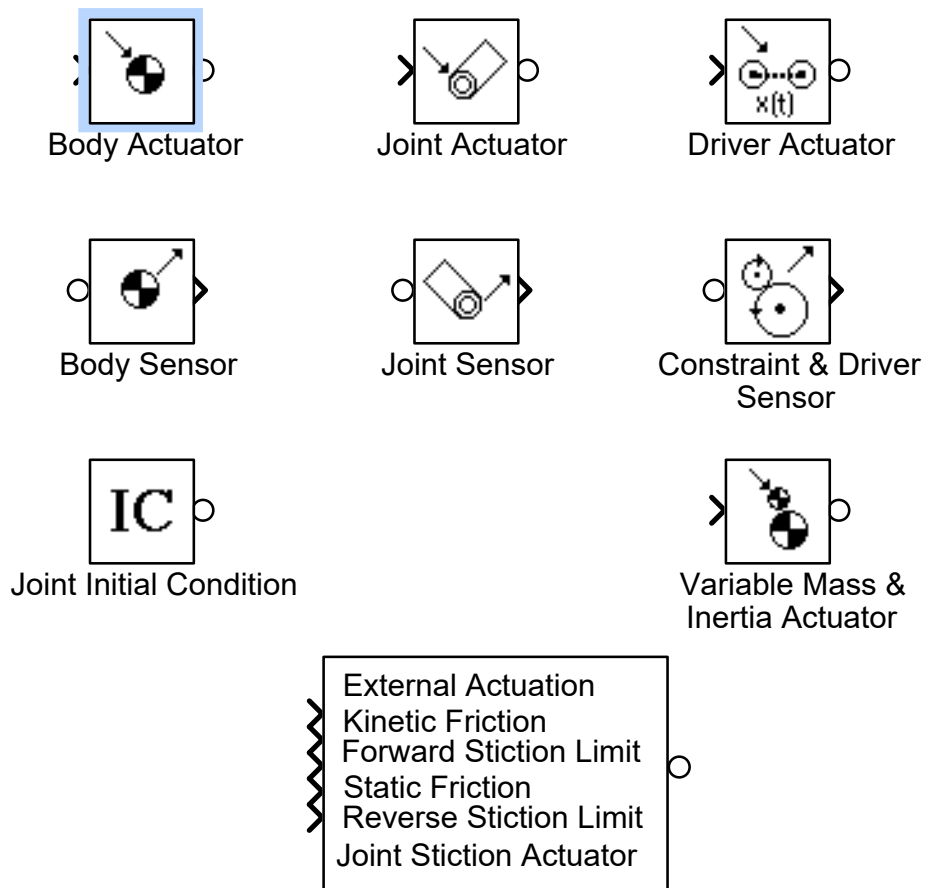


Рис. 14.17. Блоки розділу *Sensors & Actuators*

Блоки *Actuator* перетворюють силові сигнали (зусилля або моменти) або завдання на рух (лінійні чи кутові прискорення, швидкість та переміщення у вигляді векторного сигналу), сформовані за допомогою блоків *Simulink*, у сигнали, що можуть бути сприйнятими блоками *SimMechanics*. Різниця між блоками *Body Actuator* та *Joint Actuator* полягає у тому, що блок *Body Actuator* вимагає завдання відповідних сигналів у трьох напрямках Декартової сис-

теми координат або обертання навколо трьох її осей, а блок *Joint Actuator* – тільки сигналів, притаманних заданому ступеню свободи.

Блоки *Sensor* вимірюють сигнали, що виробляються тілами або суглобами, та перетворюють їх у сигнали, придатні для подальшої обробки, візуалізації або запам'ятовування їх за допомогою блоків *Simulink*. Різниця між блоками *Body Sensor* та *Joint Sensor* полягає у тому, що блоки *Body Sensor* вимірюють вектори абсолютних сигналів поступального або обертального рухів, а блок *Joint Sensor* – різниці між сигналами тіл однієї кінематичної пари у напрямку. сигнали, притаманних заданому ступеню свободи.

Блок *Joint Initial Condition* задає початкове відносне положення та початкову відносну швидкість двох частин суглобів типу *Revolute* або *Prismatic*.

Розглянутих блоків цілком достатньо для створення простих віртуальних механічних моделей.

Віртуальна фізична модель двомасової механічної системи з використанням описаних вище блоків подана на рис. 14.18. Тип руху (поступальний чи обертальний) залежить тільки від параметрів блоків.

Блок *Body1* є моделлю першої маси, а блок *Body2* – другої. Блоки *Prismatic* за допомогою встановлення параметру *Axis of Action [x y z]* у стан $[1 \ 0 \ 0]$ визначають один поступальний ступінь свободи у напрямку осі x (див. рис. 14.8) між нерухомою основою і першою масою та між першою і другою масами. Зусилля F , сформоване блоком *Step1*, прикладається до блоку *Body1* через суглоб *Prismatic1* у напрямку осі x .

Блок *Joint Sensor* вимірює силу реакції суглобу на зусилля, створене блоком *Joint Spring & Damper*, тобто є пружно-в'язким зусиллям F_{12} . Блок вимірює швидкість першої маси та зусилля, що діють на неї у трьох напрямках. Оскільки базовим блоком першої кінематичної маси є нерухома опора, то виміряні відносні сигнали суглобу збігаються з відповідними абсолютними сигналами першої маси. Для вимірювання абсолютної швидкості першої маси встановлено блок *Body Sensor*.

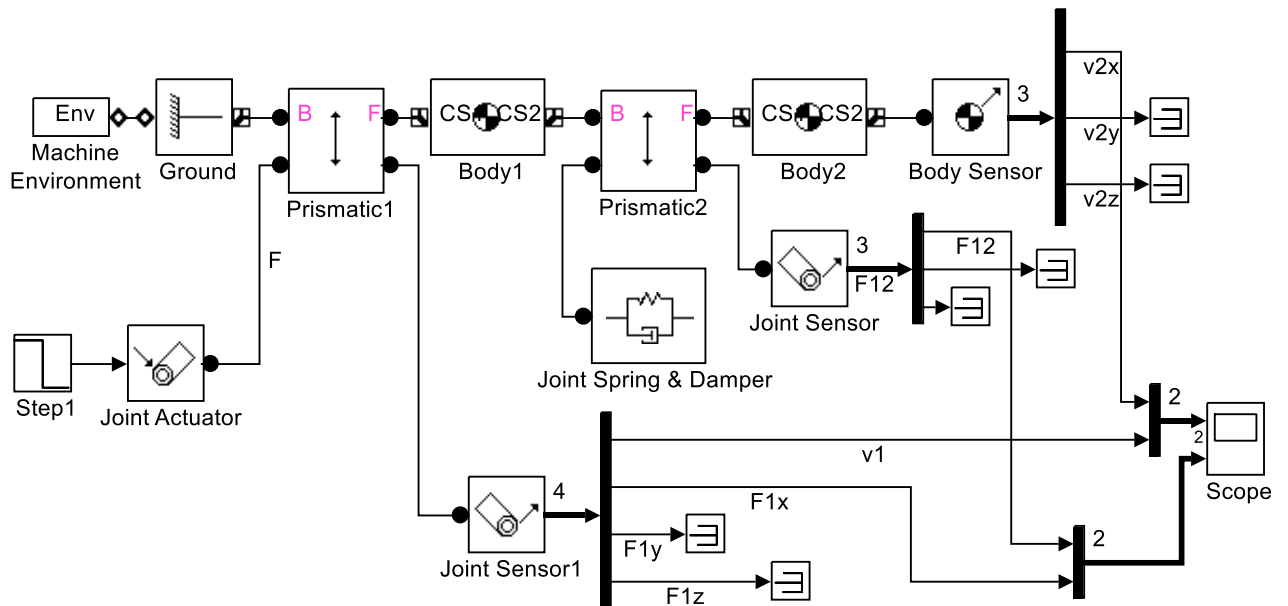


Рис. 14.18. Віртуальна фізична модель двомасової механічної системи поступального руху

14.4 Завдання

Розрахувати перехідні процеси у двомасовій механічній системі поступального руху, зображеній на рис. 14.2 при заданому характері зміни зусилля із відображених на рис. 14.19. Параметри систем надано у табл. 14.1.

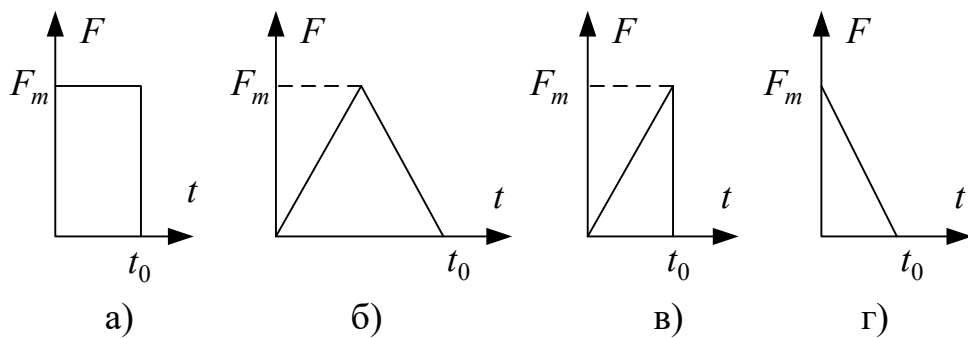


Рис. 14.19. Варіанти вхідних зусиль

Побудувати графіки переміщень і швидкостей зосереджених мас та пружно-в'язких зусиль у елементах кінематичних зв'язків у функції часу. Розрахунки виконати методами структурного математичного та віртуального фізичного моделювання. Результати порівняти.

Таблица 14.1

№ вар.	m_1 , кг	m_2 , кг	c_{12} , кг/с ²	d_{12} , кг/с	F_m , Н	t_0	$F(t)$
1,13	5	5	20	0 5 10	16	4	а
2,14	5	5	20	0 5 10	16	8	б
3,15	5	5	20	0 5 10	16	7	в
4,16	5	5	20	0 5 10	16	6	г
5,17	7.5	2.5	10	0 2 5	16	4	а
6,18	7.5	2.5	10	0 2 5	16	8	б
7,19	7.5	2.5	10	0 2 5	16	7	в
8,20	7.5	2.5	10	0 2 5	16	6	г
9,21	2.5	7.5	20	0 5 10	16	4	а
10,22	2.5	7.5	20	0 5 10	16	8	б
11,23	2.5	7.5	20	0 5 10	16	7	в
12,24	2.5	7.5	20	0 5 10	16	6	г

14.4 Порядок виконання роботи

- 1) Ознайомитися з блоками бібліотек *SimMechanics*.
- 2) Зібрати нелінійну та лінеаризовану *S*-моделі математичного маятника, розрахувати параметри коливань за формулами
- 3) За даними двигуна розрахувати необхідні параметри і встановити їх.
- 5) Виконати такі віртуальні експерименти:
 - прямий пуск двигуна від джерела, що виробляє трифазну синусоїдальну напругу, параметри якої відповідають номінальній напрузі та номінальній частоті статора; після досягнення двигуном синхронної швидкості здійснити стрибкоподібний накид навантаження;
 - поступово підвищуючи момент статичного опору на валу двигуна, знайдіть його критичний момент та критичне ковзання;
 - виконайте попередні 2 експерименти зі зменшеною на 50% частотою джерела при номінальній амплітуді та зі зменшеною на 20% амплітудою напруги джерела при номінальній частоті;
 - здійсніть пуск двигуна у протилежному напрямку.
- 6) Спробуйте побудувати статичну механічну та статичну швидкісні характеристики двигуна.
- 7) Проаналізуйте результати досліджень. Сформулюйте висновки.

14.5 Методичні вказівки

- 1) Зовнішнє зусилля формуйте блоками *Simulink* та приєднуйте його до вхідного порту блоку *Joint Actuator*. Зусилля можна прикладати і до блоку *Body Actuator*, але тоді треба формувати зусилля за трьома напрямками ($F_x = F(t)$, $F_y = 0$, $F_z = 0$), а потім поєднувати їх у один векторний сигнал.
- 2) Параметр *Inertia* блоків *Body* можна задавати матрицею з нулів розміром 3×3 (`zeros(3,3)`), тому що візок не має жодного обертового ступеня руху.

4) До блоків *Body* та блоків бібліотеки *Joints* можна приєднувати додаткові порти, розташовуючи їх за власним вибором ліворуч або праворуч, що ніяк не впливає на роботу цих блоків.

5) До непотрібних для візуалізації виходів блоків приєднайте ланки *Terminator Simulink*-бібліотеки *Sinks*.

14.6 Контрольні питання та завдання

- 1) Чим відрізняються жорсткі механічні системи від пружно-в'язких?
- 2) Як пов'язані між собою сигнали поступального та обертального рухів?
- 3) Що уявляють собою коефіцієнт пружності та коефіцієнт в'язкого тертя?
- 4) Запишіть рівняння руху першої та другої мас пружнов'язкої двомасової системи.
- 5) Як обчислити періоди й амплітуди коливань двомасової системи і другої маси у виконуваних експериментах?
- 6) Як впливає на перехідні процеси ДЕМС коефіцієнт жорсткості, коефіцієнт в'язкого тертя, коефіцієнт розподілу мас?
- 7) Що таке кінематична пара? Наведіть приклади кінематичних пар.
- 8) Як треба узгоджувати параметри тіл та суглобу, що входять до складу однієї кінематичної пари?

*Лабораторна робота №15***ВІРТУАЛЬНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ КРАНОВИХ
МЕХАНІЗМІВ****15.1 Теоретичні відомості**

Переміщення вантажів в умовах виробничого процесу неможливо здійснити без кранових установок. При цьому вантаж переміщується одночасно у декількох напрямках. Зокрема, мостові крани представляють собою сукупність трьох основних механізмів:

- моста, що пересувається по рейках, покладеним на підкранові балки уздовж прольоту відкритої або закритої вантажної площадки;
- візка з підвішеним до неї вантажем, що переміщається уздовж моста, тобто поперек прольоту площадки;
- піднімального пристрою у вигляді лебідки для намотування каната, до якого прикріплений вантаж.

Оскільки вантаж є підвішеним до візка на гнучкому канаті, то він може ще відхилятися на деякий кут від вертикалі і здійснювати коливання у вертикальній площині, просторова орієнтація якої залежить від співвідношення параметрів руху моста та візка.

Здійснимо розробку моделі процесу пересування вантажу візком мостового крану за такою методикою:

- 1) розробляємо математичний опис системи при прийнятих припущеннях;
- 2) за отриманим математичним описом розробляємо структурну модель;
- 3) виконуємо лінеаризацію математичного опису і структурної моделі;
- 4) на основі лінеаризованої моделі розраховуємо параметри характерних точок перехідних процесів;
- 5) порівнюємо результати симуляції лінійної та нелінійної структурних математичних моделей з точки зору відповідності розрахованим параметрам;

- б) розробляємо віртуальну фізичну модель досліджуваної системи і порівнюємо результати її симуляції з результатами симуляції нелінійної моделі.

Така складна методика необхідна для того, щоб переконатися у відсутності помилок при складанні математичного опису та освоїти основні прийоми розробки віртуальних математичних моделей, головними перевагами яких є складання їх при відсутності математичного опису, простота отримання анімації та простота ускладнення моделей при поступовому введенні нових ступенів свободи, врахуванні форми і геометричних розмірів тіл і т.п.

Почнемо з моделі вантажу, підвішеного до нерухомої опори, який при виведенні його із стану рівноваги, тобто при відхиленні осі канат-вантаж від вертикалі, починає коливатися. Зазвичай при математичному описі такого об'єкта його вважають математичним маятником – точковим тілом, підвішеним до нерозтяжної і невагомої нитки, що здійснює коливання в одній площині під дією сили власного тяжіння без врахування будь-яких сил опору. Така уява можлива, якщо розміри вантажу набагато менші за довжину нитки, маса вантажу набагато більша маси нитки, довжина нитки та її пружні властивості не занадто великі, щоб ними не можна б було знехтувати, сили опору коливальному руху відсутні.

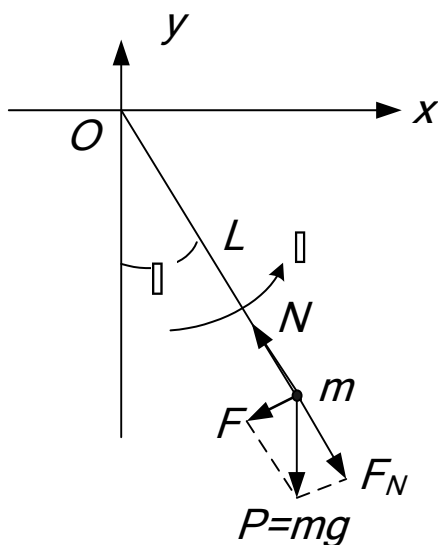


Рис. 15.1. Математичний маятник

Схематичне зображення математичного маятника масою m з довжиною нитки підвісу L подано на рис. 15.1.

Силу тяжіння $P = mg$ ($g = 9.81 \text{ м/с}^2$ – прискорення сили тяжіння) можна розкласти на 2 складові, одна з яких (F_N) урівноважується силою натягу нитки N , а друга ($F = mg \sin \varphi$) створює відносно точки закріплення нитки O момент $M = -FL$,

спрямований у бік зменшення кута φ . Тому його записуємо зі знаком «-».

Момент інерції матеріальної точки відносно точки підвісу дорівнює $J = mL^2$. Тому рівняння руху математичного маятника $M = J \cdot d\omega/dt$ можна записати у вигляді $-mgL \sin \varphi = mL^2 \cdot d\omega/dt$, або після скорочення множника mL

$$-g \sin \varphi = L \frac{d\omega}{dt}. \quad (15.1)$$

Після доповнення рівняння (2.1) диференціальним рівнянням

$$\frac{d\varphi}{dt} = \omega \quad (15.2)$$

складаємо на їх основі структурну *Simulink*-модель (*S*-модель), показану на рис. 15.2.

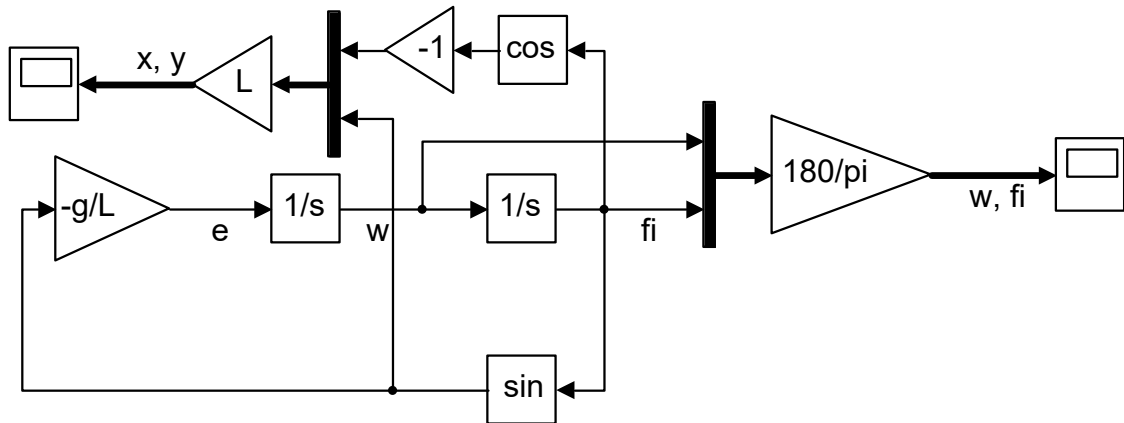


Рис. 15.2. Нелінійна структурна *S*-модель математичного маятника

При невеликих кутах відхилення маятника її можна лінеаризувати, скориставшись співвідношенням

$$\sin \varphi \approx \varphi. \quad (15.3)$$

Характеристичний поліном лінеаризованого об'єкта має вигляд

$$T_0^2 s^2 + 1 = 0,$$

де

$$T_0 = \sqrt{L/g}, \quad (15.4)$$

а перехідний процес, зумовлений початковим відхиленням маятника на кут

φ_0 ($\varphi(0) = \varphi_0$, $\omega(0) = 0$) описується рівнянням

$$\varphi(t) = \varphi_0 \cos \Omega_0 t, \quad \omega(t) = -\varphi_0 \Omega_0 \sin \Omega_0 t = -\omega_m \sin \Omega_0 t,$$

де $\Omega_0 = 1/T_0$ – кругова частота власних (невимушених) коливань лінеаризованої моделі математичного маятника, $\omega_m = \varphi_0 \Omega_0 = \varphi_0 / T_0$ – амплітуда коливань кутової швидкості.

У відповідності з (15.4) період коливань становить

$$T_{k0} = 2\pi T_0 = 2\pi \sqrt{L/g}. \quad (15.5)$$

Період коливань нелінійної моделі при $\varphi_0 \leq 60^\circ$ можна приблизно (з похибкою менше 1%) визначити за більш точною формулою

$$T_{kn} \approx 2\pi \sqrt{L/g} \left(1 + \frac{1}{4} \sin^2 \frac{\varphi_0}{2} \right). \quad (15.6)$$

Можна також розрахувати максимальні відхилення вантажу по осям x та y :

$$x_m = x_0 = L \sin \varphi_0, \quad y_m = y_0 = -L \cos \varphi_0. \quad (15.7)$$

SM-модель математичного маятника наведена на рис. 15.3.

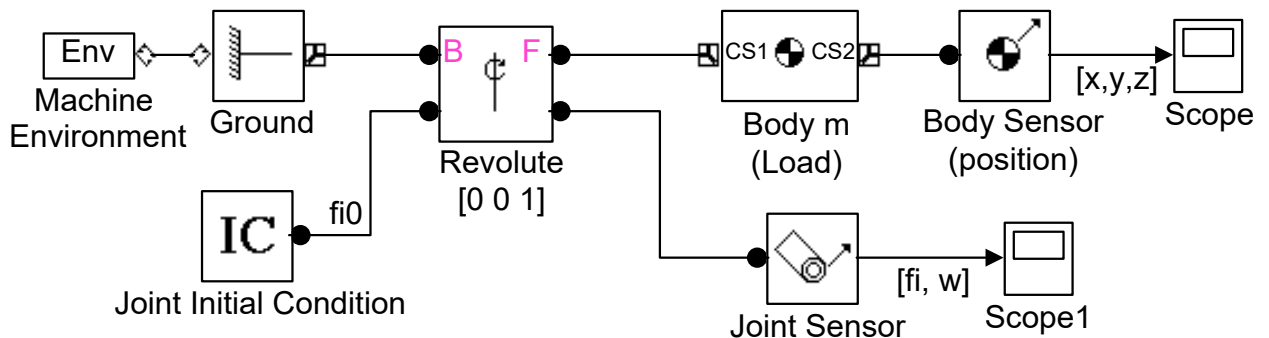


Рис. 15.3. Віртуальна фізична SM-модель математичного маятника

У даному випадку система координат основи визначена як мирова. До основи приєднано кінематичний зв'язок *Revolute*, що утворює із нерухомої опори *Ground* та тіла *Body* з точковою масою m кінематичну пару з одним обертовим ступенем свободи навкруги осі z , що задається параметром *Axis of Action* $[x \ y \ z]$, який у даному прикладі має значення $[0 \ 0 \ 1]$.

До невагомого з'єднання (нитка, канат) і тіла з точковою масою *Body m* (*Load*) приєднані відповідні датчики *Joint Sensor* та *Body Sensor*. Обрані для візуалізації сигнали фіксуються будь-якими вихідними блоками *Simulink*-бібліотеки *Sinks*. Початкове відхилення вантажу φ_0 задає блок *Joint Initial Condition (IC)*, приєднаний до блоку *Revolute*.

Для правильного спряження тіл, що утворюють між собою кінематичні пари, у блоці *Body (Вантаж)* задано в СК *World* координати таких точок: CG: [0 -L 0], Cs1: [0 0 0], Cs2: [0 -L 0]. Точка Cs2 забезпечує не тільки приєднання до вантажу блоку *Body Sensor*, але й візуалізацію невагомому канату при анімації рухів вантажу.

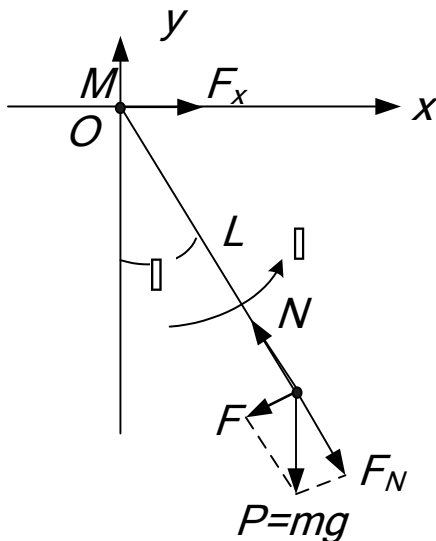


Рис. 15.4. Механічна система візок-вантаж

Тепер розробимо моделі для дослідження рухів вантажу, у якого точка підвісу прикріплена не до нерухомої опори, а до візка, що здійснює переміщення вантажу у напрямку x . Така механічна система складається з вантажу масою m , який на канаті підвішений до візка мостового крану масою M , тобто вона стає двомасовою (див. рис. 15.4). Візок, як і вантаж, будемо вважати точковою масою, зосередженою у центрі його тяжіння. Рух візка

відбувається у горизонтальному напрямку впродовж осі x під дією деякої сили F_x при постійній довжині канату.

Для складання математичного опису скористаємося рівняннями Лагранжа другого роду, яке у загальному випадку має такий вигляд:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i, \quad (15.8)$$

де $q_i, \dot{q}_i$ – узагальнені координати та їх перші похідні за часом, Q_i – узагальнені сили, $\mathcal{L} = W_k - W_p$ – функція Лагранжа, яка дорівнює різниці між кінетичною і потенціальною енергією.

Вирази для визначення просторових координат візка та вантажу в кожен момент часу мають вигляд

$$\begin{aligned}\vec{s}_M &= [s_{Mx}(t), 0] = [x(t), 0], \\ \vec{s}_m &= [s_{mx}(t), s_{my}(t)] = [x(t) + L \sin \varphi(t), -L \cos \varphi(t)],\end{aligned}\quad (15.9)$$

у результаті диференціювання яких отримаємо такі вирази для проекцій векторів швидкостей руху візка і вантажу на обрані координатні осі:

$$\begin{aligned}\vec{v}_M &= [v_{Mx}(t), 0] = [v_x(t), 0], \\ \vec{v}_m &= [v_{mx}(t), 0] [v_x(t) + L \omega(t) \cos \varphi(t), L \omega(t) \sin \varphi(t)].\end{aligned}\quad (15.10)$$

Для системи візок-вантаж формули потенціальної і кінетичної енергії мають вигляд

$$\begin{aligned}W_k &= \frac{m}{2} [(v_x(t) + L \omega(t) \cos \varphi(t))^2 + (L \omega(t) \sin \varphi(t))^2] + \frac{M}{2} v_x^2(t) = \\ &= \frac{m+M}{2} v_x^2(t) + \frac{mL}{2} (L \omega^2(t) + 2 v_x(t) \omega(t) \cos \varphi(t)), \\ W_p &= -mgL \cos \varphi(t).\end{aligned}$$

Тоді функція Лагранжа для системи «візок-вантаж» матиме вигляд:

$$\mathcal{L} = \frac{M+m}{2} v_x^2(t) + \frac{mL}{2} (2 v_x(t) \omega(t) \cos \varphi(t) + L \omega^2(t) + 2g \cos \varphi(t)). \quad (15.11)$$

Узагальненими координатами і силами для досліджуваної системи являються:

$$\begin{cases} q_1 = x, & \dot{q}_1 = v_x, & Q_1 = F_x; \\ q_2 = \varphi, & \dot{q}_2 = \omega, & Q_2 = 0. \end{cases} \quad (15.12)$$

Визначивши послідовно усі складові рівняння Лагранжа (15.10), одержуємо такі рівняння:

$$\begin{cases} (M+m) \frac{dv_x(t)}{dt} + mL \cos \varphi(t) \cdot \varepsilon(t) - mL \omega^2(t) \cdot \sin \varphi(t) = F(t), \\ L \frac{d\omega(t)}{dt} + \cos \varphi(t) \cdot a_x(t) + g \sin \varphi(t) = 0, \end{cases} \quad (15.13)$$

які доповнюємо рівняннями, що пов'язують узагальнені координати з їхніми похідними:

$$\begin{cases} \frac{dx(t)}{dt} = v_x(t), & \frac{dv_x(t)}{dt} = a_x(t); \\ \frac{d\varphi(t)}{dt} = \omega(t), & \frac{d\omega(t)}{dt} = \varepsilon(t). \end{cases} \quad (15.14)$$

де $a(t)$ – лінійне прискорення візка; $\varepsilon(t)$ – кутове прискорення вантажу;

Структурна модель, що відповідає рівнянням (15.13), (15.14), наведена на рис. 15.5.

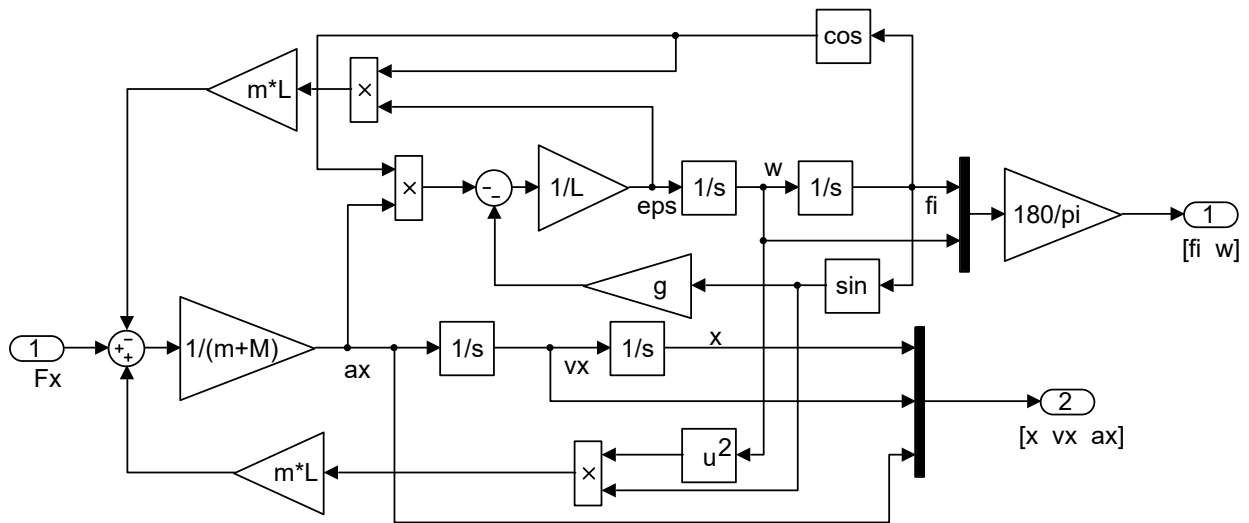


Рис. 15.5. Структурна модель системи візок- вантаж

Відповідна віртуальна фізична модель зображена на рис. 15.6

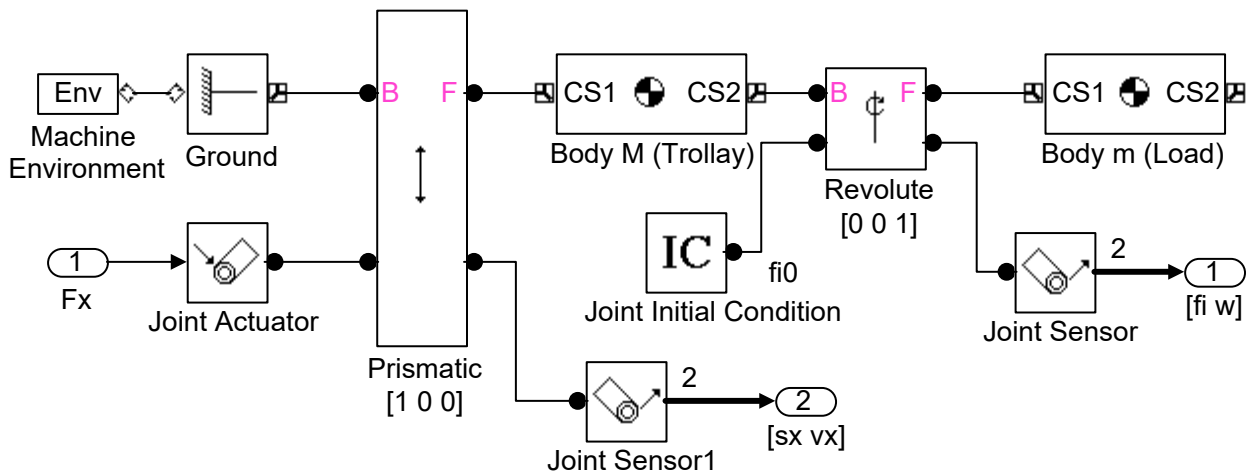


Рис. 15.6. Віртуальна фізична модель системи візок-вантаж

Вона отримана із моделі математичного маятника рис. 15.2 шляхом додавання ще одного тіла (візка) масою M , яке утворює з нерухомою опорою

кінематичну пару з одним поступальним ступенем свободи у напрямку осі x , що визначається кінематичним зв'язком *Prismatic* з параметром *Axis of Action* = $[1 \ 0 \ 0]$, а з вантажем – кінематичну пару з одним обертальним ступенем свободи обертання навколо осі z , що визначається кінематичним зв'язком *Revolute*.

Лінеаризація математичної моделі можлива при малих кутах відхилення канату від вертикалі, коли можна зробити такі припущення:

$$\cos \varphi \approx 1, \quad \sin \varphi \approx \varphi, \quad \omega^2 \approx 0. \quad (15.15)$$

В такому випадку рівняння (15.14) набувають вигляду:

$$\begin{cases} (M + m) \frac{dv(t)}{dt} = F(t) - mL\varepsilon(t), \\ L \frac{d\omega(t)}{dt} = -a(t) - g\varphi(t). \end{cases} \quad (15.16)$$

Такій системі рівнянь відповідає структурна модель, наведена на рис. 15.7.

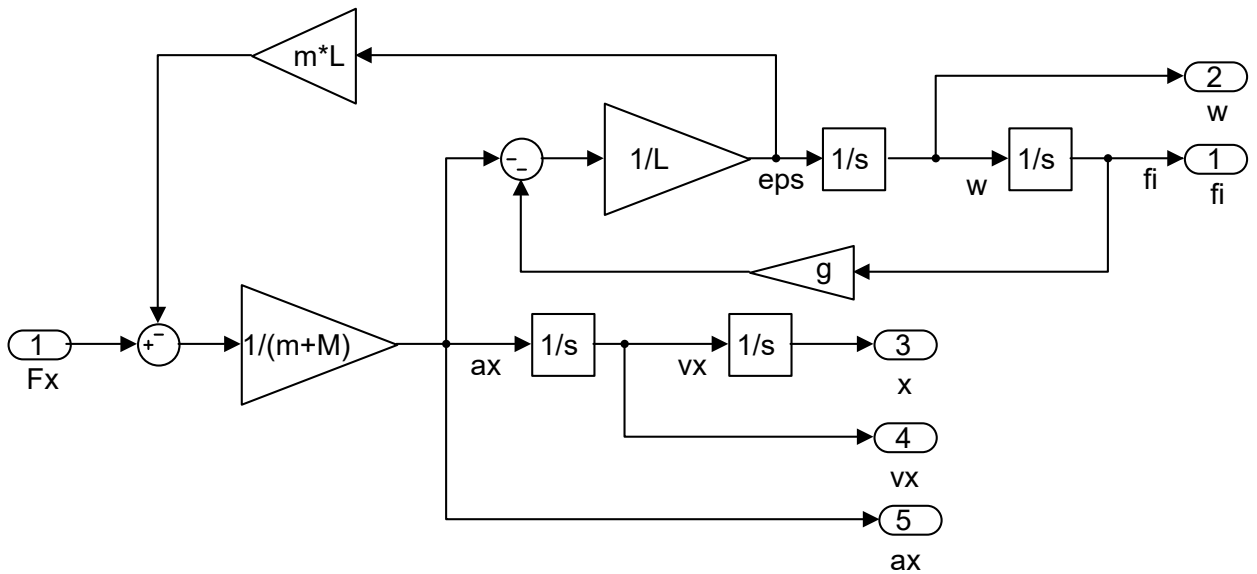


Рис. 15.7. Лінеаризована структурна модель системи візок-вантаж

Із структурної моделі рис. 15.7 знаходимо передавальні функції (ПФ) механічного об'єкта:

$$\frac{\varphi(p)}{a_x(p)} = -\frac{1}{g} \cdot \frac{1}{(T_0^2 p^2 + 1)}, \quad (15.17)$$

де $T_0 = \sqrt{L/g}$ – стала часу власних коливань вантажу;

$$\frac{\varphi(p)}{F_x(p)} = -\frac{1}{(M+m)g} \cdot \frac{1}{(T_{12}^2 p^2 + 1)}, \quad (15.18)$$

де

$$T_{12} = \sqrt{\frac{L}{g} \cdot \frac{M}{M+m}} = T_0 \sqrt{\frac{M}{M+m}}, \quad T_{k12} = 2\pi T_{12} \quad (15.19)$$

– стала часу та період коливань двомасової системи візок-вантаж.

Діленням (15.18) на (15.17) отримуємо:

$$\frac{a_x(p)}{F_x(p)} = \frac{1}{(M+m)} \cdot \frac{(T_0^2 p^2 + 1)}{(T_{12}^2 p^2 + 1)} \quad (15.20)$$

Із ПФ (15.18) знаходимо реакцію кутових координат вантажу на стрибок зусилля $F_x(t) = F_{0x} \cdot 1(t)$ при нульових початкових умовах:

$$\varphi(t) = -\frac{F_{0x}}{(M+m)g} \left(1 - \cos \frac{t}{T_{12}} \right), \quad (15.21)$$

що відповідає коливанням двомасової системи вантаж-візок з періодом T_{k12}

навколо похилої осі $\varphi_{mean} = \frac{F_0}{(M+m)g}$ від $\varphi_{min} = 0$ до

$$\varphi_{max} = -\frac{2F_{0x}}{(M+m)g}. \quad (15.22)$$

Шляхом диференціювання (15.21) знайдемо закон зміни кутової швидкості:

$$\omega(t) = \frac{d\varphi(t)}{dt} = \frac{F_{0x}}{(M+m)gT_{12}} \sin \frac{t}{T_{12}} = \frac{F_{0x}}{\sqrt{Lg(M+m)M}} \sin \frac{t}{T_{12}}, \quad (15.23)$$

звідкіля маємо

$$\omega_{max} = \frac{F_{0x}}{(M+m)gT_{12}} = \frac{\varphi_{max}}{2T_{12}}. \quad (15.24)$$

З передавальної функції (15.17) знаходимо

$$\begin{aligned}
 a(t) &= \frac{F_{0x}}{(M+m)} \left[\left(1 - \cos \frac{t}{T_{12}} \right) + T_0^2 \frac{d^2}{dt^2} \left(1 - \cos \frac{t}{T_{12}} \right) \right] = \\
 &= \frac{F_{0x}}{(M+m)} \left(1 - \cos \frac{t}{T_{12}} + \frac{T_0^2}{T_{12}^2} \cos \frac{t}{T_{12}} \right) = \frac{1}{(M+m)} \left[1 + \left(\frac{T_0^2}{T_{12}^2} - 1 \right) \cos \frac{t}{T} \right],
 \end{aligned}$$

або з врахуванням (15.18)

$$a(t) = \frac{F_{0x}}{(M+m)} \left[1 + \frac{m}{M} \cos \frac{t}{T_{12}} \right]. \quad (15.25),$$

З рівняння (15.25) знаходимо

$$a_{\max} = \frac{F_{0x}}{M}, \quad a_{\min} = \frac{F_{0x}}{M+m} \cdot \frac{M-m}{M} = a_{\max} \frac{M-m}{M+m}. \quad (15.26)$$

Отже, лінійне прискорення коливається навколо середнього значення

$$a_0 = (a_{\max} + a_{\min})/2 = F_{0x}/(M+m) \quad (15.27)$$

з амплітудою

$$A_a = \frac{a_{\max} - a_{\min}}{2} = \frac{F_{0x}m}{2M(M+m)}. \quad (15.28)$$

15.2 Завдання

1) Розрахувати перехідні процеси коливань вантажу при жорсткому зв'язку канату і моста як механічної системи типу «математичний маятник» при заданих ненульових початкових умовах.

2) Розрахувати перехідні процеси рухів візка та вантажу у таких режимах:

- реакція на стрибок зусилля;
- відпрацювання заданого положення за трапецеїдальною тахограмою із заданим прискоренням та швидкістю;
- відпрацювання заданого положення за трапецеїдальною тахограмою, у якій час розгону та час гальмування дорівнюють періоду коливань двомасової системи візок-вантаж.

Параметри систем надано у табл. 15.1.

Розрахунки виконати методами структурного математичного та віртуального фізичного моделювання. Результати порівняти.

15.3 Порядок виконання роботи

1) Зібрати нелінійну та лінеаризовану S -моделі математичного маятника, виконати їх симуляцію і побудувати графіки перехідних процесів.

Таблиця 15.1

№ вар.	m , кг	M , кг	L , м	$v_{\max}$, м/с	$a_{\max}$, м/с ²	φ_0 , град	s , м
1,13	4000	2000	12	0.5	0.5	4, 45	5
2,14	4000	3000	15	1	0.5	8, 60	10
3,15	4000	4000	10	0.75	0.4	7, 30	8
4,16	5	5	18	1	0.8	10, 50	12
5,17	7.5	2.5	9	1	1	12, 55	11
6,18	7.5	2.5	13	0.6	0.5	5, 90	4
7,19	7.5	2.5	8	0.8	0.6	11, 75	7
8,20	7.5	2.5	20	0.8	0.8	6, 40	8
9,21	2.5	7.5	7	2	2	4, 40	16
10,22	2.5	7.5	11	2	1.6	8, 80	16
11,23	2.5	7.5	15	1.8	1.6	7, 70	14
12,24	2.5	7.5	14	1.5	0.9	6, 60	15

2) Розрахувати параметри коливань маятника за формулами (15.5)-(15.7), та порівняти розраховані показники з відповідними показниками перехідних процесів.

3) Ознайомитися з блоками бібліотек *SimMechanics*.

4) Розробити віртуальну фізичну модель математичного маятника та порівняти перехідні процеси, отримані методами структурного математичного та віртуального фізичного моделювання.

5) Розрахувати характерні параметри вхідних сигналів до виконання пункту 2 завдання.

6) Розробити структурну математичну та віртуальну механічну моделі кранової механічної системи візок-вантаж, промоделювати процеси пересування візка та коливань вантажу у заданих режимах, порівняти параметри

отриманих перехідних процесів з розрахованими при виконанні попереднього пункту.

7) Проаналізуйте результати досліджень. Сформулюйте висновки.

15.4 Методичні вказівки

1) Якщо у блоці *IC* моделі рис. 15.2 встановлено початковий ненульовий кут відхилення нитки від вертикалі, то позиція центру тяжіння (*CG*) та кінця нитки, прикріпленої до вантажу задається у рівноважному стані маятника ($[0 \ -L \ 0]$); так само визначається і їх орієнтація у просторі – ($[0 \ 0 \ 0]$); якщо викинути блок *IC*, то позицію центру тяжіння слід задавати у вигляді $[L \sin \varphi_0 \ -L \cos \varphi_0, 0]$, а просторову орієнтацію – $[0 \ 0, \ \varphi_0]$; при фіксації кутових координат початковий кут відхилення треба додати до вимірюваного кута за допомогою *Simulink*-блоку *Sum*.

2) Моменти інерції треба задавати відносно осей, що перетинаються між собою у центрі гравітації; у нашому випадку (для точкової маси) це буде матриця з нульовими елементами: $\mathbf{J}_{CG} = \text{zeros}(3,3)$. Момент інерції відносно осей, паралельних осям, що проходять через центр гравітації, але зсунутих від центру гравітації на відстань d розраховується за формулою Гюйгенса-Штейнера $\mathbf{J}_{Gxyz} = \mathbf{J}_{CGxyz} + m \cdot \text{diag}([d_x^2, d_y^2, d_z^2])$ вочевидь самою системою MATLAB;

3) При виконанні пункту 1 та двох останніх пунктів пункту 2 завдання увімкніть анімацію.

15.5 Контрольні питання та завдання

1) Які проблеми виникають при пересуванні вантажів крановими механізмами?

2) Складіть математичний опис ідеального маятника.

3) За яких умов і як можна його лінеаризувати? Як розрахувати період коливань математичного маятника з малою амплітудою?

- 4) Як розрахувати період коливань механічної системи візок-вантаж.
- 5) Як можна позбавитися від коливань вантажу при русі візка з установленою швидкістю та після його зупинки?
- 6) Як увімкнути анімацію при застосуванні у моделі блоків бібліотек *SimMechanics*?

ЛІТЕРАТУРА

1. Островерхов М.Я., Пижов В.М. Моделирование электромеханических систем в Simulink: Навч. Посібник для студентів вищих навчальних закладів. – К.: ВД «Стилос», 2008. – 528 с.
2. Лазарев Ю.Ф. МАТЛАБ 5.x. – К.: Издательская группа BHV, 2000. – 384с. (Серия “Библиотека студента”)
3. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения / Дьяконов В.П. – М.: СОЛОН-Пресс, 2004. – 768 с.
4. Дьяконов В.П. Simulink 4. Специальный справочник. – СПб.: Питер, 2001. – 528с.
5. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в МАТЛАБ 6.0: Учебное пособие. – СПб.: КОРОНА принт, 2001. – 320 с.
6. Герман-Галкин С.Г. Линейные электрические цепи. Лабораторные работы на ПК. – СПб.: КОРОНА принт, 2002. – 192 с.
7. Черных И.В. Simulink: среда создания инженерных приложений. – Диалог-МИФИ, 2003. – 496 с.
8. Новгородцев А.Б. Расчет электрических цепей в MATLAB: Учебный курс. – СПб.: Питер, 2004. – 250 с.
9. SimMechanics 2. User's Guide. Matlab & Simulink. COPYRIGHT 2001–2007 by The MathWorks, Inc.
10. Мусалимов В.М., Г.Б. Заморуев, И.И. Калапышина, А.Д. Перечесова, К.А. Нуждин. Моделирование мехатронных систем в среде MATLAB (Simulink / SimMechanics): учебное пособие для высших учебных заведений. – СПб: НИУ ИТМО, 2013. – 114 с.
11. Щербаков В.С., Корытов М.С., Руппель А.А., Глушеч В.А., Милушенко С.А. Моделирование и визуализация движений механических систем в MATLAB / Учебное пособие. – Омск: издательство СибАДИ, 2007. – 84 с.

12. Медведев В.С., Потемкин В.Г. Control System Toolbox. МАТЛАБ 5 для студентов / Под общ. ред. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 1999. – 287с.
13. Избранные главы теории автоматического управления с примерами на языке MATLAB / Б.Р. Андриевский, А.Л. Фрадков. – СПб.: Наука, 2000. – 475 с.
14. Никульчев Е.В. Практикум по теории управления в среде MATLAB: Учебное пособие. М., 2004, 87 с.
15. Solving control engineering problems with MATLAB (MATLAB curriculum series) / K. Ogata. – Englewood Cliffs, NJ: Prentice Hall, 1994. – 359 p.
16. Dingyü Xue, YangQuan Chen, Derek P. Atherton. Linear Feedback Control Analysis and Design with MATLAB. – Philadelphia: Siam. – 2007, 356 P.
17. Черный А.П. и др. Моделирование электромеханических систем / Учебное пособие. – Кременчуг: КГПИ, 1999. – 204 с.
18. Шрейнер Р.Т. Системы подчиненного регулирования электроприводов. Часть 1. Электроприводы постоянного тока с подчиненным регулированием координат: Учеб. Пособие для вузов. – Екатеринбург: Изд-во Урал. гос. проф.-пед. ун-та, 1997. – 279 с.
19. Толочко О.І. Моделювання електромеханічних систем. Математичне моделювання систем асинхронного електроприводу: Навчальний посібник. – Київ: НТУ «КПІ», 2016. – 150 с.
20. Башарин А.В. Новиков В.А., Соколовский Г.Г. Управление электроприводами. – Л.: Энергоиздат. – 1982, 392 с.
21. Толочко О.І., Стяжкін В.П., Рижков О.П. Дослідження процесів руху механізмів мостового крану при їх одночасній роботі методом математичного моделювання з використанням віртуальних механічних блоків бібліотеки Simmechanics пакету Matlab // Вісник НТУ «ХПІ». Проблеми автоматизованого електропривода. – Харків: НТУ «ХПІ», 2017, вип. 27 (1249). – С.48-52

ЗМІСТ

Введення	3
<i>Лабораторна робота №1. Знайомство з середовищем програми</i>	
структурного моделювання <i>Simulink</i> системи програмування <i>Matlab</i> ...	5
1.1 Запуск <i>Simulink</i> . Перелік бібліотек та демонстрацій	5
1.2 Меню вікна <i>Simulink Library Browser</i>	16
1.3 Меню вікон <i>Simulink</i> -моделей.....	19
1.4 Меню вікон <i>Simulink</i> -бібліотек	27
1.5 Типи <i>Simulink</i> -блоків.....	27
1.6 Створення та редагування моделей	29
1.7 Завдання	33
1.8 Контрольні питання та завдання	33
<i>Лабораторна робота №2. Формування вхідних сигналів та реєстрація</i>	
вихідних сигналів в середовищі <i>Simulink</i>	35
2.1 Основні засоби формування вхідних сигналів	35
2.2 Основні засоби реєстрації та візуалізації сигналів.....	50
2.3 Завдання	58
2.4 Порядок виконання роботи.....	59
2.5 Методичні вказівки та рекомендації	61
2.6 Контрольні питання та завдання	63
<i>Лабораторна робота №3. Знайомство з бібліотечними неперервними</i>	
ланками програми <i>Simulink</i>	64
3.1 Теоретичні відомості.....	64
3.2 Відомості про блоки бібліотеки <i>Continues</i>	68
3.3 Відомості про лінійні арифметичні блоки бібліотеки <i>Math</i>	75
3.4 Завдання	76
3.5 Порядок виконання роботи	76
3.6 Методичні вказівки та рекомендації	76
3.7 Контрольні питання та завдання	77

<i>Лабораторна робота №4. Знайомство з бібліотечними нелінійними блоками програми Simulink</i>	80
4.1 Теоретичні відомості.....	80
4.2 Нелінійні блоки програми <i>Simulink</i>	81
4.3 Порядок виконання роботи	88
4.4 Завдання	88
4.5 Методичні вказівки та рекомендації	90
4.6 Контрольні питання та завдання	90
<i>Лабораторна робота №5. Структурне моделювання електричних кіл.....</i>	91
5.1 Методика структурного математичного моделювання.....	91
5.2 Приклад структурного математичного моделювання лінійного електричного кола	93
5.3 Приклад структурного математичного моделювання нелінійного електричного кола	96
5.4 Завдання	98
5.5 Порядок виконання роботи	101
5.6 Методичні вказівки та рекомендації	102
5.7 Контрольні питання та завдання	102
<i>Лабораторна робота №6. Структурне математичне моделювання двигуна постійного струму з керуванням у колі якоря</i>	103
6.1 Математичний опис об'єкту моделювання.....	103
6.2 Завдання	105
6.3 Порядок виконання роботи	107
6.4 Методичні вказівки та рекомендації	107
6.5 Контрольні питання та завдання	109
<i>Лабораторна робота №7. Структурне математичне моделювання двигуна постійного струму з керуванням у колі якоря та колі збудження.....</i>	110
7.1 Математичний опис об'єкту моделювання.....	110
7.2 Завдання	112
7.3 Порядок виконання роботи	113

7.4 Методичні вказівки та рекомендації	114
7.5 Контрольні питання та завдання	114
<i>Лабораторна робота №8. Створення підсистем та їх маскування</i>	<i>116</i>
8.1 Теоретичні відомості.....	116
8.1.1 Створення підсистем	116
8.1.2 Маскування підсистем	117
8.2 Створення кнопок, що керують виконанням модельного експерименту	125
8.3 Завдання	128
8.4 Порядок виконання роботи	129
8.4 Методичні вказівки та рекомендації	129
8.5 Контрольні питання та завдання	133
<i>Лабораторна робота №9. Еквівалентні перетворення лінійних структурних схем у середовищі MATLAB</i>	<i>134</i>
9.1 Теоретичні відомості.....	134
9.2 Еквівалентні перетворення лінійних динамічних систем в середовищі MATLAB у символьному вигляді	135
9.3 Знаходження результуючих передавальних функцій лінійних динамічних систем у чисельному вигляді	137
9.4 Завдання	138
9.5 Порядок виконання роботи	141
9.6 Методичні вказівки та рекомендації	141
9.7 Контрольні питання та завдання	142
<i>Лабораторна робота №10. Аналіз лінійних динамічних систем у середовищі пакета MATLAB</i>	<i>143</i>
10.1 Теоретичні відомості.....	144
10.1.1 Характеристика показників якості перехідних процесів	144
10.1.2 Частотний аналіз	148
10.1.3 Аналіз лінійних динамічних систем за розподіленням полюсів та нулів	153

10.2 Завдання	157
10.3 Порядок виконання роботи	158
10.4 Методичні рекомендації	158
10.6 Контрольні питання та завдання	158
<i>Лабораторна робота №11. Віртуальне фізичне моделювання електричних кіл з використанням блоків бібліотек SimPowerSystem</i>	
11.1 Основні відомості про бібліотеку SimPowerSystem та правила її застосування	159
11.2 Характеристика основних блоків для моделювання кіл постійного струму та однофазних кіл змінного струму	164
11.2.1 Джерела електричної енергії	164
11.2.2 Електротехнічні елементи	166
11.2.3 Основні вимірювальні прилади	166
11.2.4 Ключові елементи	170
11.2.5 Графічний інтерфейс користувача Powergui	173
11.3 Приклад віртуального фізичного моделювання лінійного розгалуженого електричного кола	177
11.4 Завдання	179
11.5 Порядок виконання роботи	179
11.6 Методичні вказівки та рекомендації	179
11.7 Контрольні питання та завдання	179
<i>Лабораторна робота №12. Віртуальне фізичне моделювання електричного двигуна постійного струму з незалежним збудженням</i>	
12.1 SPS-модель двигуна постійного струму	181
12.2 Завдання	185
12.3 Порядок виконання роботи	185
12.4 Методичні вказівки та рекомендації	185
12.5 Контрольні питання та завдання	186
<i>Лабораторна робота №13. Віртуальне фізичне моделювання асинхронного двигуна</i>	
	188

13.1 SPS-моделі асинхронних двигунів	188
13.2 Завдання	195
13.3 Порядок виконання роботи	195
13.4 Методичні вказівки та рекомендації	196
13.5 Контрольні питання та завдання	196
<i>Лабораторна робота №14. Структурне математичне та віртуальне фізичне моделювання багатомасових механічної систем</i>	<i>198</i>
14.1 Математичний опис жорстких механічних об'єктів з одним ступенем свободи	198
14.2. Математичний опис пружно-в'язких механічних об'єктів з одним ступенем свободи	199
14.3 Віртуальне фізичне моделювання механічних об'єктів у середовищі <i>MATLAB-Simulink-SimMechanics</i>	202
14.4 Завдання	212
14.5 Порядок виконання роботи	214
14.6 Методичні вказівки та рекомендації	214
14.7 Контрольні питання та завдання	215
<i>Лабораторна робота №15. Віртуальне фізичне моделювання кранових механізмів</i>	<i>216</i>
15.1 Теоретичні відомості.....	216
15.2 Завдання	225
15.3 Методичні вказівки та рекомендації	227
15.4 Контрольні питання та завдання	227
Література	229