

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**МОДЕЛЮВАННЯ СИСТЕМ  
АВТОМАТИЧНОГО КЕРУВАННЯ**

**ЛАБОРАТОРНИЙ ПРАКТИКУМ**

**Навчальний посібник**

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня бакалавра  
за освітньою програмою «Електромеханічні системи автоматизації, електропривод  
та електромобільність»  
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»,*

Електронне мережне навчальне видання

Укладач О.І. Толочко

Київ  
КПІ ім. Ігоря Сікорського  
2023

Рецензент: *Островерхов М.Я.*, доктор технічних наук, професор,  
завідувач кафедри теоретичної електротехніки

Відповідальний редактор: *Ковбаса С.М.*, доктор технічних наук, доцент, завідувач ка-  
федри автоматизації електромеханічних систем та електро-  
приводу

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського  
(протокол № 9 від 22.06.2023 р.)  
за поданням факультету електроенерготехніки та автоматики  
(протокол №11 від 29.05.2023 р.)*

Моделювання систем автоматичного керування. Методичні вказівки до лабораторних ро-  
біт [Електронний ресурс]: навч. посіб. для студентів спеціальності 141 «Електроенергети-  
ка, електротехніка та електромеханіка» з дисципліни «Моделювання систем автоматично-  
го керування» / О. І. Толочко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані  
(1 файл: 41928 кБ). – Київ: КПІ ім. Ігоря Сікорського, 2023. – 250 с.

Навчальний посібник присвячено питанням структурного математичного і  
імітаційного фізичного моделювання та аналізу електротехнічних, механічних і  
електромеханічних об'єктів у середовищі пакету *MATLAB*. Розглянуті питання  
математичного опису та побудови структурних моделей лінійних та нелінійних  
стаціонарних динамічних систем. Пояснені принципи переходу від математичного опису  
до побудови структурних схем. Викладено методику створення, редагування та  
експлуатації структурних математичних моделей в середовищі *MATLAB-Simulink*.  
Наведено відомості про основні блоки бібліотек *Simulink*. Викладено основні прийоми  
імітаційного фізичного моделювання електротехнічних та електромеханічних систем в  
середовищі *MATLAB-Simulink-SimPowerSystems*. У якості прикладів розглянуто електричні  
кола, електричні машини, механізми з пружно-в'язкими кінематичними зв'язками.

Посібник отримує методичні вказівки до 15 лабораторних робіт. Вказівки до  
кожного заняття складаються з теоретичних положень, завдання та методичних  
рекомендацій щодо його виконання.

Посібник призначений для використання здобувачами ступіня бакалавра за  
освітньою програмою "Електромеханічні системи автоматизації, електропривод та  
електромобільність" спеціальності 141 «Електроенергетика, електротехніка та  
електромеханіка» при вивченні дисципліни «Моделювання систем автоматичного  
керування» та можуть бути використаними при виконанні бакалаврських та магістерських  
дипломних робіт.

Реєстр. № **НП XX/XX-XXX**. Обсяг 11,7 авт. арк.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
проспект Перемоги, 37, м. Київ, 03056  
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів  
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

## 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 керування (САК) 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 імітаційне фізичне моделювання, яке не потребує від користувача знання математичного опису досліджуваного процесу, потребує менше часу для налаштування, є більш наочним, має розвинені засоби візуалізації та анімації результатів досліджень. Такі моделі можна створювати і в *Simulink* шляхом використання віртуальних електротехнічних та електромеханічних блоків бібліотек *SimPowerSystems (SPS)*. 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]. Математичний опис окремих елементів цих бібліотек приховано від користувача, завдяки чому створюється ілюзія (імітація) фізичного моделювання, тобто застосування реальних експериментальних установок.

Щоб електромеханічні, електротехнічні та енергетичні об'єкти працювали за бажаними законами та задовольняли поставлених до них вимогам, до них додаються пристрої вимірювання фізичних величин (датчики) та пристрої керування (регулятори, задавальні пристрої, пристрої ідентифікації тощо). Разом вони утворюють системи автоматичного керування координатами об'єктів. Пристрої керування можуть бути як аналоговими, так і цифровими. В останній час більш вживаними є цифрове керування, що реалізується на базі сучасних мікроконтролерів.

Отже, метою даного лабораторного практикуму є придбання практичних навичок структурного математичного та імітаційного фізичного моделювання неперервних (аналогових), дискретних (цифрових) та цифро-аналогових динамічних систем автоматичного керування. У якості реальних 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 вже 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-R13b*. Але більшість матеріалів може бути використано і при роботі з більш пізніми версіями. При інсталяції більш сучасних версій треба враховувати, що вони потребують і більш досконалих комп'ютерів, а також те, що студентам прийдеться самостійно розбиратися з інтерфейсом та деякими відмінностями від попередніх версій.

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

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

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.

**Звіт з кожної лабораторної роботи** має містити наступні матеріали:

- 1) титульний аркуш;
- 2) мета роботи;
- 3) завдання згідно з варіантом;
- 4) схеми моделей з параметрами блоків;
- 5) тексти програмних модулів;
- 6) результати досліджень;
- 7) висновки по роботі;
- 8) відповіді на контрольні питання.

**На титульному аркуші звітів з лабораторних робіт** має бути відображеною назва університету, назва дисципліни, номер та назва лабораторної роботи, шифр групи, ПІБ виконавця, ПІБ та регалії викладача.

**Відповіді на контрольні запитання і звіти з лабораторних робіт** подаються на аркушах паперу формату А4 з урахуванням **вимог ЄСКД**, а саме:

- текстова частина набирається у середовищі *Word* шрифтом *Times New Roman*, розміром 14 пт. з вирівнюванням по ширині, відстань між рядками – 1,5 інтервали, абзац – 1,25 см, розмір полів: зверху та знизу 2 см, справа – 2,5 см, зліва – 1 см; виключенням є тексти програм і результати їх виконання, для відображення яких бажано застосовувати будь-який інший шрифт розміром 11-12 пт. з відстанню між рядками 1 інтервал;
- усі рисунки (у тому числі графіки) повинні мати нумеровані підрисуночні підписи у форматі «Рисунок <номер> – <Назва>»; «крапка» після підрисуночного підпису не ставиться (це стосується і будь-яких заголовків)
- формули набираються шрифтом *Times New Roman*, розміром 14 пт. у середовищі *Equation 4.0* або *MathType* (треба попередньо установити), які відкриваються за допомогою команд меню *Word: Вставка → Об'єкт → Microsoft Equation 4.0 / MathType Equation* або *MathType → Inline* (після інсталяції *MathType* продукт *Microsoft Equation 4.0* стає недоступним); самі формули розташовують або посередині рядка, або з абзацу; якщо після фор-

мули речення продовжується, то його продовжують без абзацного відступу;

- графічні вікна пакету *MATLAB* краще копіювати не методом *ScreenShort*, а через меню цих вікон *Edit* → *Copy Figure*; для обрізки отриманого графіку та його редагування можна скористатися графічним редактором *Microsoft Visio*;
- при посиланні на рисунок та на таблиці у тексті слова «рисунок» та «таблиця» скорочують, наприклад, «на рис. 2 зображено ...», «у табл. 3 наведено ...».

Звіти, програмні файли та моделі, створені при виконанні лабораторних робіт зберігаються протягом семестру. Викладач може під час занять та контрольних заходів попросити студента показати, як працюють його файли і надати до них пояснення.

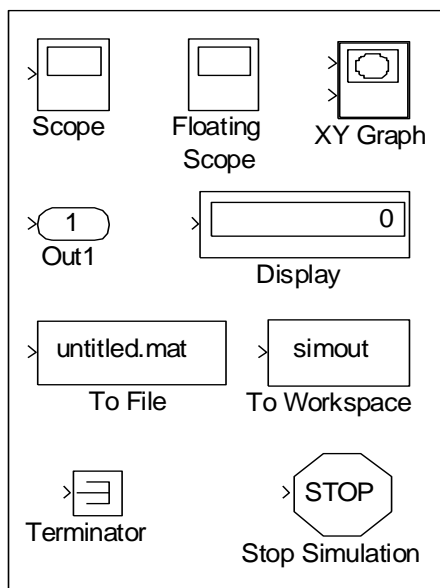
## Лабораторна робота №1

### РЕЄСТРАЦІЯ ТА ВІЗУАЛІЗАЦІЯ СИГНАЛІВ

*Мета роботи:* навчитися користуватися блоками реєстрації та візуалізації сигналів в середовищі *Simulink* та використовувати отриману інформацію при оформленні звітів та інших документів.

#### 1.1 Основні засоби реєстрації та візуалізації сигналів

Для реєстрації та візуалізації сигналів у *Simulink* існує бібліотека *Sinks*,



блоки якої подані на рис. 1.1.

Більшість з них призначені для запам'ятання 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 ***Terminator*** (*Заглушка*) приєднується до виходів блоків, не зв'язаних з іншими блоками для запобігання виведення попереджувального повідомлення (*Warning*) про наявність у моделі не приєднаних виходів.

Рис. 1.1 – Бібліотека вихідних блоків *Sinks*

Розглянемо параметри, які є спільними для декількох блоків бібліотеки *Sinks*.

**Параметр Decimation (Проріджування)** у блоках *Scope*, *To Workspace*, *Display* 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, отри-



мана на кожному п'ятому кроці. При інтегруванні з постійним кроком параметр  $d$  – це відношення кроку фіксації до кроку інтегрування:  $h_{out} = dh_{int}$ .

**Параметр *Limit data point to last*** у блоках *Scope*, *To Workspace*, *Out* визначає максимальну кількість точок для збереження сигналу (відлік ведеться від останньої розрахованої точки). Для того, щоб не втратити інформацію, значення параметра *Limit data point to last* можна установити рівним  $\infty$  (константа *Inf*) або прибрати прапорець, що робить цю опцію активною.

**Параметр *Save format*** у блоках *Scope*, *To Workspace*, *Out* визначає формат запам'ятовування даних.

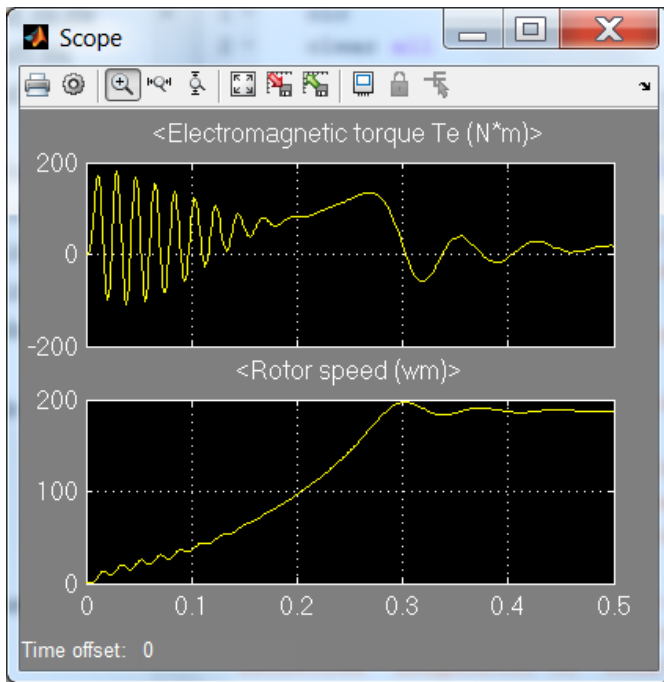
Інформація може бути збережена в наступних форматах: *Structure with time* (Структура з часом моделювання), *Structure* (Структура без часу моделювання) і *Array* (Масив).

Найпростішим типом даних є *Array*. У цьому випадку кожен сигнал записується в окремий стовбець 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  $y(i,:)$ .

### 1.1.1 Блоки візуалізації

Блок *Scope* (**Осцилограф**) у процесі моделювання відображає вихідні сигнали приєднаних до нього блоків. Для того, щоб побачити графіки перехідних процесів, необхідно відкрити його вікно (див. рис. 1.2), що поряд з полем виведення графіків містить такі «кнопки» (рис. 1.3):

- 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);

Рис. 1.2. Вікно блока *Scope*Рис. 1.4. Кнопки вікна блоку *Scope*

12 – *Dock Scope* (осцилограф виводить графіки без кнопкової панелі).

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 кожнім натисканні масштаб буде

7 – *Save current axes settings* (запам'ятовування системи координат);

8 – *Restore saved axes settings* (відновлення системи координат, наприклад, після зміни масштабу однією з клавіш *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);

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*), змінивши 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* і *Data history*.

За допомогою вкладки *General* (див. рис. 1.4) установлюються наступні параметри:

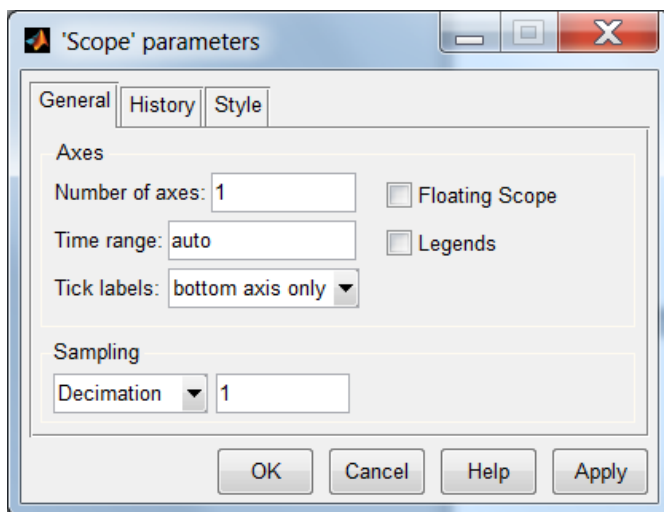


Рис. 1.4. Вкладка *General* вікна параметрів блоку *Scope*

- *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);

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

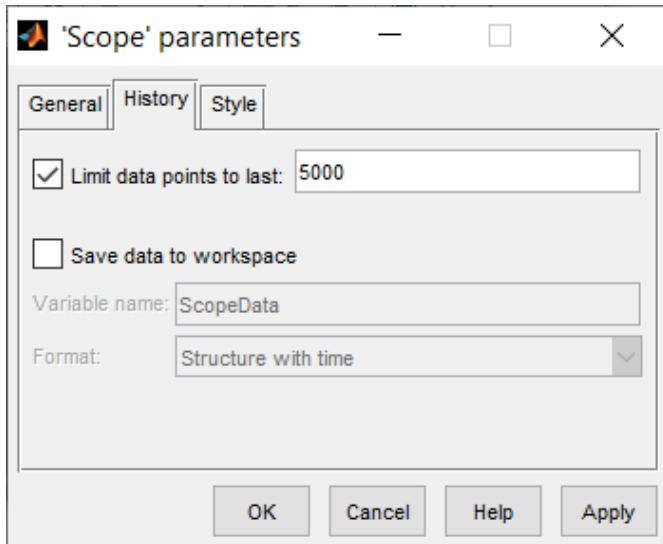


Рис. 1.5. Вкладка *History* вікна

За допомогою **вкладки *History*** (див. рис. 4.5)значаються параметри *Limit data point to last*, *Variable Name* і *Save format*. Два останніх параметри 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 застосувати оператор

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

Формат *Array* не можна використовувати для збереження інформації багатоканальним *Осцилографом*. За допомогою **вкладки *Style*** (рис. 1.6)значаються такі параметри:

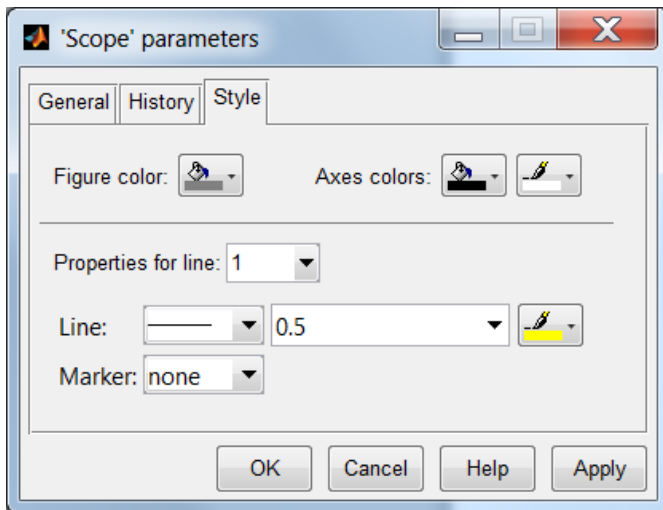


Рис. 1.6. Вкладка *Style* вікна параметрів блоку *Scope*

- *Figure Colour* – колір графічної фігури;
- *Axes Colour* – колір системи координат (колір фону і колір ліній градування);
- *Line* – стиль зображення ліній, їх товщина та колір;
- *Marker* – стиль зображення 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.

Скалярний сигнал 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*.

Приклади використання блоку *Scope* для візуалізації двох синусоїдальних сигналів різної амплітуди і частоти показано на рис. 1.7.

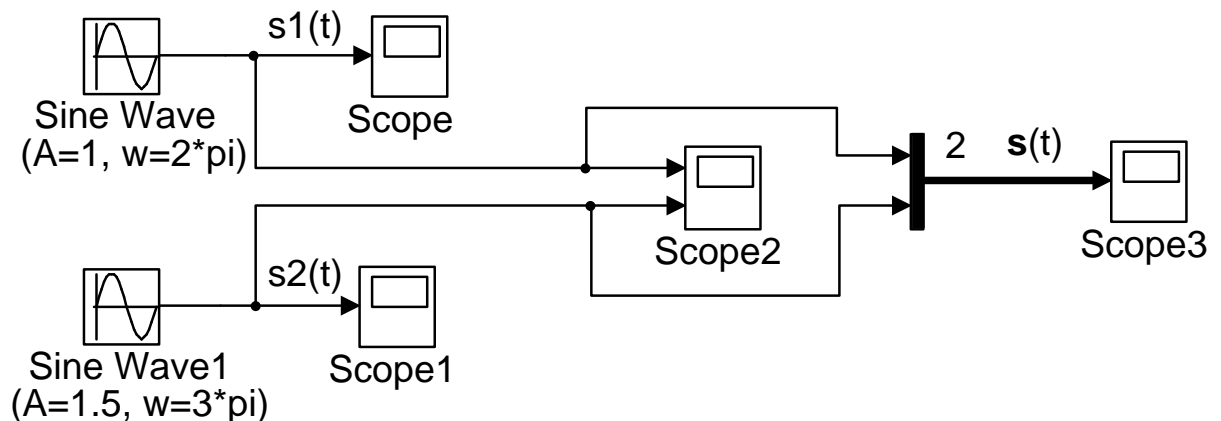


Рис. 1.7. Модель візуалізації двох сигналів блоками *Scope*

Вигляд відкритих після симуляції осцилографів показано на рис. 1.8.

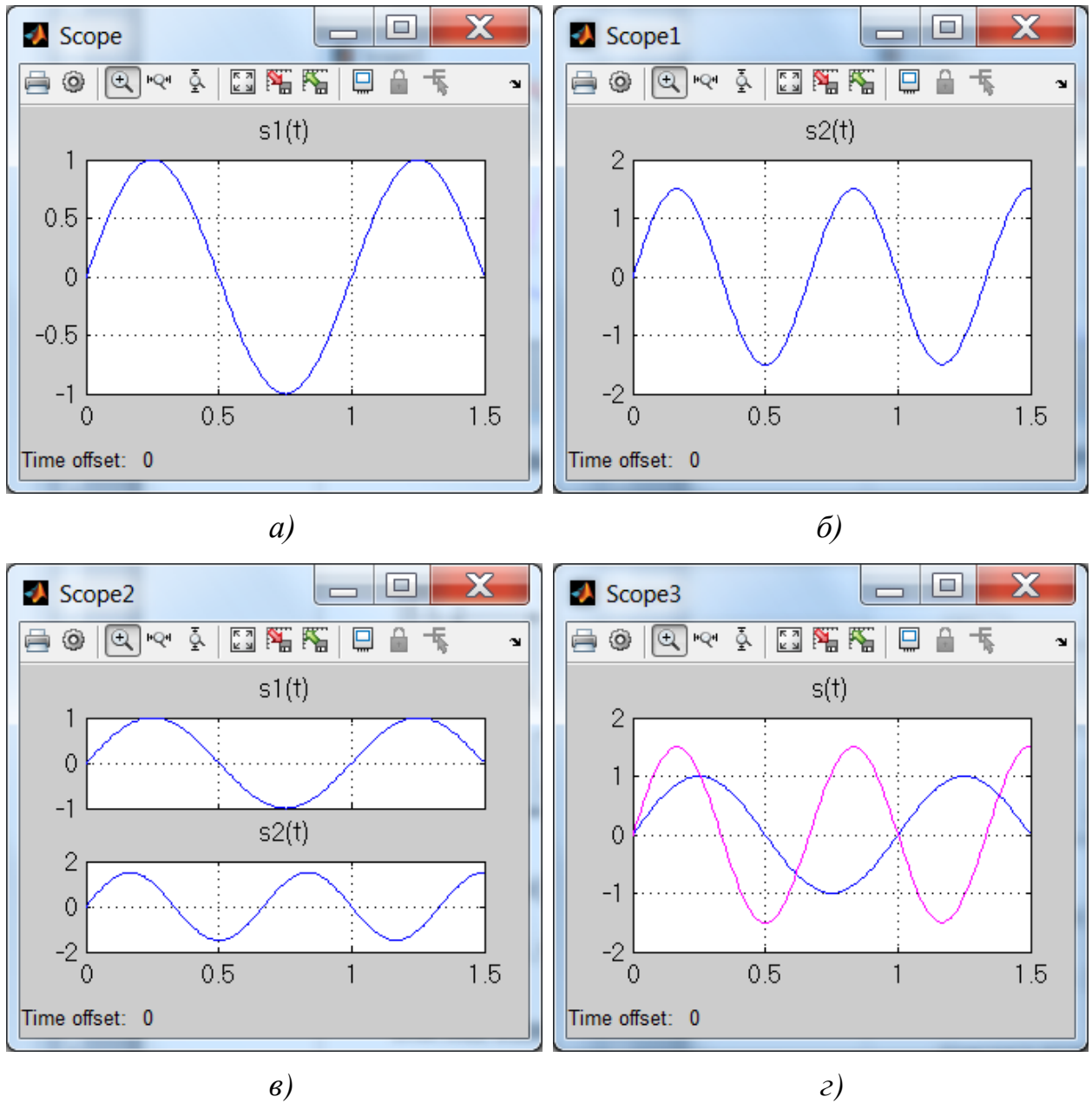


Рис. 1.8. Результати візуалізації синусоїд блоками *Scope* (а), *Scope1* (б), *Scope2* (в), *Scope3* (г)

Блоки *Scope* та *Scope1* відображують синусоїди у різних вікнах, а блоки *Scope1* та *Scope3* – в одному вікні. Різниця між останніми полягає в тому, що *Scope2* відкриває 2 системи координат, кожна у своєму підвікні, що досягається установленням у полі параметру *Number of axes* вкладки *General* вікна *Parameters* значення 2, а *Scope3* зображує обидва сигнали, попередньо

об'єднані блоком *Mux* (Мультиплексор) в один векторний сигнал, в одній системі координат.

**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 вхідних портів. Підключення до нього реєстрованих сигналів можна виконати у вікні *Signal Selector*, що відкривається кнопкою 11 (див. рис. 1.1). *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).**

Єдиною перевагою плаваючого осцилографа є економія оперативної пам'яті.

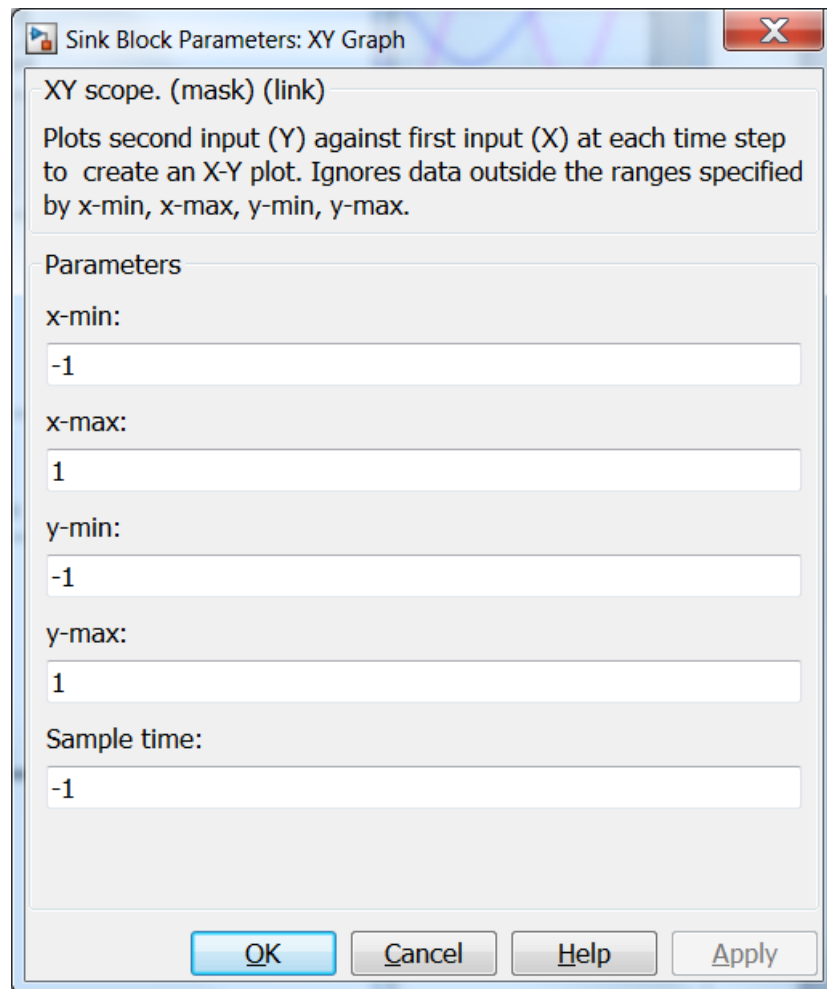
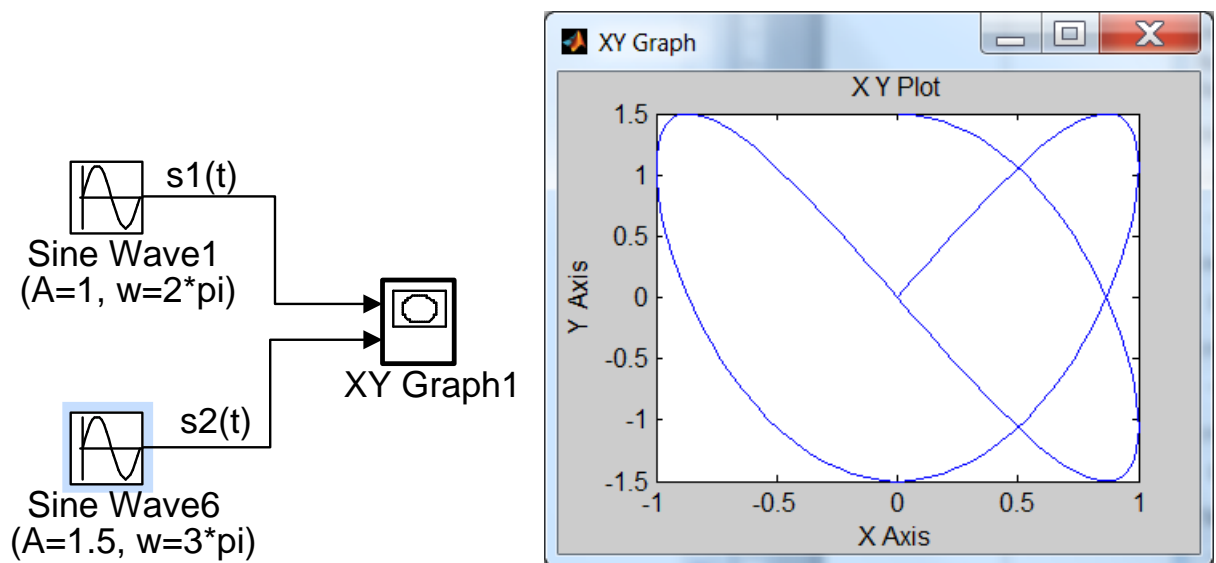
**Блок *XY Graph* (XY-графобудівник)** призначено для побудови фазових портретів, тобто для зображення одного зі збережених у пам'яті сигналів у функції іншого (а не у функції часу). Його вхідними параметрами є координати меж графіка : *x-min*, *x-max*, *y-min* і *y-max*, а також параметр дискретизації *Sample time* (див. рис. 1.9).

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

Приклад та результат використання блоку *XY Graph* показано на рис. 1.10.

До перегляду графіка треба встановити діапазони зміни його параметрів у вікні рис. 1.9.

Більш якісні графіки залежності одного сигналу від іншого можна побудувати оператором *plot* за результатами вимірювання сигналів, як це буде показано на рис. 1.14

Рис. 1.9. Вікно параметрів блоку *XY Graph*Рис. 1.10. Приклад використання блоку *XY Graph*



Блок *Display (Дисплей)* застосовується для виведення чисельних значень сигналів у заданому форматі, що обирається 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.

Приклад використання блоку наведено на рис. 1.11.

Оскільки зазвичай моделювання здійснюється дуже швидко і з малими проміжками часу між розрахунками, то користувач може побачити тільки останнє значення змінної після закінчення симуляції.

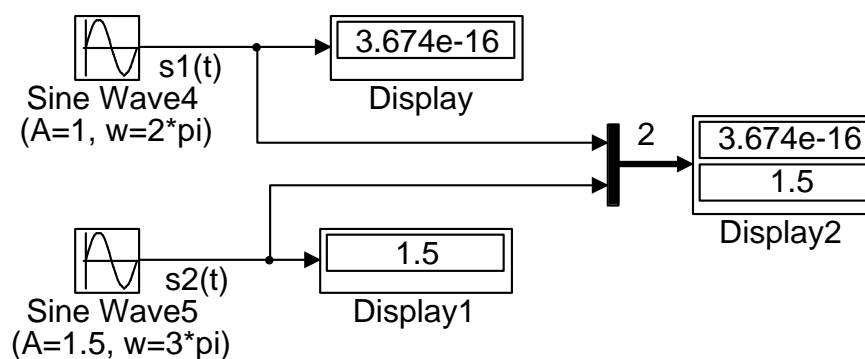


Рис. 1.11. Приклад використання блоку *Display*

### 1.1.2 Блоки запам'ятовування сигналів

Блоками *Out (Вихідний Порт)* позначають виходи моделей і підсистем.

У підсистемах вони необхідні для зв'язку підсистеми з моделлю більш високого рівня, а в моделях вищого рівня – для запам'ятовування приєднаних до них сигналів з метою подальшого використання для побудови графіків або для аналізу чисельних даних.

Для виконання останньої операції необхідно відкрити через функцію *Simulation* вікно *Model Configuration Parameters*, вибрати в ньому вкладку *Data Import/Export* та в розділі *Save to workspace* встановити прапорці в полях *Output* та *Time* і визначити імена змінних, у яких зберігатиметься інформація. За замовчанням у полі *Output* прописано ім'я *yout*, а у полі *Time* – ім'я *tout*, які, за бажанням користувача, можна змінити на будь-які інші (рис. 1.12).

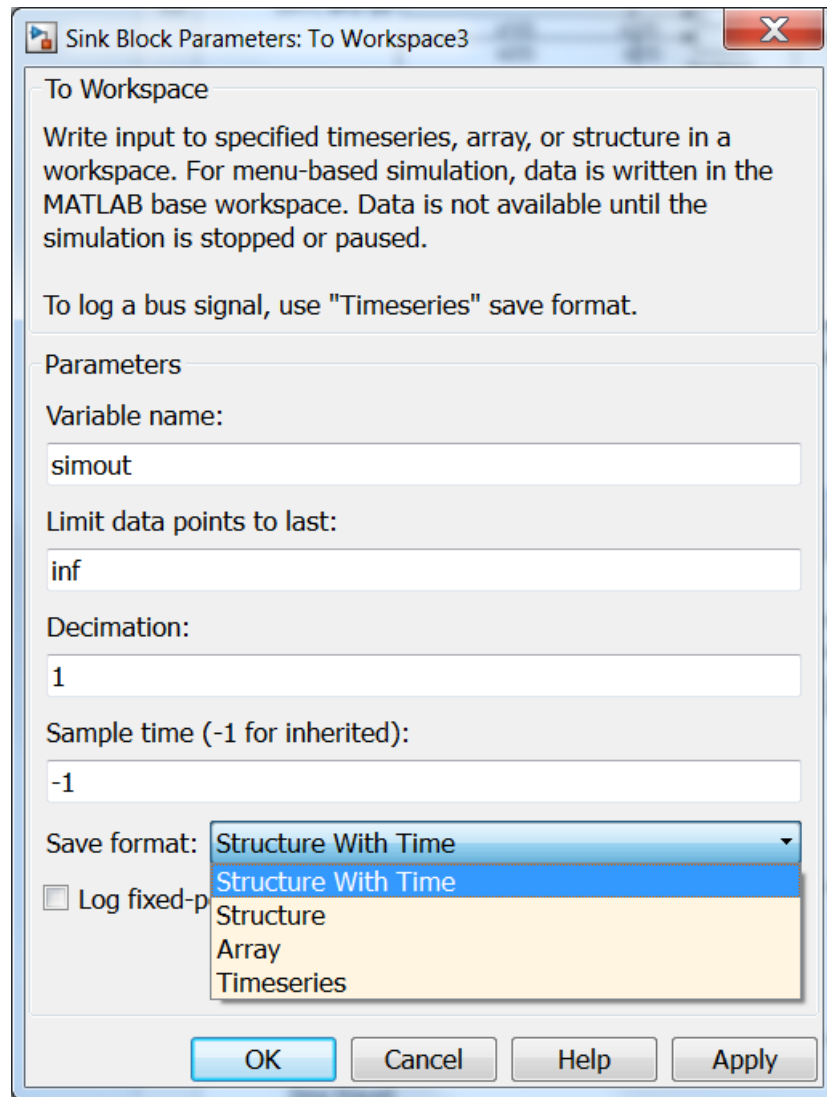
Цей блок зручно використовувати для фіксації векторних сигналів.

Блок *To Workspace (У Пам'ять)* запам'ятовує дані, що надходять на його вхідний порт у перемінній з ім'ям, завданням у поле параметра *Variable name*. Його зручно застосовувати для фіксації змінних з різними іменами.

Для запису інформації у форматі *Array (Масив)* треба не забути змінити значення за замовченням параметру *Save Format* (див. рис. 1.12).

Усякий раз *при старті процесу моделювання дані, що зберігаються, обновляються*, так що цей блок не може об'єднати результати декількох послідовних розрахунків. Результати не доступні в робочому середовищі доти, поки не закінчено або не зупинено розрахунок перехідних процесів.

Як бачимо, за замовченням параметр *Variable Name* має значення *Simout*.

Рис. 1.12. Вікно параметрів блоку *To Workspace*

**Реєстратор To File (У Файл)** записує часи моделювання і вхідні сигнали у файл з ім'ям *File Name* та розширенням *mat (\*.mat)*, який має структуру матриці, що містить у першому рядку монотонно зростаючий вектор часу, а в інших рядках відповідні йому вектори вихідних сигналів:

$$\begin{bmatrix} t_1 & t_2 & \dots & t_k \\ y_1 & y_2 & \dots & y_k \\ z_1 & z_2 & \dots & z_k \\ \dots & \dots & \dots & \dots \end{bmatrix}. \quad (4.1)$$

Ім'я матриці задається параметром *Variable name*.

Цей блок 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*.

На рис. 1.13 наведено приклад використання блоків *Out, To Workspace* та *To File* для реєстрації вихідних сигналів двох синусоїдальних сигналів, застосованих вище у моделях рис. 1.7, 1.10 та 1.11.

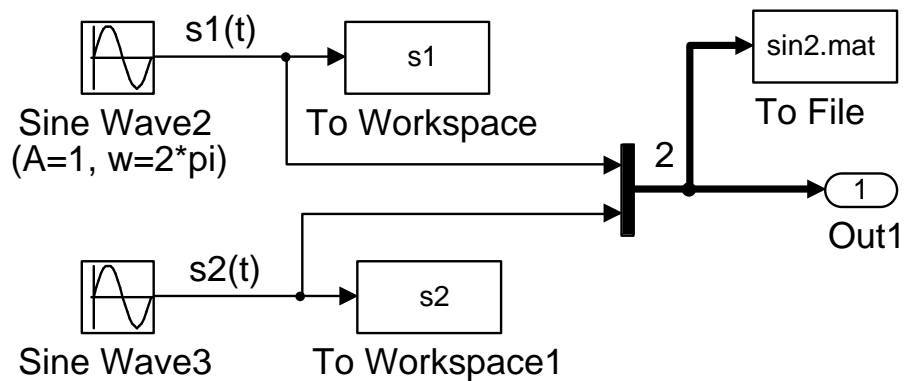


Рис. 1.14. Приклад реєстрації сигналів блоками *Out, To Workspace* та *To File*

Для того, щоб отримати графіки, аналогічні графікам, побудованим за допомогою блоків *Scope* (рис. 1.8) та *XY Graph* (рис. 1.10), використовуючи інформацію, записану у змінні `tout`, `yout`, з застосуванням блоку *Out*, треба виконати таку послідовність операторів алгоритмічної мови *MATLAB*:

```
figure (1), plot (tout, yout (:,1)), title ('s1(t)'), grid on
figure (2), plot (tout, yout (:,2)), title ('s2(t)'), grid on
figure (3), subplot (2,1,1), plot (tout, yout (:,1)), title ('s1(t)'), grid on
subplot (2,1,2), plot (tout, yout (:,2)), title ('s2(t)'), grid on
figure (4), plot (tout, yout), legend ('s1(t)', 's2(t)'), grid on
figure (5), plot (yout (:,1), yout (:,2)), title ('s2(s1)'), grid on
```

Результати виконання цієї програми показані на рис. 1.14.

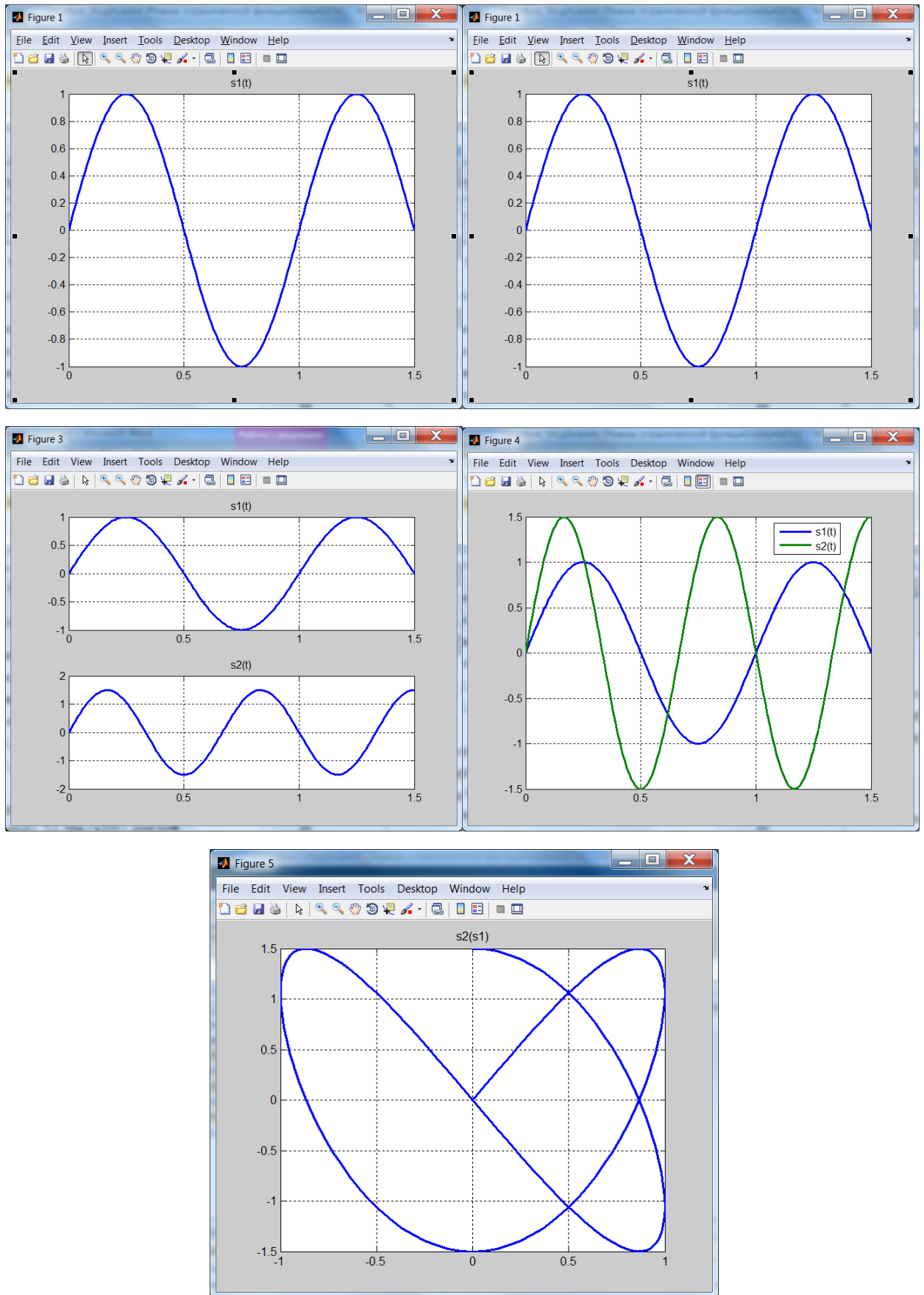


Рис. 1.14. Результати симуляції моделі рис. 1.13 та виконання наведеної вище програми

Такі ж само результати отримаємо, якщо у наведеній вище програмі замінимо  $yout(:,1)$  на  $s1$ ,  $yout(:,2)$  на  $s2$  та  $yout$  на  $[s1\ s2]$ .

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

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  $\infty$  (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 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  $y(i,:)$ .

## 1.2 Завдання

1. За допомогою 3-х блоків *Sine Wave* сформувати 3 синусоїди  $s_1(t)$ ,  $s_2(t)$  і  $s_3(t)$  з різними амплітудами, періодами та фазовими зсувами.

2. Візуалізувати ці сигнали блоками *Scope* у такі способи:

- кожний сигнал окремим блоком *Scope*,
- усі сигнали одним блоком *Scope* з трьома вхідними портами та відповідно трьома системами координат (підвікнами), кожний сигнал у своєму підвікні;
- усі сигнали одним блоком *Scope* в одній системі координат.

4. Апробувати візуалізацію різних синусоїд блоком *Floating Scope*.

4. Візуалізувати залежності  $s_1(s_2)$  і  $s_3(s_2)$  блоками *XY Graph*.

5. Отримати числові результати моделювання за допомогою одного (спільного) та трьох (окремих для кожного сигналу) блоків *To Workspace* та одного (спільного для трьох сигналів) блоку *Out*. Окремим блоком *To Workspace*, приєднаним до блоку *Clock*, отримайте вектор часу.

6. Розробити *script*-файл для побудови графіків за числовими даними, отриманими у попередньому пункті, у такий спосіб:

- кожна із залежностей  $s_1(t)$ ,  $s_2(t)$  і  $s_3(t)$  в окремому графічному вікні;
- кожна із залежностей  $s_1(t)$ ,  $s_2(t)$  і  $s_3(t)$  в окремому підвікні одного графічного вікна;
- усі графіки  $s_1(t)$ ,  $s_2(t)$  і  $s_3(t)$  в одному графічному вікні;
- графіки  $s_1(s_2)$  і  $s_3(s_2)$  в окремих графічних вікнах.

7. Отримати числові значення сигналів  $s_1(t)$ ,  $s_2(t)$  і  $s_3(t)$  у момент часу, що дорівнює двом періодам останньої синусоїди, одним спільним і трьома окремими блоками *Display*. Позначте координати зафіксованих точок на будь-якому із раніше отриманих графіків перехідних процесів.

### 1.3 Методичні рекомендації

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

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

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

4. 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, замініть 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;

5. Для реєстрації часу моделювання застосуйте 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*.

6. 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*.

7. Для об'єднання декількох скалярних сигналів в один векторний використовуйте блок *Mux* або блок *Vector Concatenate* бібліотеки *Signals Routing* (*Маршрутизація Сигналів*).

8. При відображенні моделей та графіків у звіті апробуйте такі способи:

- копіювання вікна моделі, осцилографа або графічного вікна способом *Screenshot* (клавійна команда *Alt+Print Screen*) з подальшим копіюванням його у звіт середовища *Word* командою  $\wedge V$  – вставка з буферу;
- копіювання вікна моделі або графічного вікна за допомогою функцій відповідних вікон *Edit* → *Copy Current View to Clipboard* → *Metafile* та *Edit* → *Copy Figure*; результат такого копіювання можна одразу вставляти в документ *Word*, а можна спочатку пропустити його через графічний редактор *Microsoft Visio*, де отриманий об'єкт можна розгрупувати, прибрати зайві поля та елементи, скоригувати або додати підписи тощо.



9. Для позначення на графіках точок з їх координатами додайте до інструментів графічного вікна через функцію основного меню *View* панель *Plot Edit Toolbar* і натисніть на клавішу *Data Cursor* цієї панелі, що приведе до появи у вікні графічного курсора. За допомогою миші переміщайте курсор в бажану точку графіка, після чого натисніть праву клавішу миші. Якщо потрібно виділити ще декілька точок у графічному вікні, то повторюйте послідовність описаних вище операцій з натискноюю клавішею *Alt*.

10. Щоб зафіксувати увагу на якомусь фрагменті графіка, виділіть її еліпсом (*Insert*→*Ellipse*) або прямокутником (*Insert*→*Rectangle*).

## Лабораторна робота №2

### ФОРМУВАННЯ ВХІДНИХ СИГНАЛІВ

*Мета роботи:* навчитися формувати різноманітні типи вхідних сигналів в середовищі *Simulink*, встановлювати і коригувати їх параметри, комбінувати декілька блоків для виконання поставленої задачі.

#### 2.1 Загальна характеристика блоків бібліотеки *Sources*

Вхідні сигнали, що діють на системи автоматичного керування, здебільш є функціями часу. Їх можна розподілити на такі групи: постійні; ступеневі; лінійні; нелінійні неперіодичні, що описуються аналітичними виразами; гармонічні; періодичні; випадкові.

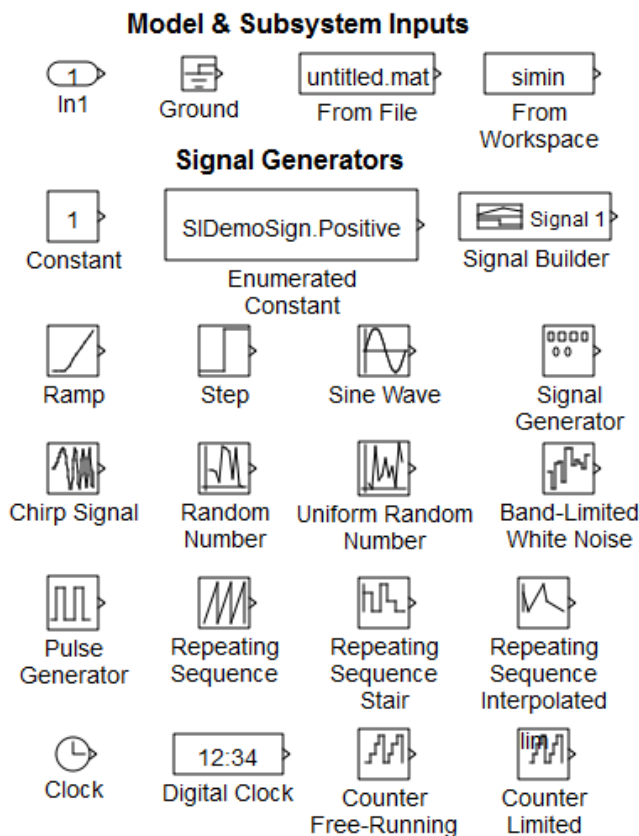


Рис. 2.1. Бібліотека джерел вхідних сигналів *Sources*

Основу формування вхідних сигналів складають блоки бібліотеки джерел *Sources*. 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 неприєданого 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*, *Random Number*, *Uniform Random Number* і *Band Limited White Noise*).

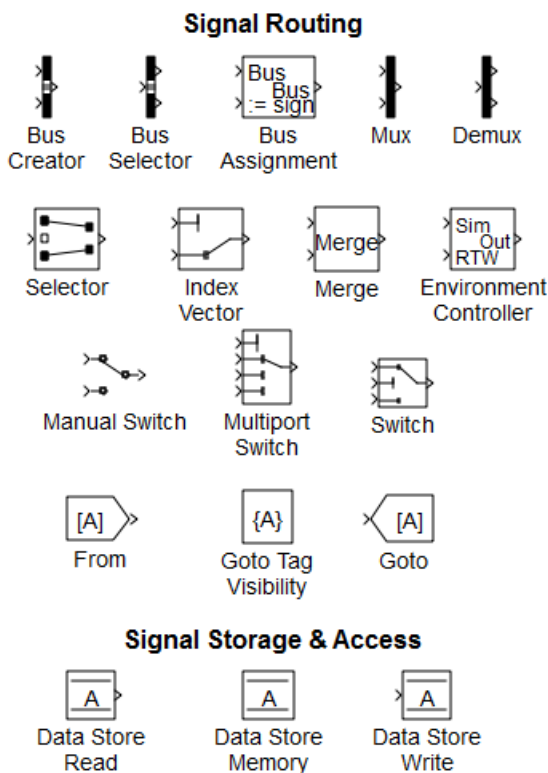


Рис. 2.2. Бібліотека *Signals Routing* (Маршрутизація сигналів)

Векторні параметри джерел можуть бути задані у вікнах налаштування 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*, *Random Number* і *Uniform Random 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 ( $T_s=0$ ) або залишити у стані наслідування ( $T_s=-1$ ), якщо наступний блок також буде неперервним.

Якщо джерело повинне формувати дискретний сигнал, то в поле параметра *Sample Time* може вводитися 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

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

## 2.2 Основні засоби формування вхідних сигналів

Виходом блоку *Constant (Константа)* можуть бути визначені параметром *Constant value* і не залежні від часу скалярна константа  $y = c = \text{const}$ , вектор констант (рядок  $y = [c_1 \ c_2 \ \dots \ c_n]$  або стовбець  $y = [c_1 \ c_2 \ \dots \ c_n]^T$ ) та матриця констант. Елементи масивів можуть бути як дійсними, так і комплексними числами (в тому числі і зарезервованими константами *MATLAB*).

Джерело *Step (Стрибок, Сходінка)* забезпечує стрибкоподібну зміну вихідного сигналу між двома постійними рівнями  $y_{In}$  (*Initial value*) і  $y_{Fin}$  (*Final value*) у заданий момент часу  $t_s$  (*Step time*):

$$y_{Step}(t) = (y_{Fin} - y_{In}) \cdot 1(t - t_s) + y_{In} = \begin{cases} y_{In} & \text{при } t \leq t_s, \\ y_{Fin} & \text{при } t > t_s. \end{cases} \quad (2.2)$$

де  $1(t - t_s)$  – одинична функція Хевісайда з запізненням на час  $t_s$ . Загальний вигляд скалярного ступінчатого сигналу показано на рис. 2.3.

Декілька блоків *Step*, приєднаних до входів алгебраїчного суматора *Sum* з математичної бібліотеки *Math operations* створюють багатосходінковий сигнал. Враховуючи можливість блоку *Step* формувати векторні сигнали, та можливість блоку *Sum* сумувати усі елементи векторного сигналу, багатосходінковий сигнал можна створити послідовним з'єднанням одного блоку *Step* із суматором.

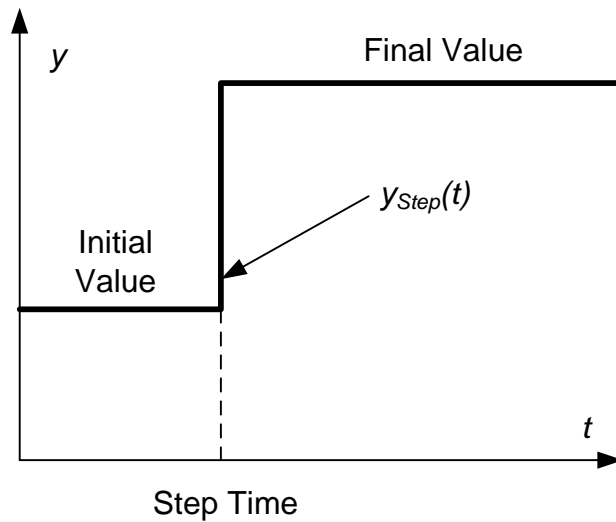


Рис. 2.3. Загальний вигляд скалярного ступінчатого сигналу

Блоки *Sum* (Суматор) та його аналоги *Add* (Додавати), *Subtract* (Віднімати), *Sum of Elements* можуть мати круглу (*round*) або прямокутну (*rectangular*) форму, яка визначається параметром *Icon shape*.

Вони підсумовують 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 „|”.

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

$$y = \text{sum}(u) = \sum_{i=1}^k u_i . \quad (2.3)$$

При цьому в піктограмі блоку відображається не знак вхідного сигналу, а символ  $\Sigma$ . Блоки *Sum* та *Add* з одним входом можна замінити блоком *Sum of Elements*.

Можливі різновиди іконок блоків *Sum*, *Subtract*, *Add* та *Sum of elements* в залежності від значень їх параметрів наведені на рис. 2.4.

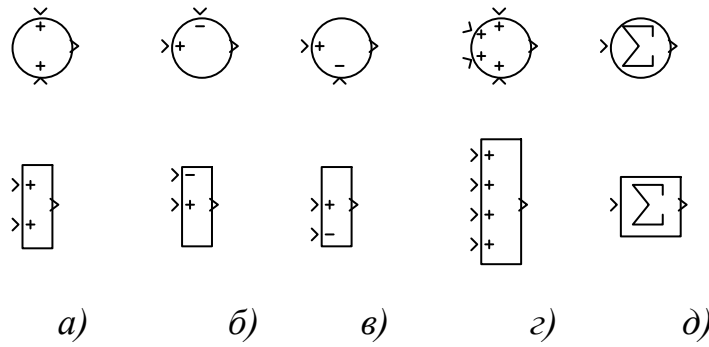


Рис. 2.4. . Вигляд іконок блоків *Sum*, *Subtract*, *Add* та *Sum of elements* при таких значеннях параметру *List of signs*:  
 а) ++; б) -+; в) |+; г) 4; д) 1 або +

На рис. 2.5 показано багатоступінчатий сигнал, а на рис. 2.6 – варіанти його формування. Параметри блоків *Step* з рис. 2.6 наведені у табл. 2.1.

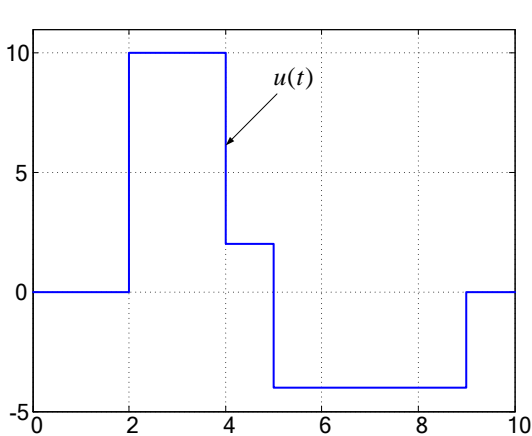


Рис. 2.5. Багатоступінчатий неперіодичний сигнал

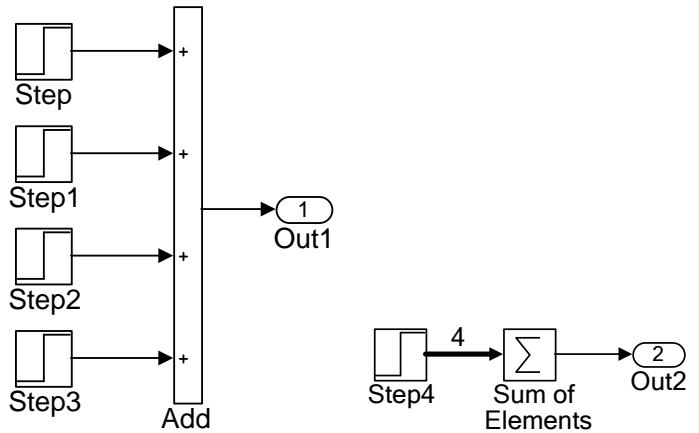


Рис. 2.6. Варіанти формування багатоступінчатого сигналу

Таблиця 2.1

Ім'я блоку	<i>Step Time</i>	<i>Initial Value</i>	<i>Final Value</i>
<i>Step</i>	2	0	10
<i>Step1</i>	4	0	-8
<i>Step2</i>	5	0	-6
<i>Step3</i>	9	0	4
<i>Step4</i>	[2 4 5 9]	[0 0 0 0]	[10 -8 -6 4]

Блок *Clock* (Годинник) забезпечує відлік і відображення часу моделювання:  $y(t) = t$ .

Джерело *Ramp* (Рампа, Похила 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):

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

Загальний вигляд скалярного ступінчатого сигналу показано на рис. 2.7. Розгорнута модель цього блоку рис. 2.8 можна подивитися під маскою (*Ctrl+U*).

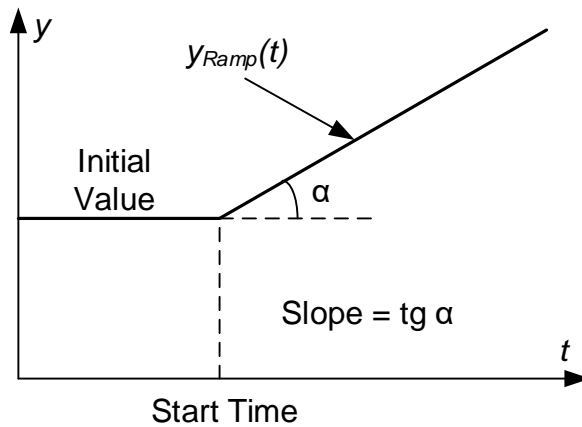


Рис. 2.7. Загальний вигляд скалярного сигналу *Ramp*

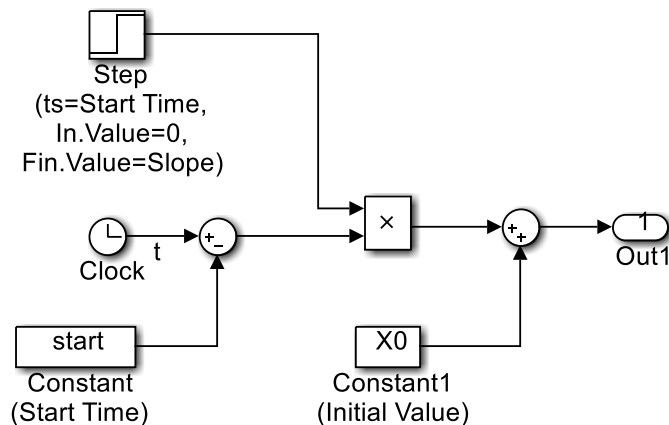
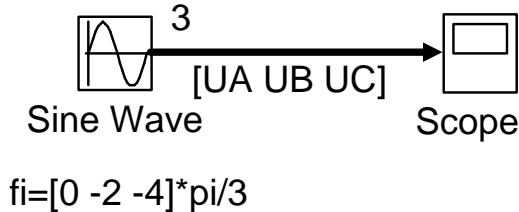


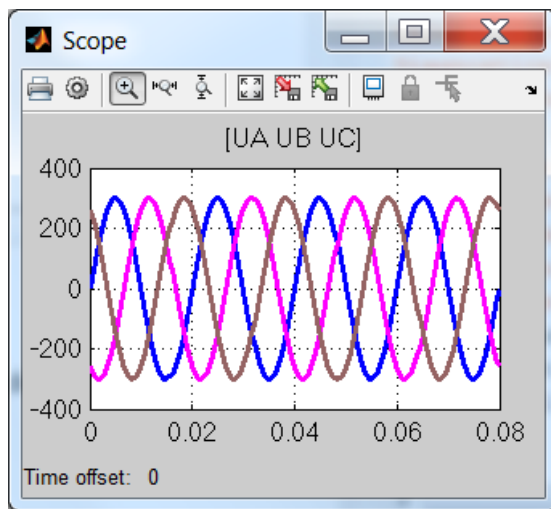
Рис. 2.8. Розгорнута модель блоку *Ramp*

Блок *Sine Wave (Синусоїда)* може формувати як неперервний, так і дискретний вихідний сигнал. У неперервному режимі ( $T_S = 0$ ) блок генерує синусоїду у функції часу відповідно до рівняння:

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



а)



б)

Рис. 2.9. Модель (а) формування трифазних напруг та результати її симуляції (б)

де

$A$  – амплітуда (*Amplitude*);

$\omega$  – кругова частота, рад/с (*Frequency*);

$\varphi_0$  – фазовий зсув, рад. (*Phase*);

$y_0$  – вертикальний зсув (*Bias*).

Не варто забувати, що блок *Sine Wave* здатний формувати векторні сигнали. Цю властивість зручно використовувати при моделюванні трифазних кіл змінного струму. Для прикладу на рис. 2.9 наведено модель формування трифазних напруг змінного струму частотою 50 Гц і амплітудою 300 В, зсунутих одна від одної на кут 120 градусів і результат її симуляції.

Блок *Pulse Generator (Генератор Прямокутних Імпульсів)* формує періодичний сигнал, що складається з однополярних (додатних) прямокутних імпульсів з такими параметрами:

$ht$  – висота імпульсу (*Amplitude*);

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

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

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

Блок *Repeating Sequence (Повторювана Послідовність)* виконує повторення циклу, заданого таблицею двох параметрів: *Time values* ( $t_v$ ) – ве-



ктор 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:  $T=tv(end)-tv(1)$ . Блок перемальовує свою піктограму відповідно до вигляду вихідного сигналу. Він реалізований за допомогою замаскованої підсистеми, принцип роботи якої пояснює рис. 2.10.

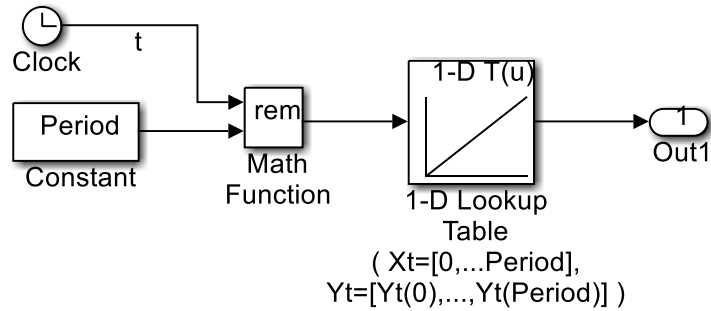
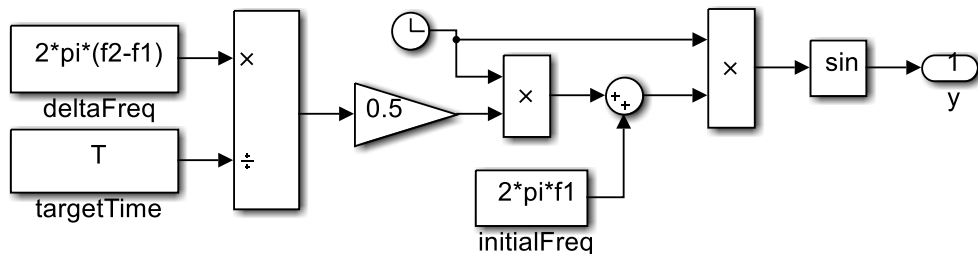


Рис. 2.10 – Принцип роботи замаскованого блоку *Repeating Sequence*

Блок *Math Function* моделі рис. 2.10 визначає залишок від цілочисельного ділення часу моделювання на період, а блок *Look-Up Table* виконує кусочно-лінійну апроксимацію табличної функції, заданої параметрами *Vector of input values* і *Table data*.

Джерело *Chirp Signal (Синусоїда зі Змінною Частотою)*, реалізоване за допомогою замаскованої підсистеми рис. 2.11, формує синусоїду з однічною амплітудою і частотою, що змінюється за лінійним законом.



```
freqSlope = deltaFreq/targetTime;
instantaneousFreq(t) = freqSlope*t + initialFreq;
instantaneousPhase(t) = integral(instFreq) = ...
    0.5*freqSlope*t^2 + initialFreq*t;
outputSignal(t) = sin(instantaneousPhase(t));
```

Рис. 2.11 – Розгорнута модель блоку *Chirp Signal*

Отже, вихідний сигнал цього блоку можна описати рівнянням

$$y = \sin(0.5At^2 + Bt), \quad (2.6)$$

де

$$B=2 \pi f_1, \quad A=2 \pi (f_2 - f_1) / T. \quad (2.7)$$

Вхідними параметрами є:

- $f_1$  (*Initial frequency [Hz]*) – початкова частота [Гц];
- $T$  (*Target time [secs]*) – заданий час [с];
- $f_2$  (*Frequency at target time [Hz]*) – частота в заданий момент часу [Гц].

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 випадваючого меню (*Parameters, Wave form*). Числовими 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* (одиниця вимірювання). 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*, коли реакцію системи на різні типи вхідного сигналу потрібно одержати швидко.

Випадкові сигнали створюються генераторами випадкових чисел.

Джерела випадкових чисел з нормальним (**Random Number**) та рівномірним (**Uniform Random Number**) розподілами генерують відповідно нормально розподілений (Гаусовський) і рівномірно розподілений випадкові сигнали для деяких заданих стартових чисел *Initial seed*, що ініціалізують 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.

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<sub>s</sub> (Sample Time)* – період дискретності;
- Seed* – стартове число генератора випадкових чисел.

Вектори *Noise Power* і *Seed* можуть бути однієї довжини.

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

$$\begin{bmatrix} t_1 & y_1 & z_1 & \dots & w_1 \\ t_2 & y_2 & z_2 & \dots & w_2 \\ \dots & \dots & \dots & \dots & \dots \\ t_k & y_k & z_k & \dots & w_k \end{bmatrix}. \quad (2.8)$$

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 настроювання блоку *From Workspace* ім'я матриці *D*.

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 Fi-*

*nal Value* – затримка останнього табличного значення; *Cyclic Repetition* – циклічне повторення.

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

Цей формат ідентичний формату блоку *To File*, що забезпечує сумісність файлів.

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

```
save FileName var1 var2 ...
```

Наприклад,

```
save data1 D
```

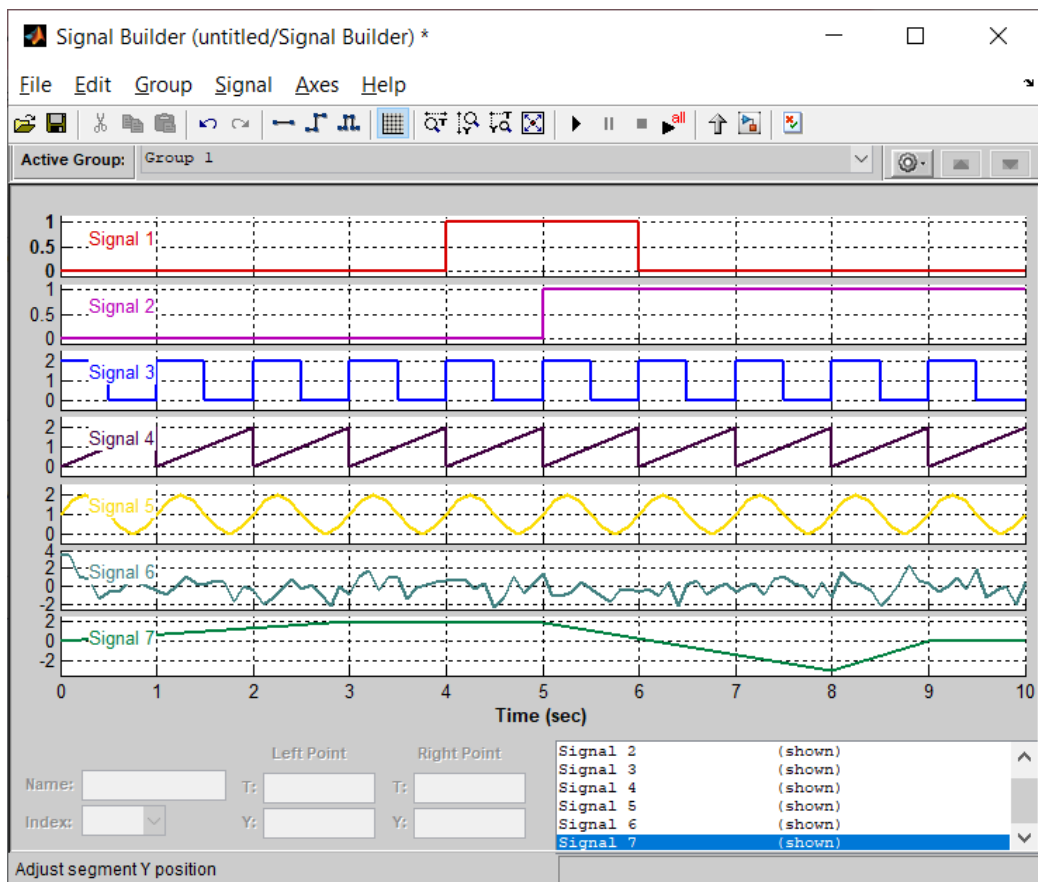
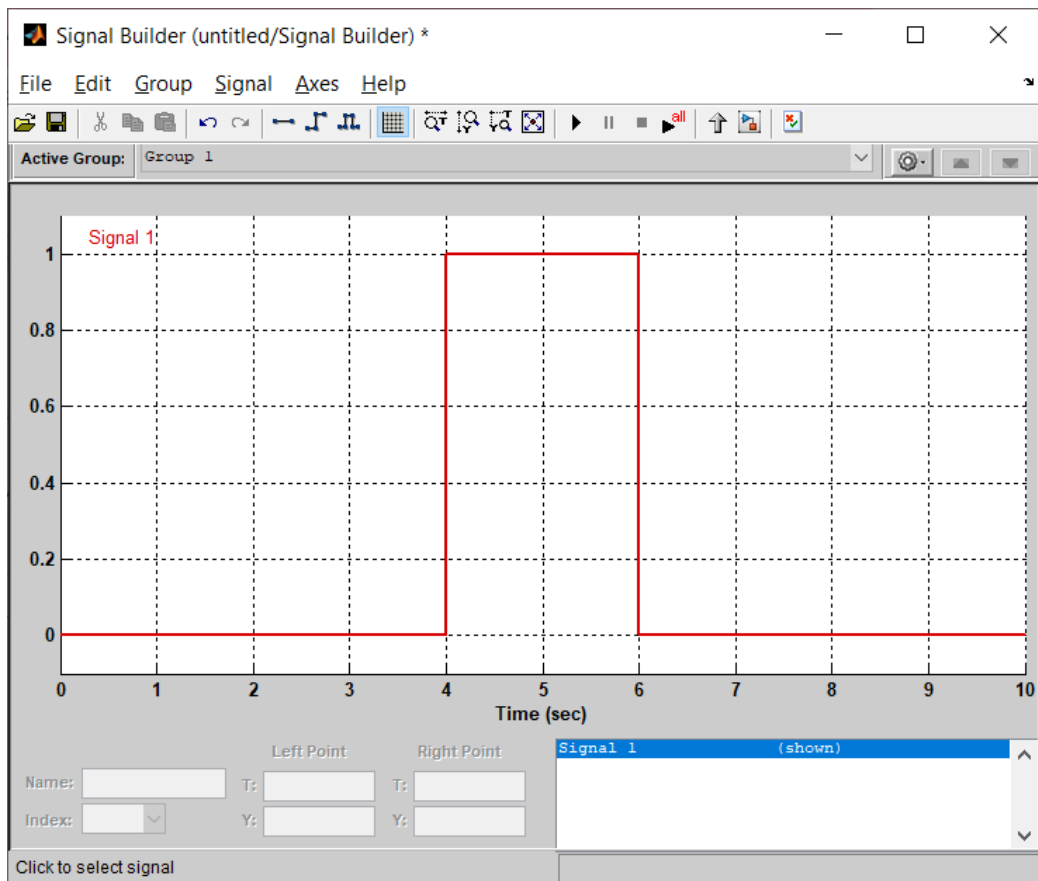
збереже значення змінної *D* у файлі *data1.mat*. Файл даних для блоку *From File* може бути сформований блоком *To File* при моделюванні іншої системи.

Комплексне джерело ***Signal Bilder*** може генерувати один або декілька сигналів (див. рис. 2.12) з переліку (*Signal*→*New*→) *Constant, Step, Pulse...*, *Square...*, *Triangle...*, *Sampled Sin...*, *Sampled Gaussian Noise...*, *Pseudorandom Noise...*, *Poisson Random Noise...*, *Custom...* з різними параметрами.

Отримані сигнали можна коригувати переміщенням фрагментів та/або точок зламу кусково-лінійних функцій та змінювати їх властивості (*Signal* → *Rename, Color, Line Style, Line Width, Change Index*). Крім того, можна коригувати властивості системи координат (*Axes*→*Change Time Range, Set Y Snap Grid, Set T Snap Grid, Set Y Display Limits, Set T Display Limits*), ховати (*Signal*→*Hide*) та показувати (*Signal*→*Show*) графік сигналу, прибирати та показувати координатні сітки (кнопка *Toggle Grid Lines*).

Блок ***In (Вхідний Порт)*** помічає входи моделі.

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.

Рис. 2.12. Вікна блоку *Signal Builder*

### 2.3 Використання блоку *Lookup Table* при формуванні кусково-лінійних вхідних сигналів

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

$$y = \frac{y_i - y_{i-1}}{u_i - u_{i-1}} \cdot (u - u_{i-1}) + y_{i-1}, \quad (2.15)$$

де  $i=1\dots n$  – номери вузлових точок таблиці.

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

Піктограма блоку *Lookup 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.

### 2.4 Завдання

1. За допомогою блоків *Constant* сформууйте довільні дійсну і комплексну константи, константу  $2\pi/3$ , довільні вектор і матрицю констант; результати фіксуйте блоком *Display*.

2. Сформууйте багатоступінчаті сигнали, зображені на рис. 2.13, використовуючи блоки *Step*, *Sum*, *Add* та *Quantizer*.

4. Сформууйте декілька лінійних сигналів у такий спосіб:

- послідовним з'єднанням блоків *Clock* і *Gain*;
- блоком *Ramp*, змінюючи значення і знаки його параметрів;
- послідовним з'єднанням блоків *Constant* і *Integrator* з початковими умовами;

- послідовним з'єднанням блоків *Step* і *Integrator* з початковими умовами і обмеженням вихідного сигналу.

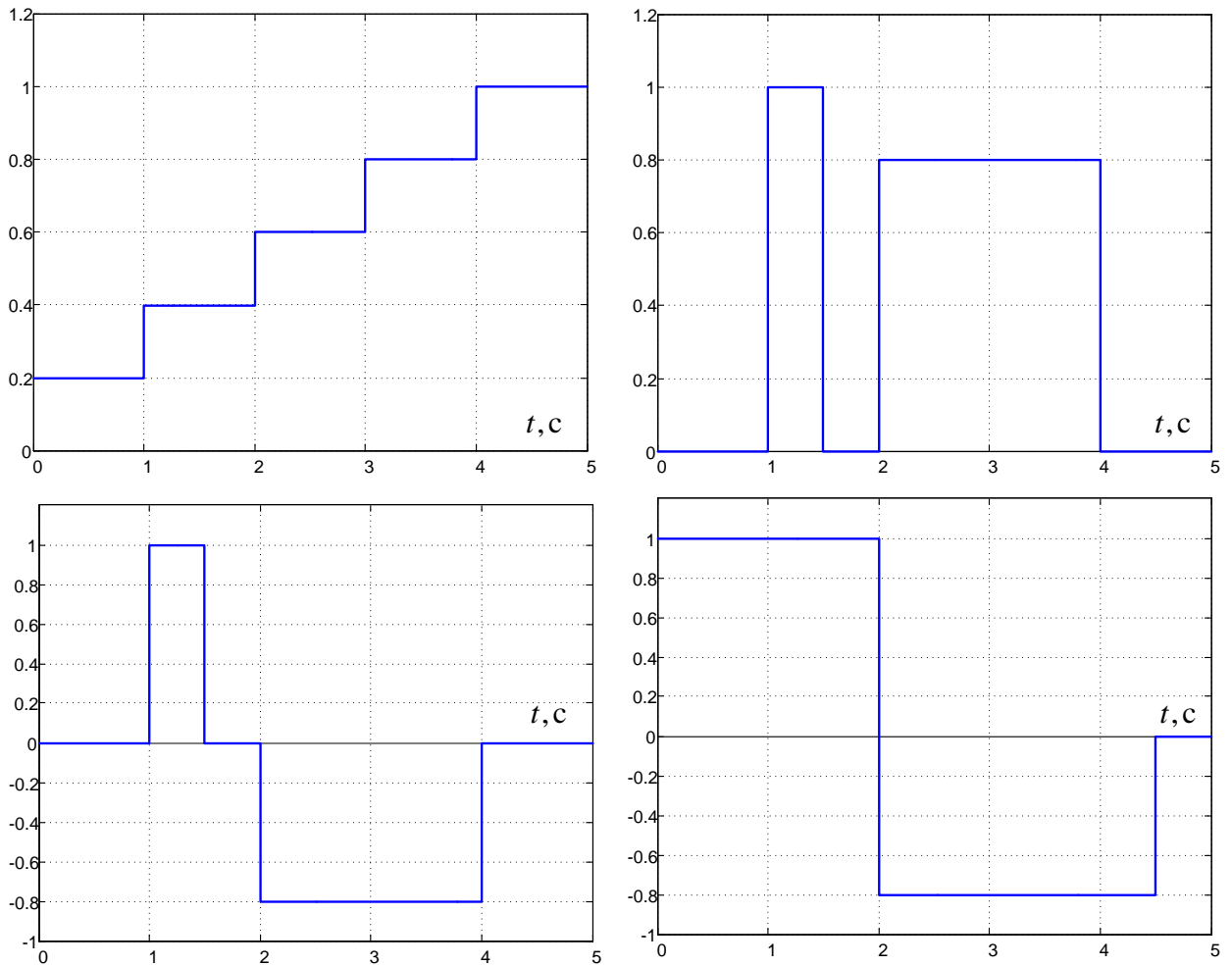


Рис. 2.13. Багатоступінчаті сигнали до завдання 2

4. За допомогою блоку *1D-Lookup Table* сформуєте довільну кусково-лінійну функцію часу із 4-5 ділянок.
5. Сформувати періодичні входні сигнали, подані на рис. 2.14, за даними табл. 2.1.
6. Сформуєте неперервну та дискретну синусоїди, приєднавши блоки *Sine Wave* до блоку фіксації або візуалізації через блок *Manual Switch*.
7. Апробуйте блок *Pulse Generator*. Сформуєте однополярну та біполярну прямокутні хвилі.
8. Апробуйте блок *Chirp Signal*.

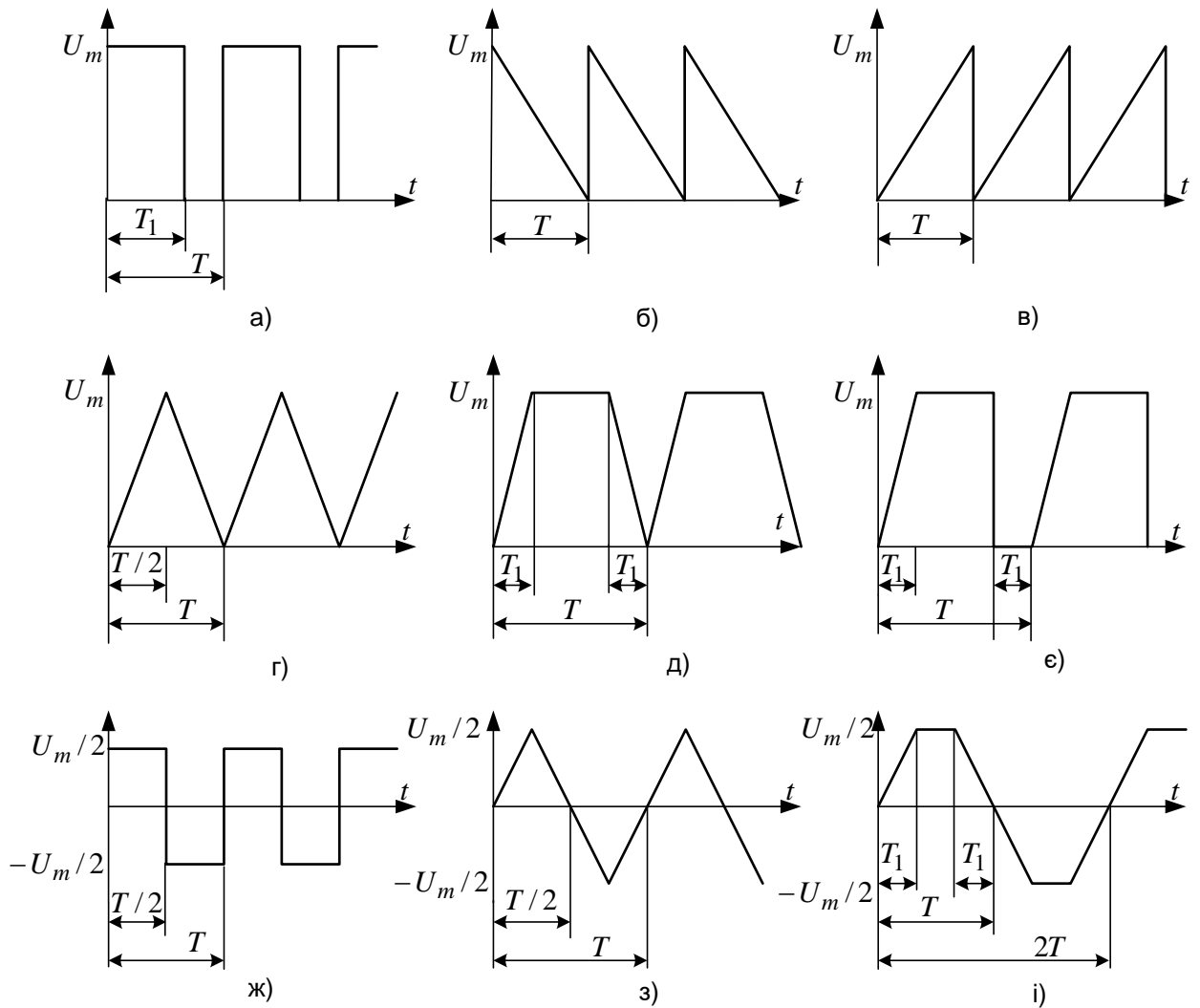


Рис. 2.14. Графіки вхідних сигналів до завдання 4

9. Ознайомтеся з блоком *Signal Builder*. Апробуйте функції “створити новий сигнал” (*Signal→New*), “сховати графік” (*Hide*), “показати графік” (*Show*), “замінити графік” (*Replace with*). Поміняйте властивості графіка (*Rename, Color, Line Style, Line Width, Change Index*), відредагуйте графік в інтерактивному режимі пересуванням ліній та точок зламу.

10. Сформууйте в одному блоці *Signal Builder* 3 сигнали: *Constant*, *Step*, *Pulse* та довільну кусково-лінійну залежність, а в другому – прямокутну хвилю (*Square*), трикутну хвилю (*Triangle*), синусоїду (*Sampled Sin*) та випадковий Гаусівський сигнал (*Sampled Gaussian Noise*).

11. Сформууйте симетричну трифазну напругу, що здійснює пуск асинхронного двигуна (АД) для таких випадків:



- а) прямий пуск –  $f=50\text{ Hz}=\text{const}$ ,  $U_m=220\text{ V}=\text{const}$ , –;
- б) плавний пуск (*Soft Start*) –  $f=50\text{ Hz}=\text{const}$ ,  $U_m(t)=\frac{220}{0.5}t$ ,  $U_m(t)\leq 220\text{ V}$  –;
- в) частотний пуск за законом Костенко  $\frac{U}{f}=\text{const}$  –  $f=100t$ ,  $f(t)\leq 50\text{ Hz}$ ,  
 $U_m(t)=440t=\text{var}$ ,  $U_m(t)\leq 220\text{ V}$ .

## 2.5 Методичні рекомендації

1. При виконанні цієї лабораторної роботи блок *Scope* використовуйте тільки при налаштуванні моделі, а у звіт поміщайте графіки, побудовані за допомогою *plot* або *stairs* з використанням, де треба, допоміжних графічних функцій.

2. При формуванні першого із багатоступінчатих сигналів пункту 3 застосовуйте не *Step*, а *Quantizer*, а для побудови отриманого графіка застосуйте функцію *stairs*, а не *plot*.

3. При використанні блоку *Look-up Table* не забувайте видалити в параметрах функцію *atan*.

4. При наявності в моделях векторних сигналів застосовуйте опції *Wide Nonscalar Lines* та *Signal Dimensions* функції *Display*→*Format*→*Signals & Ports*.

5. При виконанні завдання 11 майте на увазі, що формули

$$\begin{cases} U_A(t)=U_m \sin \omega t, \\ U_B(t)=U_m \sin(\omega t-2\pi/3), \\ U_C(t)=U_m \sin(\omega t-4\pi/3), \end{cases} \quad (2.16)$$

де

$$\omega=2\pi f, \quad (2.17)$$

справедливі тільки при  $\omega=\text{const}$ . Добуток  $\omega t=\varphi$  є з одного боку фазовим кутом, а з іншого – площиною прямокутника під прямою постійної кругової частоти, тобто визначеним інтегралом. Отже в загальному випадку (коли частота може змінюватися у часі) рівняння (2.16) набувають вигляду

$$\begin{cases} U_A(t) = U_m \sin\left(\int_0^t \omega(t) dt\right), \\ U_B(t) = U_m \sin\left(\int_0^t \omega(t) dt - 2\pi/3\right), \\ U_C(t) = U_m \sin\left(\int_0^t \omega(t) dt - 4\pi/3\right). \end{cases} \quad (2.18)$$

Для інтегрування в *Simulink* застосовують блок *Integrator* із бібліотеки *Continuous*, для формування синусоїд – блок *Trigonometric Fn* із бібліотеки *Math Operations*, для перемноження двох змінних сигналів – блок *Product* із бібліотеки *Math Operations*. Варіант *Simulink*-моделі, що формує напруги статора при плавному пуску (нижнє положення ключа) та при частотному пуску (верхнє положення ключа) показано на рис.2.15, а графіки отриманих напруг – на рис. 2.16 та 2.17 відповідно.

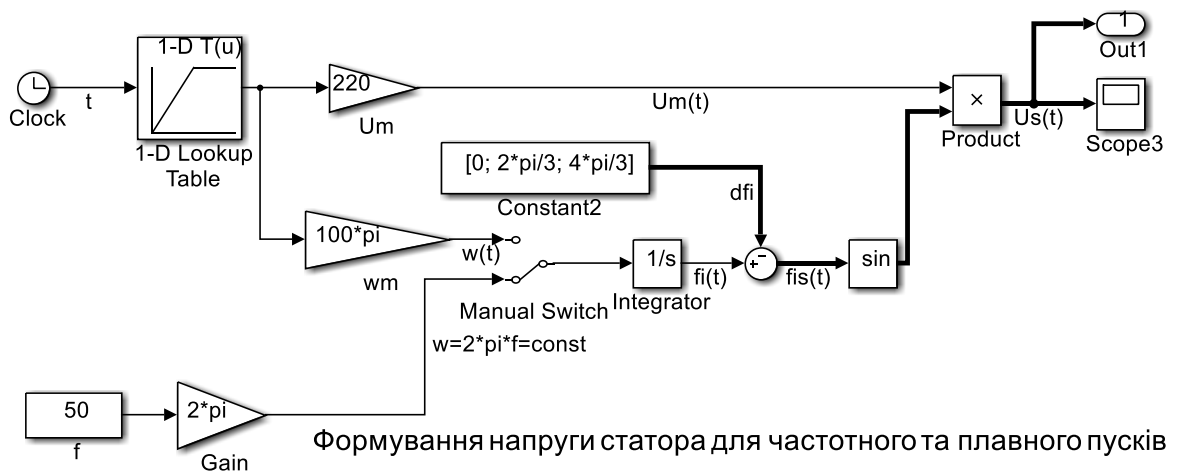


Рис. 2.15. Модель для формування напруг до завдань 11б, 11в

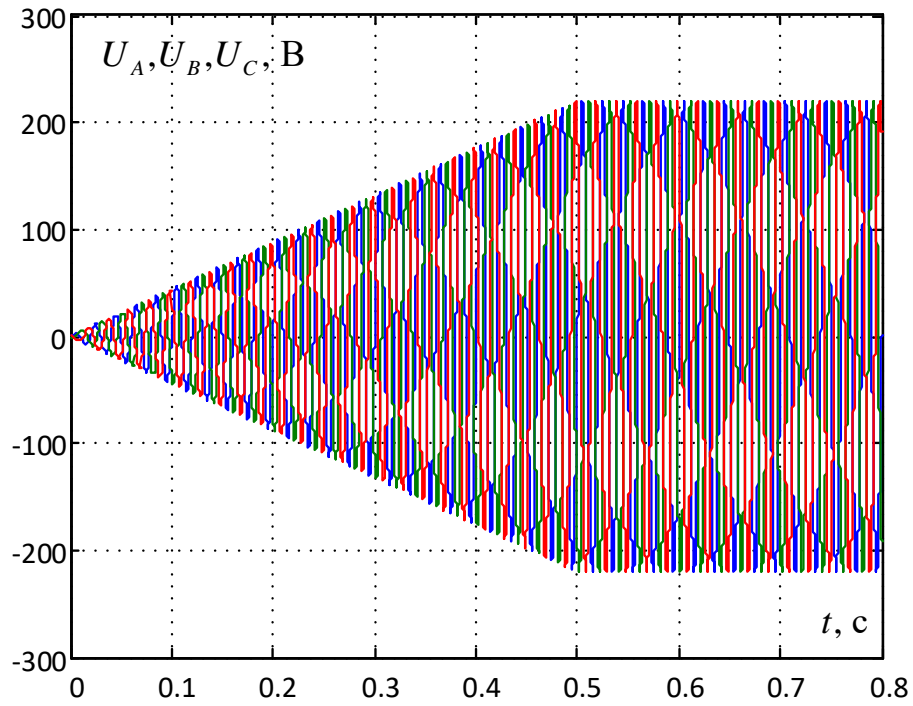


Рис.2.16. Графіки напруг статора при плавному пуску

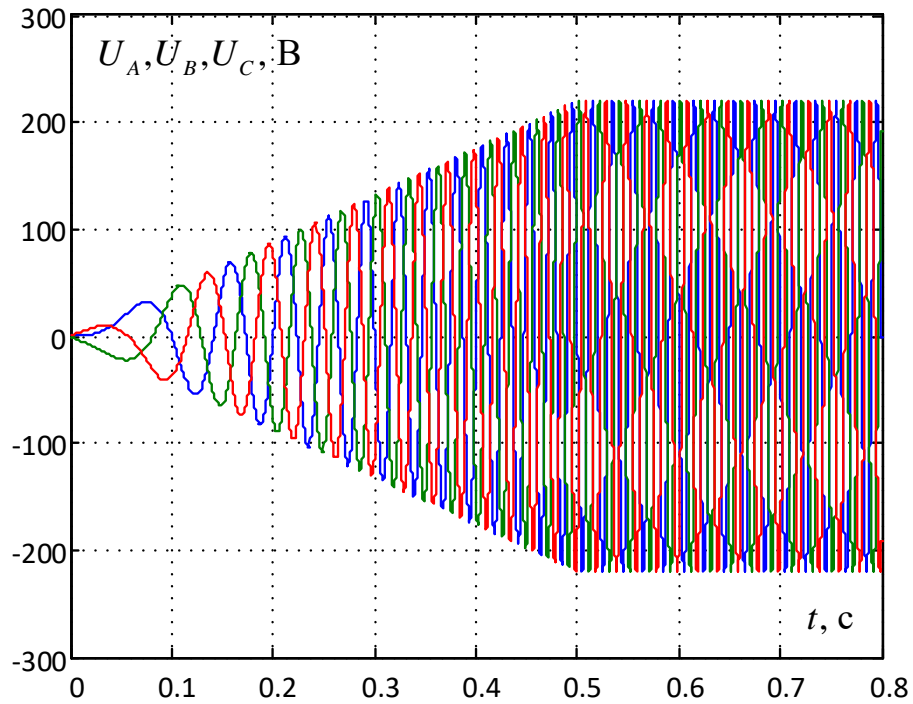


Рис. 2.17. Графіки напруг статора при частотному пуску,  $\frac{U}{f} = const$

### *Лабораторна робота №3*

## **ЗНАЙОМСТВО З БІБЛІОТЕЧНИМИ НЕЛІНІЙНИМИ БЛОКАМИ ПРОГРАМИ *Simulink***

*Мета роботи:* навчитися моделювати в середовищі *Simulink* аналогові та дискретні функціональні перетворювачі з різноманітними статичними характеристиками вхід-вихід.

### **3.1 Класифікація нелінійних блоків**

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

- 1) блоки множення-ділення декількох сигналів;
- 2) типові нелінійності;
- 3) функціональні перетворювачі, описувані нелінійними аналітичними виразами;
- 4) нелінійні функціональні перетворювачі, параметрами яких є координати вузлових точок характеристики «вхід-вихід»;
- 5) періодичні функціональні перетворювачі, для яких є відомим аналітичний опис функції на періоді або координати вузлових точок характеристики «вхід-вихід» на періоді.

***Типовими нелінійностями*** називають нелінійності з досить простими кусково-лінійними характеристиками «вхід-вихід», що часто зустрічаються на практиці.

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

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 нелінійностями є блоки *Петля гістерезису* та *Зазор в кінематичній передачі*.

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

При апроксимації найчастіше використовують *метод найменших квадратів (МНК)* або *кусково-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. Найбільш простим способом інтерполювання є кусково-лінійна апроксимація. Для підвищення точності можна використовувати глобальну інтерполяцію або інтерполяцію кубічними сплайнами. Але застосування останніх методів може значно збільшити час розрахунку перехідних процесів.

### 3.2 Блоки множення та ділення

Блоки, що здійснюють операції множення та ділення декількох сигналів знаходяться у бібліотеці *Math Operations* (див. рис. 3.1).

Блоки *Product (Множення)*, *Divide (Ділення)* та *Product of Elements (Добуток елементів)* є одним і тим же блоком, що здійснює різні операції і змінює свою піктограму (вигляд, іконку) в залежності від значення параметрів *Number of inputs*, *Multiplication*, *Multiply over* та *Dimension*.

У полі параметру *Number of inputs* можна вводити кількість входів або чергувати знаки операцій множення (\*) і ділення (/) без пробілів між ними, подібно до чергування операцій додавання (+) та віднімання (-) у блоках *Sum*, *Add* і *Subtract*. Наприклад, при послідовності «\*/\*» для скалярних вхідних сигналів блоки розраховують  $y = u_1 u_3 / u_2$ , а при значенні цього параметру «4» –  $y = u_1 u_2 u_3 u_4$ . Значення «1» або «\*» мають сенс тільки для векторних або матричних вхідних сигналів. У цьому разі до списку параметрів досліджуваних

блоків додається параметр *Multiply over*, який може приймати значення або *Specified Dimension*.

Якщо обрано опцію *All Dimensions*, то блок розраховує добуток усіх елементів вхідного масиву. Якщо обрано опцію *Specified Dimension*, то з'являється ще й параметр *Dimension*, значеннями якого є цілочислові константи, що визначають порядковий номер розмірності (для матриці значення «1» примушує блок розраховувати добутки усіх елементів кожного стовпця, а значення «2» – добутки усіх елементів кожного рядка).

Параметр *Multiplication* може приймати значення *Element-wise* (.\*), та *Matrix* (\*). Опція *Matrix* має сенс при наявності мінімум двох входів.

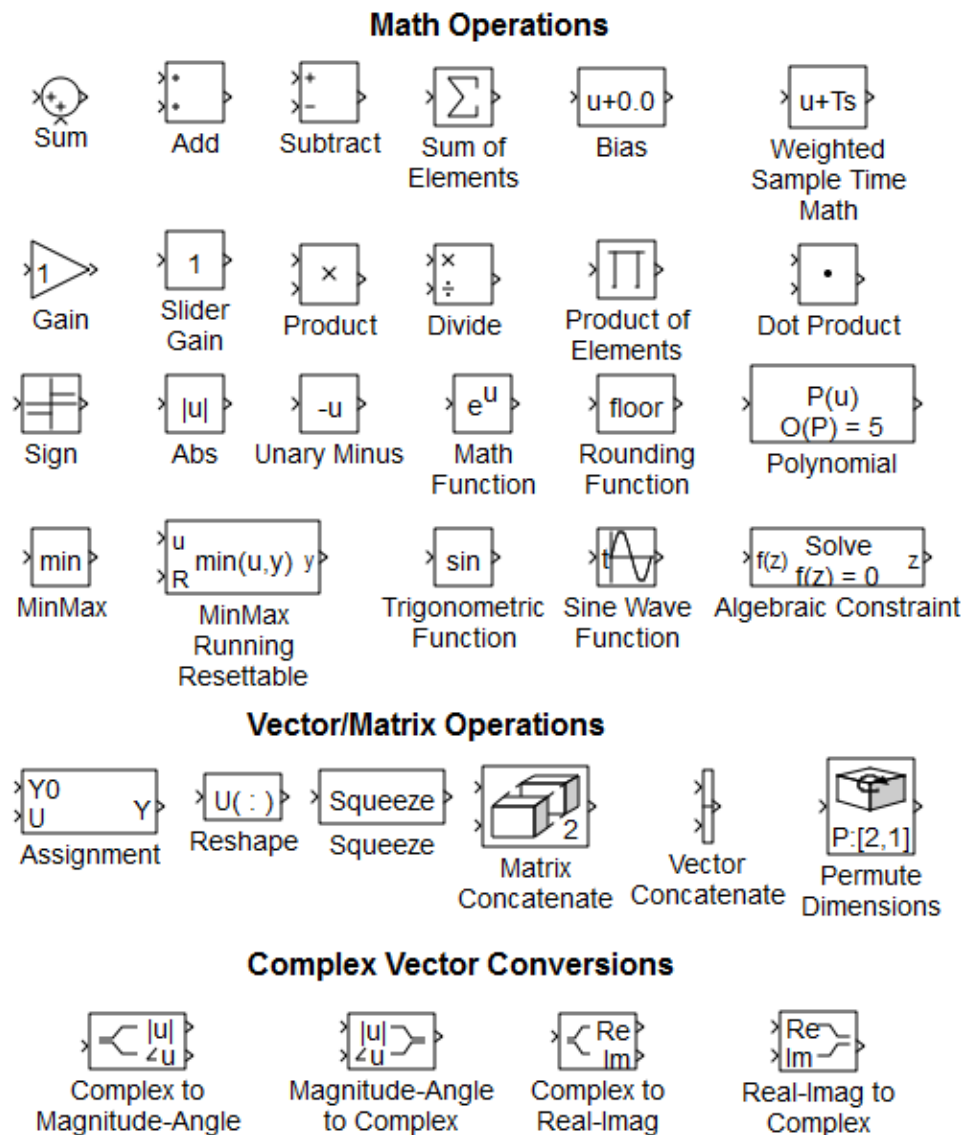


Рис. 3.1. Бібліотека математичних блоків *Math Operations*

Варіанти операцій, що можуть виконувати ці блоки, можна зрозуміти із розглядання рис. 3.2 та інформації щодо цього рисунку, наведеної у табл. 3.1.

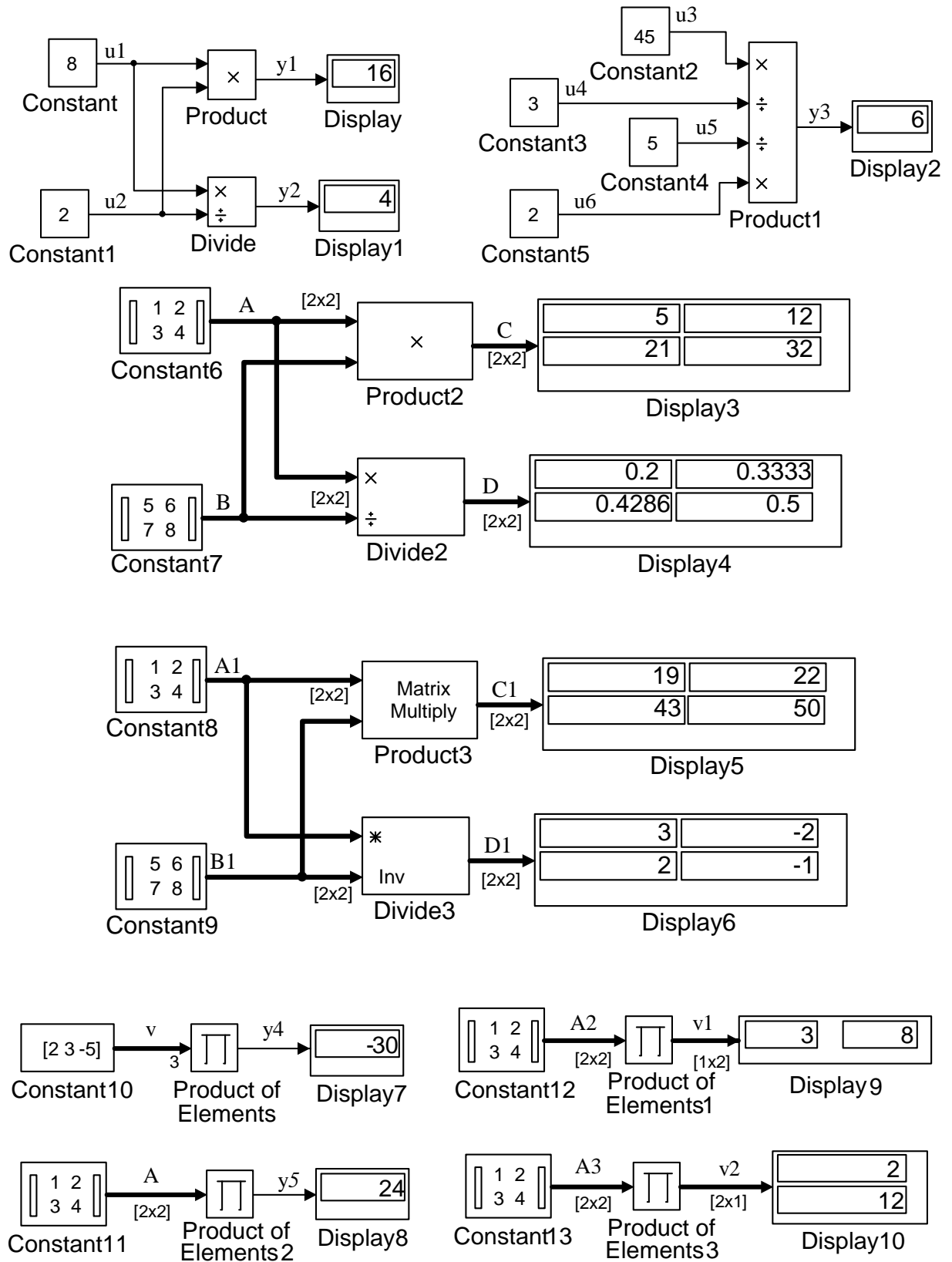


Рис. 3.2. Приклади використання блоків множення та ділення у *Simulink*

Таблиця 3.1

Ім'я блоку	Number of inputs	Multiply	Multiplication over	Dimension	Операція MATLAB
Product	2	Element-wise	-	-	$u1*u2$
Divide	*/	Element-wise	-	-	$u1/u2$
Product1	*//*	Element-wise	-	-	$u1/u2/u3*u4$
Product2	2	Element-wise	-	-	$A.*B$
Divide2	*/	Element-wise	-	-	$A./B$
Product3	2	Matrix	-	-	$A*B$
Divide3	*/	Matrix	-	-	$A/B=A*inv(B)$
Product of elements	1 або *	Element-wise	All Dimensions	-	$prod(v)$
Product of elements1	1 або *	Element-wise	All Dimensions	-	$prod(prod(A))$
Product of elements2	1 або *	Element-wise	Specified Dimension	1	$prod(A)$
Product of elements3	1 або *	Element-wise	Specified Dimension	2	$prod(A')$

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

$$y = \text{dot}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} = \sum_{i=1}^n u_i v_i . \quad (3.3)$$

Якщо вхідні сигнали – скаляри, то обчислюється їхній добуток. Приклади використання блоку наведені на рис. 3.3.

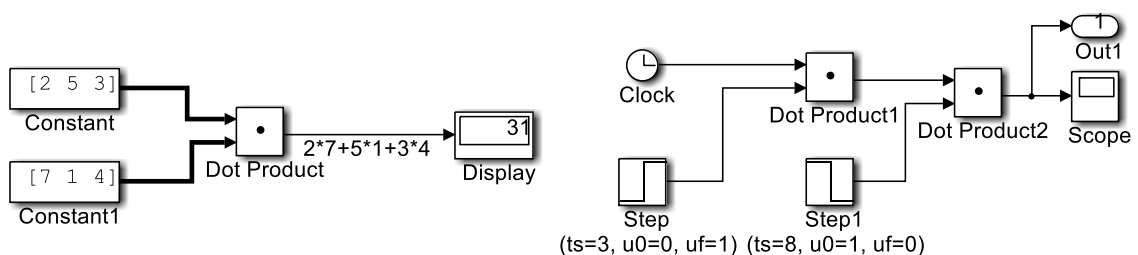


Рис. 3.3. Приклади використання блоку *Dot Product* для векторних та скалярних вхідних сигналів



### 3.3 Блоки з нелінійними статичними характеристиками, заданими аналітично

Нелінійні статичні характеристики, задані аналітичними виразами, в *Simulink* можна реалізувати блоками функціональними блоками *Abs*, *Sign*, *Sqrt*, *Signed Sqrt*, *Reciprocal Sqrt*, *Math Function*, *Trigonometric Function* і *Polynomial* з бібліотеки *Math Operations* (див. рис. 3.1) і блоками бібліотеки *User defined functions*, наведеною на рис. 3.4.

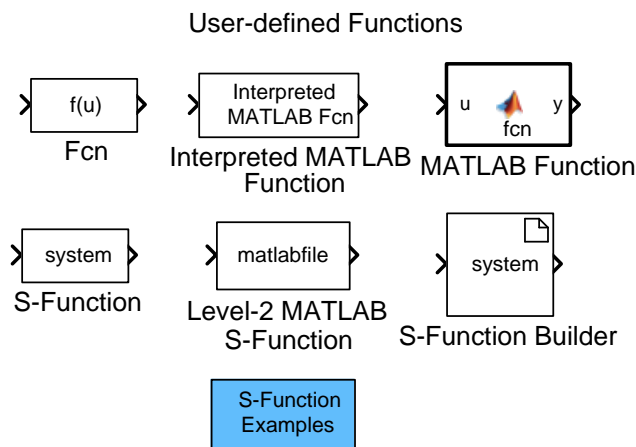


Рис. 3.4. Бібліотека *User defined functions*

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

Блоки *Sqrt*, *Signed Sqrt*, та *Reciprocal Sqrt* – це однакові блоки, які в залежності від вибору параметру *Function*, який може приймати значення *Sqrt*, *ReSqrt* та *SignSqrt*, виконують такі опе-

рації:

$$y = \sqrt{u}, \quad (3.4)$$

$$y = 1/\sqrt{u} \quad (3.5)$$

та

$$y = \sqrt{|u|} \cdot \text{sign}(u).$$

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

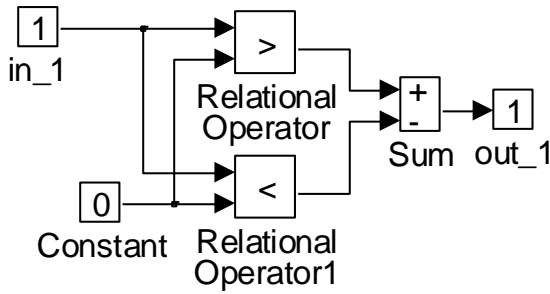


Рис. 3.5. Структура блоку *Sign*

Блок *Sign* (Знакова Функція)

формує вихідний сигнал, величина якого визначається знаком вхідного сигналу:

$$y = \begin{cases} 1 & \text{при } u > 0, \\ 0 & \text{при } u = 0, \\ -1 & \text{при } u < 0. \end{cases} \quad (3.6)$$

Блок *Trigonometric Function* відтворює одну із елементарних тригонометричних (*sin*, *cos*, *tan*), зворотних тригонометричних (*asin*, *acos*, *atan*, *atan2*), гіперболічних (*sinh*, *cosh*, *tanh*), та зворотних гіперболічних функцій (*asinh*, *acosh*, *atanh*) від вхідних сигналів шляхом вибору значення параметра *Function*.

Блок *Polynomial* (Поліном) обчислює значення степеневого полінома, в якому незалежною змінною є вхідний сигнал *u*, а вектор коефіцієнтів  $\mathbf{A} = [a_n, a_{n-1}, \dots, a_1, a_0]$  задається у полі параметра *Polynomial coefficients*:

$$P_n(u, \mathbf{A}) = a_n u^n + a_{n-1} u^{n-1} + \dots + a_1 u + a_0, \quad (3.7)$$

що відповідає виконання в MATLAB оператора  $y = \text{polyval}(\mathbf{A}, u)$ .

Блок *Math Function*

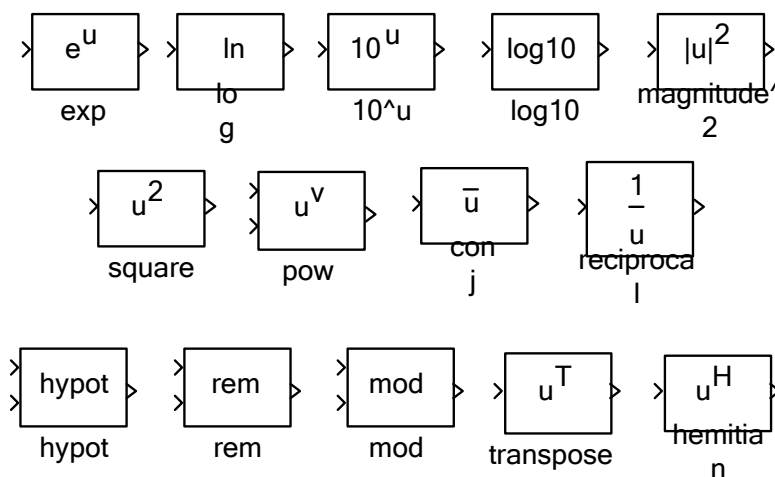


Рис. 3.6. Блоки *Math Function*

*tion* може відтворювати елементарні одно- та двоаргументні функції від вхідних сигналів. Вибір функції здійснюється через меню. Він впливає на вигляд піктограми блоку. На рис. 3.6 наведені можливі різно-

видності цього блоку. Для наочності імена блоків замінені іменами функцій меню. Двоаргументна функція *hypot* (*u, v*) здійснює операцію

$$y_i = \sqrt{u_i^2 + v_i^2}, \quad i = 1, 2, \dots, n, \quad (3.8)$$

функція  $mod(u, v)$  округляє результат ділення першого аргументу на другий до найближчого цілого з недоставкою, а функція  $rem(u, v)$  – визначає остачу від цілочисленного ділення цих аргументів. Інші функції не потребують пояснень.

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

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

Наприклад,  $sin(4*u).*exp(-u/k)$ ,  $rem(u(1), u(2))$ ,  $u(1)*u(2)>=u(3)+u(4)$ .

Блок **Interpreted MATLAB Fcn (Інтерпретована МАТЛАВ-Функція)** обчислює задану **MATLAB**-функцію (стандартну або створену користувачем) від вхідного сигналу  $u$  (скалярного або векторного). Цей блок є пасивним двобічним 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 **Interpreted 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 *MATLAB Fcn (MATLAB-Функція)* включає в *Simulink*-модель С-код сформованої користувачем *MATLAB*-функції, текстовий редактор для формування якої з заданим шаблоном відкривається при подвійному щиглику мишею на іконці блоку:

```
1 function y = fcn(u)
2 %#codegen
3
4 y = u;
```

### 3.4 Типові нелінійності та їх формування блоками Simulink

При моделюванні нелінійних систем використовують термін «*типові нелінійності*» стосовно кусково-лінійних характеристик з невеликою кількістю лінійних ділянок, що часто зустрічаються на практиці, як це вже було відзначено у підрозділі 3.1.

Серед вже розглянутих вище нелінійних залежностей до них відноситься функція  $y = |u|$ , що реалізуються блоком *Abs* та має характеристику, зображену на рис. 3.7а. У якості прикладу реального пристрою з такою статичною характеристикою можна навести діодний міст (див. рис. 3.7б).

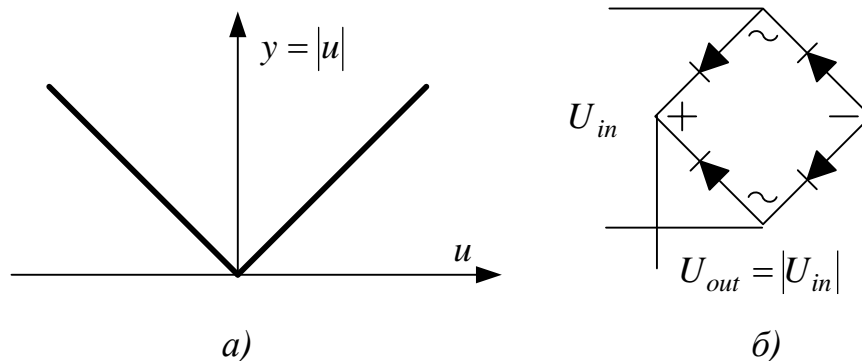


Рис. 3.7. Статична характеристика блоку (а) та принципова схема діодного моста (б)

За допомогою функції  $y = \text{sign}(u)$  (3.6), графік якої подано на рис. 3.8, можна описати типові нелінійності «*Ідеальне реле*» та «*Сухе тертя*».

Ідеальна релейна характеристика описується рівнянням

$$y = y_0 \text{sign}(u), \quad (3.9)$$

сила сухого тертя  $F_{fr}$  (*Coulombic friction*) для тіл з поступальним рухом – рівнянням

$$F_{fr} = F_0 \text{sign}(v), \quad (3.10)$$

а момент сухого тертя  $M_{fr}$  для тіл, що обертаються, –

$$M_{fr} = M_0 \text{sign}(\omega). \quad (3.11)$$

Отже, *сухим (Кулоновським) тертям* називають тертя, величина якого не залежить від абсолютної величини швидкості (лінійної  $v$  або кутової  $\omega$ ). При зміні напрямку швидкості сила або момент тертя також змінюють свій знак. Для прикладу на рис. 3.8 показано графік залежності моменту ідеального сухого тертя від кутової швидкості та модель сухого тертя, реалізовану за допомогою блоку **Sign**. Слід підкреслити неможливість перестановки у моделі сухого тертя блоків **Sign** та **Gain**.

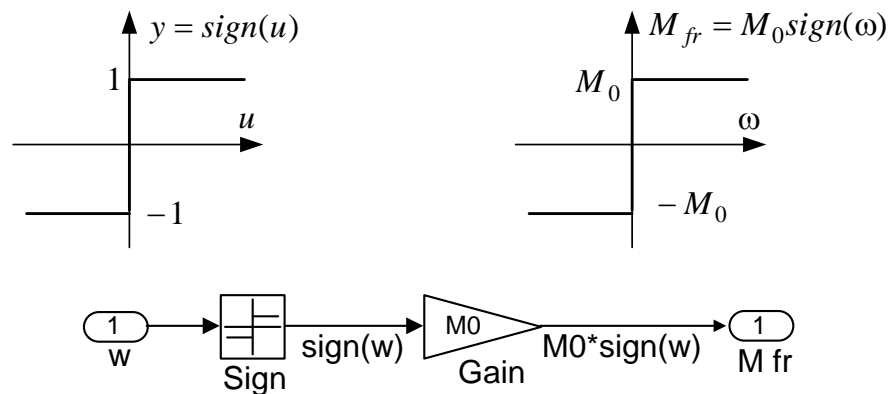


Рис. 3.8. Графіки функцій (3.9), (3.10) та модель сухого тертя

Крім сухого тертя, у природі існує так зване в'язке тертя, величина якого змінюється при варіації швидкості. При наявності і сухого, і в'язкого тертя, пропорційного швидкості, отримуємо ще одну типову нелінійність.

Для її відображення в *Simulink* передбачено блок **Coulombic & Viscous Friction (Сухе та В'язке Тертя)** зі статичною характеристикою, поданою на рис. 3.9, яка описується рівнянням

$$y = (y_0 \pm K \cdot |u|) \cdot \text{sign}(u), \quad (3.12)$$

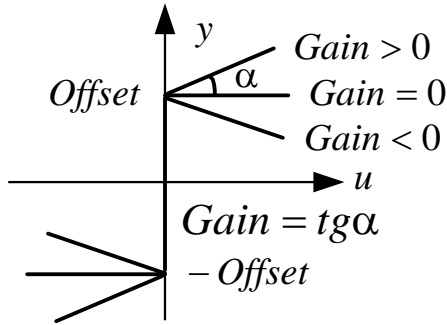


Рис. 3.9. Статичні характеристики блоку *Coulombic & Viscous Friction*

де

$y_0$  – *Coulomb friction value (Offset)* – амплітуда Кулонівського тертя;

$K$  – *Coefficient of viscous friction (Gain)* – коефіцієнт в'язкого тертя.

При  $K > 0$  блок реалізує **додачне в'язке тертя**, що має місце, наприклад, при руху в рідинному або газовому середовищах, при  $K < 0$  – **від'ємне в'язке тертя**, яке характерне для тертя ковзання між двома поверхнями, а при  $K = 0$  – **сухе тертя**.

При зміні знаку цього параметру змінюється і вигляд іконки блоку (див. рис. 3.10).

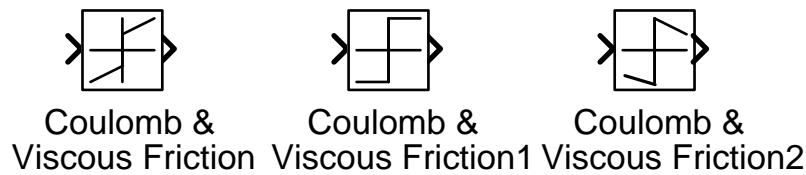


Рис. 3.10. Вигляд блоків *Coulombic & Viscous Friction* при різних значеннях параметру *Coefficient of viscous friction (Gain)*

Блок *Coulombic & Viscous Friction*, як і більшість інших типових нелінійностей, знаходяться в бібліотеці ***Discontinuities***, представленій на рис. 3.11. Піктограми нелінійних блоків зображують загальний вигляд їх статичних характеристик.

На рис. 3.12 показано перехідні процеси перетворення синусоїди блоком *Coulombic & Viscous Friction*.

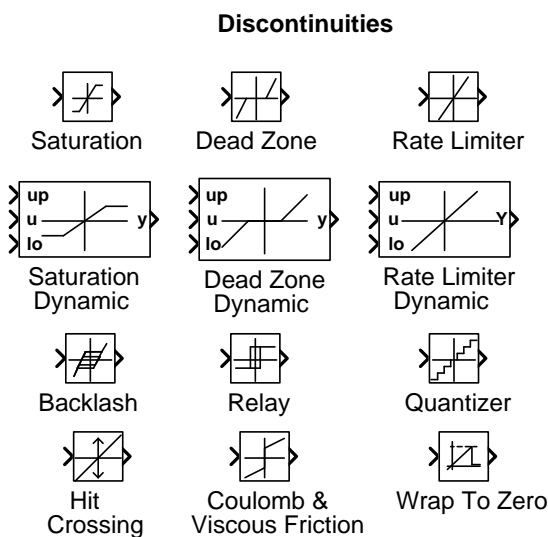


Рис. 3.11. Бібліотека *Discontinuities*

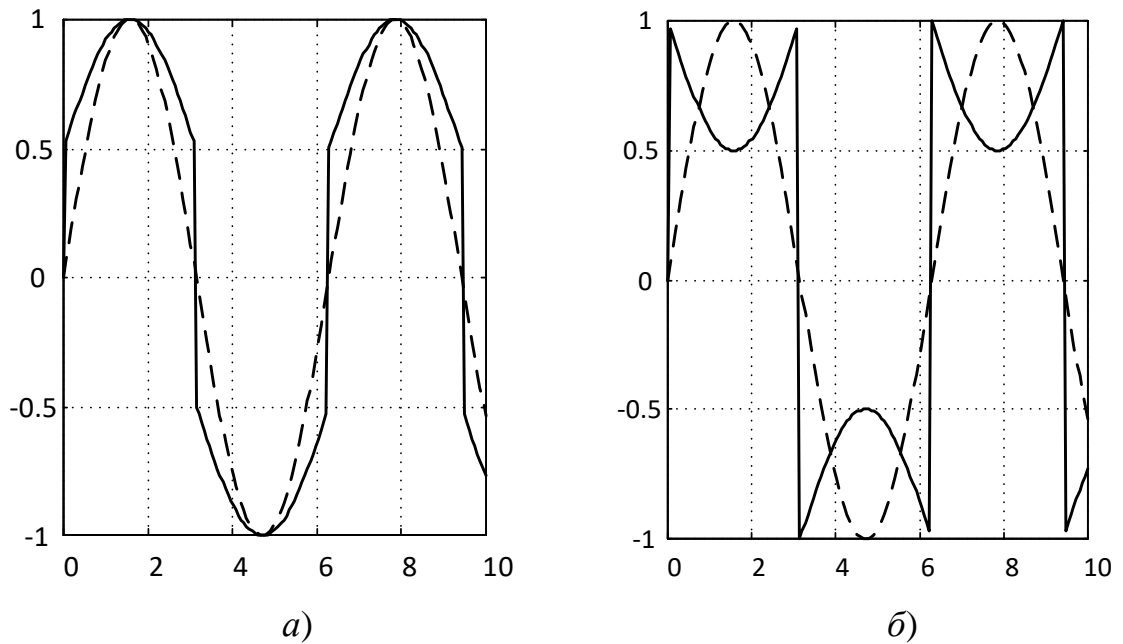


Рис. 3.12 – Перетворення синусоїди ланкою *Coulumbic & Viscous Friction*:  
 а) *Offset* = 0.5, *Gain* = 0.5; б) *Offset* = 1, *Gain* = -0.5

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

$$y = \begin{cases} u & \text{при } L \leq u \leq U, \\ U & \text{при } u > U, \\ L & \text{при } u < L. \end{cases} \quad (3.13)$$

Типова нелінійність «**Обмеження координат**» має надзвичайне велике поширення на практиці, тому що вихідні координати реальних пристроїв не можуть змінюватися до безкінечності. Прикладами таких пристроїв є операційні підсилювачі, тиристорні підсилювачі, різноманітні датчики. Найчастіше характеристики «Обмеження координат» є симетричними, але, на відміну від характеристики блоку *Saturation*, коефіцієнт підсилення лінійної ділянки в них може бути довільним. Отже, для узгодження цих характеристик на вході блоку *Saturation* треба встановити блок *Gain* з бажаним коефіцієнтом, як це показано на рис. 3.13.

Як і в схемі рис. 3.8, в моделі рис. 3.13 послідовно з'єднані блоки *Gain* і *Saturation* не можна міняти місцями.

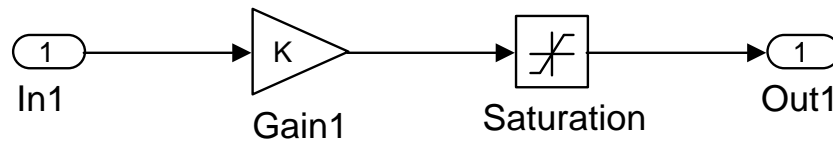


Рис. 3.13. Модель типової нелінійності «Обмеження координат»

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

$$y = \begin{cases} 0 & \text{при } L \leq u \leq R, \\ u + L & \text{при } u < L, \\ u - R & \text{при } u > R, \end{cases} \quad (3.14)$$

де  $L, R$  – ліва (*Start of dead zone*) і права (*End of dead zone*) границі зони нечутливості відповідно. Ліву зону цієї ланки можна збільшувати до  $-\infty$  (*-Inf*), а праву – до  $+\infty$  (*Inf*). Єдине обмеження, що накладається на ці параметри – це  $L < R$ .

Якщо треба змінити коефіцієнт підсилення лінійних ділянок, то блок *Gain* необхідно встановити на виході блоку *Dead Zone* (див. рис. 3.14).

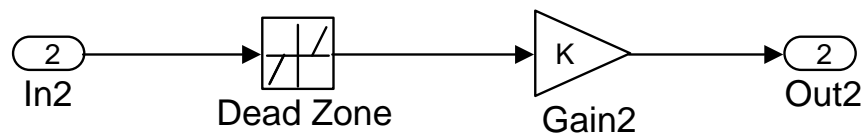


Рис. 3.14. Модель типової нелінійності «Зона нечутливості»

Реалізація типових нелінійностей «Обмеження координат» та «Зона нечутливості» на операційних підсилювачах показана на рис. 3.15.

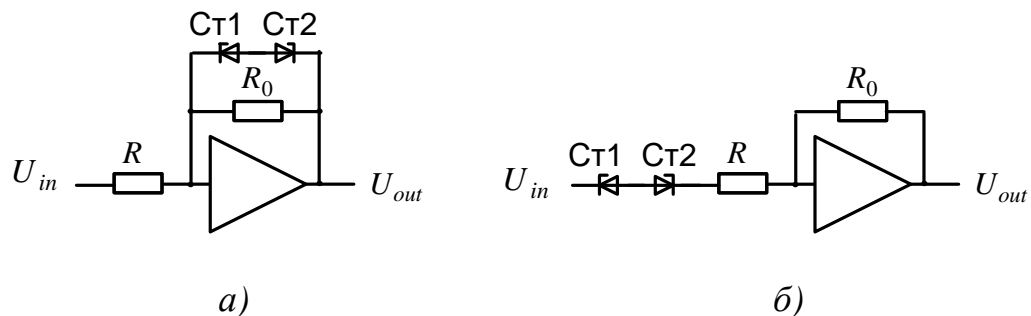


Рис. 3.15. Реалізація типових нелінійностей «Обмеження координат» (а) та «Зона нечутливості» (б) на операційних підсилювачах



Обмеження та нечутливість у цих схемах досягається завдяки відповідному включенню стабілітронів Ст1 та Ст2.

Процес перетворення синусоїдального сигналу ланками *Saturation* та *Dead Zone* відображено на рис. 3.16, 3.17.

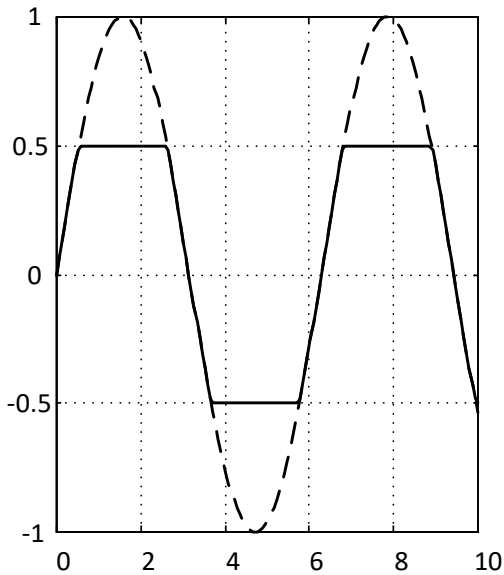


Рис. 3.16 – Перетворення синусоїди ланкою *Saturation*

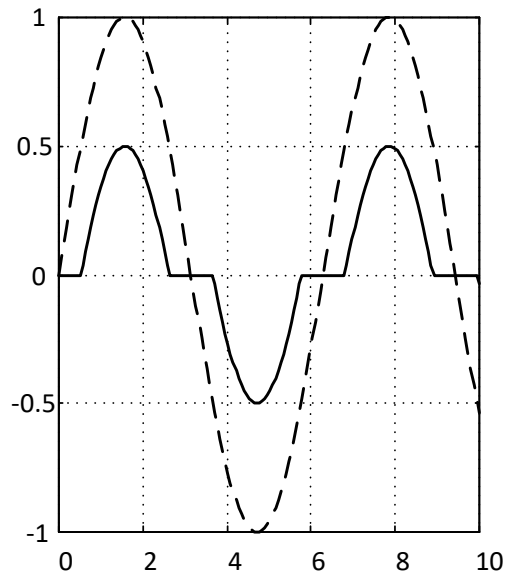


Рис. 3.17 – Перетворення синусоїди ланкою *Dead Zone*

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*).

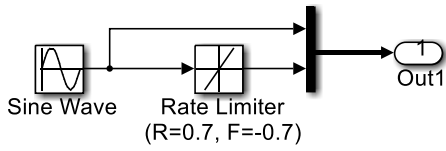
Модель та графік обмеження темпів наростання та спадання синусоїдального сигналу показано на рисунку 3.18, а стрибкоподібних – на рис.3.19.

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

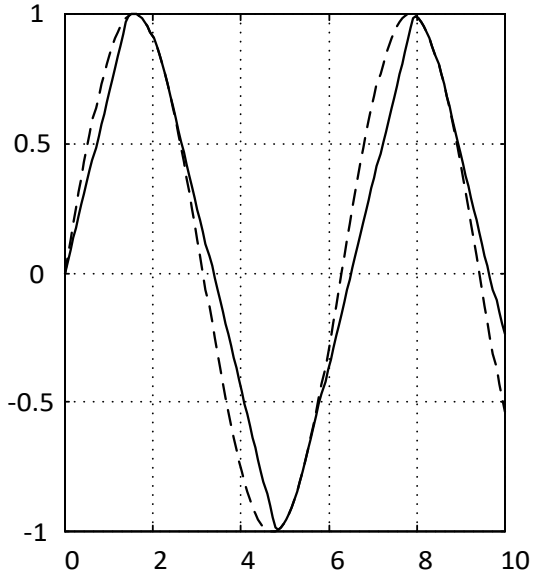
$$y_i = \begin{cases} u - \frac{\delta}{2} & \text{при } u - \frac{\delta}{2} > y_{i-1}, \\ u + \frac{\delta}{2} & \text{при } u + \frac{\delta}{2} < y_{i-1}, \\ y_{i-1} & \text{при } u - \frac{\delta}{2} \leq y_{i-1} \leq u + \frac{\delta}{2}, \end{cases} \quad (3.15)$$

де  $\delta$  – величина зазору (*Deadband width*).

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

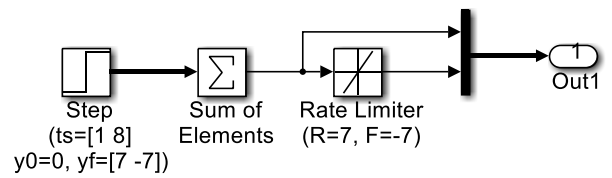


a)

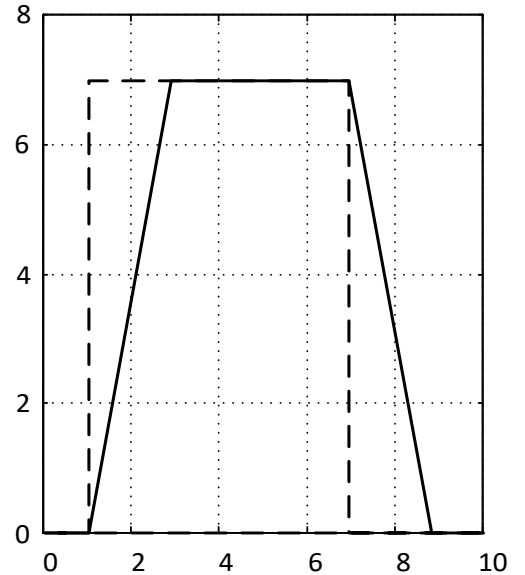


б)

Рис. 3.18. Модель(а) та перетворення(б) синусоїди ланкою *Rate Limiter*



a)



б)

Рис. 3.19. Модель (а) та перетворення (б) стрибкоподібних сигналів ланкою *Rate Limiter*

Статична характеристика блоку *Backlash* та реакція його на синусоїдальний вхідний сигнал показані на рис. 3.20, 3.21 відповідно.

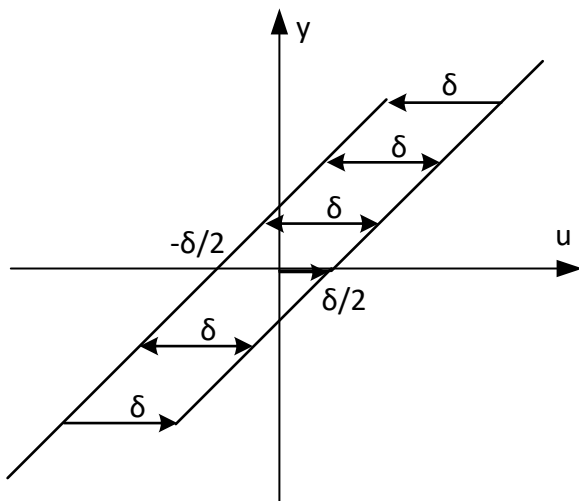


Рис. 3.20. Статична характеристика блоку *Backlash*

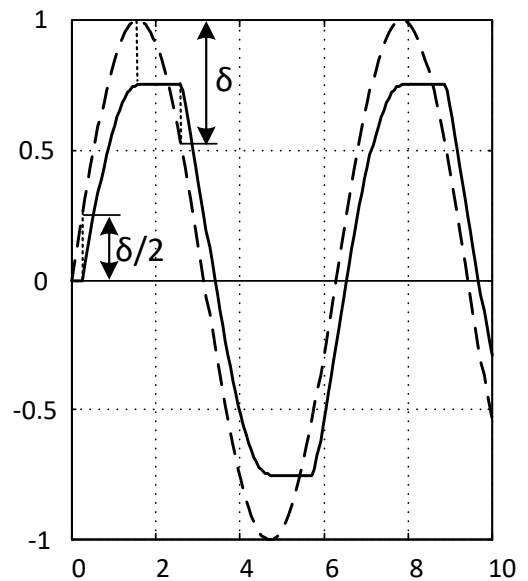


Рис. 3.21. Реакція блоку *Backlash* на вхідну синусоїду

Блок *Relay (Реле з Гістерезисом)* дозволяє перемикати 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 гістерезису (див. рис. 3.21), а при  $x_{on} = x_{off}$  – ідеальне реле (див. рис. 3.22).

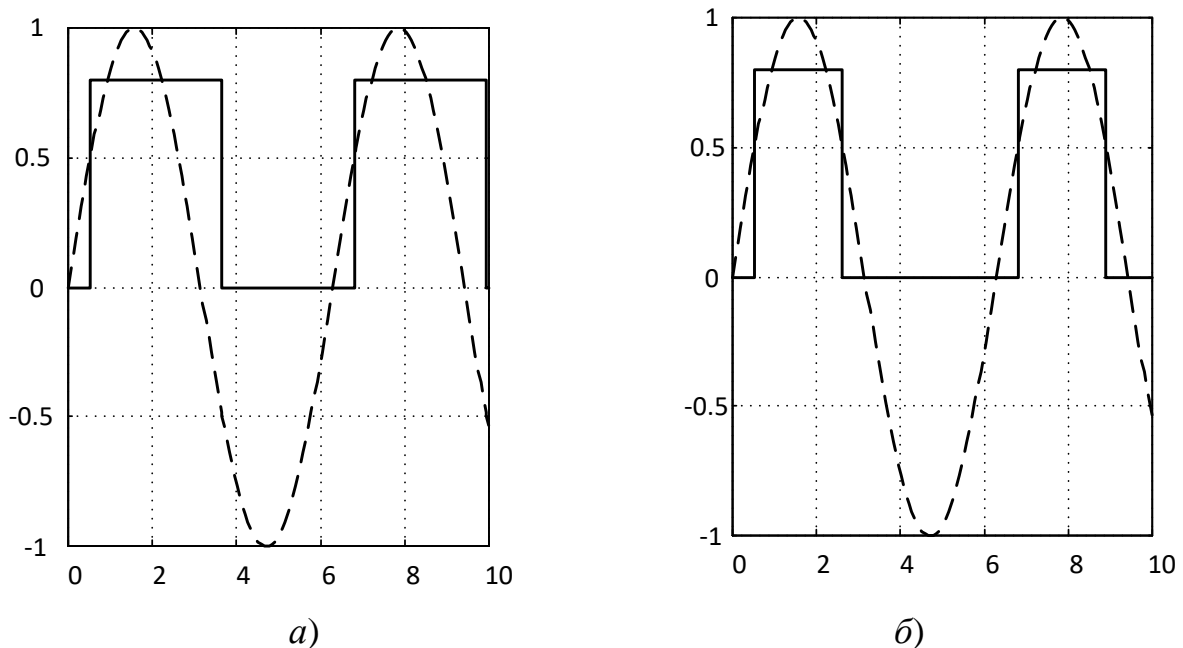


Рис. 3.22. Перетворення синусоїди ланкою *Relay*:  
 а)  $x_{on} = 0.5$ ,  $x_{off} = -0.5$ ; б)  $x_{on} = x_{off} = 0.5$

Для того, щоб рівні обмеження та межі зони нечутливості можна було б задавати не тільки константами, а і змінними сигналами у бібліотеці *Discontinuities* передбачено 3 динамічних блоки: *Saturation Dynamic*, *Dead Zone Dynamic* та *Rate Limiter Dynamic*.

### 3.5 Моделювання нелінійних функціональних перетворювачів з характеристиками, заданими в табличній формі

У природі існують багато об'єктів з нелінійними статичними характеристиками «вхід-вихід», для яких не знайдено аналітичного опису. Зазвичай такі характеристики знімають експериментально. Інформація про них має ви-

гляд таблиць, у яких кожному дискретному значенню вхідного сигналу відповідає певне значення вихідного сигналу. Отже, інформація про таку нелінійну залежність подається у вигляді двох одномірних масивів однакової довжини:

$$y_i = f(u_i); \quad i=1, 2, \dots, n. \quad (3.16)$$

Кожна пара чисел цих масивів визначає одну точку нелінійності. Абсциси точок таких залежностей називають вузлами таблиці. Задача полягає у розрахунку вихідних сигналів таких нелінійних залежностей у точках, відрізнених від вузлових.

При розв'язанні цієї задачі використовують такі методи обчислювальної математики як інтерполювання та апроксимацію.

**Інтерполювання рухомими степеневими поліномами** різних порядків в пакеті MATLAB здійснює функція `interp1`:

$$y = \text{interp1}(U_t, Y_t, u, \text{method}),$$

де  $U_t, Y_t$  – вектори аргументів та значень табличної функції;

$u$  – точка або масив точок, у яких необхідно обчислити значення інтерполюваної функції  $y$ ;

`method` – метод інтерполювання, може приймати наступні значення:

- 'linear' – лінійна інтерполяція (рухомими поліномами першого порядку);
- 'pchip' – кубічна інтерполяція (рухомими поліномами третього порядку);
- 'spline' – кубічна сплайн-інтерполяція (рухомими поліномами третього порядку, що забезпечують неперервність не тільки функції, але і її першої та другої похідних у вузлових точках);
- 'nearest' – інтерполяція за найближчим сусіднім вузлом (поліномами нульового порядку).

У *Simulink* інтерполяцію рухомими степеневими поліномами нульового та першого порядку, та інтерполяцію, та інтерполяцію кубічними сплайнами здійснює блок *ID-Look-Up Table (Одномірний Функціональний Перетворювач)* з параметрами *Breakpoints 1* (табличні значення вхідного сигналу) та *Table data* (табличні значення вихідного сигналу). Вектор аргумен-

тів повинен бути монотонно зростаючим.

Вікно введення параметрів цього блоку зображено на рис. 3.21.

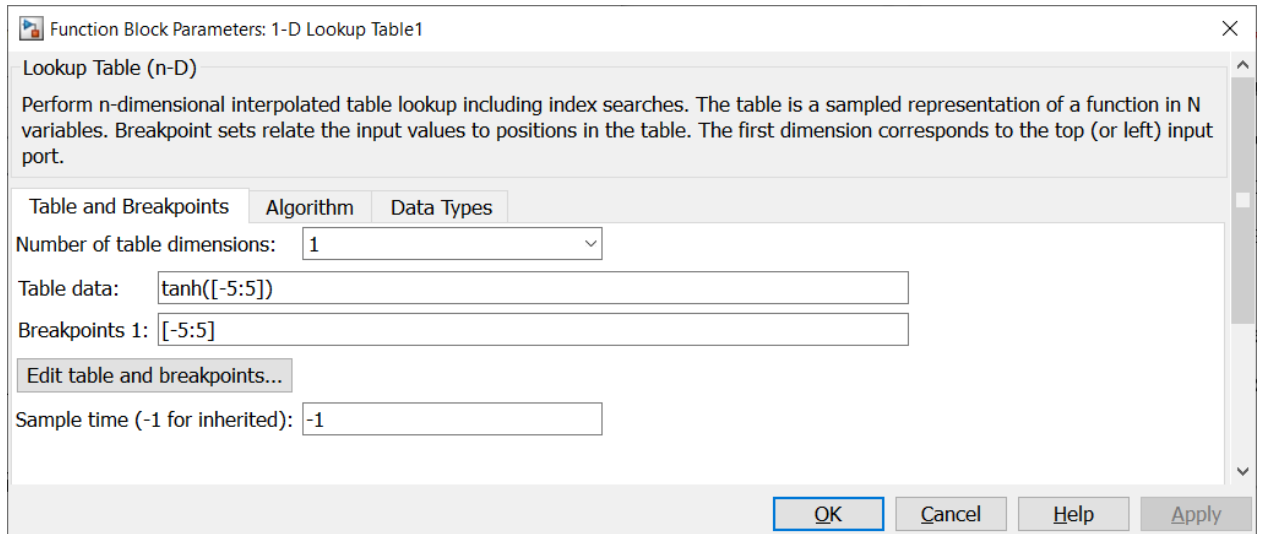


Рис. 3.21. Вікно введення параметрів блоку *1D-Look-Up Table*

Тип інтерполювання обирається у вкладці *Algorithm* (див. рис. 3.22) з випадаючого меню *Interpolation method*, в якому в початковому стані встановлено метод *Linear*, що означає інтерполяцію рухомими поліномами першого порядку, яку ще називають *кусочно-лінійною апроксимацією*.

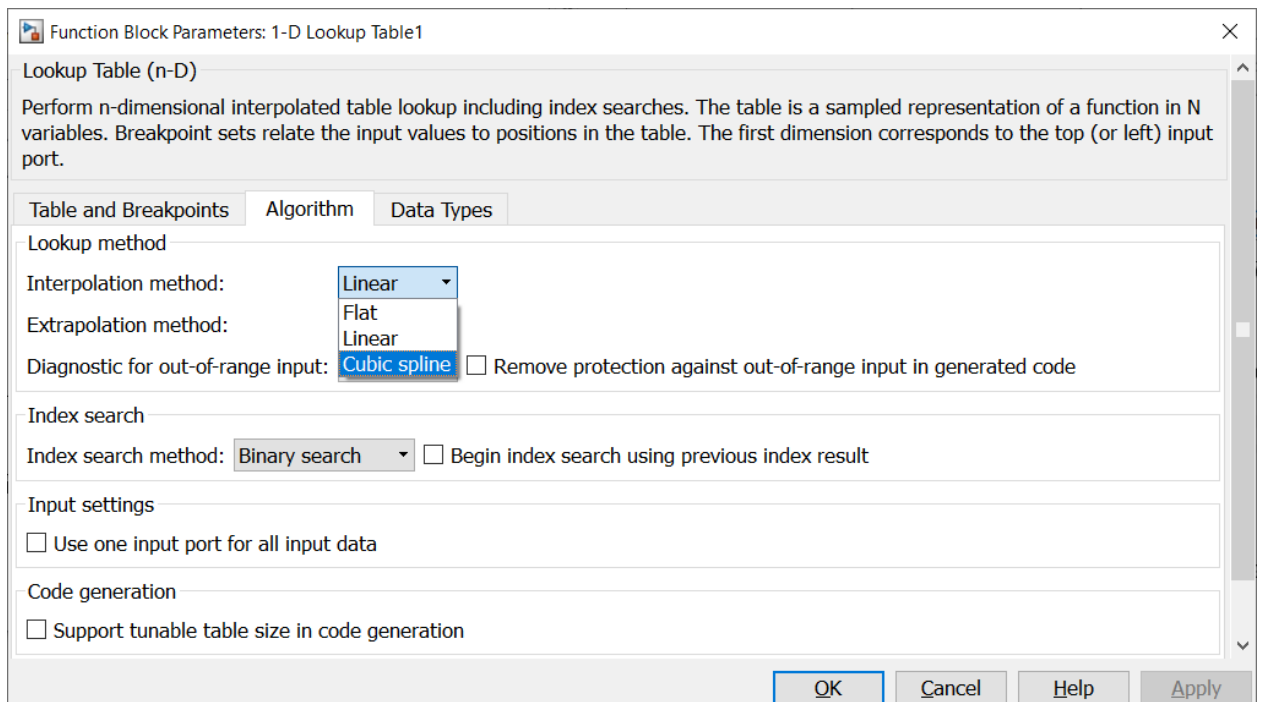


Рис. 3.22. Вкладка *Algorithm* вікна параметрів блоку *1D-Look-Up Table*

У цьому випадку блок розраховує вихідний сигнал  $y = f(u)$  за формулою

$$y(u) = \frac{y_i - y_{i-1}}{u_i - u_{i-1}} \cdot (u - u_{i-1}) + y_{i-1}, \quad (3.17)$$

де  $i$  – номер вузла, найближчого до поточного значення аргументу праворуч, тобто

$$u_{i-1} \leq u \leq u_i. \quad (3.18)$$

При виборі методу *Flat* виконується інтерполювання рухомими поліномами нульового порядку, прив'язаними до найближчої ліворуч вузлової точки:

$$y(u) = y_{i-1}, \quad u_{i-1} \leq u \leq u_i, \quad (3.19)$$

а при виборі методу *Cubic spline* – спеціальними рухомими поліномами третьої степені, що зветься кубічними сплайнами. Їх особливість полягає у тому, що вони забезпечують не тільки рівність значень табличної та апроксимуючої функцій у вузлових точках, а ще й неперервність першої та другої похідних у цих точках, що забезпечує плавність вихідної функції

Вихідні значення для вхідних сигналів, що попадають за межі заданих у таблиці, виводять за допомогою екстраполювання по перших (екстраполювання ліворуч) або останніх (екстраполювання праворуч) табличних точках, кількість яких на одиницю перевищує порядок інтерполяції. Тип екстраполяції визначається параметром *Extrapolation method*, що може приймати значення: *Clip* (обрізати), *Linear* (продовжити лінійно) або *Cubic spline* (продовжити за рівнянням кубічного сплайну).

Піктограма блоку *1D-Lookup Table* відображає графік залежності вхід-вихід тільки для кусково-лінійної апроксимації. У цьому випадку при зміні значень параметрів у діалоговому вікні блоку графік автоматично перемальовується. Якщо значення параметру змінюється в робочому середовищі пакета *MATLAB*, то іконка блоку перемальовується тільки після відкриття його діалогового вікна. При інтерполяції кубічними сплайнами та кусково-ступінчатій апроксимації іконка блоку має вигляд, який не залежить від па-

раметрів табличної функції (див. рис. 3.23).

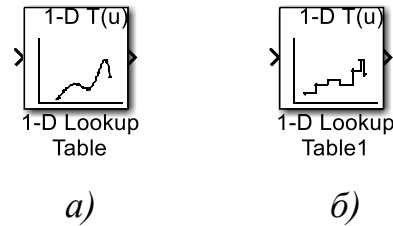


Рис. 3.23. Вигляд іконки блоку *1D-Look-Up Table* при виборі методу інтерполяції *Cubic spline* (а) та *Flat* (б)

Недоліком інтерполяції кубічними сплайнами є складність визначення їх коефіцієнтів. Більш простим методом отримання доволі плавних функцій є проста кубічна інтерполяція *Pchip*, яку можна реалізувати за допомогою блоку *Interpreted MATLAB Fn*, звертаючись через нього до функції

$$\text{interp1} (U_t, Y_t, U, 'pchip').$$

Якщо звернутися у блоці *Interpreted MATLAB Fn* до функції

$$\text{interp1} (U_t, Y_t, u, 'nearest', 'extrap'),$$

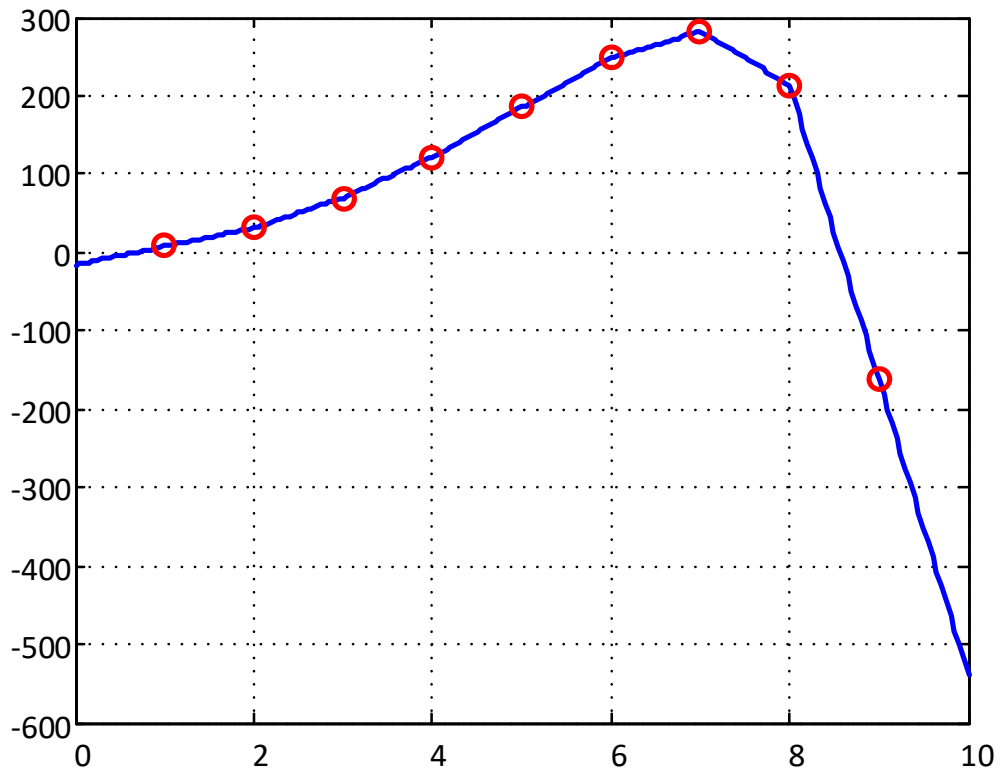
то отримаємо ступінчасту інтерполяцію поліномами 0-го порядку типу «найближчий сусід»:

$$y(u) = \begin{cases} y_{i-1}, & \text{якщо } (u - u_{i-1}) < (u_i - u), \\ y_i, & \text{якщо } (u - u_{i-1}) \geq (u_i - u); \end{cases} \quad u_{i-1} \leq u \leq u_i, \quad (3.20)$$

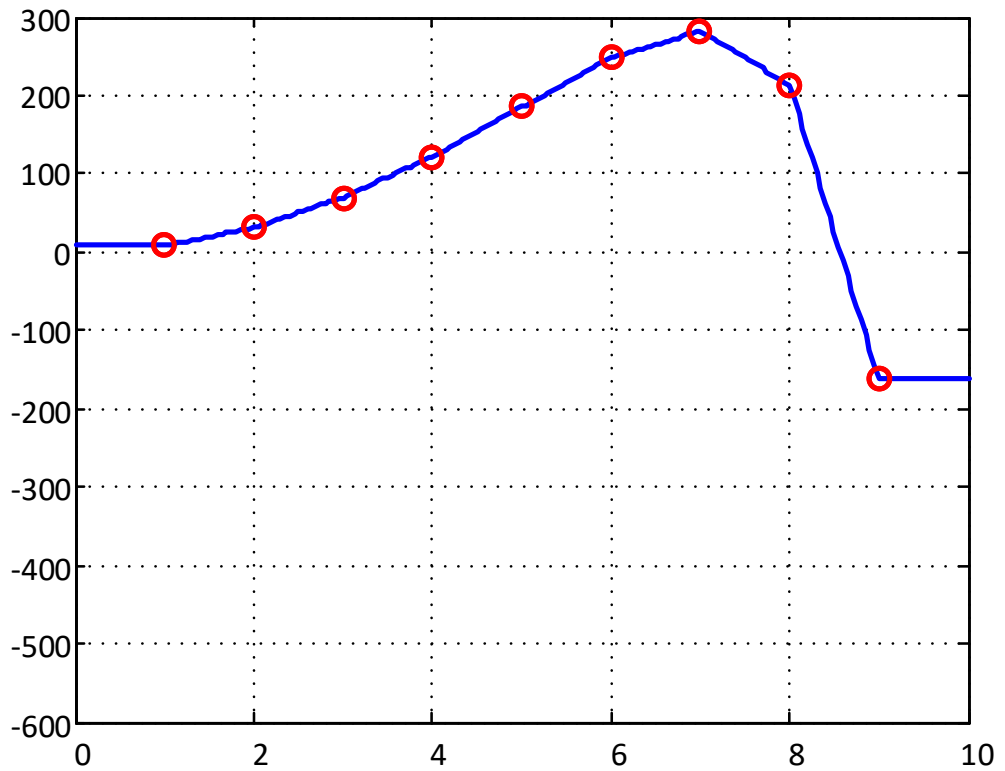
яка дещо відрізняється від ступінчастої інтерполяції (3.19).

На рис. 3.24-3.27 подані графіки описаних вище варіантів інтерполяції та екстраполяції деякої нелінійної функції, заданої у вигляді таблиці на інтервалі  $u \in [1, 9]$ .

Кружечками на графіки зображено вузлові точки табличної функції. Вхідний сигнал сформовано блоком *Clock*, а час моделювання дорівнює 10, отже поточне значення вхідного сигналу повністю охоплює інтервал інтерполяції і дає можливість здійснювати екстраполяцію.



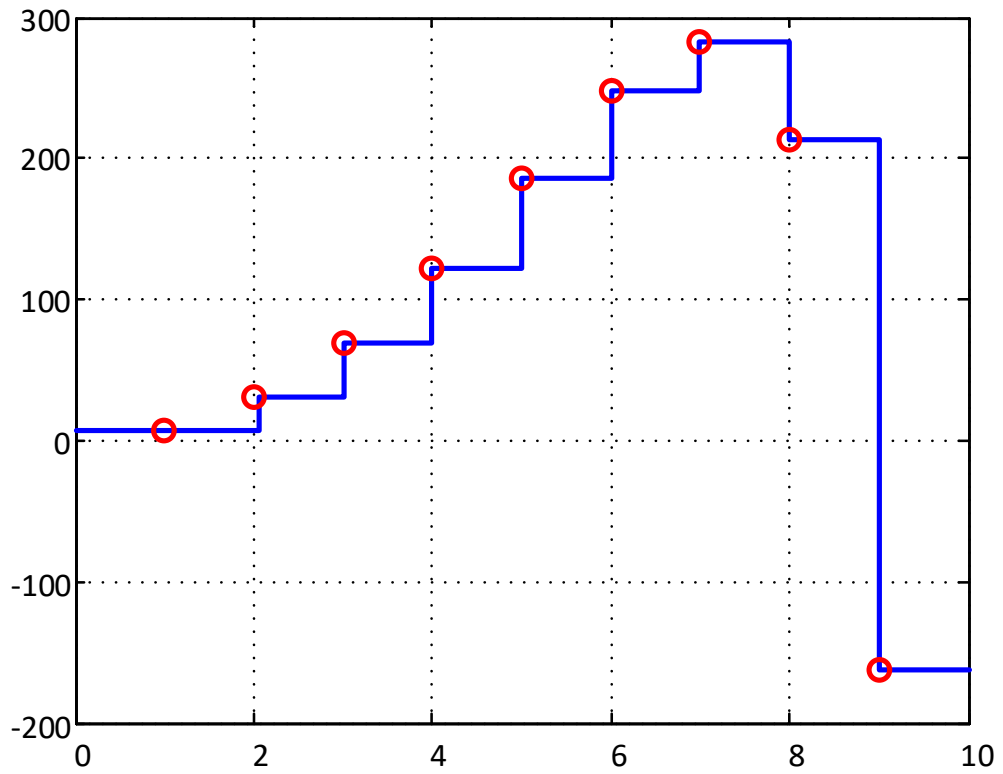
a)



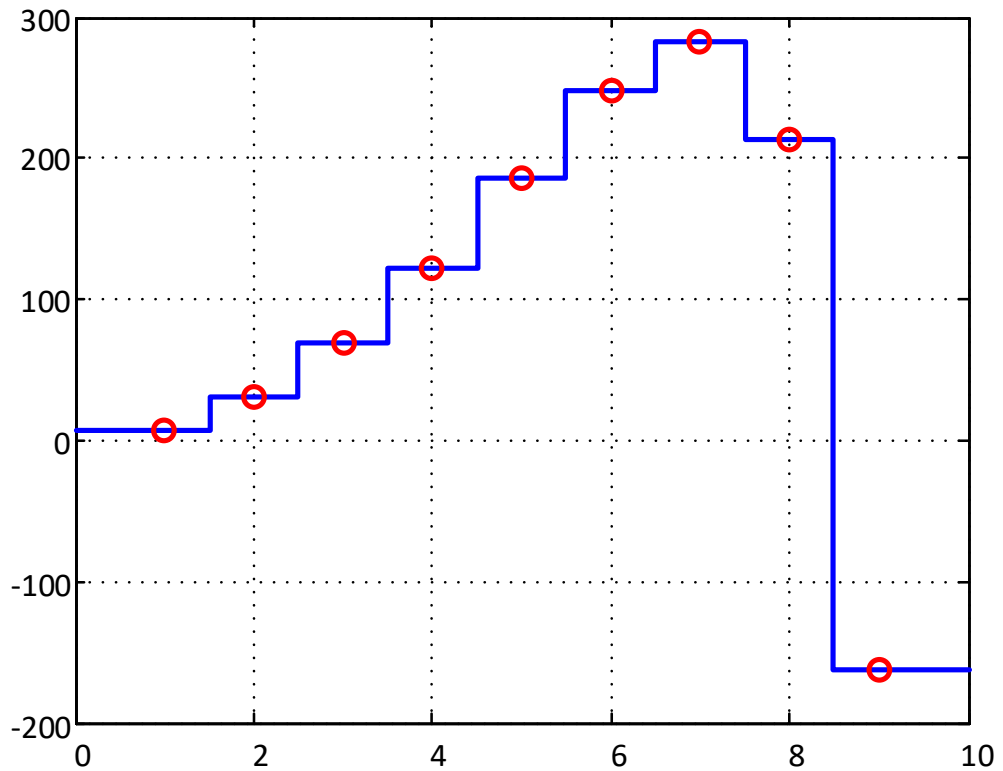
б)

Рис. 3.24. Графіки з виходу блоку *1D-Look-Up Table* при  
*Interpolation method = Linear*:  
а) *Extrapolation method = Linear*; б) *Extrapolation method = Clip*





a)



б)

Рис. 3.25. Кусково-ступінчата апроксимація, здійснена блоком *1D-Look-Up Table* при *Interpolation method = Flat* (a) та блоком *Interpreted MATLAB Fn* при зверненні `interp1 (Ut, Yt, u, 'nearest', 'extrap')` (б)

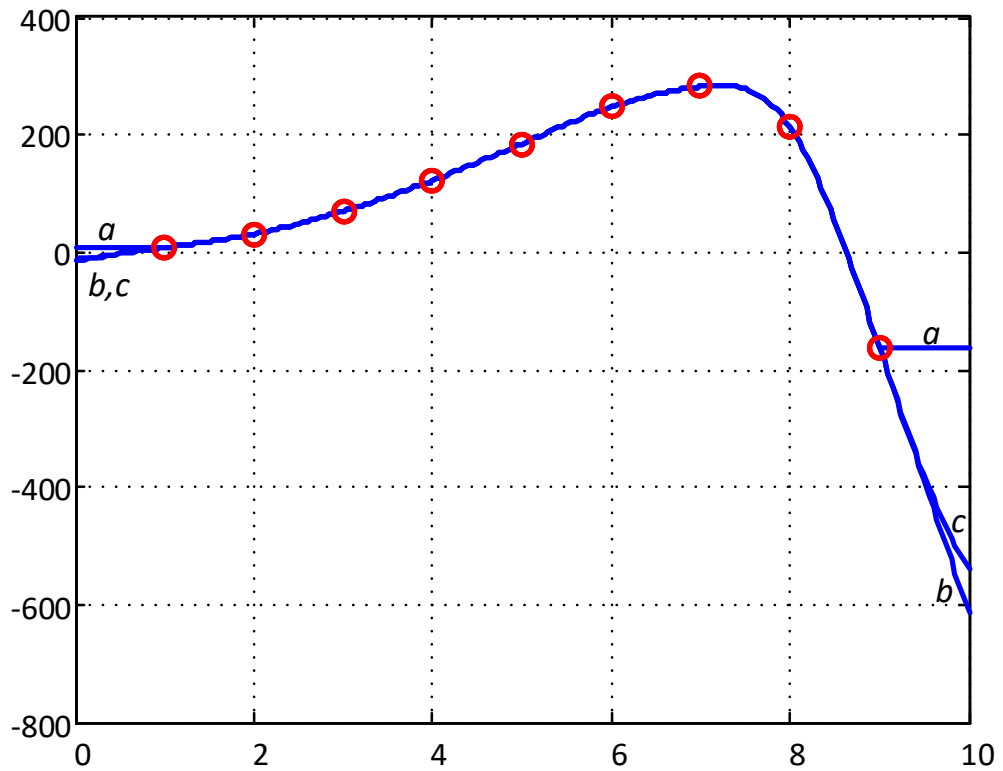


Рис. 3.26. Інтерполювання кубічними сплайнами, здійснене блоком *1D-Look-Up Table* при *Interpolation method = Cubic spline* та *Extrapolation method = Clip (a), Linear (b), Cubic spline (c)*

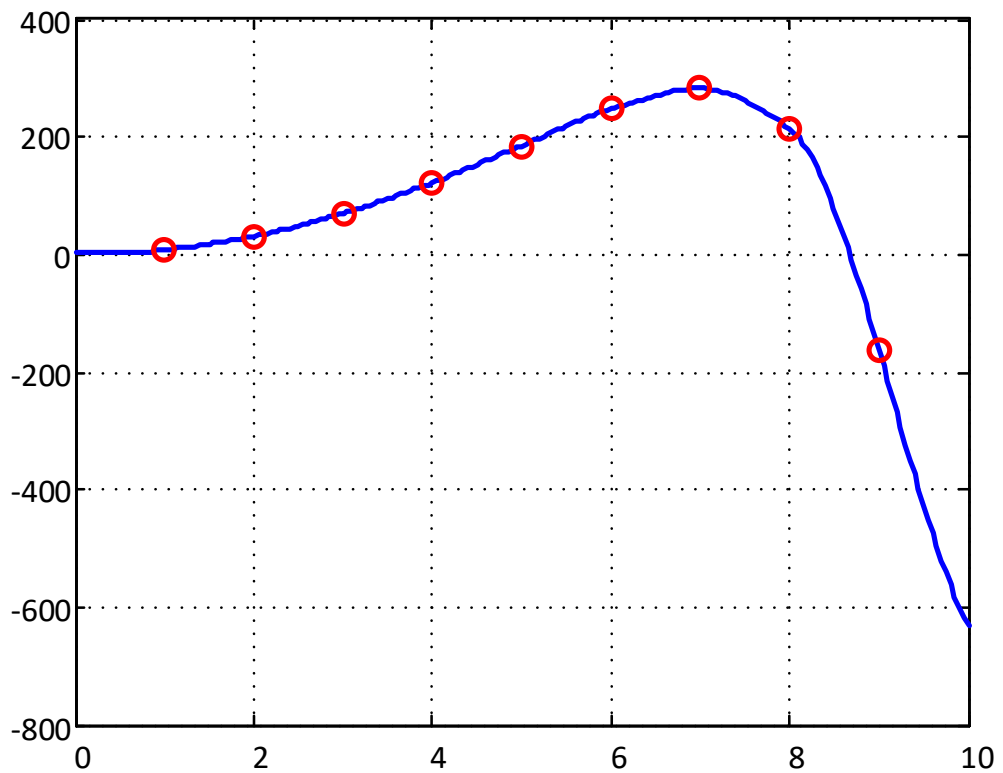


Рис. 3.27. Кусково-кубічне інтерполювання, здійснене блоком *Interpreted MATLAB Fn* при зверненні *interp1 (Ut, Yt, u, 'pchip')*

**Апроксимація** табличної функції (3.16) **методом найменших квадратів** полягає у знаходженні таких параметрів деякої аналітичної функції, які забезпечують мінімальну суму квадратичних відхилень апроксимуючої функції від заданої табличної у вузлових точках:

$$\Phi = \sum_{i=1}^n (F(u_i) - y_i)^2. \quad (3.21)$$

Якщо в якості функції, що апроксимує, взяти степеневий поліном

$$F(x) = P_k(x) = C_0 + C_1x + \dots + C_kx^k = \sum_{j=0}^k C_jx^j, \quad (3.22)$$

то задача зводиться до визначення вектору коефіцієнтів  $C=(C_0, C_1, \dots, C_k)$ . Цю операцію в пакеті MATLAB здійснює функція `polyfit`:

$$C = \text{polyfit}(U_t, Y_t, k),$$

де  $C$  – вектор-рядок коефіцієнтів апроксимуючого полінома, упорядкований за зменшенням степені  $x$  ;

$k$  – порядок апроксимуючого полінома.

У пакеті *MATLAB* значення степеневого поліному з коефіцієнтами  $C$  у точках  $U$  вираховує функція

$$y = \text{polyval}(C, u),$$

де  $u$  – точки, у яких треба обчислити значення СП;

$y$  – масив значень степеневого поліному, розмірність і розмір якого співпадає з відповідними параметрами вхідного аргументу  $u$ .

У *Simulink* останню операцію здійснює блок *Polynomial* (див. підрозділ 3.3).

Результати такої апроксимації поліномами 3-го, 4-го та 8-го порядків показані на рис. 3.28. Як бачимо, при збільшенні порядку степеневого поліному  $k$  точність апроксимації підвищується. При  $k = n-1$  ( $n$  – кількість табличних точок) апроксимуючий поліном стає глобальним інтерполянтом, тобто значення апроксимуючої функції у вузлових точках збігаються зі значеннями вихідної табличної функції.

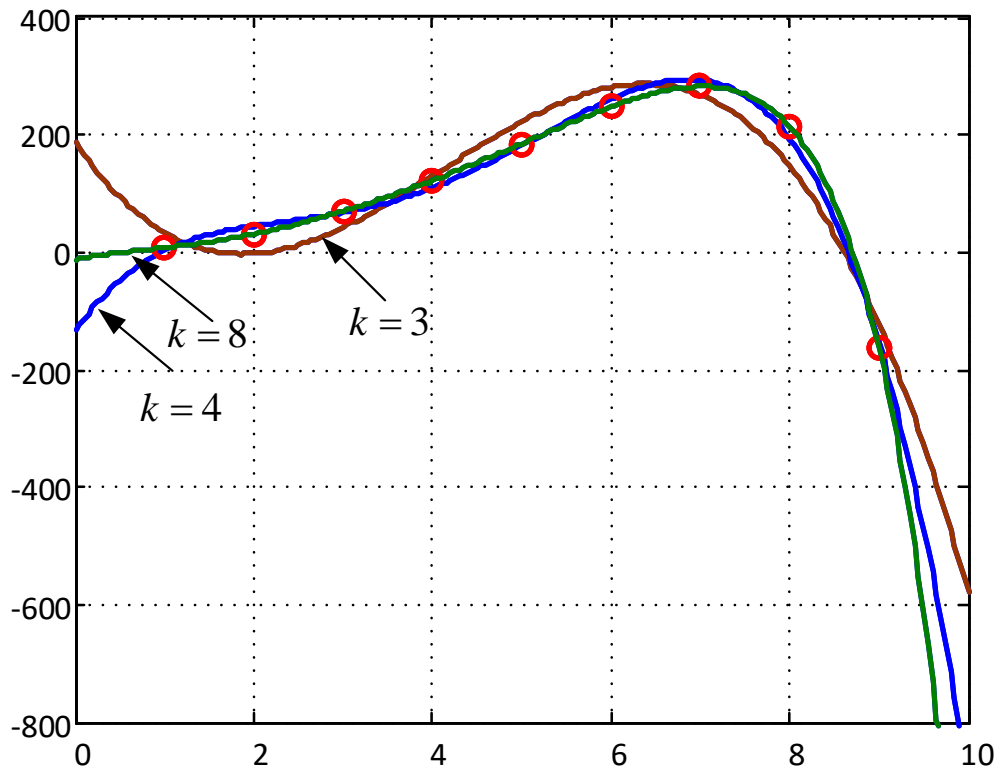


Рис. 3.28. Апроксимація табличної функції степеневими поліномами 3-го, 4-го та 6-го порядків методом найменших квадратів

### 3.6 Моделювання періодичних нелінійностей, заданих таблично або аналітично на періоді

Якщо для повторюваної нелінійної залежності є відомим її математичний опис або таблиця значень абсцис і ординат на першому періоді

$$y = f(u), \quad 0 \leq u \leq \Delta u \quad (3.23)$$

або

$$y_i = f(u_i), \quad i = 1, 2, \dots, n, \quad u_1 = 0, \quad u_n = \Delta u, \quad (3.24)$$

то для періодичного її повторення достатньо на вхід моделі, що формує залежність (6.22) або апроксимовані значення залежності (6.23), подати залишок від ділення вхідного сигналу  $u$  на його період  $\Delta u$ :

$$u_p = \text{rem}(u, \Delta u). \quad (3.25)$$

Прикладом такого підходу є блок *Repeating Sequence* бібліотеки вхідних блоків *Sources*, структуру якого можна побачити, зазирнувши під маску. Цей блок уявляє собою окремий випадок досліджуваної повторюваної послідовності, в якому вхідним сигналом є час, а закон зміни повторюваної послі-

довності на першому періоді утворює блок *ID Look-up-Table*. На рис. 3.29 наведено модель повторення залежності  $y = (u - 1)^2$  з періодом  $\Delta u = 2$ , а на рис. 3.30 – статична характеристика вхід-вихід цієї моделі.

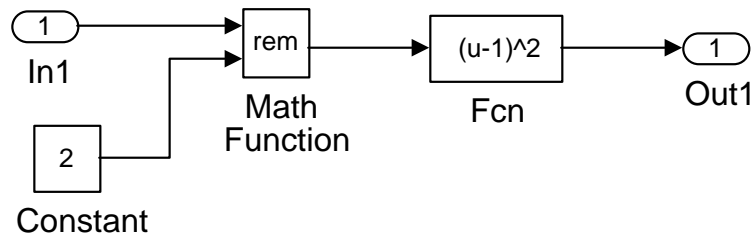


Рис. 3.29. Модель, що утворює повторювану послідовність парабол

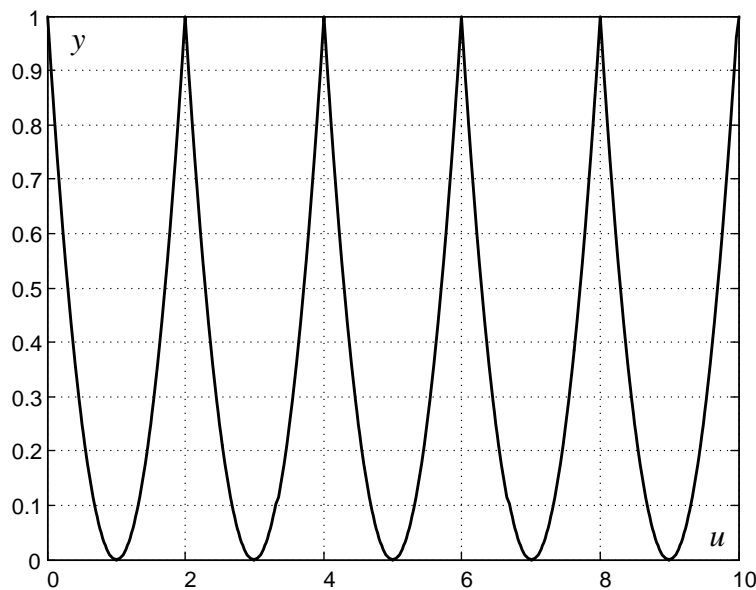


Рис. 3.30. Результат симуляції моделі рис. 3.29

### 3.7 Завдання

1. Навести приклади використання блоків множення та ділення.
2. Сформувати сигнали за формулами  $U = f(t)$ , поданими у табл. 3.2, за допомогою блоку *Fn*.
3. Одержати перехідні процеси перетворення синусоїдального сигналу типовими нелінійностями та статичні характеристики цих блоків.
4. Реалізувати різні способи інтерполяції та екстраполяції табличних нелінійностей, заданих в табл. 3.3.

Таблиця 3.2

№ вар	$U = f(t), t \in [0, 10]$	№ вар	$U = f(t), t \in [0, 10]$
1	$U = 150 \sin(60t + \pi/10)$	11	$U = 200 \cos(\ln(t+1))$
2	$U = 125 \cos(50t - \pi/12)$	12	$U = 100 \sin(10t) e^{-t/4}$
3	$U = 150 \frac{1}{1 - 0.5 \sin t}$	13	$U = 100 \frac{t^2}{1+t^3}$
4	$U = 100 e^{-t/4}$	14	$U = 200 \cos^2(t) \cos(2t)$
5	$U = 200 \sqrt{t}$	15	$U = 200 \sin^2(t) \sin(2t)$
6	$U = 100 \sqrt[3]{t}$	16	$U = e^{\sin 2t} \sin t$
7	$U = \frac{100}{t+0.1}$	17	$U = \frac{150t}{t+1}$
8	$U = 200 \sqrt[3]{t} \cdot \sin(3t)$	18	$U = 100 \sqrt[2]{t} \cdot \cos(2t)$
9	$U = 3 \sin(4t) \cdot e^{-\frac{t}{4}}$	19	$U = 5 \cos(2t) \cdot e^{-\frac{t}{2}}$
10	$U = 100 \sin(\lg(t+1))$	20	$U = 10t^2 + 50t + 4$

Таблиця 3.3

№ вар.	Табличні функції									
	$x_t$	-1	1	3	5	7	9	11	13	15
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

5. Здійснити апроксимацію степеневими поліномами табличних нелінійностей, заданих в табл. 4.3, методом найменших квадратів для трьох значень порядку апроксимуючого поліному.

6. Утворіть повторювані на періоді  $T = 2$  функції часу  $e^t$ ,  $e^{-t}$ ,  $t^2$ ,  $\sqrt{t}$ ,  $2^t$ ,  $\min(t^2, 2)$ ,  $\min(\sqrt{t}, 0.2)$ .

### 3.8 Методичні рекомендації

#### 1. При виконанні пункту 3

- зберіть 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 поєднайте мультіплексором з вхідним синусоїдальним сигналом, і отриманий векторний сигнал приєднайте до осцилографу та блоку *Out*;
- для фіксації статичних характеристик типових нелінійностей додайте до моделі блок *XY Graph*;

- оберіть амплітуду і частоту синусоїди та задайте час моделювання рівним 2-м періодам синусоїди;
- шляхом керування блоком *Multiport Switch* у ручному режимі налаштуйте по черзі кожен тип нелінійності, узгоджуючи її параметри з обраною амплітудою синусоїди у такий спосіб, щоб результати моделювання наочно демонстрували особливості кожного блоку;
- розробіть 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.

Приклад виконання цього пункту для трьох типових нелінійностей подано на рис. 3.31 (модель) та рис. 3.32-3.34 (графіки).

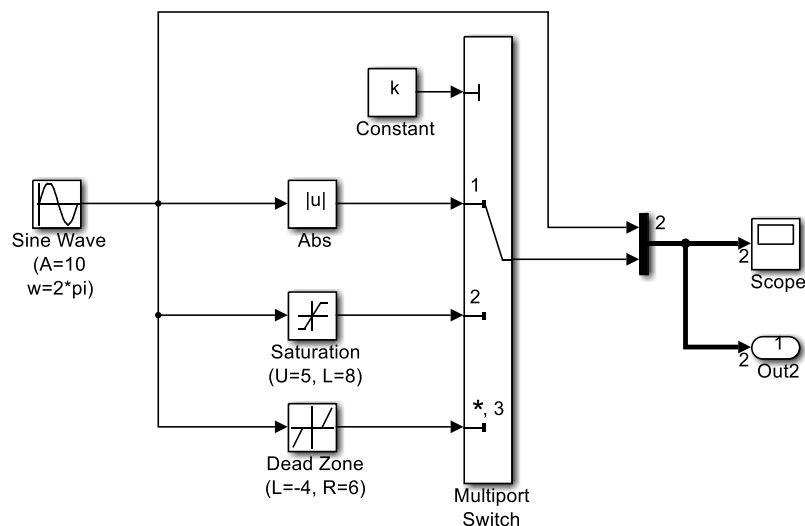


Рис. 3.31. Приклад моделі *NonLin.slx* до виконання завдання 3 для трьох типових нелінійностей

Текст програми, що забезпечила симуляцію та побудову графіків, виглядає так:

```

for k=1:3
    sim('NonLin')
    figure(k)
    subplot(2,1,1)
    plot(tout, yout), grid on
    subplot(2,1,2)
    plot(yout(:,1), yout(:,2)), grid on, ylim([-10 10])
end

```



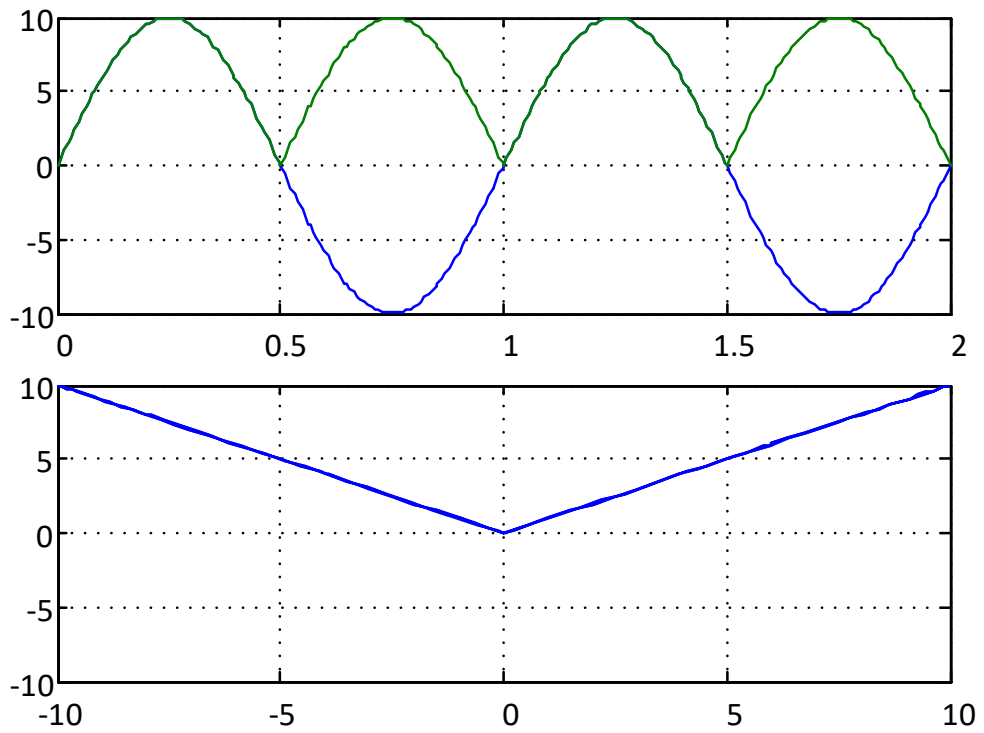


Рис. 3.32. Реакція на синусоїду і статична характеристика блоку *Abs*

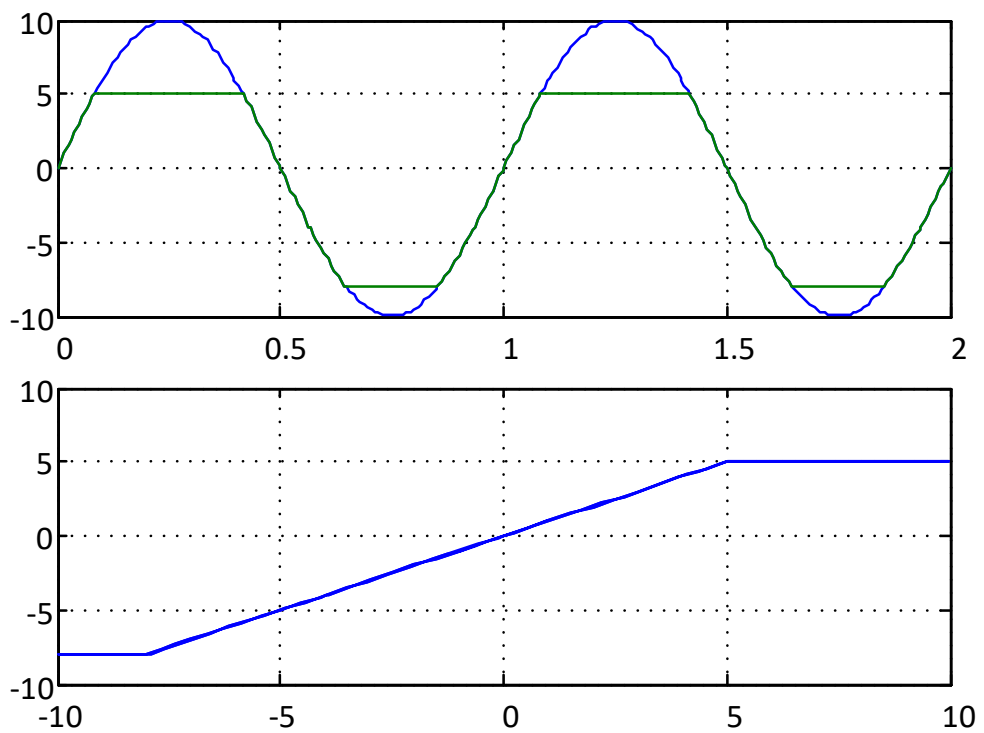


Рис. 3.33. Реакція на синусоїду і статична характеристика блоку *Saturation*

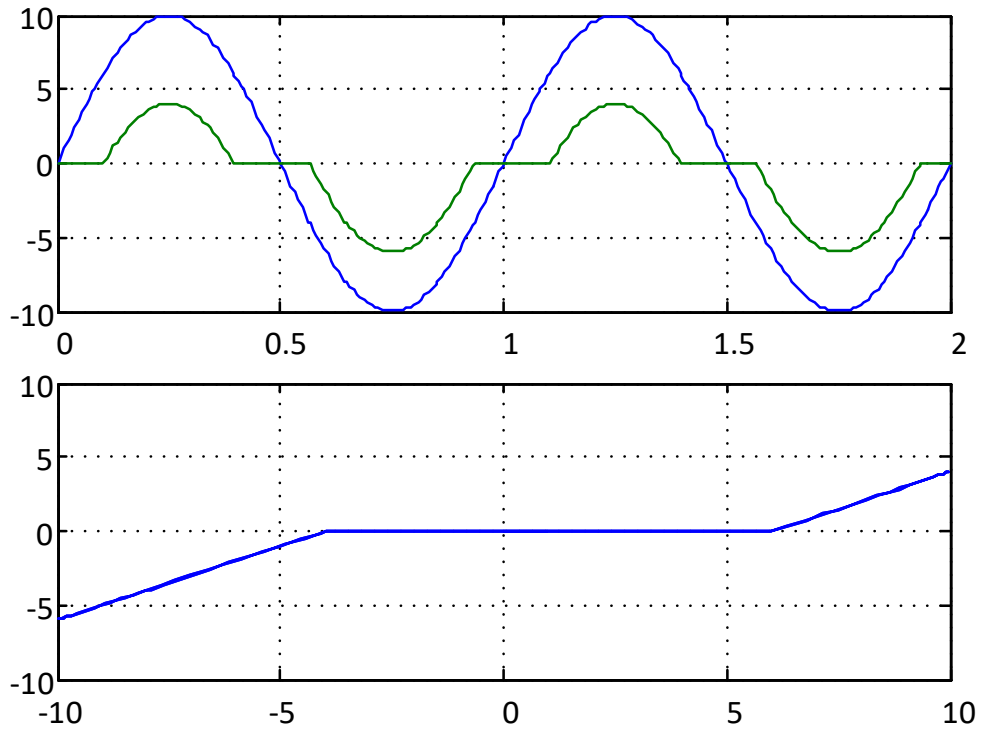


Рис. 3.34. Реакція на синусоїду і статична характеристика блоку *Dead Zone*

2. При виконанні пункту 5 час моделювання і вхідний сигнал змінюйте так, щоб діапазон його зміни охоплював і дещо перевищував (ліворуч та праворуч) діапазон зміни аргументів табличної функції. Цього можна досягти застосуванням блоку *Ramp* з обмеженням вихідного сигналу.

### Лабораторна робота №4

## ЗНАЙОМСТВО З БІБЛІОТЕЧНИМИ НЕПЕРЕРВНИМИ ДИНАМІЧНИМИ ЛАНКАМИ ПРОГРАМИ *Simulink*

*Мета роботи:* навчитися моделювати в середовищі *Simulink* неперервні лінійні динамічні об'єкти з різними формами їх математичного опису та перетворювати математичний опис з однієї форми в іншу.

### 4.1 Форми математичного опису неперервних лінійних систем

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

Неперервні *LTI*-системи описуються звичайними лінійними диференціальними рівняннями з постійними коефіцієнтами.

За кількістю вхідних та вихідних величин системи розподіляються на такі групи:

- *SISO (Single Input Single Output)* – системи, що мають один вхід та один вихід;
- *SIMO (Single Input Many Output)* – системи, що мають один вхід та багато виходів;
- *MIMO (Many Input Many Output)* – системи, що мають багато входів та багато виходів.

В середовищі *Simulink* застосовують три форми подання звичайних ДР, що описують неперервні *LTI*-системи:

- *Transfer Function Polynomial* – передавальні функції у поліноміальній формі;
- *Zero-Pole-Gain* – передавальні функції, розкладені за нулями та полюсами;
- *State Space* – математичний опис у просторі станів.

Перші 2 форми в теорії автоматичного керування (ТАК) відносяться до класичних форму запису, які застосовують для опису *SISO*-систем.

Третя форма використовується у сучасній теорії керування і є найбільш універсальною, тому що може описати як *SISO*-, так і *SIMO*- та *MIMO*-системи.

Нехай довільна неперервна стаціонарна динамічна система  $n$ -го порядку має  $k$  входів та  $r$  виходів. Її можна описати системою лінійних диференціальних рівнянь

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (4.1)$$

з початковими умовами

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (4.2)$$

та лінійним матричним рівнянням виходу

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (4.3)$$

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  $r \times n$ ;

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

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

Для *SIMO*-систем ( $k = 1$ ) матриці  $\mathbf{B}$  і  $\mathbf{D}$  вироджуються у вектори-стовпці:

$$\mathbf{B}=\mathbf{b}=[b_1 \quad b_2 \quad \dots \quad b_n]^T, \quad \mathbf{D}=\mathbf{d}=[d_1 \quad d_2 \quad \dots \quad d_n]^T, \quad (4.4)$$

а для *SISO*-систем ( $k = 1, r = 1$ ) – матриця  $\mathbf{B}$  залишається вектором-стовпцем, матриця  $\mathbf{C}$  стає вектором-рядком розміром  $1 \times n$ , а матриця  $\mathbf{D}$  вироджується у скаляр:

$$\mathbf{C}=\mathbf{c}=[c_1 \quad c_2 \quad \dots \quad c_n], \quad \mathbf{D}=d. \quad (4.5)$$

Для переходу до операторної форми запису рівнянь (4.1), (4.2) в них треба замінити операцію диференціювання вектору станів  $\mathbf{x}(t)$  операцією множення на оператор Лапласа  $s$  на зображення цього вектору  $\mathbf{x}(s)$ :

$$s\mathbf{x}(s) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s), \quad (4.6)$$

$$\mathbf{y}(s) = \mathbf{C}\mathbf{x}(s) + \mathbf{D}\mathbf{u}(s). \quad (4.7)$$

Математичному опису у формі (4.6), (4.7) відповідає структурна схема, що зображена на рисунку 4.1.

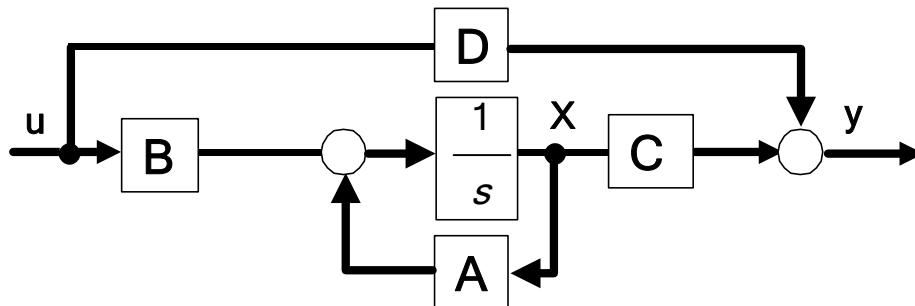


Рис. 4.1. Структурна схема *LTI-MIMO*-системи у просторі стану

З рівнянь (4.6), (4.7) можна знайти матричні передавальні функції (ПФ) від вектора входу до вектору змінних стану розміром  $n \times k$  та до вектору виходу розміром  $r \times k$ :

$$\mathbf{W}_x(s) = \frac{\mathbf{x}(s)}{\mathbf{u}(s)} = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \frac{Adj(s\mathbf{I} - \mathbf{A})}{det(s\mathbf{I} - \mathbf{A})}\mathbf{B}, \quad (4.8)$$

$$\mathbf{W}_y(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \mathbf{C} \frac{Adj(s\mathbf{I} - \mathbf{A})}{det(s\mathbf{I} - \mathbf{A})}\mathbf{B} + \mathbf{D}, \quad (4.9)$$

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

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

$$G(s) = \det(s\mathbf{I} - \mathbf{A}) = s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_1s + \alpha_0, \quad (4.10)$$

а рівняння

$$G(s) = 0$$

– *характеристичним рівнянням*.

Корені характеристичного рівняння

$$\mathbf{P} = [p_1 \quad p_2 \quad \dots \quad p_n] \quad (4.11)$$

називають *полюсами системи* або *власними числами матриці A*.

Чисельник кожної з скалярних ПФ, що є компонентами матричних ПФ (4.9) та (4.10), також уявляє собою степеневий поліном відносно змінної Лапласа, який називають *поліномом впливу*:

$$H(s) = \beta_m s^m + \beta_{m-1} s^{m-1} + \dots + \beta_1 s + \beta_0. \quad (4.12)$$

Його корені

$$\mathbf{Z} = [z_1 \quad z_2 \quad \dots \quad z_m] \quad (4.13)$$

звуть *нулями системи*.

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

$$m \leq n. \quad (4.14)$$

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 елемента матриці обходу  $\mathbf{D}$ .

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

$$W_{ij}(s) = \frac{y_i(s)}{u_j(s)} = \frac{H_{ij}(s)}{G(s)} = \frac{\beta_m s^m + \beta_{m-1} s^{m-1} + \dots + \beta_1 s + \beta_0}{s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0}. \quad (4.15)$$

Як бачимо, ХП, визначений за формулою (4.10), і застосований у ПФ (4.15) 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,

$$W_{ij}(s) = \frac{y_i(s)}{u_j(s)} = k \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + 1}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + 1} = k \frac{H_{0ij}(s)}{G_0(s)}. \quad (4.16)$$

Коефіцієнти ПФ (4.16) і (4.17) пов'язані між собою співвідношеннями:

$$a_n = \frac{1}{\alpha_0}, \quad a_i = \frac{\alpha_i}{\alpha_0}, \quad b_i = \frac{\beta_i}{\beta_0}, \quad k = \frac{\beta_0}{\alpha_0}. \quad (4.17)$$

Скалярній передавальній функції (4.16) відповідає ДР

$$\begin{aligned} \frac{d^n y_i(t)}{dt^n} + \alpha_{n-1} \frac{d^{n-1} y_i(t)}{dt^{n-1}} + \dots + \alpha_1 \frac{dy_i(t)}{dt} + \alpha_0 y_i(t) = \\ = \beta_m \frac{d^m u_j(t)}{dt^m} + \beta_{m-1} \frac{d^{m-1} u_j(t)}{dt^{m-1}} + \dots + \beta_1 \frac{du_j(t)}{dt} + \beta_0 u_j(t). \end{aligned}$$

Використовуючи розкладення поліномів у чисельнику та знаменнику передавальної функції (4.16) або (4.17) на множники, отримуємо математичний опис у формі *Zero-Pole-Gain*

$$W_{ij}(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}, \quad (4.18)$$

де

$$K = \beta_m = k \frac{b_m}{a_n}. \quad (4.19)$$

## 4.2 Перетворення математичного опису з форми *Transfer Function* у форму *State Space*

Для такого перетворення зручно використовувати ПФ у формі (4.15). В теорії автоматичного керування (ТАК) виведені залежності матриць стану, входу, виходу і обходу від коефіцієнтів такої ПФ:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_{n-2} & -\alpha_{n-1} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix}; \quad (4.20)$$

при  $m = n$

$$\mathbf{C} = [\beta_0 - \beta_n \alpha_0 \quad \beta_1 - \beta_n \alpha_1 \quad \dots \quad \beta_{n-1} - \beta_n \alpha_{n-1}] \quad , \quad d = \beta_0; \quad (4.21)$$

а при  $m < n$  –

$$\mathbf{C} = [\beta_0 \quad \beta_1 \quad \dots \quad \beta_m] \quad , \quad d = 0. \quad (4.22)$$

Для перетворення ланок першого другого порядків можна спочатку деталізувати їх, тобто перетворити їх структурні схеми у такий спосіб, щоб вони склались тільки з Інтеграторів, Пропорційних Ланок та Алгебраїчних суматорів, а потім записати рівняння стану та виходу у скалярній формі і виділити з них матриці стану, входу, виходу і обходу. Деталізовані структурні моделі таких ланок та їх опис у просторі станів подані в табл. 4.1.

### 4.3 Відомості про блоки бібліотеки *Continuous*

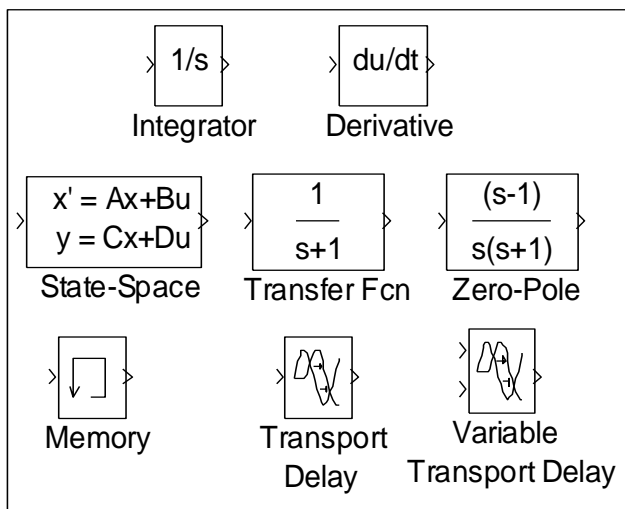


Рис. 4.2 – Бібліотека неперервних динамічних блоків (*Continuous*)

Бібліотека *Continuous*, представлена на рис. 4.2, 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*).



Таблиця 4.1

Передавальна функція	Деталізована структурна схема	Математичний опис у просторі станів
$\frac{k}{Ts + 1}$		$A = -1/T; \quad B = k/T;$ $C = 1; \quad D = 0$
$\frac{T_1 s}{T_2 s + 1}$		$A = -1/T_2; \quad B = 1/T_2;$ $C = -T_1/T_2; \quad D = T_1/T_2$
$k \frac{T_1 s + 1}{T_2 s + 1}$		$A = -1/T_2; \quad B = k/T_2;$ $C = 1 - T_1/T_2; \quad D = kT_1/T_2$
$= \frac{1}{T_1 T_2 s^2 + T_1 s + 1}$		$A = \begin{bmatrix} 0 & 1/T_1 \\ -1/T_2 & -1/T_2 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ k/T_2 \end{bmatrix};$ $C = [1 \quad 0]; \quad D = 0$
$\frac{T_3 s}{T_1 T_2 s^2 + T_1 s + 1}$		$A = \begin{bmatrix} 0 & 1/T_1 \\ -1/T_2 & -1/T_2 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 1/T_2 \end{bmatrix};$ $C = [0 \quad T_3/T_1]; \quad D = 0$

Блок **Integrator (Інтегратор)** інтегрує вхідний скалярний  $u$  або векторний  $\mathbf{u} = [u_1, u_2, \dots, u_k]$  сигнал із заданими початковими умовами (*Initial condition*)

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{u}(t) dt \quad (4.23)$$

і має передавальну функцію

$$W_i(s) = \frac{1}{s}. \quad (4.24)$$

Блок **Derivative (Похідна)** виконує чисельне диференціювання вхідного сигналу:

$$y(t_i) = \frac{\Delta u_i}{\Delta t_i} = \frac{u_i - u_{i-1}}{t_i - t_{i-1}} \approx \frac{du}{dt}. \quad (4.25)$$

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

Перед початком моделювання значення вхідного сигналу дорівнює нулю.

Блок **Transfer Fcn (Передавальна Функція)** реалізує передавальну функцію лінійної неперервної динамічної ланки загального вигляду в поліноміальній формі:

$$W(s) = \frac{B_m(s)}{A_n(s)} = \frac{\sum_{i=0}^m b_i s^i}{\sum_{j=0}^n a_j s^j}, \quad m \leq n. \quad (4.26)$$

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 степені  $n$ . Порядок знаменника повинний бути більшим від порядку чисельника або дорівнювати йому. Блок має нульові початкові умови.

Блок *Zero-Pole (Нулі-Полюси)* являє собою ланку з передавальною функцією (4.18) у формі розкладення поліномів у чисельнику та знаменнику на множники з такими параметрами:

- $K$  – посилення (*Gain*);
- $Z$  – вектор нулів (*Zeros*);
- $P$  – вектор полюсів (*Poles*).

*При відсутності нулів їх задають пустою матрицею [].*

Нулі і полюси можуть мати не тільки дійсні, але і комплексно-спряжені значення.

*При наявності комплексно спряжених пар нулів або полюсів в піктограмі блоку співмножники передавальної функції виводяться не у вигляді добутку двох поліномів першого порядку з комплексними вільними членами, а у вигляді одного поліному другого порядку з дійсними коефіцієнтами.* Наприклад, якщо

$$p_{1,2} = [\alpha + j\beta, \alpha - j\beta], \quad (4.27)$$

то відповідний множник характеристичного поліному знаходиться як

$$G_{1,2}(s) = (s + \alpha - j\beta)(s + \alpha + j\beta) = (s + \alpha)^2 + \beta^2 = s^2 + 2\alpha s + (\alpha^2 + \beta^2). \quad (4.28)$$

Блок *State-Space (Простір Стану)* забезпечує моделювання неперервної динамічної ланки загального виду за її математичному описом у просторі стану (4.1)-(4.3). Параметрами цього блоку є матриці коефіцієнтів **A**, **B**, **C**, **D** та вектор початкових умов (*Initial conditions*).

Блок *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.

Вихідний сигнал блоку *Transport Delay (Чисте Запізнювання)* повторює вхідний сигнал з заданим у якості основного параметру запізненням  $\tau$  (*Time Delay*):

$$y(t) = \begin{cases} y_0 & \text{при } t \leq \tau \\ u(t - \tau) & \text{при } t > \tau \end{cases}, \quad (4.24)$$

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

Передавальна функція ланки має вигляд:

$$\frac{y(p)}{u(p)} = e^{-s\tau}. \quad (4.25)$$

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

#### 4.4 Відомості про лінійні арифметичні блоки бібліотеки *Math*

Описані в попередньому підрозділі блоки використовуються при моделюванні лінійних неперервних систем та їх окремих елементів.

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. 4.3).

Блок *Sum* та його аналоги описано в підрозділі 2.2 (див. рис. 2.4).

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* ( $K \cdot u$ ), ( $u$  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.

#### 4.5 Завдання

1. За допомогою блоків *Transfer Function*, *Zero-Pole* та *State Space* реалізуйте моделі, задані передавальними функціями у табл. 4.2 і отримайте їх реакцію на стрибкоподібний сигнал. Впевніться у збігу результатів моделювання.

2. Доповніть модель з пункту 1 послідовним з'єднанням блоків і та побудуйте залежність від часу кроків чисельного інтегрувань диференціальних рівнянь при використанні методів *ode45*, *ode23*, *ode23s*, *ode15s*.

3. Отримайте реакцію блоку запізнення на синусоїдальний вхідний сигнал.

#### 4.6 Методичні рекомендації

1. Для визначення нулів та полюсів передавальної функції скористуйтеся функціями *roots*, а для визначення коефіцієнтів поліномів у чисельнику та знаменнику ПФ за нулями та полюсами системи – функцію *poly*.

№ 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		$T_1 = 5, T_2 = 3, T = 2$

2. Для математичного опису лінійної системи, заданої передавальною функцією, у просторі станів скористуйтеся деталізованою структурною схемою.

3. Для дослідження передавальних функцій скористайтеся структурною схемою на рис. 4.6.

*Лабораторна робота №5***МОДЕЛЮВАННЯ ДВИГУНА ПОСТІЙНОГО СТРУМУ З  
КЕРУВАННЯМ У КОЛІ ЯКОРЯ**

*Мета роботи:* на прикладі моделювання двигуна постійного струму з незалежним збудженням придбати навички математичного опису електромеханічних об'єктів, розробки структурних математичних моделей на основі математичного опису об'єкту, складання плану модельних експериментів для отримання потрібної інформації про властивості досліджуваної системи, закріпити знання про двигун постійного струму, отримані при вивченні дисципліни «Теорія електроприводу», шляхом дослідження його математичної моделі.

**5.1 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 диференційними та алгебраїчними рівняннями [17, 18, 20]:

$$U_{\text{я}}(t) - E_{\text{д}}(t) = \Delta U_{\text{я}}(t) = I_{\text{я}}(t)R_{\text{я}} + L_{\text{я}} \frac{dI_{\text{я}}(t)}{dt}, \quad (5.1)$$

$$M(t) = cI_{\text{я}}(t), \quad (5.2)$$

$$M(t) - M_{\text{с}}(t) = M_{\text{ж}}(t) = J \frac{d\omega(t)}{dt}, \quad (5.3)$$

$$E_{\text{д}}(t) = c\omega(t), \quad (5.4)$$

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;

$$c = k\Phi_{\text{зн}}; \quad (5.5)$$

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

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

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

Рівняння (5.1) називають *рівнянням електромагнітної рівноваги якірного кола*, що складається за другим законом Кірхгофа, рівняння (5.3) – *рівнянням руху* двигуна, що складається за другим законом Кірхгофа для тіл обертального руху. Рівняння моменту (5.2) та ЕРС (5.4) 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 застосуємо 2 підходи.

Згідно з першим напишемо ДР у нормальній формі Коші:

$$\frac{dI_{\text{я}}(t)}{dt} = \frac{U_{\text{я}}(t) - E_{\text{д}}(t) - I_{\text{я}}(t)R_{\text{я}}}{L_{\text{я}}}, \quad \frac{d\omega(t)}{dt} = \frac{M(t) - M_{\text{с}}(t)}{J}, \quad (5.6)$$

або в операторній формі –

$$sI_{\text{я}}(s) = \frac{U_{\text{я}}(s) - E_{\text{д}}(s) - I_{\text{я}}(s)R_{\text{я}}}{L_{\text{я}}}, \quad s\omega(s) = \frac{M(s) - M_{\text{с}}(s)}{J}. \quad (5.7)$$

Згідно з таким перетворенням ДР отримаємо структурну схему ДПС у просторі станів, як це показано на рис. 5.1.

Рис. 5.1. Структурна схема ДПС у просторі станів

Об'єднаємо змінні стану та вхідні сигнали у вектори:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \omega \\ I_{\text{я}} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} U_{\text{я}} \\ M_{\text{с}} \end{bmatrix}. \quad (5.8)$$

Тоді рівняння стану матимуть вигляд:

$$\begin{bmatrix} s\omega \\ sI_{\text{я}} \end{bmatrix} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} = \begin{bmatrix} \frac{c}{J} & 0 \\ -\frac{c}{L_{\text{я}}} & -\frac{R_{\text{я}}}{L_{\text{я}}} \end{bmatrix} \cdot \begin{bmatrix} \omega \\ I_{\text{я}} \end{bmatrix} + \begin{bmatrix} 0 & -\frac{1}{J} \\ \frac{1}{L_{\text{я}}} & 0 \end{bmatrix} \cdot \begin{bmatrix} U_{\text{я}} \\ M_{\text{с}} \end{bmatrix}. \quad (5.9)$$

Інший підхід полягає у тому, що ми в операторних рівняннях виконуємо приведення подібних сигналів та перегрупуємо їх так, щоб з одного боку від знаку рівності був вихідний сигнала з іншого – вхідний:

$$U_{\text{я}}(s) - E_{\text{д}}(s) = I_{\text{я}}(s)(R_{\text{я}} + L_{\text{я}}s) = R_{\text{я}}I_{\text{я}}(s) \left( 1 + \frac{L_{\text{я}}}{R_{\text{я}}}s \right), \quad (5.10)$$

$$M(s) - M_{\text{с}}(s) = M_{\text{ж}}(s) = Js\omega(s). \quad (5.11)$$

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

$$W_{\text{я}}(s) = \frac{I_{\text{я}}(s)}{U_{\text{я}}(s) - E_{\text{д}}(s)} = \frac{I_{\text{я}}(s)}{\Delta U_{\text{я}}(s)} = \frac{1/R_{\text{я}}}{T_{\text{я}}s + 1}, \quad (5.12)$$

$$\text{де } T_{\text{я}} = L_{\text{я}} / R_{\text{я}} \quad (5.13)$$

– електромагнітна стала часу якірного кола.

$$W_M(s) = \frac{\omega(s)}{M(s) - M_c(s)} = \frac{\omega(s)}{M_j(s)} = \frac{1}{Js}. \quad (5.14)$$

Передавальні функції від струму якоря до електромагнітного моменту та від кутової швидкості до проти-ЕРС двигуна є пропорційними ланками (див. рівняння (5.2), (5.4)):

$$W_{IM}(s) = \frac{M(s)}{I_a(s)} = c, \quad W_{\omega E}(s) = \frac{\omega(s)}{E_d(s)} = c. \quad (5.15)$$

Структурна схема, побудована з сукупності передавальних функцій (5.12)-(5.15), показана на рис. 5.2.

Обидві структурні схеми в загальному випадку є взаємозамінними. Перша з них отримує інформацію не тільки про струм якоря та кутову швидкість двигуна, а ще й про їх похідні, тобто про темп зміни струму  $dI_a(t)/dt$  та про кутове прискорення  $d\omega(t)/dt = \varepsilon(t)$ . Крім того, *Simulink*-модель, побудована за першою схемою, може працювати з ненульовими початковими умовами.

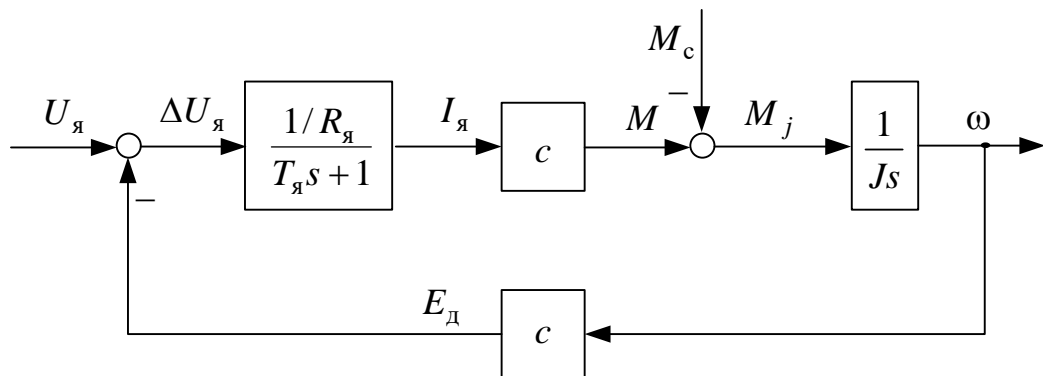


Рис. 5.2. Структурна схема ДПС, складена з передавальних функцій

## 5.2 Нормування структурних схем

Ще більше спростити структурну схему (СС) та зменшити в ній кількість параметрів можна шляхом нормування, тобто переходу від абсолютних одиниць до відносних. У якості базових величин обирають значення регульованих сигналів у характерних статичних режимах. Усі базові величини повинні бути пов'язані між собою достатньо простими рівняннями статички. Розглянемо статичні характеристики ДПС (рис.5.3) при постійній напрузі збудження ( $\Phi_3 = \Phi_{3H} = const$ ).

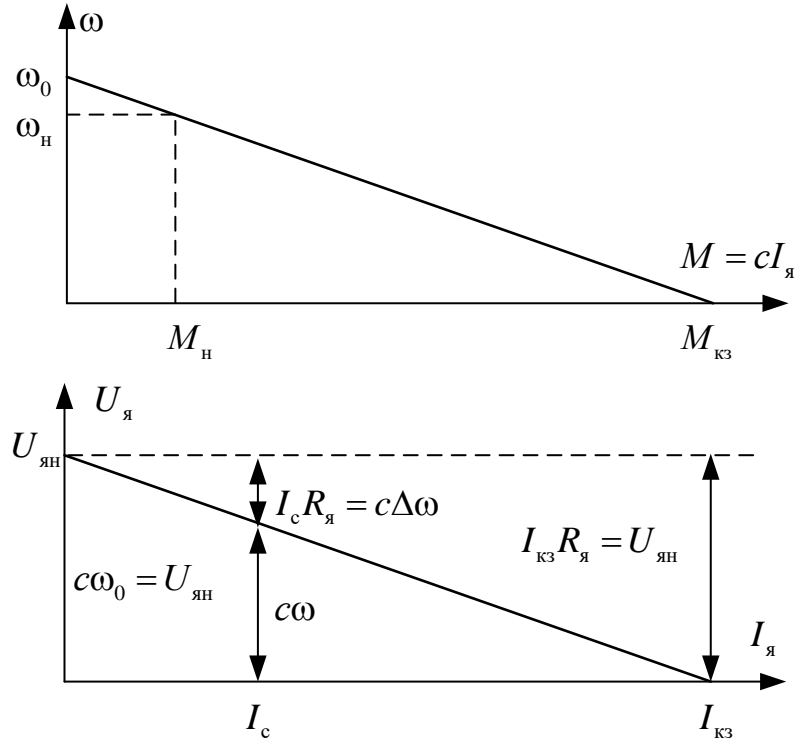


Рис. 5.3. Статичні характеристики ДПС при  $\Phi_3 = \Phi_{3H} = const$

З розгляду статичних характеристик обираємо такі базові величини:

$$\omega_6 = \omega_0, \quad U_{я6} = E_{д6} = U_{яH} = c\omega_0, \quad I_{я6} = I_{кз} = \frac{U_{яH}}{R_я}, \quad M_6 = M_{кз} = cI_{кз}. \quad (5.16)$$

При цьому утворюється така система відносних одиниць:

$$\bar{\omega} = \frac{\omega}{\omega_0}, \quad \bar{U} = \frac{U}{U_{яH}}, \quad \bar{E}_д = \frac{E_д}{U_{яH}}, \quad \bar{I}_я = \frac{I_я}{I_{кз}}, \quad \bar{M} = \frac{M}{M_{кз}}. \quad (5.17)$$

Передавальні функції у в.о. розраховуємо за формулою

$$\bar{W}(s) = W(s) \frac{u_6}{y_6}. \quad (5.18)$$

Відповідно до неї знаходимо:

$$\bar{W}_{IM}(s) = c \frac{I_{кз}}{M_{кз}} = 1, \quad \bar{W}_{\omega E}(s) = W_{\omega E}(s) = c \frac{\omega_0}{E_{д0}} = 1. \quad (5.19)$$

$$\bar{W}_я(s) = W_я(s) \frac{U_{яH}}{I_{кз}} = \frac{1/\mathcal{R}_я}{T_я s + 1} \cdot \mathcal{R}_я = \frac{1}{T_я s + 1}; \quad (5.20)$$

$$\bar{W}_M(s) = W_M(s) \frac{M_{кз}}{\omega_0} = \frac{M_{кз}}{J\omega_0 s} = \frac{1}{T_M s}. \quad (5.21)$$

В останньому рівнянні

$$T_M = \frac{J\omega_0}{M_{кз}} = \frac{JR_{я}}{c^2} \quad (5.22)$$

– *електромеханічна стала часу двигуна.*

*Фізичний сенс цієї сталої часу впливає із рівняння руху двигуна: електромеханічна стала часу дорівнює часу, за який двигун розігнався б до швидкості ідеального холостого ходу, якщо б до нього було прикладено момент короткого замикання.*

На рис. 5.4 проілюстровано наведене визначення та показано, як його можна використати для розрахунку часу розгону приводу з заданим моментом.

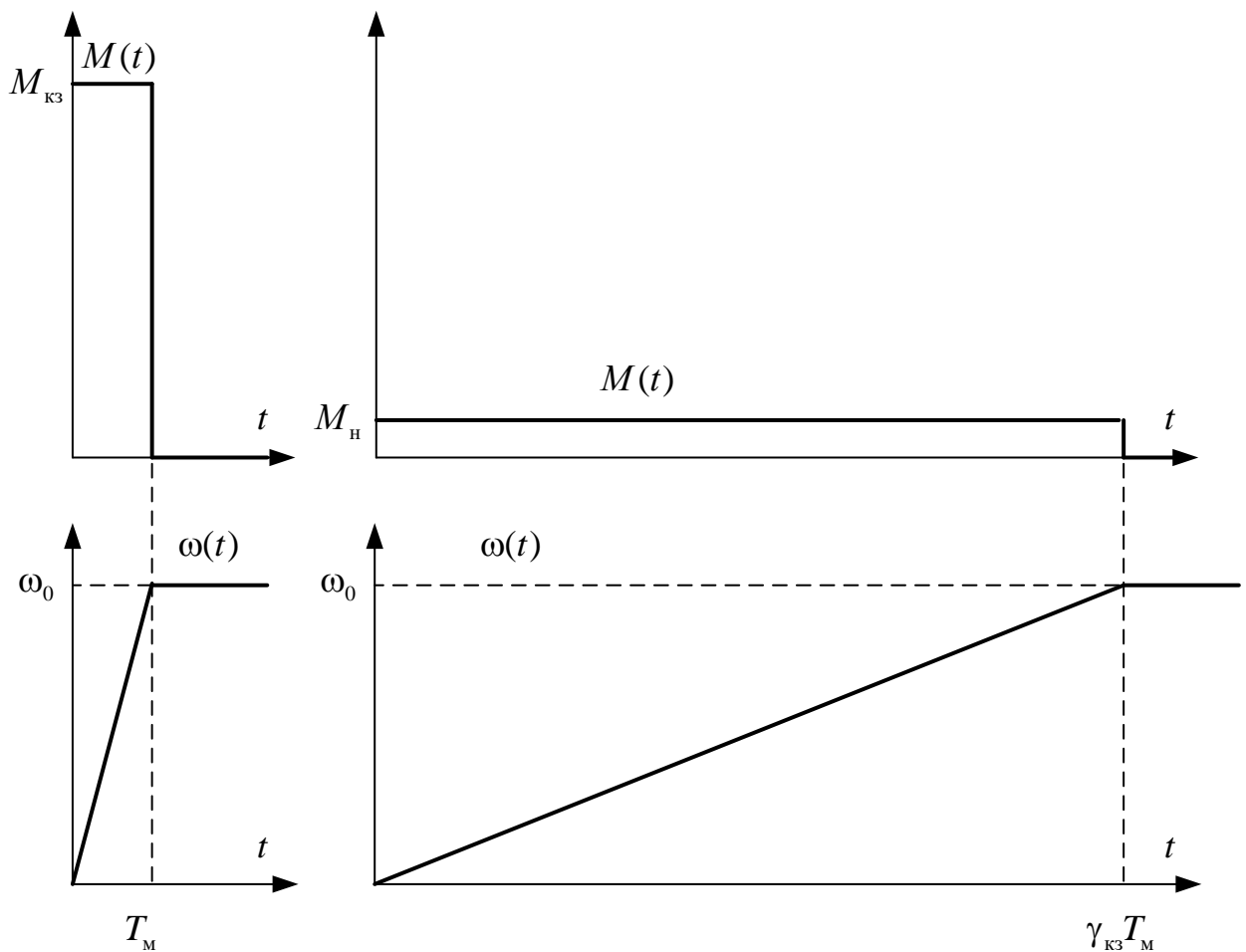


Рис. 5.4. Ілюстрація фізичного сенсу електромеханічної сталої часу

На цьому рисунку використано параметр  $\gamma_{кз}$ , який називають кратністю моменту (або струму) короткого замикання:

$$\gamma_{кз} = \frac{M_{кз}}{M_H} = \frac{I_{кз}}{I_{яH}}. \quad (5.23)$$

Цей параметр визначає жорсткість механічної характеристики:

$$\frac{\omega_0 - \omega_H}{\omega_0} = \frac{\Delta\omega_{сн}}{\omega_0} = \frac{1}{\gamma_{кз}}. \quad (5.24)$$

Структурна схема ДПС у в.о. зображена на рис. 5.5.

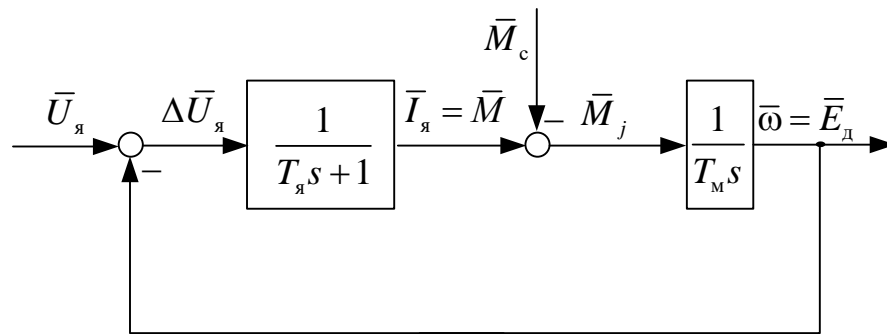


Рис. 5.5. Структурна схема ДПС у відносних одиницях (5.13)

Перевага останньої схеми – сама проста, мінімальна кількість параметрів. Недолік – струм та момент подані в долях струму та моменту КЗ. Якщо ми хочемо задавати статичний момент та візуалізувати струм і момент в долях номінальних значень, тобто

$$\bar{i}_я = \frac{I_я}{I_{яH}}, \quad \bar{m} = \frac{M}{M_H}, \quad (5.25)$$

то перехід від однієї системи в.о. до іншої здійснюємо за формулами:

$$\bar{i}_я = \bar{I}_я \gamma_{кз}, \quad \bar{M}_c = \frac{\bar{m}_c}{\gamma_{кз}}. \quad (5.26)$$

Структурна схема, в якій використано дві системи в.о., показана на рис. 5.6, а СС у другій системі в.о. – на рис. 5.7,

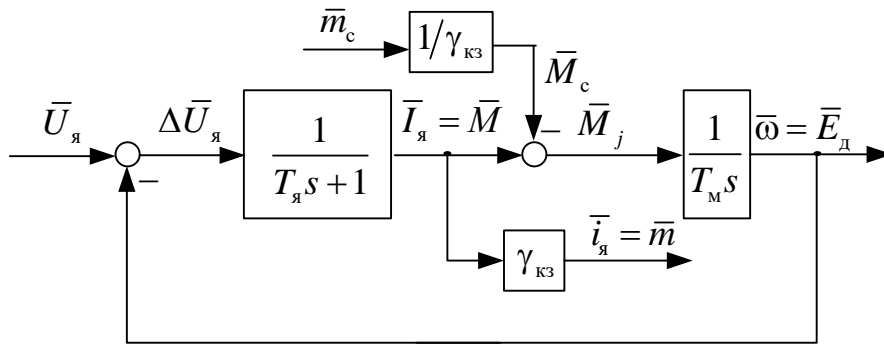


Рис. 5.6. Нормована структурна схема ДПС з двома системами в.о.

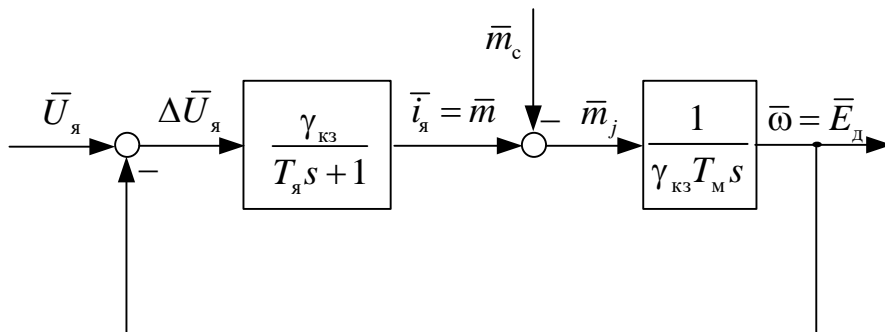


Рис. 5.7. Структурна схема ДПС у відносних одиницях (5.25)

### 5.3 Визначення та аналіз передавальних функцій

Для оцінювання адекватності розроблених моделей та виявлення у них логічних помилок визначимо передавальні функції досліджуваного двигуна в абсолютних одиницях:

$$\frac{E_d(s)}{U_я(s)} = \frac{M(s)}{M_c(s)} = \frac{1}{T_M T_я s^2 + T_M s + 1}, \quad \frac{\omega(s)}{U_я(s)} = \frac{1/c}{T_M T_я s^2 + T_M s + 1}, \quad (5.27)$$

$$\frac{M(s)}{U_я(s)} = \frac{(J/c)s}{T_M T_я s^2 + T_M s + 1}, \quad \frac{I_я(s)}{U_я(s)} = \frac{(J/c^2)s}{T_M T_я s^2 + T_M s + 1}, \quad (5.28)$$

$$\frac{\omega(s)}{M_c(s)} = -\frac{R_я}{c^2} \cdot \frac{(T_я s + 1)}{T_M T_я s^2 + T_M s + 1}, \quad (5.29)$$

З (5.29) випливає, що в усталеному режимі ( $s=0$ )

$$\frac{\Delta \omega_c}{M_c} = -\frac{R_я}{c^2} = -\frac{R_я}{(k\Phi)^2}.$$

Знайдемо корені характеристичного рівняння

$$T_M T_{\text{я}} s^2 + T_M s + 1 = 0.$$

$$s_{1,2} = \frac{-T_M \pm \sqrt{T_M^2 - 4T_M T_{\text{я}}}}{2T_{\text{я}}} = \frac{-T_M \pm \sqrt{T_M(T_M - 4T_{\text{я}})}}{2T_{\text{я}}}. \quad (5.30)$$

$$\frac{1}{T_M T_{\text{я}} s^2 + T_M s + 1} = \frac{1}{T_k^2 s^2 + 2\xi T_k s + 1}$$

$$T_M T_{\text{я}} = T_k^2, \quad T_M = 2\xi T_k$$

$$T_k = \sqrt{T_M T_{\text{я}}}, \quad \xi = \frac{T_M}{2T_k} = \frac{T_M}{2\sqrt{T_M T_{\text{я}}}} = \frac{1}{2} \sqrt{\frac{T_M}{T_{\text{я}}}}$$

З порівняння передавальних функцій випливає:

- реакція електромагнітного моменту на стрибок моменту навантаження співпадає з реакцією швидкості на стрибок напруги якоря та з реакцією моменту на зміну напруги якоря по лінійному закону;
- в усталеному режимі ( $s=0$ )  $M_{\infty} = M_c$ ,  $\omega_{\infty} = \frac{U_{\text{я}}}{c} - \frac{M_c R_{\text{я}}}{c^2} = \omega_0 - \Delta\omega_c$ .
- запізнення швидкості та ЕРС двигуна від напруги якоря, що змінюється за лінійним законом, дорівнює електромеханічній сталій часу  $T_m$ .

Із (5.30) випливає, що *при  $T_M \geq 4T_{\text{я}}$  ДПС має 2 дійсних від'ємних полюси, тобто є аперіодичною ланкою другого порядку, для якої характерний перебіг перехідних процесів без перерегулювань. При  $T_M < 4T_{\text{я}}$  полюси стають комплексно спряженими, що перетворює двигун на коливальну ланку з коефіцієнтом демпфірування*

$$\xi = \frac{1}{2} \cdot \sqrt{\frac{T_M}{T_{\text{я}}}}, \quad (5.31)$$

який дорівнює косинусу від кутів, що утворюють радіус-вектори полюсів з дійсною віссю, і визначає величину перерегулювання перехідної функції коливальної ланки:

$$\varphi = \arccos(\xi); \quad \sigma = \exp(-\pi \cdot \text{ctg}(\varphi)) \cdot 100\%. \quad (5.32)$$

Графік залежності перерегулювання від коефіцієнта демпфірування коливальної ланки наведено на рис. 5.8.



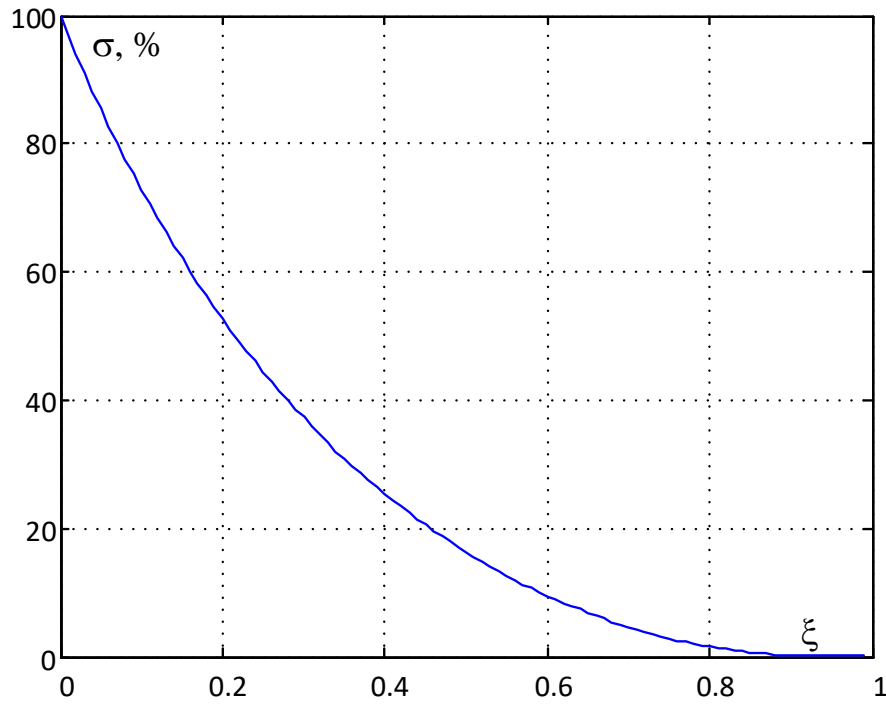


Рис. 5.8. Залежність перегулювання перехідної функції ДПС від його коефіцієнта демпфірування

#### 5.4. Моделювання та аналіз перехідних процесів ДПС в режимі прямого пуску

Simulink-модель ДПС в а.о. для дослідження реакції на стрибок напруги якоря з 0 до номінального значення показана на рис. 5.9, а перехідні процеси в цьому режимі – на рис. 5.10.

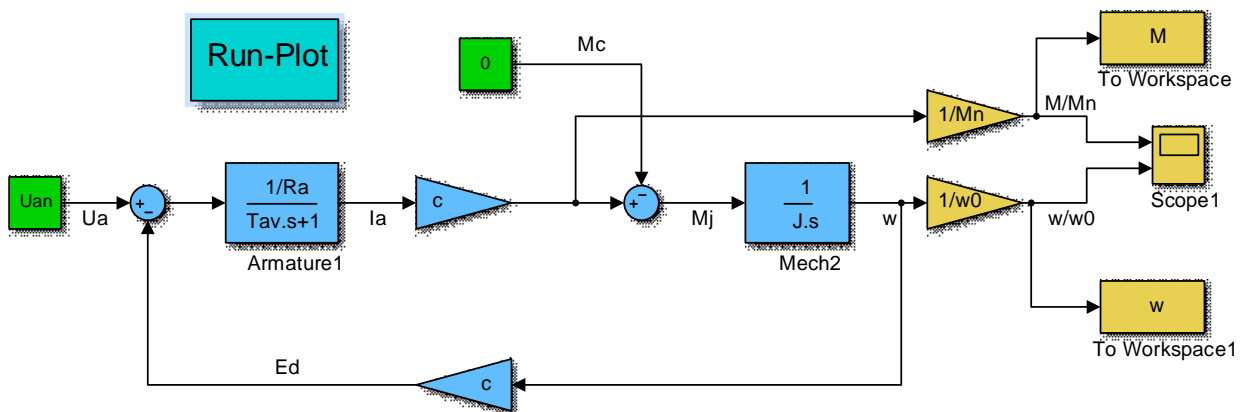


Рис. 5.9. Симулінк-модель ДПС для дослідження прямого пуску

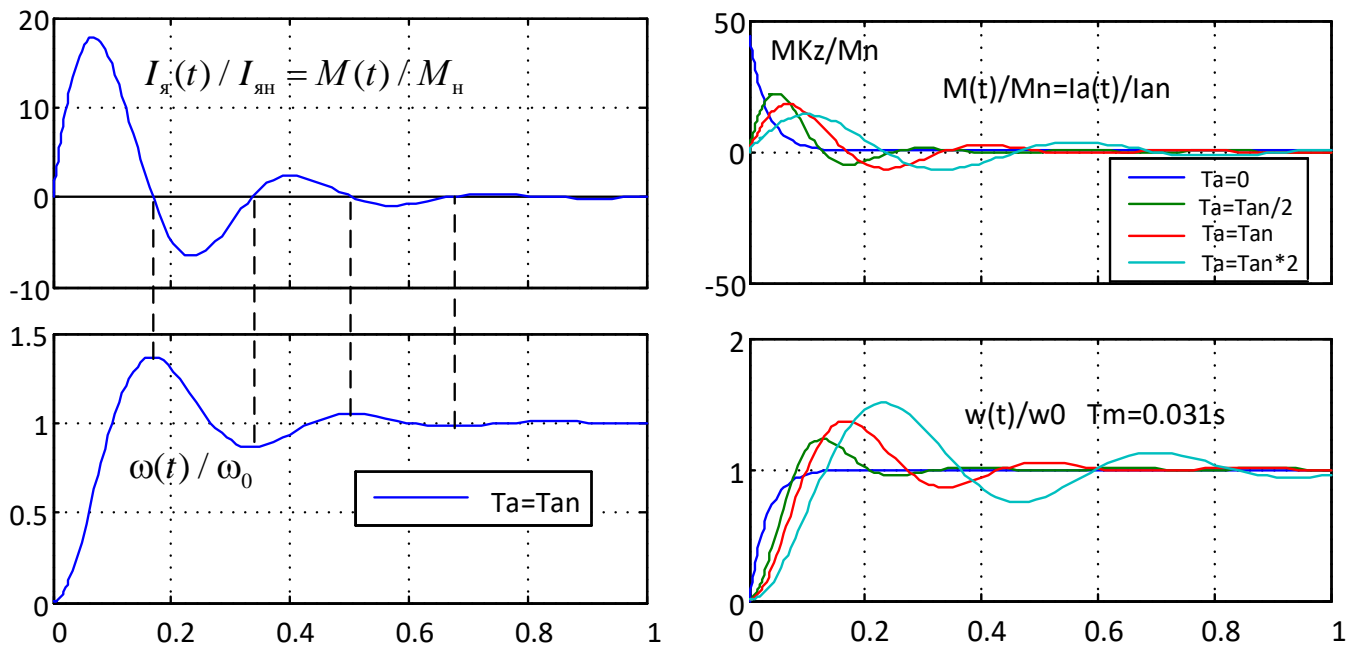


Рис. 5.10. Перехідні процеси при прямому пуску ДПС

Як бачимо, в досліджуваному режимі струм якоря і момент при зменшенні індуктивності якоря наближається до відповідних величин у режимі КЗ. Збільшення індуктивності знижує темп наростання цих величин і зменшує пікові значення, але вони все ж залишаються не припустимо великими. Цей віртуальний експеримент доводить необхідність регулювання напруги якоря.

### 5.5. Моделювання ДПС при використанні задатчиків інтенсивності для формування напруги якоря

Одним з найпростіших законів зміни напруги, що дозволить знизити момент і струм у пуско-гальмівних режимах є лінійний закон. *Пристрій, що формує завдання на швидкість з обмеженням прискорення, називають задатчиком інтенсивності (ЗІ).*

Для застосування такого методу керування необхідно живити двигун від підсилювача потужності. В сучасних систем регульованого електроприводу таким підсилювачем є тиристорний перетворювач постійного струму (ТП), який з точки зору ТАК можна представити у вигляді аперіодичної лан-

ки з великим коефіцієнтом підсилювання (20÷40) та малою сталою часу (4÷6 мс):

$$W_{\text{тп}}(s) = \frac{U_{\text{я}}(s)}{U_{\text{к}}(s)} = \frac{k_{\text{тп}}}{T_{\text{тп}}s + 1}$$

При дослідженні такого способу регулювання швидкості можна знехтувати інерційністю ТП. Модель двигуна, що відпрацьовує задану трапецоїдальну тахограму показано на рис. 5.11, а перехідні процеси – на рис. 5.12.

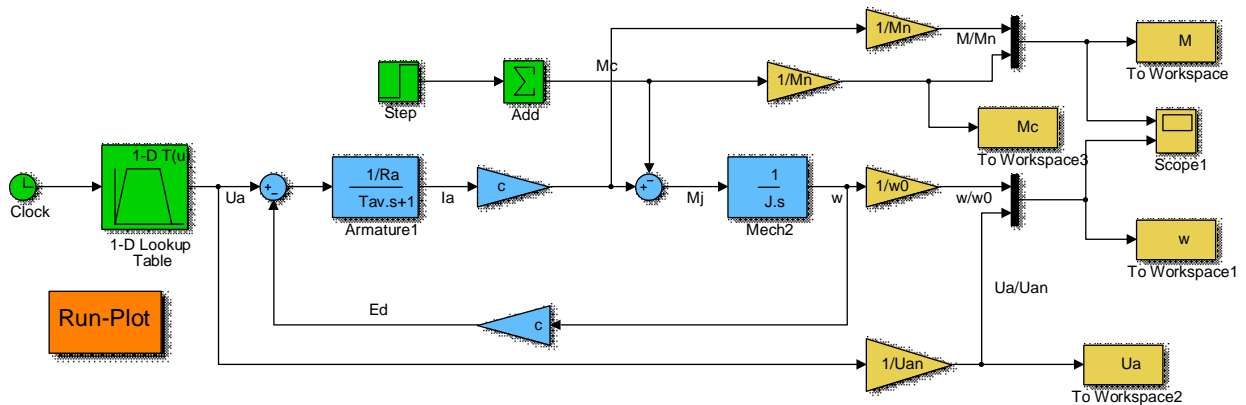


Рис. 5.11. Модель роботи ДПС за трапецоїдальною тахограмою

Час зміни напруги від 0 до номінального значення при розгоні і навпаки при гальмуванні  $t_{p0}$  визначає задане прискорення  $\varepsilon_0 = \omega_0 / t_{p0}$  і величину динамічного моменту  $M_{j0} = J\varepsilon_0 = J\omega_0 / t_{p0}$  в пуско-гальмівних режимах.

Отже

$$t_{p0} = \frac{J\omega_0}{M_{j0}}. \quad (5.33)$$

Якщо потрібна інформація не тільки про напругу якоря, яка по суті є завданням на швидкість, а ще й про завдання на прискорення двигуна, то модель задавального пристрою може мати вигляд рис. 5.13.

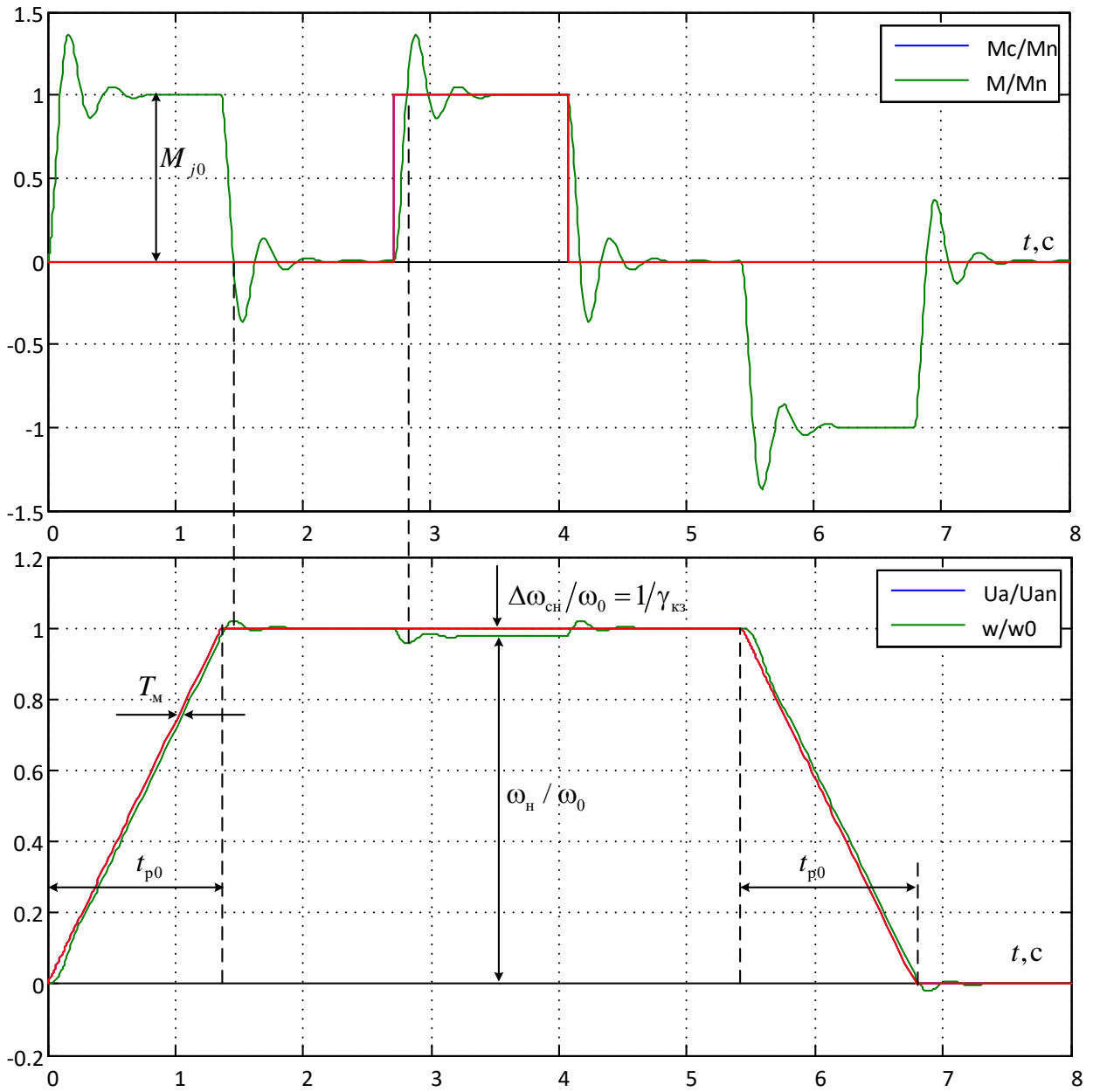


Рис. 5.12. Перехідні процеси в ДПС при лінійній зміні напруги і стрибкоподібній зміні моменту навантаження

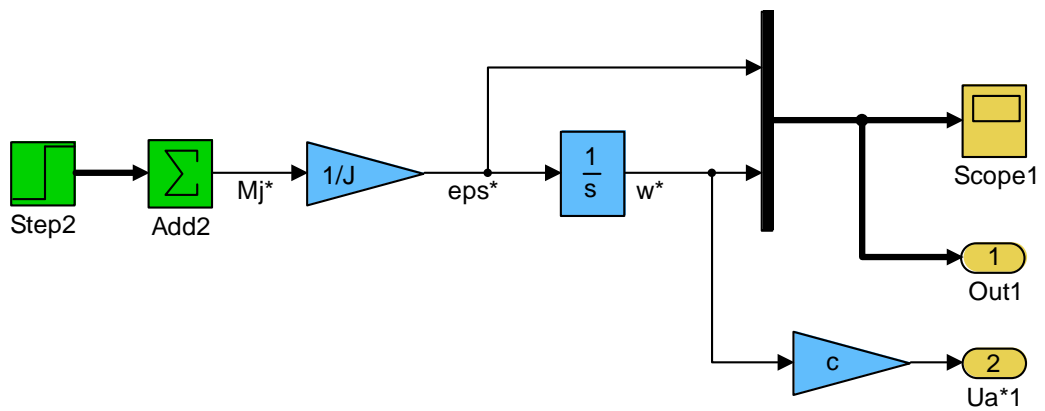


Рис. 5.13. Розімкнена модель типового ЗІ

Перехідні процеси в такому ЗІ показані на рис. 5.14.

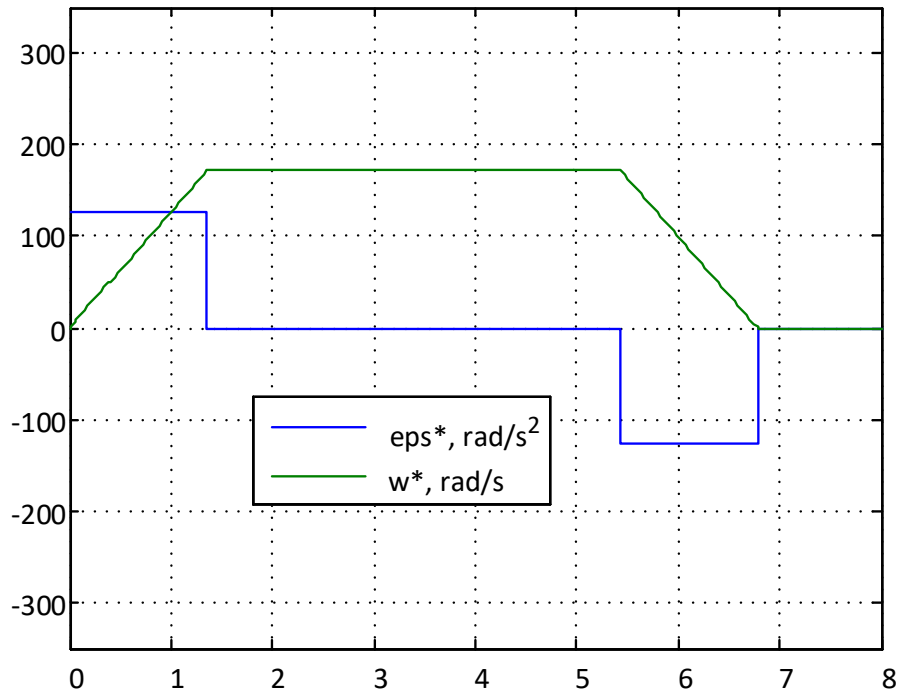


Рис. 5.14. Перехідні процеси типового ЗІ, що формує трапецоїдальну тахограму

Можна сформувати завдання на зміну напруги якоря так, щоб обмежувати не тільки прискорення, а й ривок (рис. 5.15):

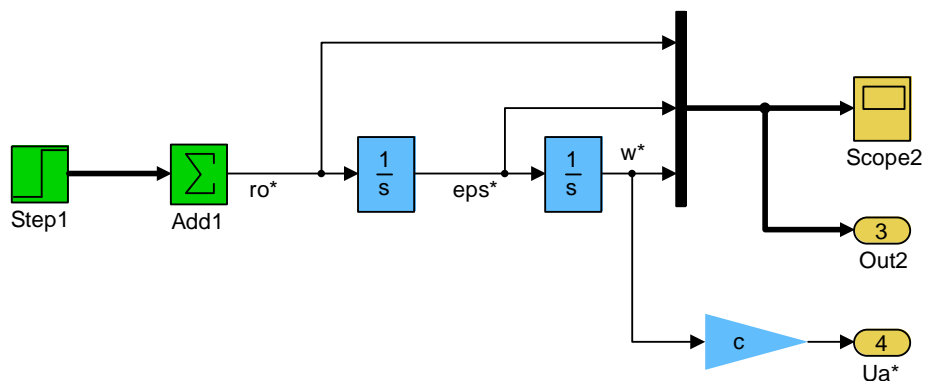


Рис. 5.15. Розімкнена модель ЗІ з обмеженням ривка і прискорення

Перехідні процеси в ЗІ з обмеженням ривка і прискорення показані на рис. 5.16. Останню *тахограму з комбінацією лінійних та параболічних ділянок називають також S-подібною трапецоїдальною.*

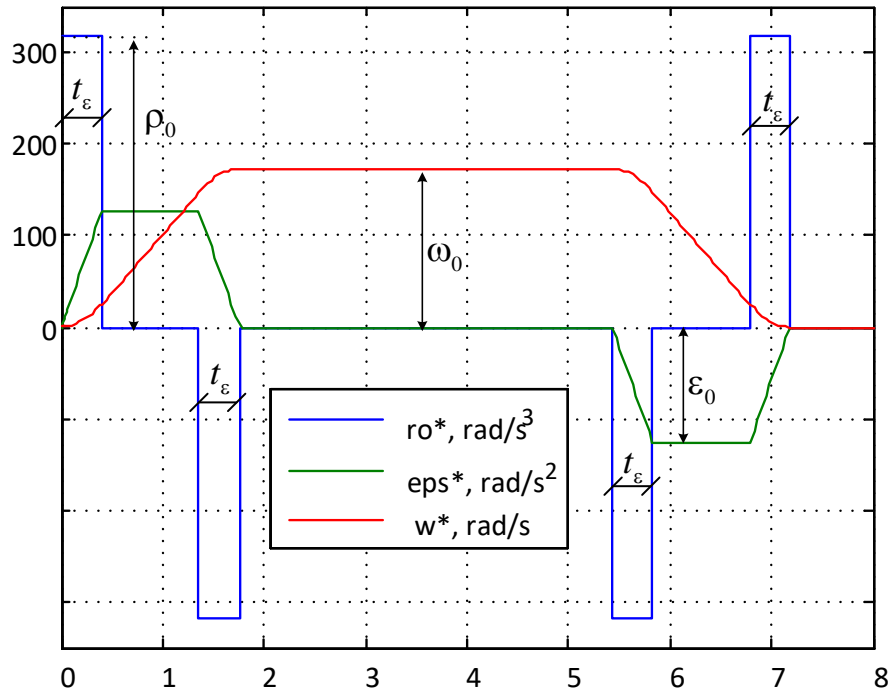


Рис. 5.16. Перехідні процеси в ЗІ з обмеженням прискорення і ривка, що формує S-подібну тахограму

Розрахунок бажаного ривка виконується з умови обмеження похідної від прискорення і відповідно похідних від динамічних складових моменту і струму якоря:

$$\rho_0 = \frac{\epsilon_0}{t_\epsilon} = \frac{M_{j0}}{Jt_\epsilon}, \quad (5.34)$$

де  $t_\epsilon$  – час зміни завдання на прискорення за лінійним законом.

Перехідні процеси в ДПС при відпрацюванні S-подібної тахограми показані на рис. 5.17.

*Наявність в ЗІ інтеграторів дає можливість подавати на вхід системи не тільки основне завдання, а ще і його похідні, що дає можливість компенсувати інерційність об'єкту керування і підвищити точність відпрацювання об'єктом заданого входного сигналу (часто кажуть «точність відпрацювання заданих траєкторій»).*

Структурна схема такого ЗІ показана на рис. 5.18.

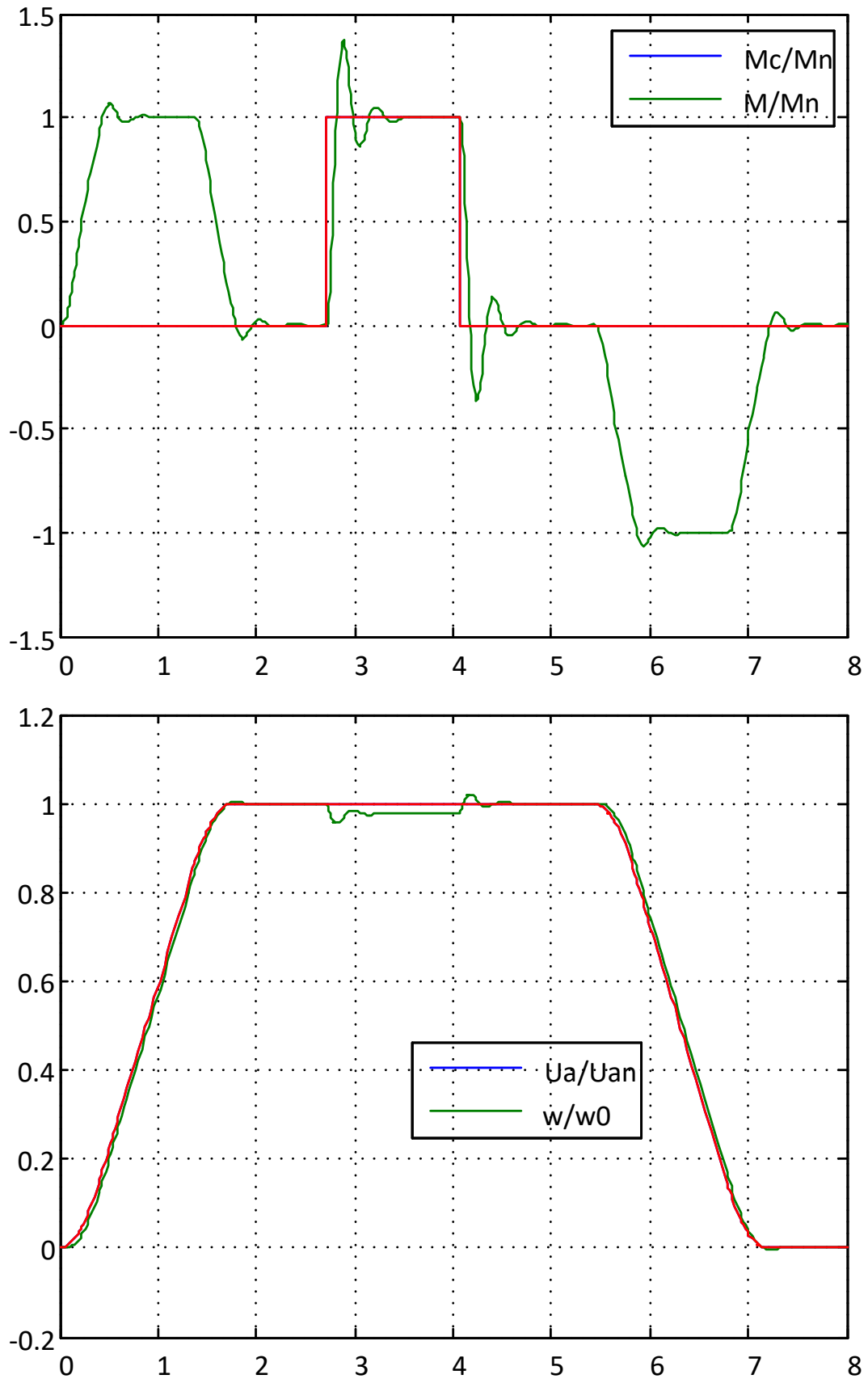


Рис. 5.17. Перехідні процеси в ДПС при відпрацюванні S-подібної тахограми при керуванні за відхиленням

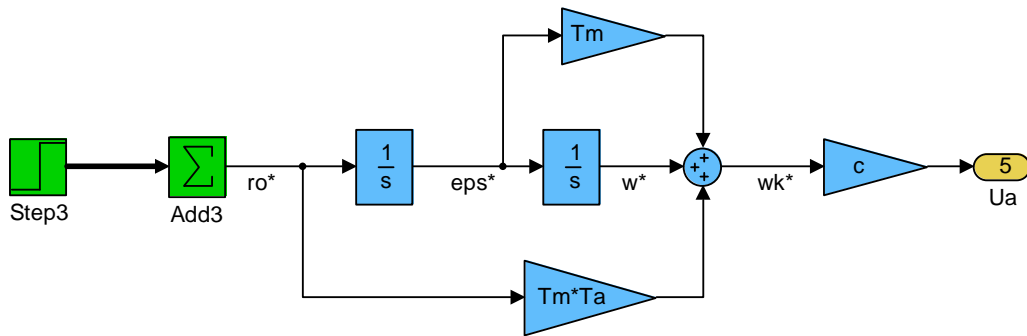


Рис. 5.18. Розімкнена модель ЗІ з обмеженням ривка і паралельними коригуючими зв'язками, що реалізують принцип комбінованого керування за задавальною дією

*Такий спосіб керування в ТАК називають «комбіноване керування за задавальною дією» (на відміну від принципу керування за відхиленням – керування з від'ємним зворотним зв'язком без паралельних коригуючих зв'язків).*

Оскільки ДПС є об'єктом другого порядку, а ЗІ має 2 інтегратори, то інерційність об'єкту можна компенсувати не частково, а повністю:

Передавальна функція представленого ЗІ має вид:

$$W_{zi}(s) = \frac{T_M T_{я} s^2 + T_M s + 1}{s^2}. \quad (5.35)$$

Як бачимо поліном в чисельнику дорівнює характеристичному поліному двигуна і повністю його компенсує.

Перехідні процеси в ДПС при керуванні від ЗІ рис. 5.18 показані на рис. 5.19.

Як бачимо, у цьому випадку, задавальна дія відпрацьовується двигуном без похибки.

Для того, щоб не розраховувати часи перемикання тахограм, коли помилка в розрахунках веде до похибки формування усталеного значення швидкості, застосовують ЗІ з замкненою структурою, моделі яких показано на рис. 5.20, 5.21.



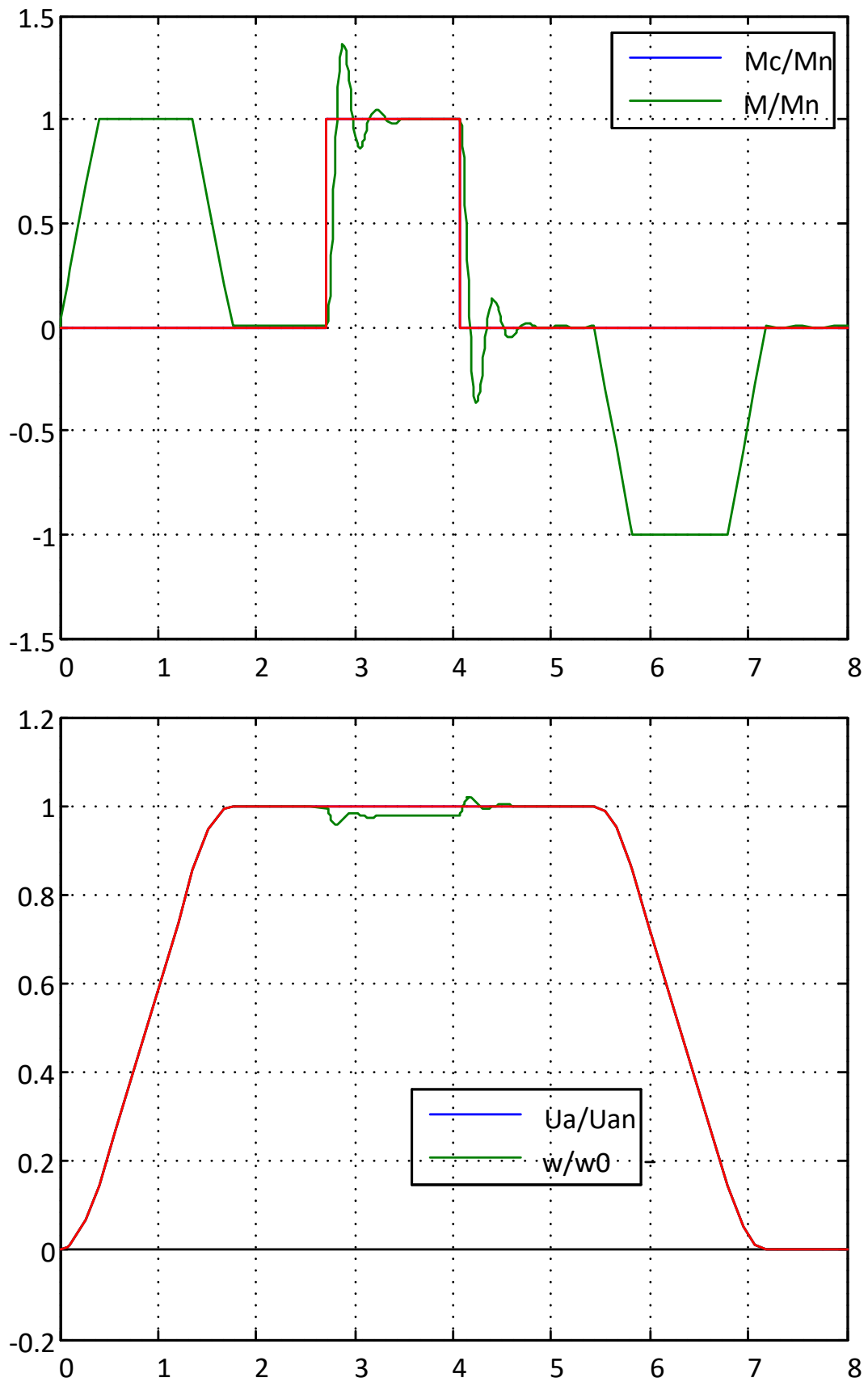


Рис. 5.19. Перехідні процеси в ДПС при відпрацюванні S-подібної тахограми при комбінованому керуванні

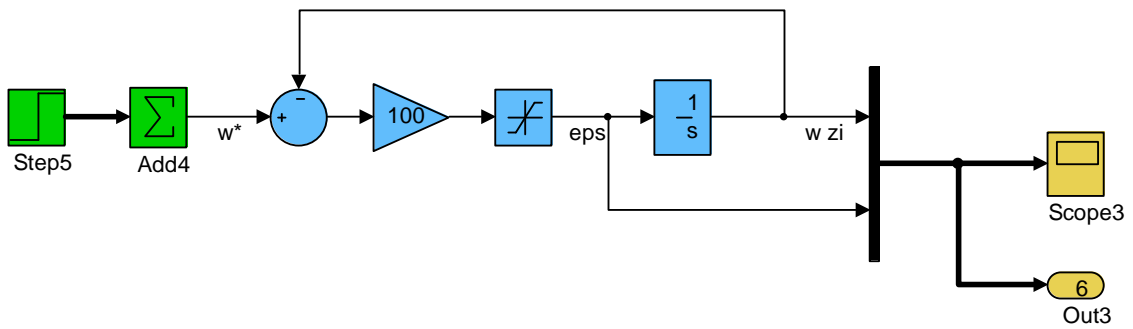


Рис. 5.20. Модель замкнутого ЗІ з обмеженням прискорення

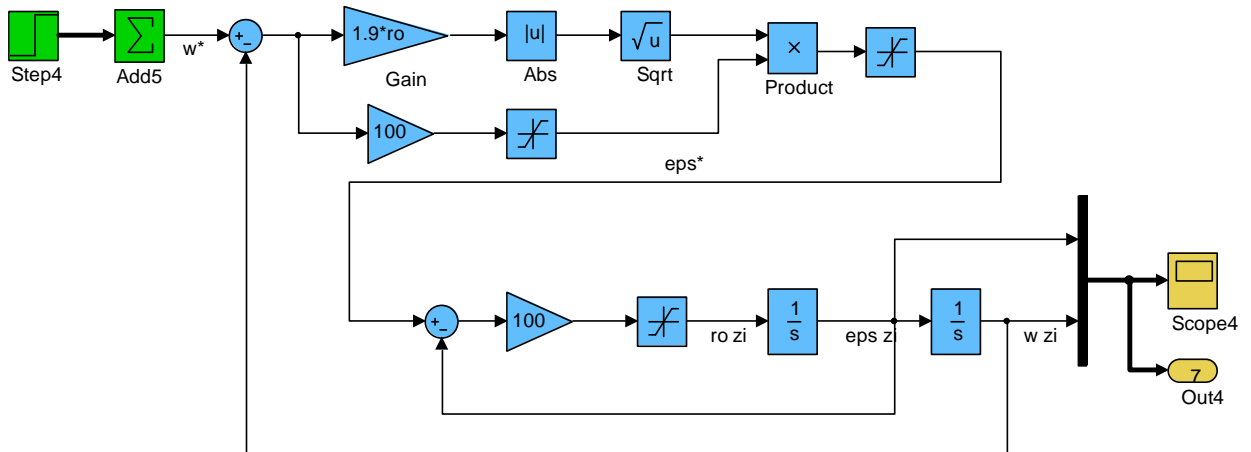


Рис. 5.21. Модель замкнутого ЗІ з обмеженням прискорення і ривка

## 5.6. Врахування тертя при моделюванні електричних двигунів

В усіх попередніх моделях не враховано наявність *моменту холостого ходу* двигуна  $M_f$ , який витрачається на подолання *моменту сухого тертя*  $M_{cf}$  (*Coulombic friction*):

$$M_f = M_{cf} \cdot \text{sign}(\omega). \quad (5.35)$$

Структурна схема механічної частини будь-якого двигуна з урахуванням моменту сухого тертя представлена на рис. 5.22а, де  $M_L$  - момент навантаження (від англ. *Load*), що утворюється при виконанні електроприводом корисної роботи (наприклад, підймання вантажу, обтискання металу на прокатних станах тощо).  $M_c$  - сумарний момент статичного опору (з урахуванням тертя).

Однак при застосуванні такої СС при цифровому математичному моделюванні із-за дискретності за часом і із-за відсутності в арифметиці з плаваючою точкою нуля момент холостого ходу на кожному кроці чисельного інтегрування перемикається між рівнями  $+M_{cf}$  і  $-M_{cf}$ .

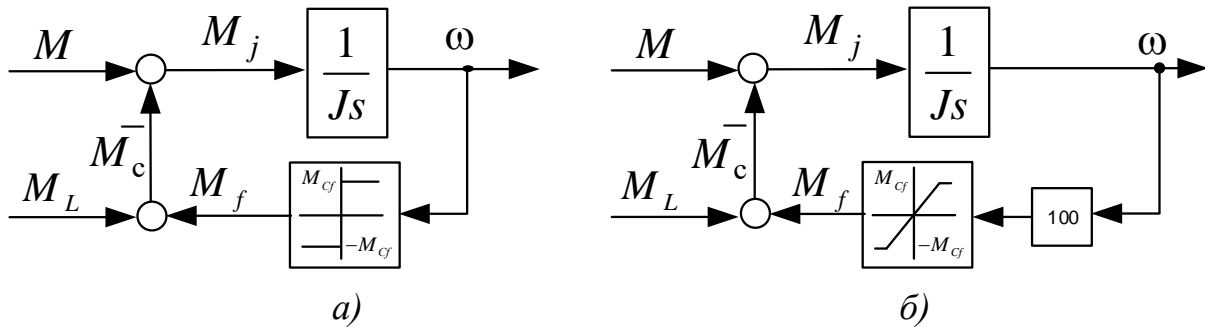


Рис. 5.22. Структурні схеми механічної частини двигуна зі врахуванням наявності моменту сухого (Кулоновського) тертя

Така різниця між значеннями однієї координати в сусідніх точках часу приводить до того, що ітераційний процес розв'язання системи диференціальних рівнянь «зациклюється». Щоб запобігти цього явища треба або застосувати метод чисельного інтегрування з постійним кроком, що приведе до збільшення часу моделювання, або замінити елемент з безкінечним підсиленням елементом з великим але кінцевим коефіцієнтом підсилення (50-100) на лінійній ділянці, тобто застосувати послідовне з'єднання блоків *Gain* та *Saturation*, як це показано на структурній схемі рис. 5.22б. Перехідні процеси зображені на рис. 5.23.

Якщо враховувати різницю між *сухим тертям у стані спокою*  $M_{bf}$  (*Breakway friction*) та у стані руху  $M_{cf}$ , то замість рівняння (5.35), слід застосувати систему рівнянь

$$M_f = \begin{cases} M & \text{при } |\omega| = 0 \text{ \& } |M| < M_{bf}, \\ M_{cf} \cdot \text{sign}(\omega) & \text{при } |\omega| > 0. \end{cases} \quad (5.36)$$

Реалізація такого математичного опису є більш складною, тому що потребує застосування та налаштування керованого ключа та блоків зрівняння і логіки.

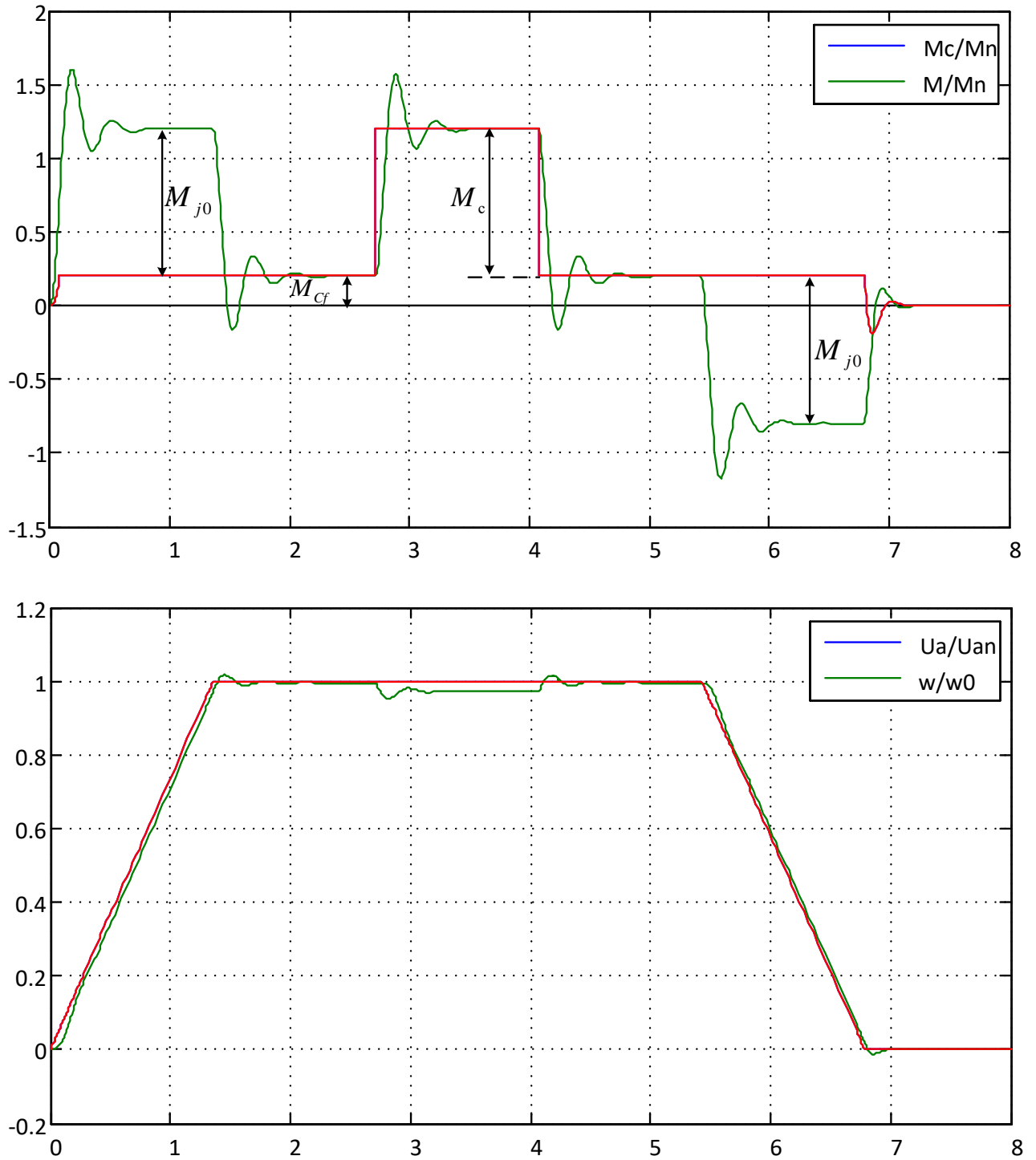


Рис. 5.23. Перехідні процеси в ДПС при врахуванні наявності сухого тертя моделлю рис. 5.22б

Ще більш точна залежність моменту тертя від швидкості описується за рівняннями:

$$M_f = \begin{cases} \frac{M_{bf}}{\omega_{th}} \omega, & \text{при } |\omega| \leq \omega_{th}, \\ [M_{cf} + M_{sf} e^{(-c_s|\omega - \omega_m|)} + f_v |\omega - \omega_{th}|] \text{sign}(\omega), & \text{при } |\omega| > \omega_{th}, \end{cases} \quad (5.37)$$

де  $M_{sf} = M_{cf} - M_{bf}$  – **компонента Штрібека (Stribeck Friction)**, яка існує при умові врахування різниці між тертям спокою та тертям руху [48];  $c_s$  – **коефіцієнт експоненціальної складової тертя (Transitional Approximation Coefficient)**;  $f_v$  – **коефіцієнт в'язкого тертя (Viscous Friction Coefficient)**;  $\omega_{th}$  – **межа лінійної ділянки (Linear Region Velocity Threshold) статичної характеристики тертя  $M_f(\omega)$** , зображеної на рис. 5.24.

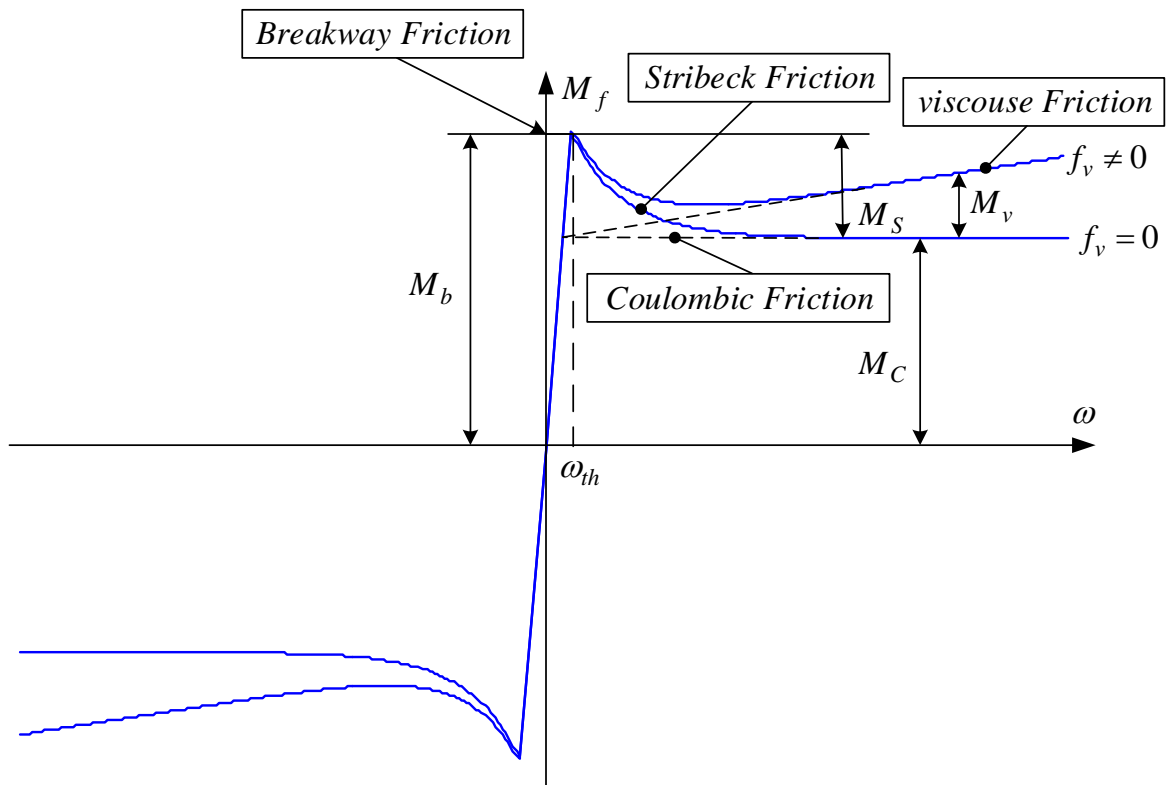


Рис. 5.24. Статична механічна характеристика тертя зі врахуванням наявності тертя спокою, сухого і в'язкого тертя руху та експоненціальної компоненти Штрібека

Але, зважаючи на те, що момент тертя складає невелику долю від номінального моменту двигуна, то у більшості випадків, цілком припустимо застосовувати спрощену модель тертя.

### 5.7 Завдання

Промоделюйте двигун постійного струму з незалежним збудженням (ТП-Д) при постійному потоку збудження з даними, які наведені в табл. 5.1, в таких режимах:

- прямий пуск (стрибкоподібна зміна напруги двигуна від 0 до номінального значення);
- робота за трапецоїдальною тахограмою при стрибкоподібному накиданні і скиданні номінального навантаження на усталеній швидкості (без врахування моменту холостого ходу і з врахуванням,  $M_{\text{хх}} = 0.1M_{\text{н}}$ );
- робота за S-подібною трапецоїдальною тахограмою при стрибкоподібному накиданні і скиданні номінального навантаження на усталеній швидкості;
- робота за S-подібною трапецоїдальною тахограмою з комбінованим керуванням за задавальною дією при стрибкоподібному накиданні і скиданні номінального навантаження на усталеній швидкості.

Таблиця 5.1

№ вар.	Тип	$P_{\text{н}}$	$U_{\text{ян}}$	$U_{\text{зн}}$	$n_{\text{н}}$	$I_{\text{ян}}$	$R_{\text{я}}$	$2p$	$R_{\text{з}}$	$\Phi_{\text{н}}$	$J$
		кВт	В	В	об/хв	А	Ом	–	Ом	мВб	кг·м <sup>2</sup>
1	П81	32	440	220	1500	83	0,250	4	208,4	14,4	0,68
2	П82	25	440	220	1000	66	0,373	4	137,6	19,8	0,78
3	П91	55	440	220	1500	143	0,143	4	129,2	19,6	1,48
4	П101	42	440	220	750	107	0,206	4	130,0	29,0	2,58
5	П81	19	440	220	1000	52	0,630	4	208,4	14,4	0,68
6	П101	100	440	220	1500	256	0,051	4	130,0	29,9	2,58
7	П102	125	440	220	1500	316	0,034	4	101,5	38,3	3,00
8	П111	95	440	220	1000	240	0,067	4	87,0	19,0	5,10
9	П111	160	440	220	1500	400	0,025	4	87,0	20,3	5,10
10	П112	85	440	220	750	220	0,075	4	82,0	25,3	5,57
11	П112	180	440	220	1500	450	0,023	4	82,0	16,2	5,75
12	П112	200	440	220	1500	1000	0,050	4	24,0	16,2	5,75
13	П52	14	220	110	3000	74	0,027	4	94,8	7,2	0,10

14	П61	19	220	110	3000	100	0,057	4	104,0	8,2	0,14
15	П72	25	220	110	1500	132	0,083	4	67,0	12,5	0,40
16	П111	160	220	110	1500	809	0,064	4	28,0	18,6	5,10
17	П102	125	220	110	1500	632	0,079	4	32,5	13,0	3,00
18	П101	100	220	110	1500	508	0,013	4	37,8	13,7	2,58
19	П92	75	220	110	1500	381	0,014	4	31,8	20,1	1,75
20	П22	1	220	110	1500	5,9	4,170	2	600,0	3,2	0,01
21	П42	7,4	220	110	3000	39,8	0,205	4	136,0	5,1	0,05

У таблиці позначено:

$P_n, \Phi_n, n_n$  – номінальні значення потужності, потоку збудження та кутової швидкості у 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.

#### 5.4 Методичні 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

$$\omega_0 = \frac{U_{ян}}{R_{я}}. \quad (5.38)$$

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 СІ) здійснюйте за формулою

$$\omega_0 = \frac{U_{ян}}{c}, \quad \omega_n = \frac{\pi n_n}{30}. \quad (5.39)$$

3. Розрахуйте номінальний момент

$$M_n = \frac{P_n}{\omega_n}, \quad (5.40)$$

струм короткого замикання

$$I_{\text{кз}} = \frac{U_{\text{ян}}}{R_{\text{я}}}, \quad (5.41)$$

коефіцієнт моменту та ЕРС

$$c = \frac{M_{\text{н}}}{I_{\text{ян}}}, \quad (5.42)$$

момент короткого замикання

$$M_{\text{кз}} = cI_{\text{кз}} \quad (5.43)$$

та статичну просадку швидкості при номінальному навантаженні

$$\Delta\omega_c = \frac{I_{\text{ян}}R_{\text{я}}}{c} \quad (5.44)$$

4. Індуктивність якоря розрахуйте за емпіричною формулою Уманського-Лінвіля

$$L_{\text{я}} = k \frac{U_{\text{ян}}}{I_{\text{ян}}\omega_{\text{н}}p}, \quad (5.45)$$

де  $k = 0.6$  для двигунів з компенсаційною обмоткою,  $k = 0.25$  для двигунів без компенсаційної обмотки, після чого визначте електромагнітну сталу часу.

5. Розрахуйте електромагнітну  $T_a$  та електромеханічну  $T_m$  сталі часу двигуна за формулами (5.13) та (5.22).

6. Розрахуйте коефіцієнт демпфірування  $\xi$  та перерегулювання  $\sigma$  перехідної функції двигуна за формулами (5.31) та (5.32).

7. Задайтесь бажаною величиною динамічного моменту у пуско-гальмівних режимах ( $M_{j0} = (0,5 \div 1,5)M_{\text{н}}$ ) та розрахуйте час зміни напруги якоря  $t_{p0}$  за формулою (5.33).

8. Задайтесь бажаною величиною зміни прискорення за лінійним законом ( $t_{\varepsilon} = (0,2 \div 0,4)\text{с}$ ) та розрахуйте обмеження на ривок  $\rho_0$  за формулою (5.34).

9. Після завершення моделювання порівняйте отримані статичні та динамічні показники якості з розрахованими.



*Лабораторна робота №6***МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ДВИГУНА ПОСТІЙНОГО СТРУМУ З КЕРУВАННЯМ У КОЛІ ЯКОРЯ ТА КОЛІ ЗБУДЖЕННЯ**

*Мета роботи:* дослідити статичні та динамічні властивості двигуна постійного струму з керуванням у колі якоря та у колі збудження при роботі у першій та другій зонах регулювання швидкості.

**6.1 Математичний опис об'єкту моделювання**

У ДПС зі змінним потоком збудження рівняння електричної рівноваги та рівняння руху не відрізняються від відповідних рівнянь (6.1) та (6.2) при постійному потоці, але рівняння моменту і ЕРС набувають вигляду [18, 20]

$$M(t) = k \Phi_3(t) I_{\text{я}}(t), \quad (6.1)$$

$$E_{\text{д}}(t) = k \Phi_3(t) \omega(t). \quad (6.2)$$

До них ще додається рівняння електричної рівноваги кола збудження:

$$U_3(t) = i_3(t)R_3 + L_3 \frac{di_3(t)}{dt}. \quad (6.3)$$

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

$$L_3 = k_3 \frac{d\Phi_3(t)}{di_3(t)}, \quad (6.4)$$

де

$$k_3 = 2p\xi w_n, \quad (6.5)$$

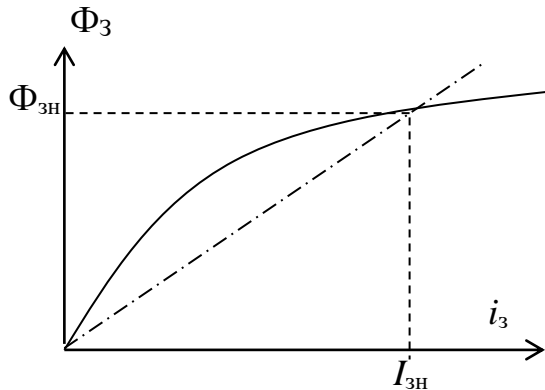
$w_n$  – кількість витків обмотки збудження на полюс,

$\xi=1+(0.5\div 0.7)(\sigma-1)$  – 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 (6.4) у(6.3) отримуємо коефіцієнт

$$U_3(t) = i_3(t)R_3 + k_3 \frac{d\Phi_3(t)}{dt}. \quad (6.6)$$



Загальний вигляд кривої намагнічування подано на рис. 6.1. Зазвичай проектувальники визначають її експериментально і подають у довідниках у вигляді таблиці.

Рис. 6.1. Крива намагнічування ДПС

Коло збудження характеризується *сталюю часу збудження*

$$T_3 = \frac{L_3(i_3, \Phi_3)}{R_3} = \frac{k_3}{R_3} \cdot \frac{d\Phi_3(t)}{di_3(t)} = \text{var}. \quad (6.7)$$

*Усереднене значення сталої збудження* визначають із лінеаризованої характеристики намагнічування за формулою

$$T_{3н} = \frac{L_{3н}}{R_3} = \frac{k_3}{R_3} \cdot \frac{\Phi_{3н}}{i_{3н}} = \frac{k_3 k_{\Phi н}}{R_3} = \text{const}, \quad k_{\Phi н} = \frac{\Phi_{3н}}{i_{3н}}. \quad (6.8)$$

Структурна схема ДПС, складена за рівняннями (5.1), (6.1), (5.3), (6.2) і (5.9) зі врахуванням позначення (5.10), зображена на рис. 6.2.

Треба відзначити, що, додатково до припущень, прийнятих у лабораторній роботі №5, модель рис. 6.2 не враховує дію вихрових струмів, які виникають у масивних частинах магнітної системи потужних двигунів при зміні магнітного потоку.

Як бачимо модель на рис. 6.2 є двоканальною, тобто має 2 канали для регулювання швидкості.

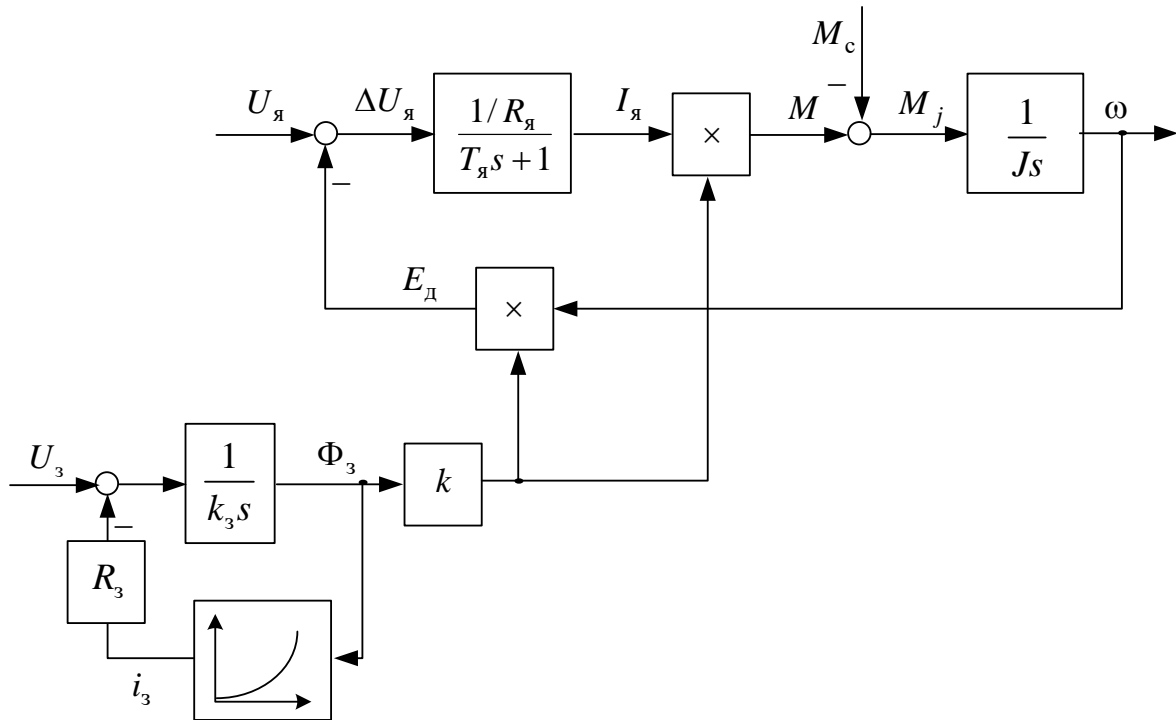


Рис. 6.2. Структурна схема двигуна постійного струму з керуванням у колах якоря і збудження

*Діапазон регулювання швидкості, що забезпечується зміною напруги якоря при постійній напрузі збудження (від 0 до номінального значення) називають першою зоною, а діапазон, що забезпечується зміною напруги збудження, при постійній напрузі якоря (вище номінального значення) – другою зоною.*

## 6.2 Розробка *Simulink*-моделей для дослідження ДПС з двозонним регулюванням швидкості

Для дослідження властивостей ДПС при роботі його у першій та другій зонах регулювання швидкості можна використати діаграми сигналів завдання у відносних одиницях, наведені на рис. 6.3, де позначено:

$t_3 = 4T_{3н}$  – час збудження двигуна за експоненційним законом ;

$t_{p0}$  – бажаний час розгону ДПС у першій та другій зонах;

$t_{по}$  – час початку ослаблення поля.

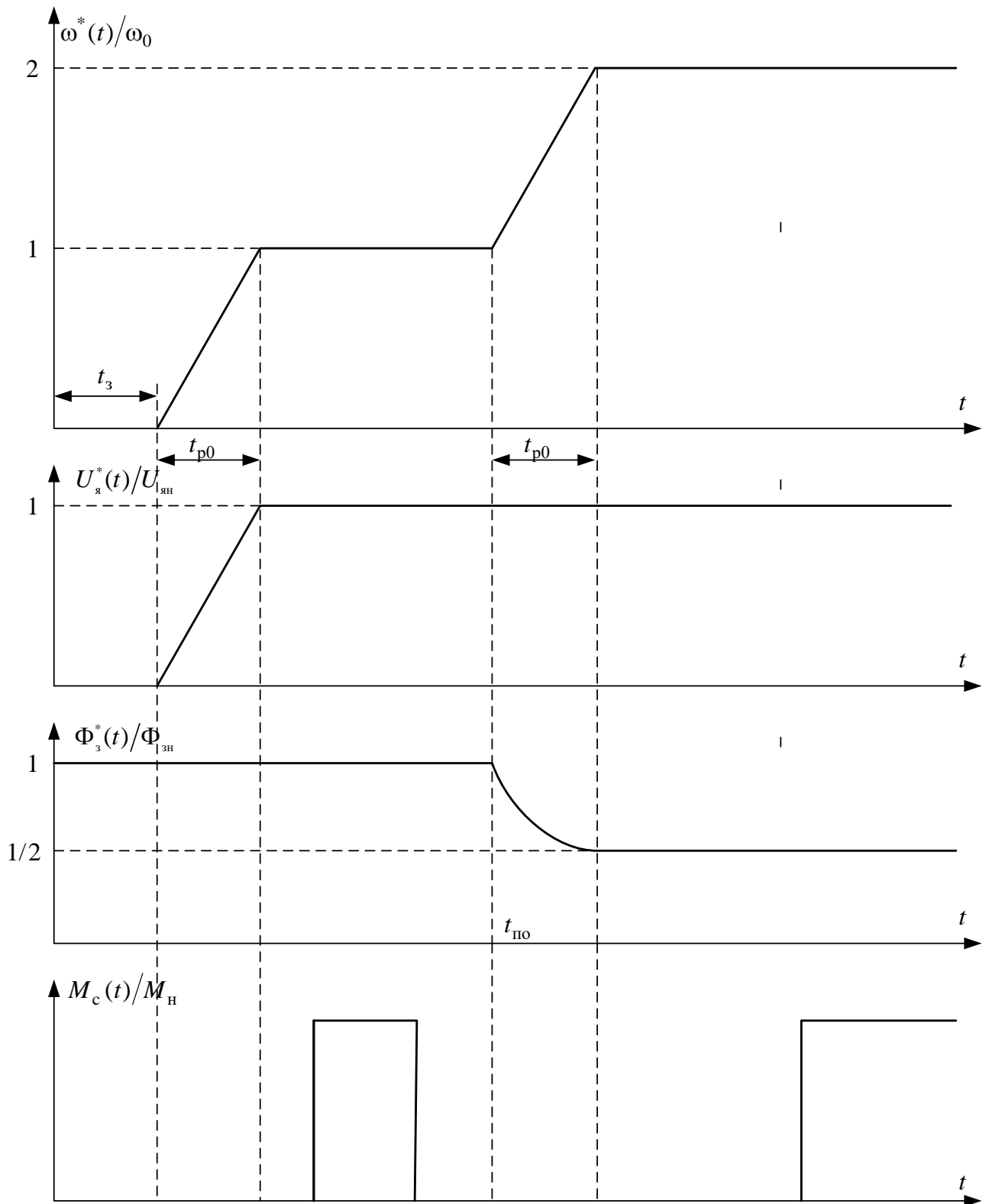


Рис. 6.3. Діаграми сигналів завдання до двозонної системи регулювання швидкості ДПС у відносних одиницях

У відповідності з рис. 6.3 спочатку необхідно стрибком подати на двигун напругу збудження  $U_3 = U_{3н}$ . Після того, як під дією цього сигналу потокозчеплення ДПС досягне номінального значення  $\Phi_3 = \Phi_{3н}$ , можна почати змінювати за лінійним законом напругу якоря до номінального значення. Після розгону двигуна до швидкості ідеального холостого ходу спостерігаємо реакцію системи на стрибкоподібну зміну моменту статичного опору  $M_c$ . Потім збільшуємо швидкість у 2 рази за рахунок зменшення напруги збудження і відповідного ослаблення поля за гіперболічним законом при постійній напрузі якоря, і на усталеній швидкості знову накидаємо навантаження.

Діаграми задавальних сигналів, зображені на рис. 6.3, можна реалізувати в *Simulink* декількома способами, один з яких представлено на рис. 6.4.

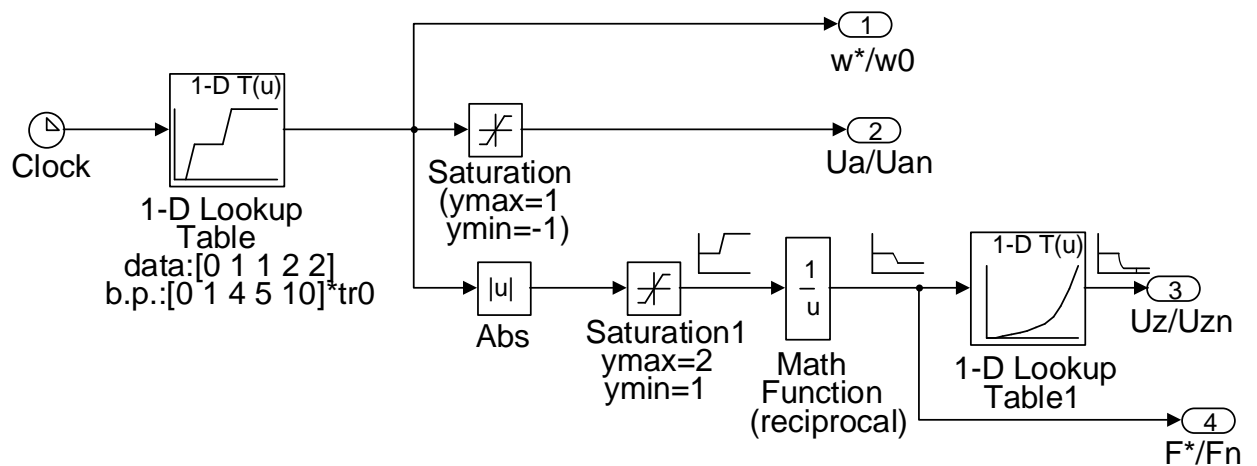


Рис. 6.4. *Simulink*-модель формування у в.о. вхідних сигналів для ДПС, що керується зміною напруги якоря та напруги збудження

*Simulink*-модель ДПС з перетворенням відносних вхідних сигналів у абсолютні та абсолютних вихідних сигналів у відносні зображена на рис. 6.5, а загальна структурна модель для досліджень, в якій модель двигуна рис. 6.5 подана у вигляді незамаскованої підсистеми – на рис. 6.6.

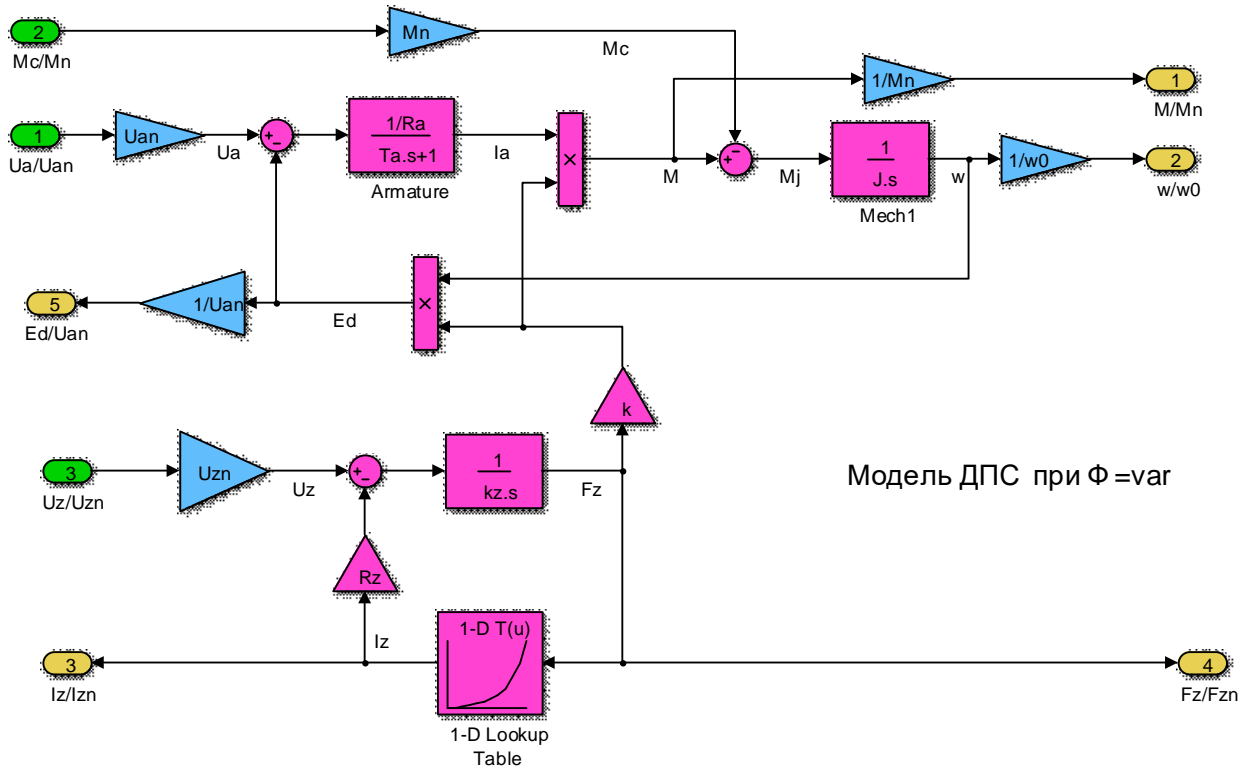


Рис. 6.5. Структурна модель ДПС, що керується зміною напруги якоря та напруги збудження

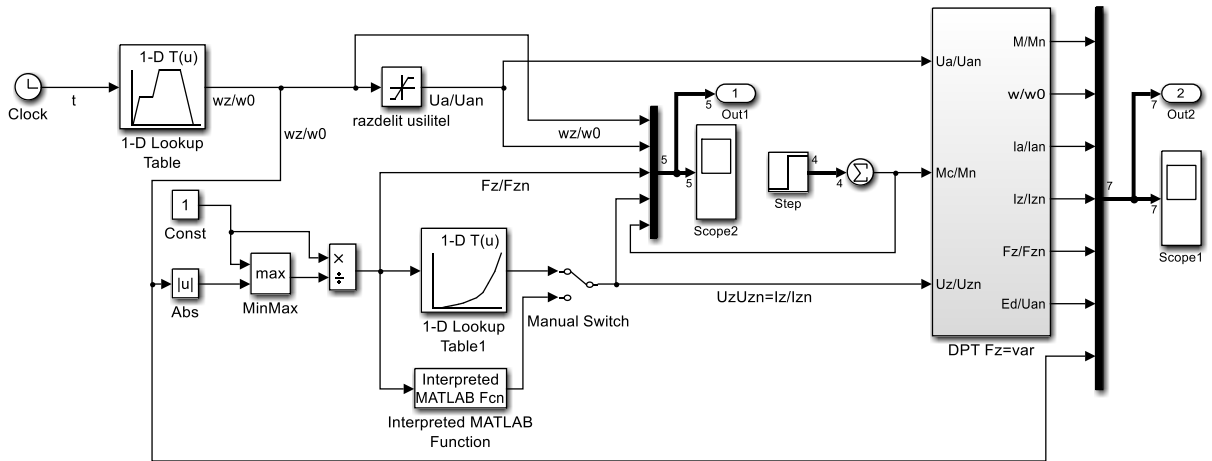


Рис. 6.6. Структурна модель для дослідження ДПС

Щоб підтримувати прискорення у другій зоні на тому ж рівні, що і у першій, треба завдання на магнітний потік у в.о. розраховувати за формулою

$$\Phi^*/\Phi_H = \begin{cases} 1 & \text{при } |\omega^*/\omega_0| \leq 1, \\ \frac{1}{|\omega^*/\omega_0|} & \text{при } |\omega^*/\omega_0| > 1. \end{cases} \quad (6.9)$$

Для формування нелінійної функціональної залежності  $i_3^* = f(\Phi^*)$  в моделі рис. 6.6 використано кусково-лінійну інтерполяцію та кубічну інтерполяцію. Графіки перехідних процесів представлені на рис. 6.7-6.9. Можна ще здійснити апроксимацію цієї залежності поліномом 5-го порядку.

На моделі рис. 6.6 завдання на швидкість доповнено ділянкою гальмування двигуна зі швидкості  $2\omega_0$  до 0.

### 6.3 Аналіз перехідних процесів

На рис. 6.7 зображені у відносних одиницях графіки сигналів, підключених до вихідного порту на моделі рис. 6.6: завдання на швидкість  $wz^*/w0$  (бажана тахограма), необхідні для забезпечення такої тахограми завдання на напругу якоря  $Ua^*/Uan$ , струм і напругу збудження  $Uz^*/Uzn=iz^*/izn$ , потік збудження  $Fz^*/Fzn$  та момент момент статичного опору  $Mc/Mn$ .

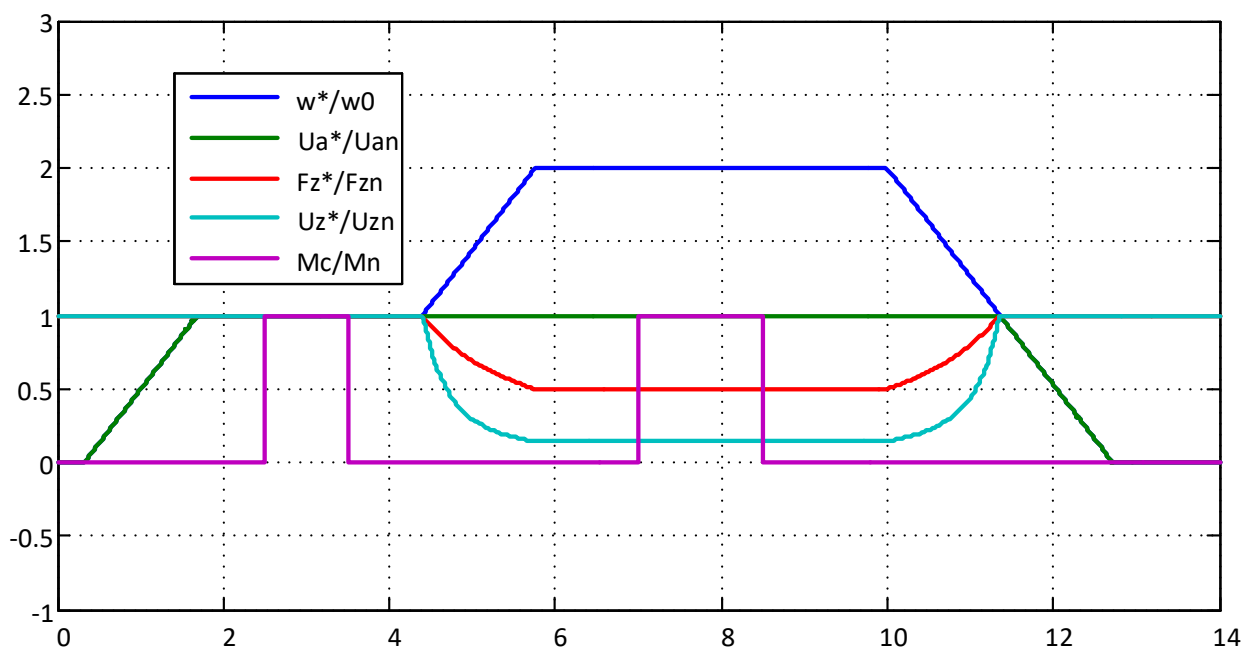


Рис. 6.7. Завдання на швидкість, напругу якоря потік збудження, напругу збудження та момент навантаження у відносних одиницях

З рис. 6.7 видно, що спочатку подається завдання на збудження двигуна, а потім завдання на швидкість. На ділянках руху з постійною швидкістю

відбувається стрибкоподібний накид номінального навантаження та його скидання.

Бажана тахограма і навантажувальна діаграма на рис. 6.7 визначають послідовність таких режимів:

- намагнічування двигуна;
- розгін двигуна до швидкості ідеального холостого ходу з постійним збудженням за рахунок зміни напруги якоря від 0 до номінального значення;
- рух з постійною швидкістю  $\omega = \omega_0$ ;
- накидання та скидання номінального навантаження при збудженні двигуна на номінальним потоком;
- розгін від швидкості до швидкості  $\omega_0$  до швидкості  $2\omega_0$  з постійною (номінальною) напругою за рахунок ослаблення поля;
- рух з постійною швидкістю  $\omega = 2\omega_0$ ;
- накидання та скидання номінального навантаження при ослабленому полі;
- гальмування від швидкості  $2\omega_0$  до 0.

Реакція двигуна на входні сигнали рис. 6.7 при кусково-лінійній інтерполяції кривої намагнічування показана на рис. 6.8

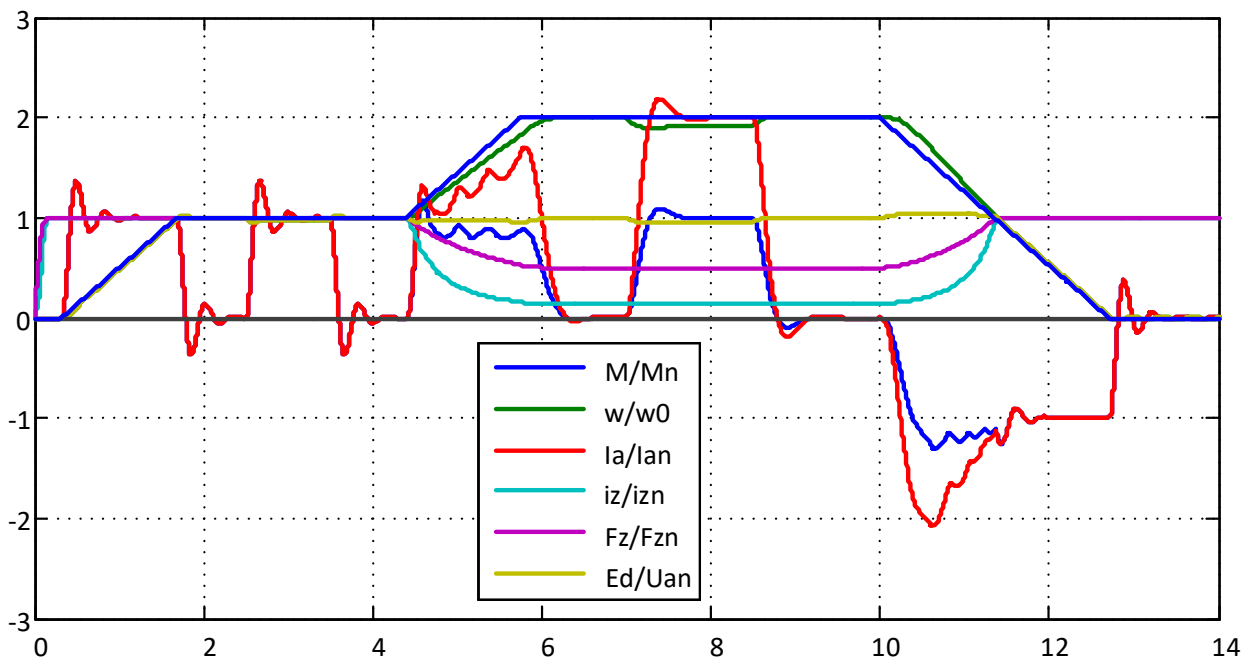


Рис. 6.8. Реакція двигуна постійного струму на входні сигнали рис. 6.7 при кусково-лінійній інтерполяції кривої намагнічування



Із рис. 6.8 видно, що перехідні процеси у другій зоні стають менш коливальними та більш повільними ніж у першій зоні. Це пояснюється зменшенням магнітного потоку в кінці другої зони в  $d_{\text{осл}} = \frac{\Phi_{\text{н}}}{\Phi_{\text{осл}}}$  раз ( $d_{\text{осл}}$  – діапазон ослаблення поля) та електромеханічної сталої часу – у  $d_{\text{осл}}^2$  раз:

$$T_{\text{мосл}} = \frac{JR_{\text{я}}}{\Phi_{\text{осл}}^2} = \frac{JR_{\text{я}}}{\Phi_{\text{н}}^2} d_{\text{осл}}^2 = T_{\text{мн}} d_{\text{осл}}^2. \quad (6.9)$$

Те ж саме стосується і статичної просадки швидкості при нахиді навантаження

$$\Delta\omega_{\text{с осл}} = \frac{I_{\text{с}} R_{\text{я}}}{k\Phi_{\text{осл}}} = \frac{M_{\text{с}} R_{\text{я}}}{(k\Phi_{\text{осл}})^2} = \Delta\omega_{\text{сн}} d_{\text{осл}}^2. \quad (6.10)$$

Це підтверджується порівнянням статичних механічних характеристик, наведених на рис. 6.9.

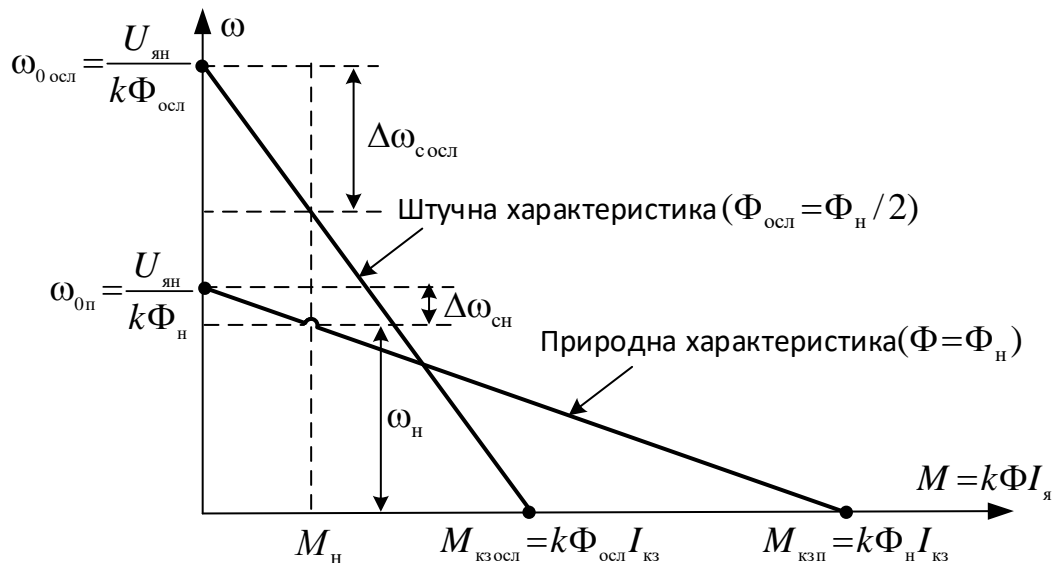


Рис. 6.9. Природна та штучна механічні характеристики ДПС

Ще варто звернути увагу на те, що в першій зоні (при  $\Phi = \Phi_{\text{н}} = \text{const}$ ) згідно з рівняннями (6.2) і (6.4) відношення електромагнітного моменту до струму якоря та ЕРС двигуна до його кутової швидкості залишаються постійними, а відносні значення цих величин – однаковими:  $M/M_{\text{н}} = I_{\text{я}}/I_{\text{ян}}$ ,  $E_{\text{д}}/U_{\text{ян}} = \omega/\omega_0$ .

У другій зоні швидкість двигуна збільшується при майже постійній ЕРС, а струм якоря збільшується при майже постійному моменті за рахунок зменшення магнітного потоку.

Коливання електромагнітного моменту і струму якоря на ділянках регулювання магнітного потоку на рис. 6.8 пов'язані з кусково-лінійною апроксимацією кривої намагнічення, зумовлені розривним характером похідної від нелінійної функції в точках зламу кривої  $i_3(\Phi_3)$ . Щоб позбутися цього явища, треба застосувати інтерполяцію кубічними сплайнами. Для цього необхідно в ланці *I-D Look up Table* якщо у змінити алгоритм інтерполювання (параметр *Algorithm*  $\rightarrow$  *Interpolation method*), який за замовченням має значення *Linear*, на алгоритм *Cubic spline*. При цьому розрахунок здійснюється набагато швидше, ніж при використанні блоку *Interpreted MATLAB Function*.

Графіки перехідних процесів у досліджуваній системі при заміні методу інтерполювання показані на рис. 6.10.

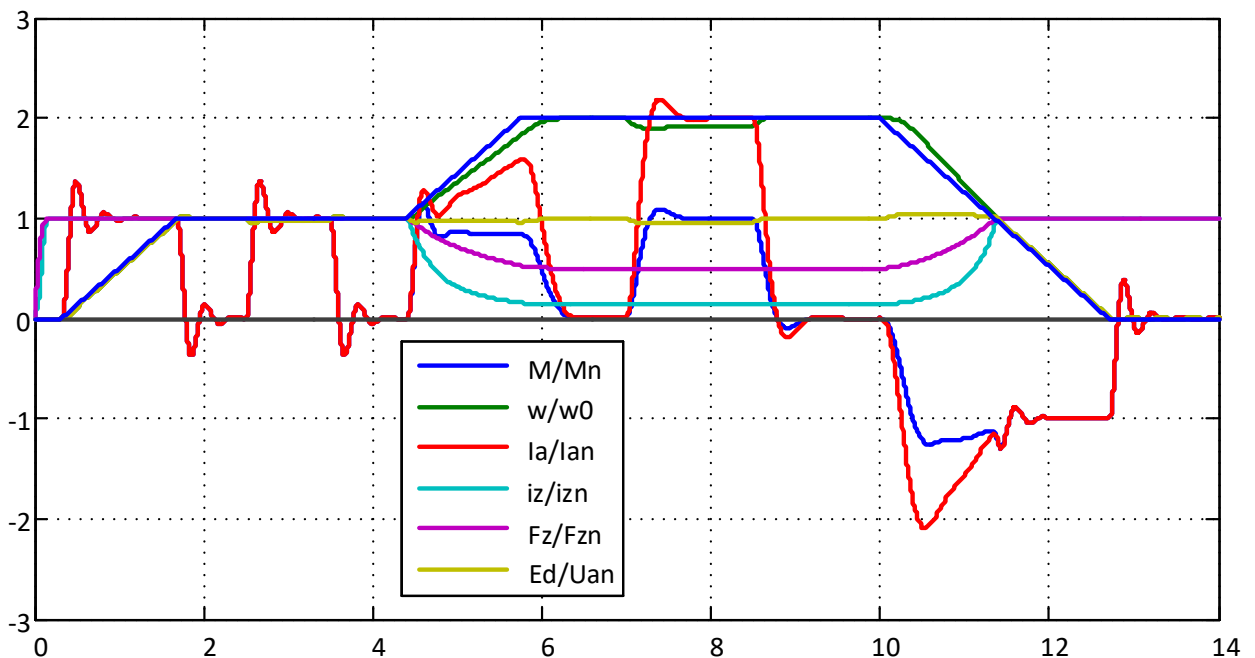


Рис. 6.10. Реакція двигуна постійного струму на вхідні сигнали рис. 6.7 при інтерполяції кривої намагнічування кубічними сплайнами

### 6.4 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 накиньте номінальне навантаження, а після закінчення перехідного процесу – скиньте його.
- після повернення двигуна в усталений режим холостого ходу змінійте напругу збудження у такий спосіб, щоб завдання на ослаблення магнітного поля відбулося за законом

$$\Phi_3^*(t) = \frac{U_{\text{ян}}}{k \omega^*(t)}, \quad (6.11)$$

де

$$\omega^*(t) = \begin{cases} \omega_0 = \text{const} & \text{при } 0 \leq t \leq t_{\text{по}}, \\ \frac{\omega_0}{t_{\text{р0}}} t & \text{при } t_{\text{по}} \leq t \leq t_{\text{по}} + t_{\text{р0}}, \\ 2\omega_0 = \text{const} & \text{при } t > t_{\text{по}} + t_{\text{р0}}, \end{cases} \quad (6.12)$$

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

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

Проаналізуйте статичні та динамічні властивості ДПС з ослабленим потоком збудження, порівняйте їх з результатами попередньої роботи.

Координати вузлових точок кривої намагнічування у відносних одиницях подані у табл. 6.1.

Таблиця 6.1

$\bar{I}_3$	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

### 6.5 Методичні рекомендації

1. 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.

2. Час намагнічування двигуна визначте за приблизною формулою

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

а час наростання напруги якоря від 0 до номінального значення – за формулою (5.33). Інші часові інтервали обирайте у процесі виконання модельного експерименту.

3. Для формування нелінійної функціональної залежності  $i_3^* = f(\Phi^*)$  спробуйте використати кусково-лінійну інтерполяцію, кубічну інтерполяцію та апроксимацією степеневим поліномом 5-го порядку. Порівняйте результати. Обґрунтуйте обраний варіант.

## *Лабораторна робота №7*

### СТВОРЕННЯ ПІДСИСТЕМ ТА ЇХ МАСКУВАННЯ

*Мета роботи:* навчитися створювати не замасковані та замасковані підсистеми, редагувати параметри замаскованих підсистем, створювати віртуальні «кнопки», що виконують ініціалізацію моделі, побудову графіків, деякі модельні експерименти тощо.

#### 7.1 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 *Subsystem*, котрий при подвійному щиглику мишею по його піктограмі відчиняє власне вікно, в якому збирається модель підсистеми. До входів підсистеми приєднують вхідні (*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 вхідні та вихідні порти додаються до фрагменту, що утворює підсистему, автоматично.

Спосіб позначення вхідних та вихідних портів блоку *Subsystem* визначається користувачем через меню *Format* → *Port Labels*, де можна обрати опції *None*, *From Port Icon*, *From Port Block Name* та *From Signal Name*.

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 вхідних та вихідних змінних моделі. Створену підсистему також бажано перейменувати у відповідності з назвою об'єкту, структуру якого вона відображує. Побачити зміст незамаскованої підсистеми можна подвійним щигликом миші (*double click*) по її іконці.

На рис. 7.1а показана розгорнута підсистема двигуна постійного струму, а на рис. 7.1б згорнуті підсистеми зі схованими і видимими іменами портів та підсистема в режимі *Format* → *Content preview*, на якій крізь іконку згорнутої підсистеми просвічується розгорнута підсистема.

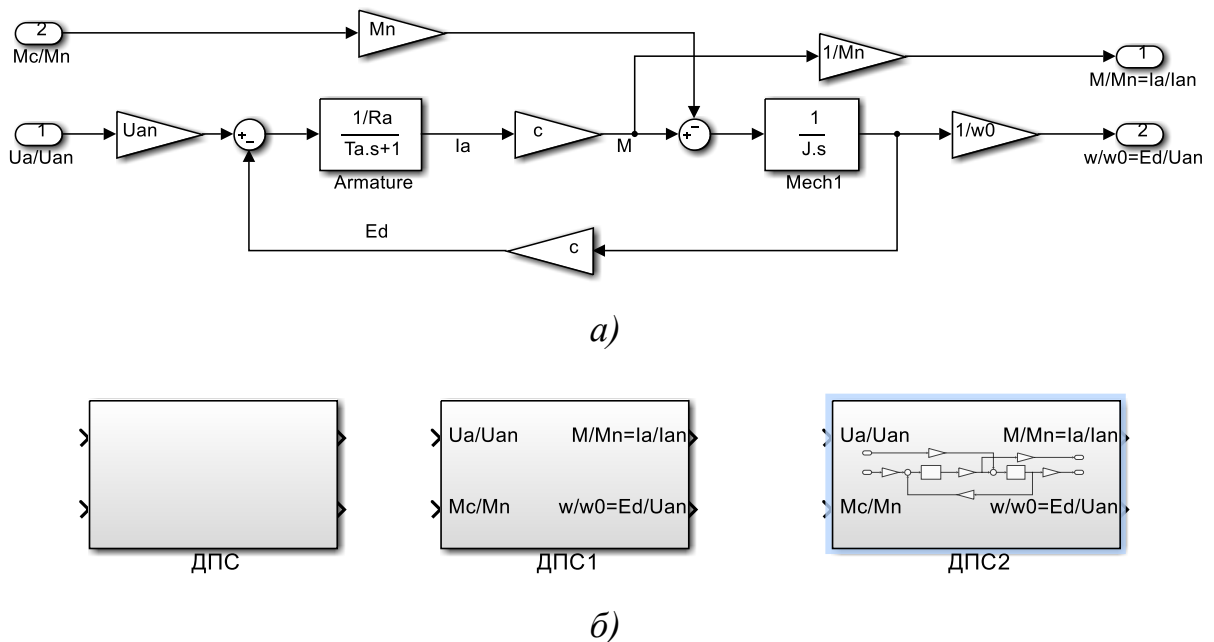


Рис. 7.1. Розгорнута (а) та згорнуті (б) підсистеми двигуна постійного струму

## 7.2 Маскування підсистем

Будь-яку підсистему можна замаскувати.

Маскування підсистеми починають командою *Edit* → *Mask Subsystem* (^M), на яку система реагує відкриттям вікна маскування *Mask Editor*, що має 4 вкладки (панелі): *Icon & Ports*, *Parameters&Dialog*, *Initialization* та *Documentation*, перша з яких показана на рис. 7.2.

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.

**Вкладка *Icon & Ports*** (див. рис. 7.2) призначена для зміни піктограми (іконки) підсистеми та позначень портів. Зовнішнім видом іконки керують команди, занесені в поле *Icon Drawing commands*.

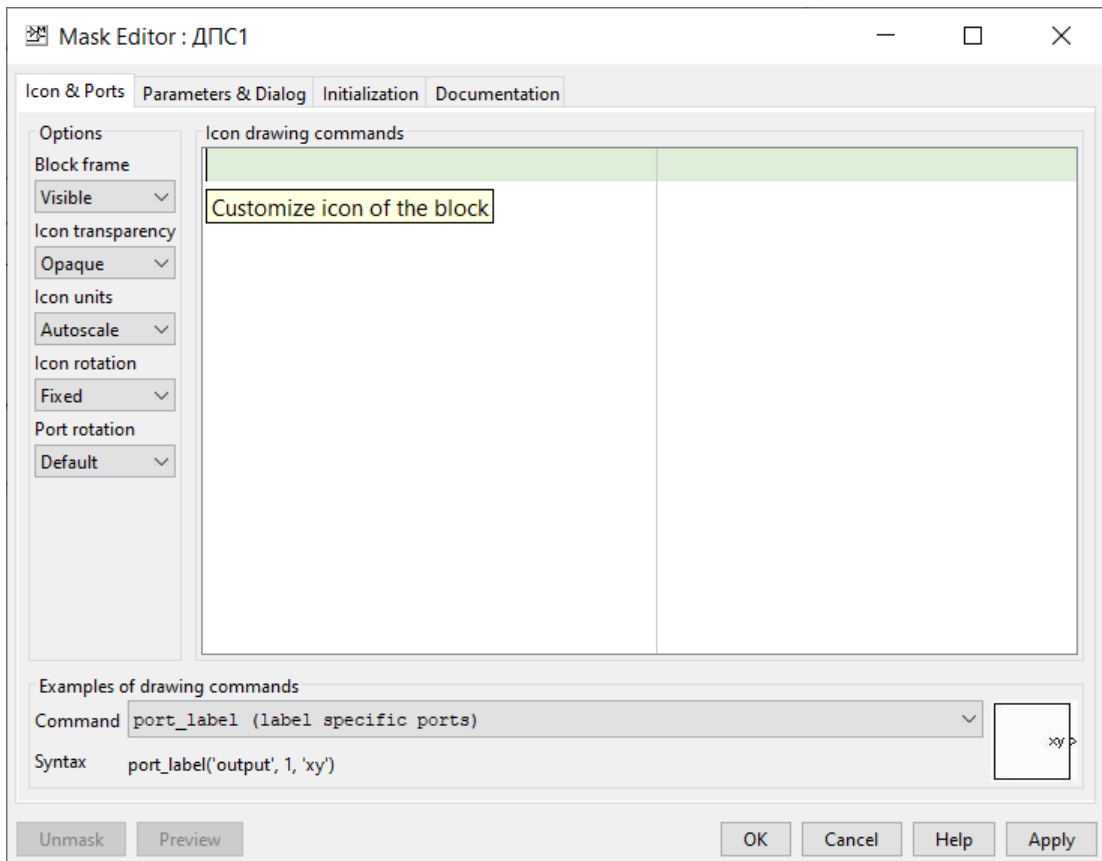


Рис. 7.2. Вкладка *Icon & Ports* редактора маскування

Принципово у піктограмі блоку може бути виведеним будь-який текст, передавальна функція, графік статичної або динамічної характеристики та не дуже складне графічне зображення.

Для виведення тексту можна використовувати один з наступних операторів:

`disp ('text')`

`text (x,y,'text'),`

`text (x,y,'text','HorizontalAlignment',HorAl 'VerticalAlignment',VertAl)`

`port_label (port_type, port_number,label)`

Для розподілу тексту на рядки 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. 7.3.

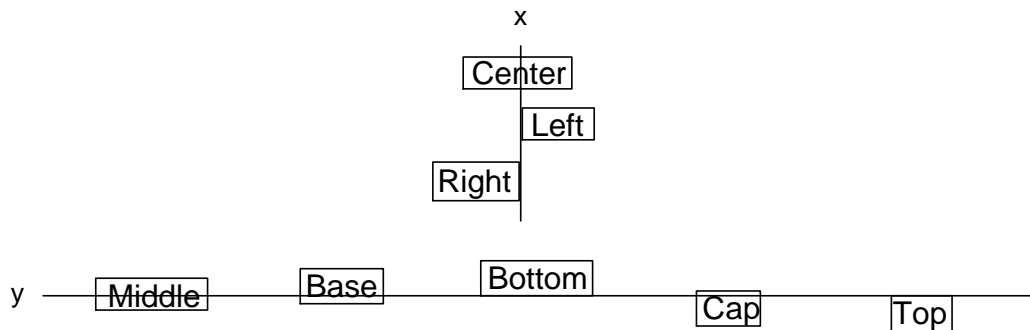


Рис. 7.3. Демонстрація дії опцій вирівнювання текстів в іконці

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. Необов'язковий символний параметр `char` має за замовчанням значення `'s'`. Для виведення дискре-



тних ПФ його можна змінити на 'z' або 'z^-1'.

Параметри команд `dpoly` і `droots` звичайно визначаються через параметри підсистеми у полі *Initialization commands* вкладки *Initialization*. Для прикладу на рис. 7.4 показана у розгорнутому (а) та згорнутому (б, в) виглядах підсистема *Aperiodic link with initial value* (Аперіодична ланка з початковими умовами). На рис. 7.4б іконка замаскованої підсистеми сформована командою `dpoly`, а на рис 7.4в – командою `droots`, для чого у полі *Initialization commands* записані оператори присвоювання

```
num = [k]; den = [T 1];
```

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

відповідно.

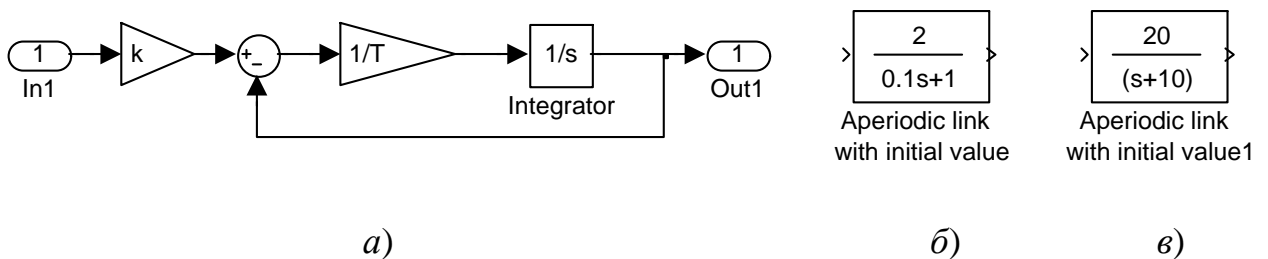


Рис. 7.4. Розгорнута (а) та згорнуті (б, в) підсистеми *Aperiodic link with initial value*

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 (;).

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),
```

де  $xlb, ylb, xrt, yrt$  – явно визначають координати лівого нижнього ( $lb$  – *left, bottom*) та правого верхнього ( $rt$  – *right, top*) кутів іконки.

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

$$\text{patch}(x, y, [r \ g \ b])$$

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

$$\text{patch}([-1 \ 0 \ 1], [0 \ 1 \ 0], [0 \ 1 \ 1])$$

створює піктограму у вигляді трикутника, зафарбованого бірюзовим кольором, зображену на рис. 7.5.

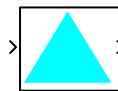


Рис. 7.5. Замаскована підсистема, іконка якої створена командою `patch`

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

$$\text{image}(\text{imread}(\text{'FileName.Extension'}))$$

де `FileName` – ім'я графічного файлу, а `Extension` – його розширення.

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 іконки керують опції, імена яких розташовані ліворуч, а значення обираються за допомогою *popup-menu*. Інформація про ці опції наведена в табл. 7.1.

Таблиця 7.1

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>Icon Units</i> (Система Координат)	<i>Autoscale</i>	$x_{min} = \min(\mathbf{X})$ , $x_{max} = \max(\mathbf{X})$ , $y_{min} = \min(\mathbf{Y})$ , $y_{max} = \max(\mathbf{Y})$
	<i>Normalized</i>	$x_{min} = 0$ , $x_{max} = 1$ , $y_{min} = 0$ , $y_{max} = 1$
	<i>Pixel</i>	$x_{min} = 0$ , $x_{max} = \text{block width}$ , $y_{min} = 0$ , $y_{max} = \text{block height}$

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);  height = pos(4) – pos(2);
```

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

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

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 такої замаскованої підсистеми треба „зазирнути під маску” виконанням команди *Edit* → *Look Under Mask* (^U).

У будь-який момент з будь-якої вкладки можна подивитися, як впливають дії, виконані в процесі маскування, на вигляд вікна *Block Parameters*. Для цього треба натиснути віртуальну кнопку *PreView* у нижній частині вікна редактора маскування.

**Визначення параметрів замаскованої підсистеми здійснюється у вкладці *Parameters&Dialog*** (рис. 7.6).

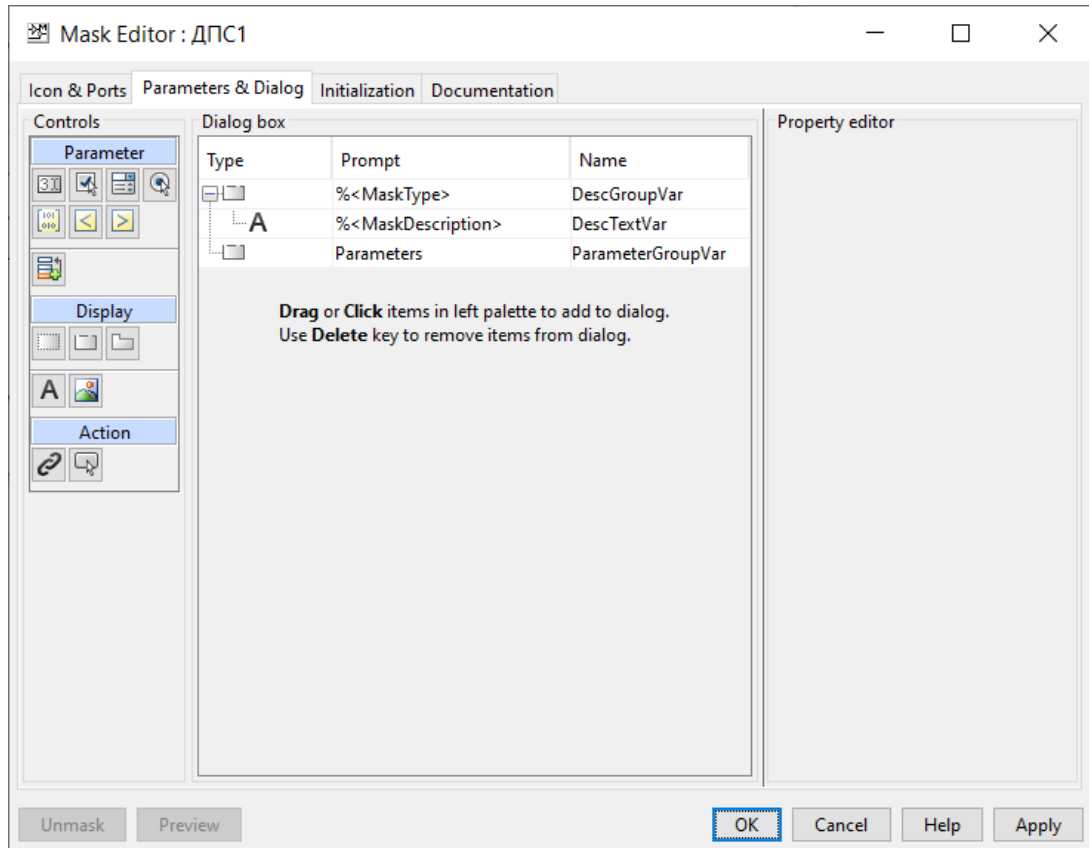
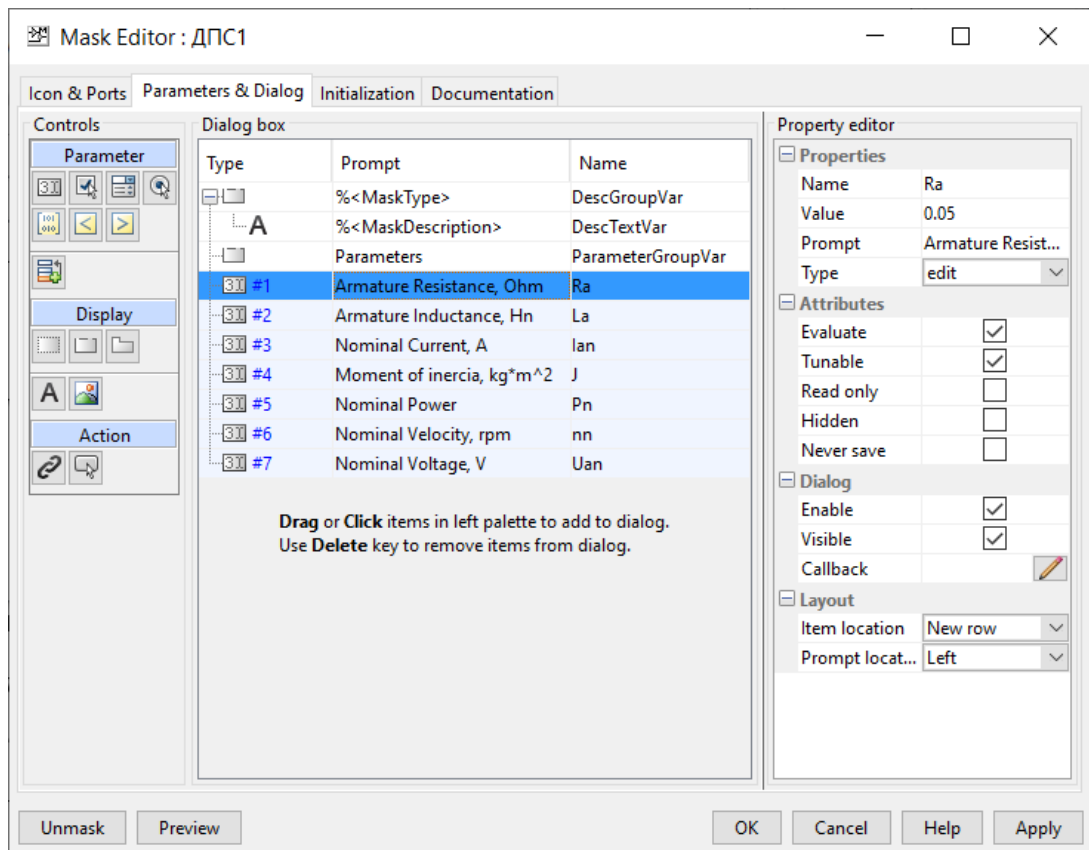
З лівого боку цієї вкладки знаходиться панель керування *Controls* параметрами (*Parameters*), дисплеєм (*Display*) та діями (*Actions*).

Керування основними типами параметрів здійснюють 3 кнопки:

- *Edit* – введення константи, змінної або математичного 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;
- *PopUp* – вибір варіанту з випадаючого меню.

Після вибору типу параметру у першому стовпчику *Type* таблиці *Dialog box* (центральна частина вкладки) з’являється номер параметру (наприклад, #1) з іконкою, що відповідає його типу. Тепер у другому стовпчику цього рядка необхідно ввести запрошення (*Prompt*) до введення параметру у вигляді тексту, а у третьому стовпчику (*Name*) – ім’я змінної, яка отримає значення після введення його у відповідь на запрошення у вікні параметрів замаскованого блоку.

Приклад вкладки для маскування моделі двигуна постійного струма, зображеної на рис. 7.1a, наведено на рис. 7.7.

Рис. 7.6. Вкладка *Parameters & Dialog* редактора маскуванняРис. 7.7. Приклад вкладки *Parameters & Dialog* для маскування моделі двигуна потійного струму рис.7.1а

Значення обраного або введеного параметра може використовуватись як у чисельній формі, так і у форматі рядка символів. Спосіб обчислення значення параметра в режимі *Evaluate* (обчислити значення) пояснюється у табл. 7.2.

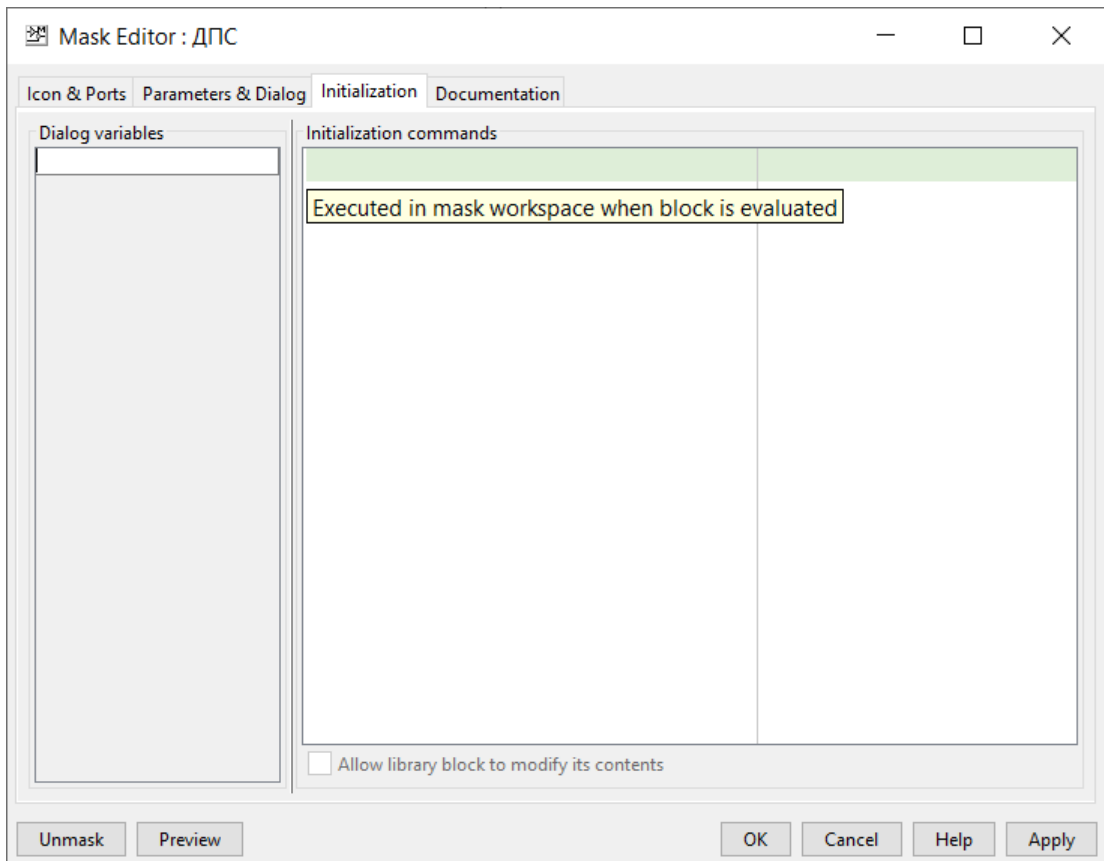
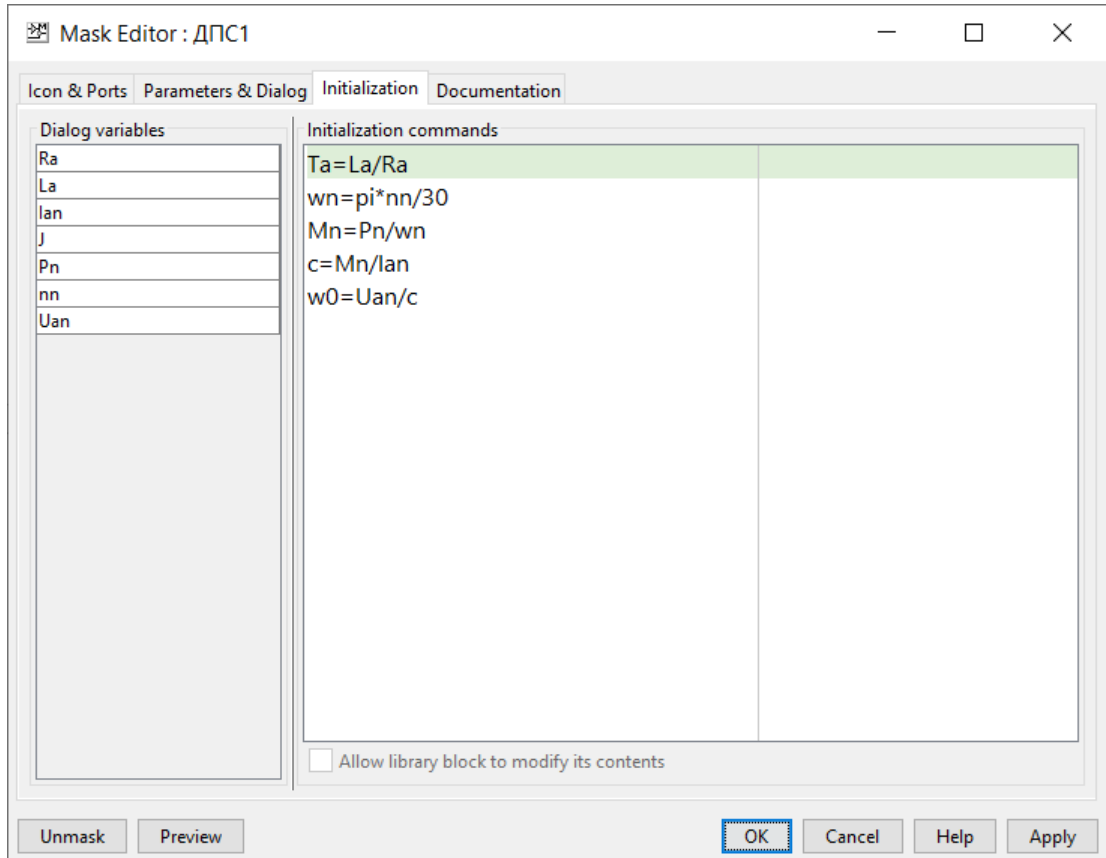
Таблиця 7.2

<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

Результати діалога (*Dialog box*) вкладки *Parameters & Dialog* відображаються в опції *Properties* панелі *Property editor*, розташованій праворуч. Інші опції цієї панелі *Attribute*, *Dialog* і *Layout*, призначені за замовчанням, можна змінити, але ці налаштування не є принциповими і краще їх залишити недоторканими. Те ж саме відноситься і до опцій керування *Display* та *Actions* панелі *Controls*.

У вкладці *Initialization commands* (рис. 7.8) вводяться команди ініціалізації, що пов'язують задані іменами перемінних параметри замаскованої підсистеми з параметрами блоку, що її маскує. Справа у тому, що змінні замаскованої підсистеми є локальними и не можуть бути визначені у робочому просторі (*Workspace*) пакете, якщо тільки вони не описані явно як глобальні змінні. Відповідно вони не можуть змінити значень однойменних змінних основного робочого середовища.

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.

Рис. 7.8. Панель *Initialization* редактора маскуванняРис. 7.9. Приклад вкладки ініціалізації замаскованої моделі  
двигуна постійного струму

Імена змінних, які використовуються в командах ініціалізації, відображаються у панелі *Dialog variables* (ліворуч) вкладки ініціалізації.

Приклад вкладки ініціалізації замаскованої підсистеми двигуна постійного струму зі входними параметрами, визначеними на рис. 7.7, наведено на рис. 7.9.

У вкладці **Documentation** (рис. 7.10) визначаються 3 параметри: *Mask type*, *Mask description* та *Mask Help*.

У полі **Mask type** необхідно ввести тип блока. Пізніше цей тип з поміткою (*Mask*) буде відображено у вікні параметрів замаскованого блока.

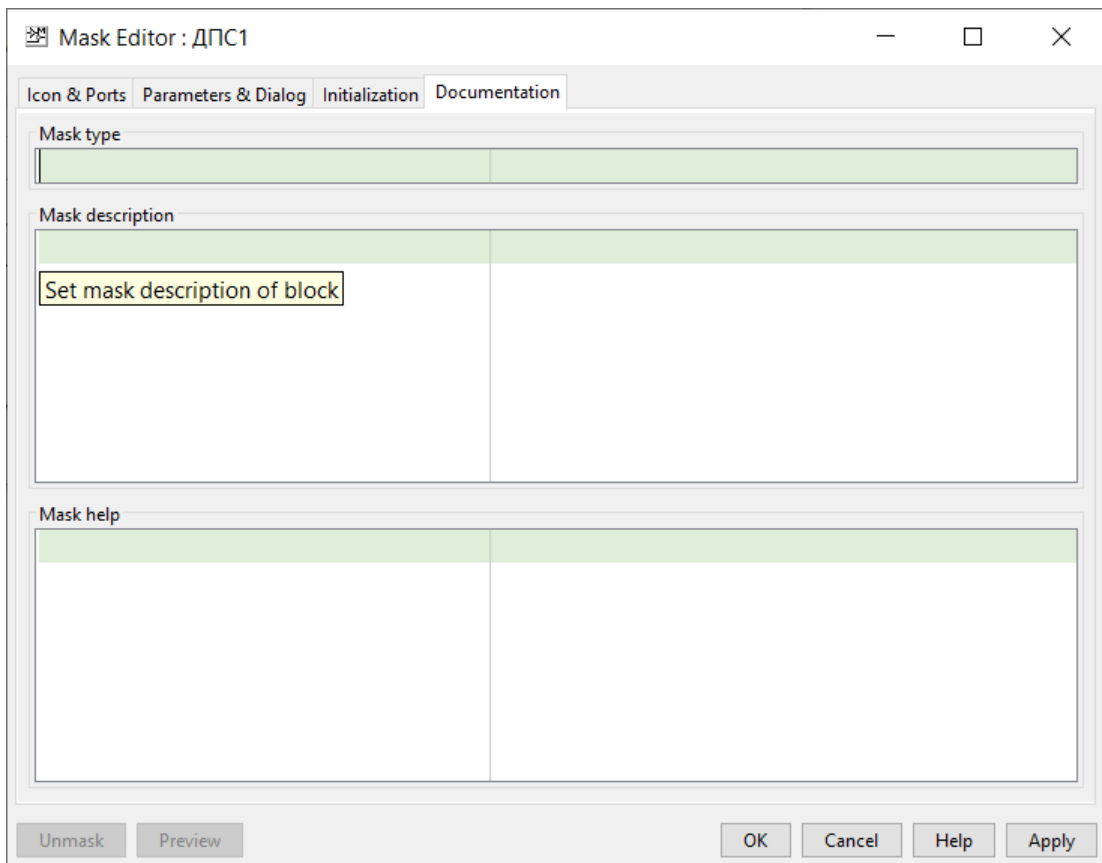


Рис. 7.10. Панель *Documentation* редактора маскуваня

В поле **Mask description** можна ввести короткий коментар про призначення блока та про його параметри, який буде відображений у вікні *Block Parameters*, а в поле **Block Help** – більш докладний опис блока, який буде виводитися на екран в *html*-форматі при натисканні клавіші *Help* в однойменному



вікні. Усі ці параметри є інформаційними і ніяк не впливають на роботу підсистеми.

Відмінити маскування можна командою *Unmask*, яку можна активізувати з кожної вкладки редактора маскування. У полі *Mask type* необхідно ввести тип блока. Пізніше цей тип з поміткою (*Mask*) буде відображено у вікні параметрів замаскованого блока.

Як відображаються результати маскування на вікні введення параметрів замаскованого блока, продемонстровано на рис. 7.11 на прикладі блока *From Workspace*, який має усі три перелічені вище типи визначення параметрів.

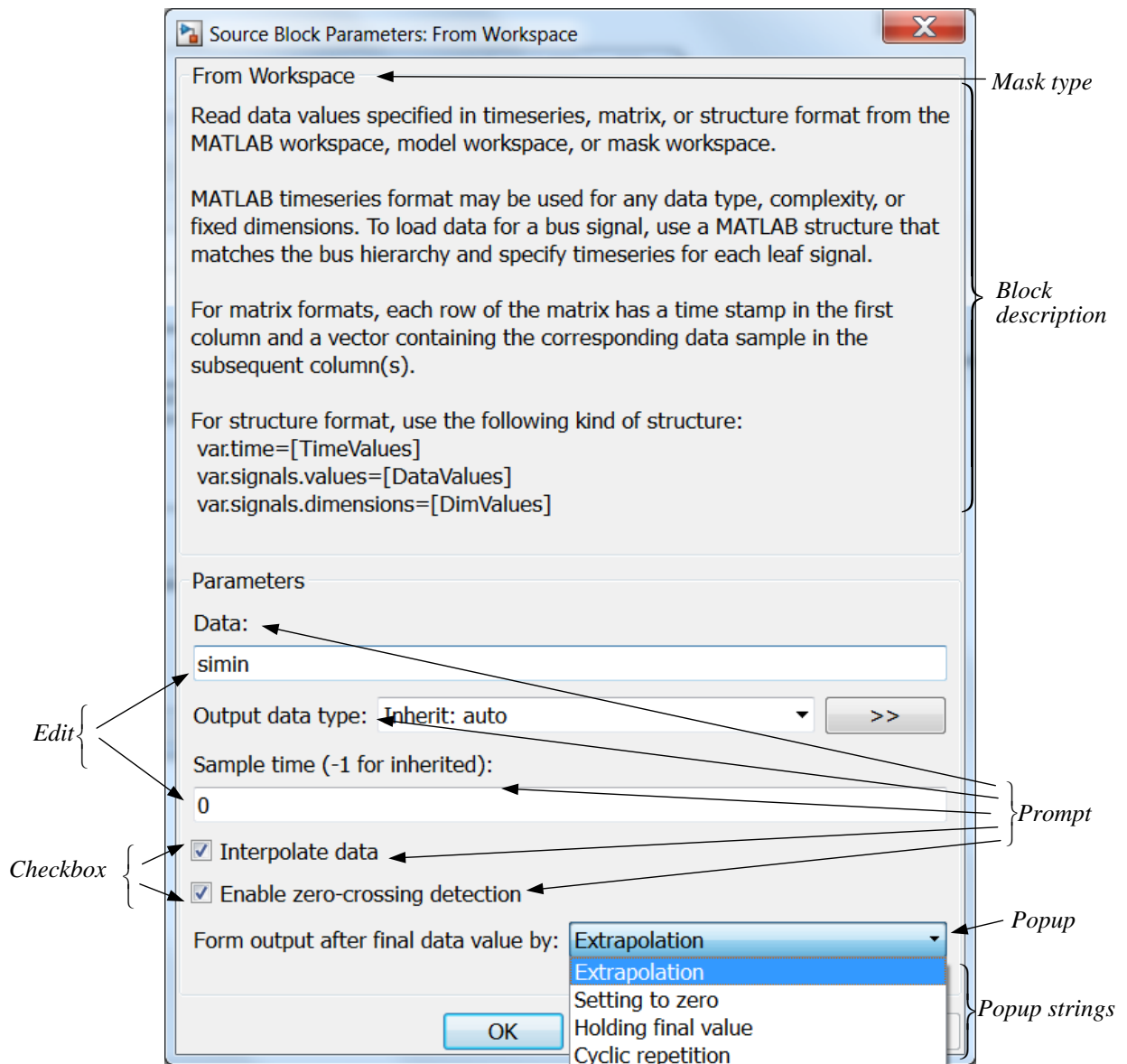


Рис. 7.11. Вікно введення параметрів блока *From Workspace*

На жаль блок *From Workspace* є вбудованим, отже подивитися його вікно маскування не представляється можливим.

Як видно з рис. 7.11, вигляд вікна *Block Parameters* замаскованої підсистеми визначається змістом комірок *Mask type*, *Block description*, *Prompt*, *Variable* та *Control type*.

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

### 7.3 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 модельного 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 ( $M$ ) 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. 7.12).

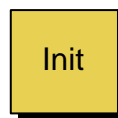


Рис. 7.12. Вигляд „кнопки” *Init*, створеної за наведеною методикою

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

## 7.4 Завдання

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 незамасковані підсистеми ДПС з постійною напругою збудження, ДПС з регульованою напругою збудження та задавального пристрою для двозонного регулювання швидкості ДПС у форматі *Show Port Label*.
6. Створити замасковані підсистеми ДПС для однозонного та двозонного регулювання швидкості.
7. Створити інформаційні „кнопки” та „кнопки” для керування моделями, розробленими при виконанні попередніх лабораторних робіт за вказівкою викладача. Перевірте працездатність створених підсистем та кнопок.

Варіанти завдань до перших чотирьох пунктів завдання відображені у табл. 7.3.

### 7.5 Методичні вказівки та рекомендації

1. Для створення задавальних пристроїв скористуйтеся моделями, поданими на рис. 7.13

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;

$t_p$  – час наростання прискорення до рівня обмеження.

Для перевірки працездатності моделей скористайтесь такими параметрами:  $Wz=1$ ;  $Fiz=1.2$ ;  $tr=100$ ;  $tn=20$ ;  $Wo=1$ ;  $eo=Wo/tr$ ;  $Ro=eo/tn$ ;  $Tv=4$ ;  $k2=b2/Tv$ ;  $k1=b1-k2$ ;  $b2=50$ ;  $b1=10$ .

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

Таблиця 7.3

№ варіантів	Задавальний пристрій (п. 1)	Регулятор з обмеженням (п. 2)	Фільтр (п. 3)			Нелінійні блоки (п. 4)
			Тип	Тип	Маск.блок	Іконка
1, 7, 13, 19	I-ЗІ	Дискретний ПІ	Баттерворта	<i>Transfer Function</i>	Перехідна функція	<i>MATLAB Function</i>
2, 8, 14, 20	I <sup>2</sup> -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

2. Для створення аналогових регуляторів з передавальними функціями

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

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

та обмеженням вихідного сигналу скористайтесь деталізованими моделями, наведеними на рис. 7.14.

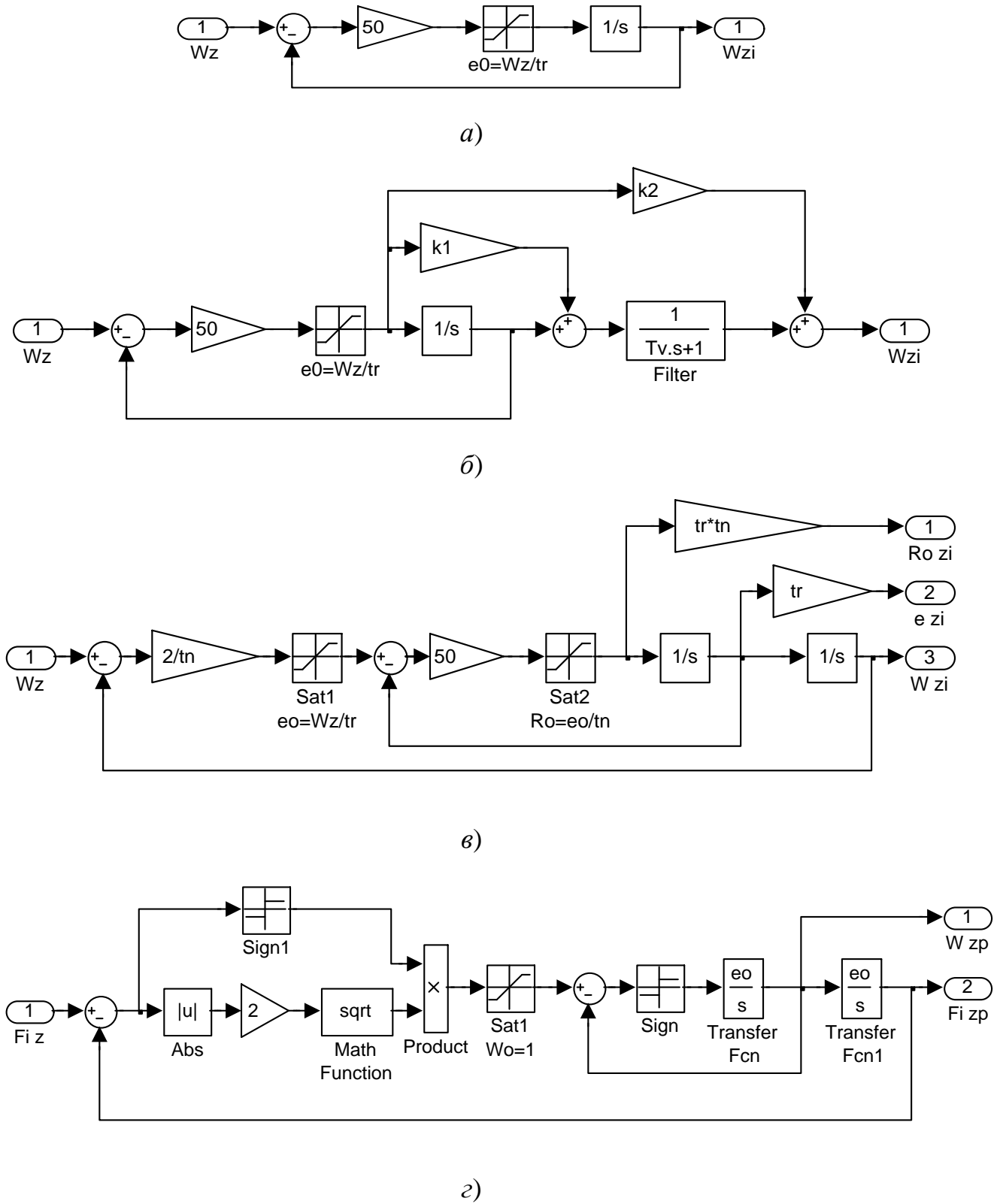


Рис. 7.13. Моделі задавальних пристроїв:  
 а) І-ЗІ, б) ІПД-ЗІ, в) І<sup>2</sup>-ЗІ, г) ЗП

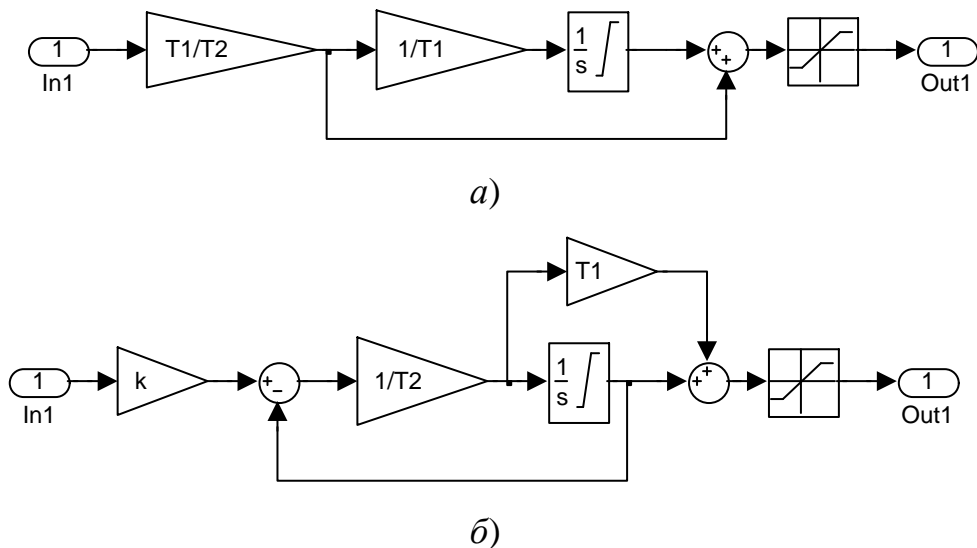


Рис. 7.14. Моделі регуляторів з обмеженням вихідного сигналу:  
а) ПІ; б) ІФ

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. 7.3 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 скористайтеся моделлю, зображеною на рис. 7.15.

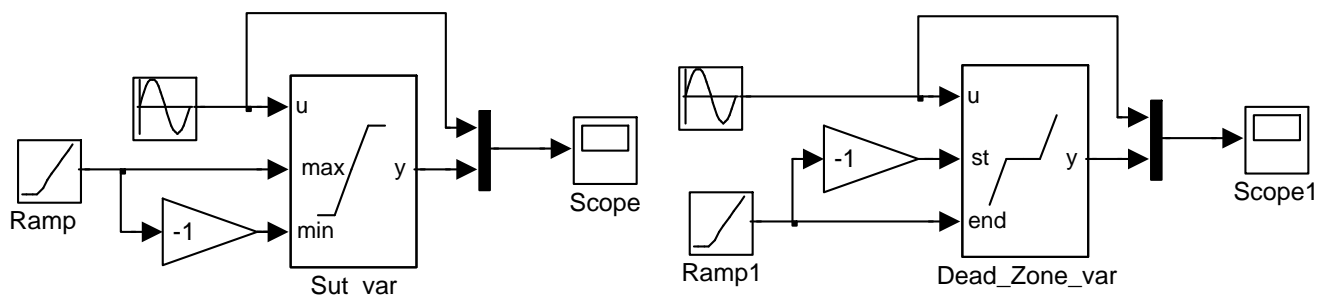


Рис. 7.15. Модель для перевірки адекватності підсистем „Змінне обмеження координат” та „Змінна зона нечутливості”

*Лабораторна робота №8***ІМІТАЦІЙНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ ЕЛЕКТРИЧНИХ  
КІЛ З ВИКОРИСТАННЯМ БЛОКІВ БІБЛІОТЕК *SimPowerSystems***

*Мета роботи:* навчитися моделювати розгалужені електричні кола постійного та змінного струму шляхом імітації фізичного експерименту за допомогою використання блоків бібліотек додатку *SimPowerSystems* пакету *MATLAB*.

**8.1 Основні відомості про бібліотеку *SimPowerSystems*  
та правила її застосування**

Програма *Simulink* поряд з власними блоками, розглянутими у роботах [1-5] може використовувати блоки бібліотеки *SimPowerSystems* (*SPS*), призначеної для імітаційного фізичного моделювання електротехнічних, електромагнітних, електронних та електромеханічних пристроїв та схем [5-8].

В *MATLAB R2013* дві версії (*Second Generation* і *Third Generation*) бібліотеки *SimPowerSystems* можна знайти через *Simulink Library Browser* в додатку імітаційного фізичного моделювання *SimScope*. В цьому посібнику розглянемо тільки другу версію, яка є більш простішою.

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 імітаційного 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*. Починаючи з версії *R2013b*, *Simulink*-моделі блоків *SimPowerSystems* можна побачити зазирнувши під маску.

*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 другої генерації бібліотеки *SimPowerSystems* показано на рис. 8.1.

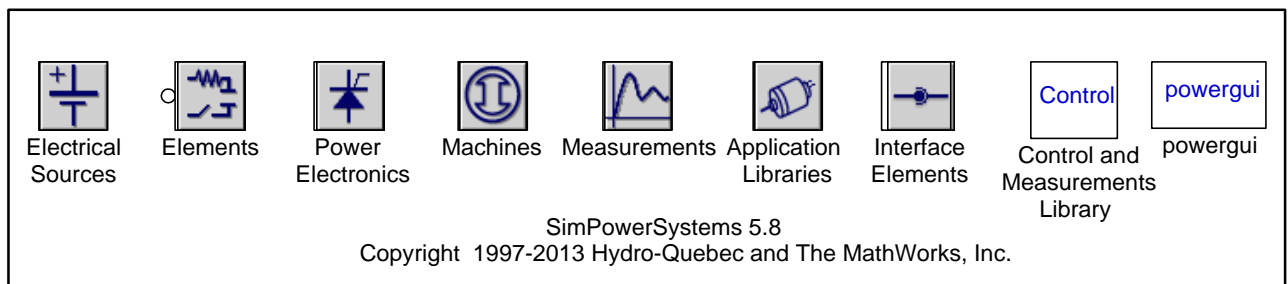


Рис. 8.1 Розділи другої генерації бібліотеки *SimPowerSystems*

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 8.2. Вони знаходяться у різних розділах бібліотеки *SPS*.

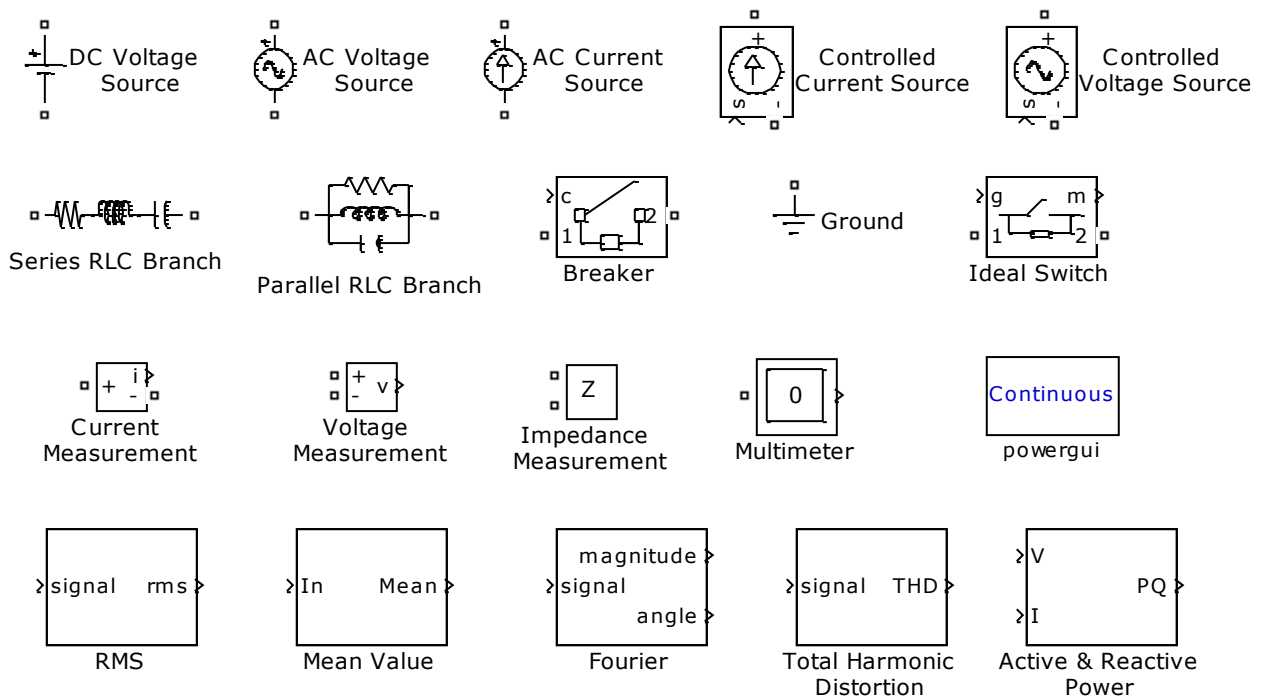


Рис. 8.2. *SPS*-блоки для імітаційного фізичного моделювання простих електричних кіл

Подані блоки мають порти двох типів: *SPS*-порти, позначені дрібними квадратами та *S*-порти, позначені кінцівками стрілок, що допомагає запобігти помилкам 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 імітаційних 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.

## 8.2 Характеристика основних блоків для моделювання кіл постійного струму та однофазних кіл змінного струму

Розглянемо призначення та основні параметри блоків, наведених на рис. 8.2.

### 8.2.1 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  $\varphi^\circ$  (град.),
- *Frequency (Hz)* – частота  $f$  (Гц), що формує напругу

$$U(t) = U_m \sin\left(2\pi ft + \frac{\varphi^\circ}{180^\circ} \pi\right). \quad (8.1)$$

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  $\varphi^\circ$  (град.),
- *Frequency (Hz)* – частота  $f$  (Гц), що формує струм

$$I(t) = I_m \sin\left(2\pi ft + \frac{\varphi^\circ}{180^\circ} \pi\right). \quad (8.2)$$

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 імітаційні 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 імітаційних 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. 8.3.

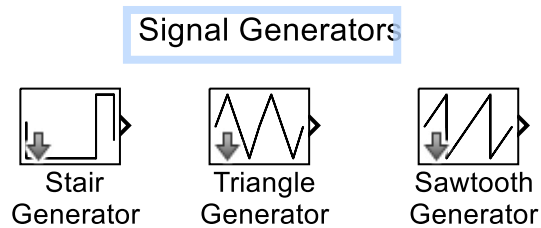


Рис. 8.3. Блоки із групи *Pulse & Signal Generators*

Блок *Stair Generator* (*Ступінчатий генератор*) формує послідовність ступінчатих сигналів на основі блоку *Look-Up Table*, як це показано на рис. 8.4. 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*.

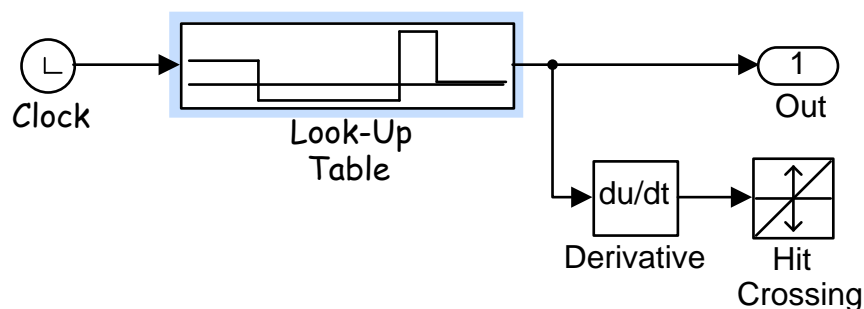


Рис. 8.4. Структура блоку *Stair Generator*



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. 8.5.

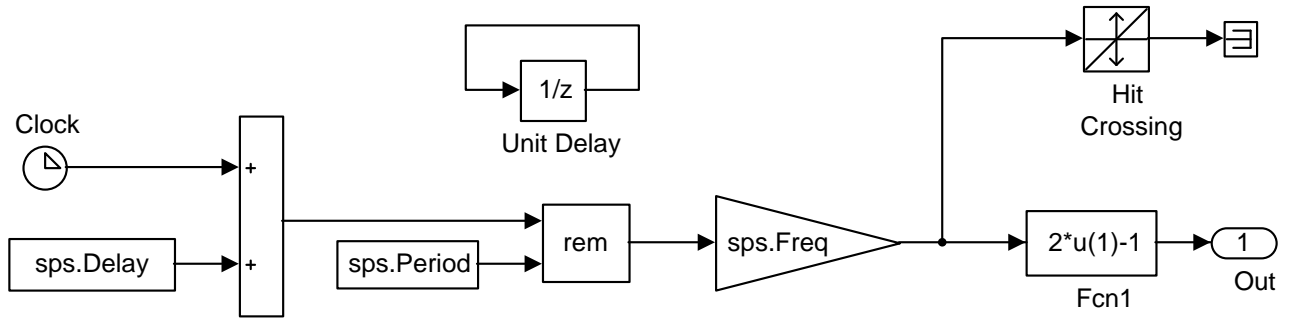


Рис. 8.5. Структура блоку *Sawtooth Generator*

## 8.2.2 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. 8.6.

2) *Parallel RLC Branch* – паралельне *RLC*-з'єднання з такими ж основними параметрами, як і блок *Series RLC Branch*.

У кожному з цих блоків можна установити початковий струм на індуктивному елементі (*Set the initial inductor current*) та початкову напругу на конденсаторі (*Set the initial capacitor voltage*), а також зробити доступними або недоступними для вимірювання блоком *Multimetr* струми та (або) напруги з'єднання (параметр *Measurements*).

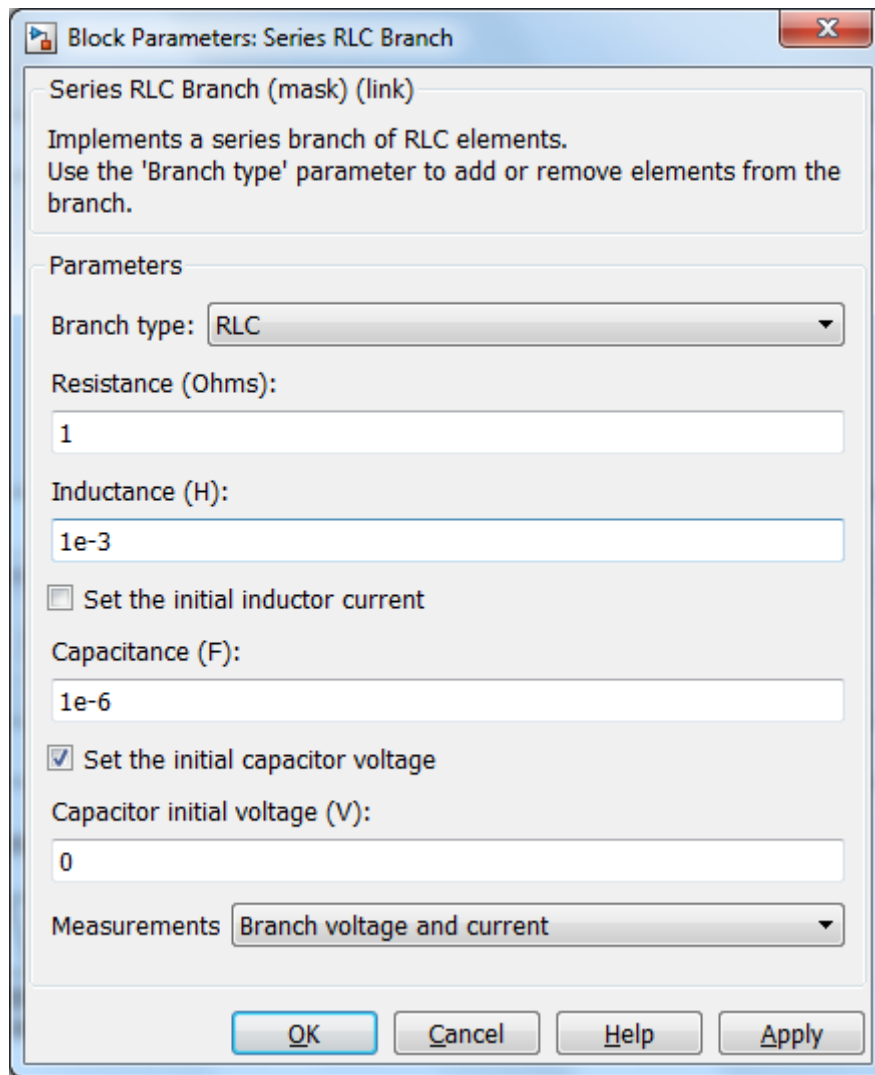


Рис. 8.6. Вікно параметрів блоку *Series RLC Branch*

### 8.2.3 Основні вимірювальні прибори

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.

### 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 8.2.

Якщо такі сигнали існують, то після подвійного щиглика по піктограмі (іконці) блока відчиняється вікно, зображене на рис. 8.7.

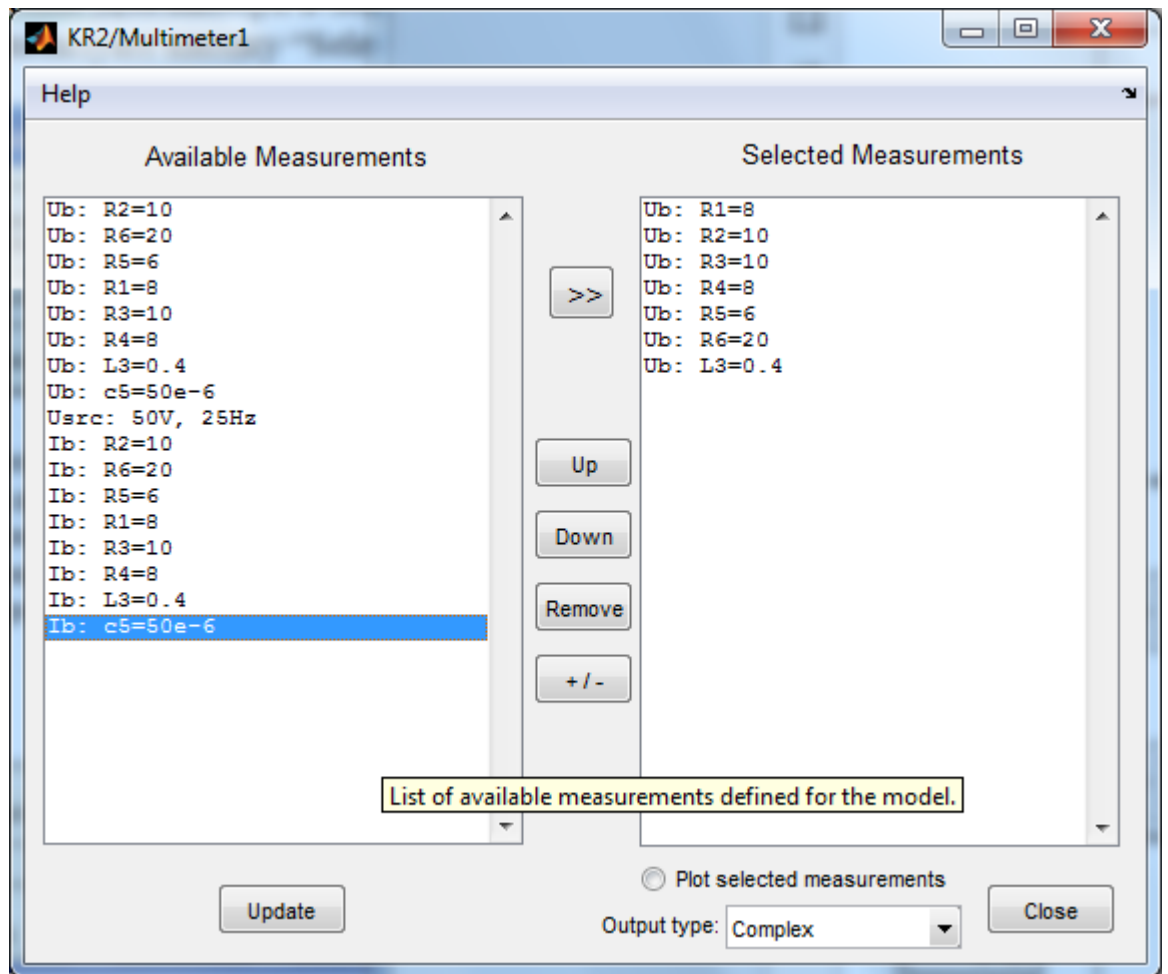


Рис. 8.7. Вікно налаштування блоку *Multimetr*

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 кно-

пками *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 імітаційну 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.

## 8.2.4 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. 8.8).

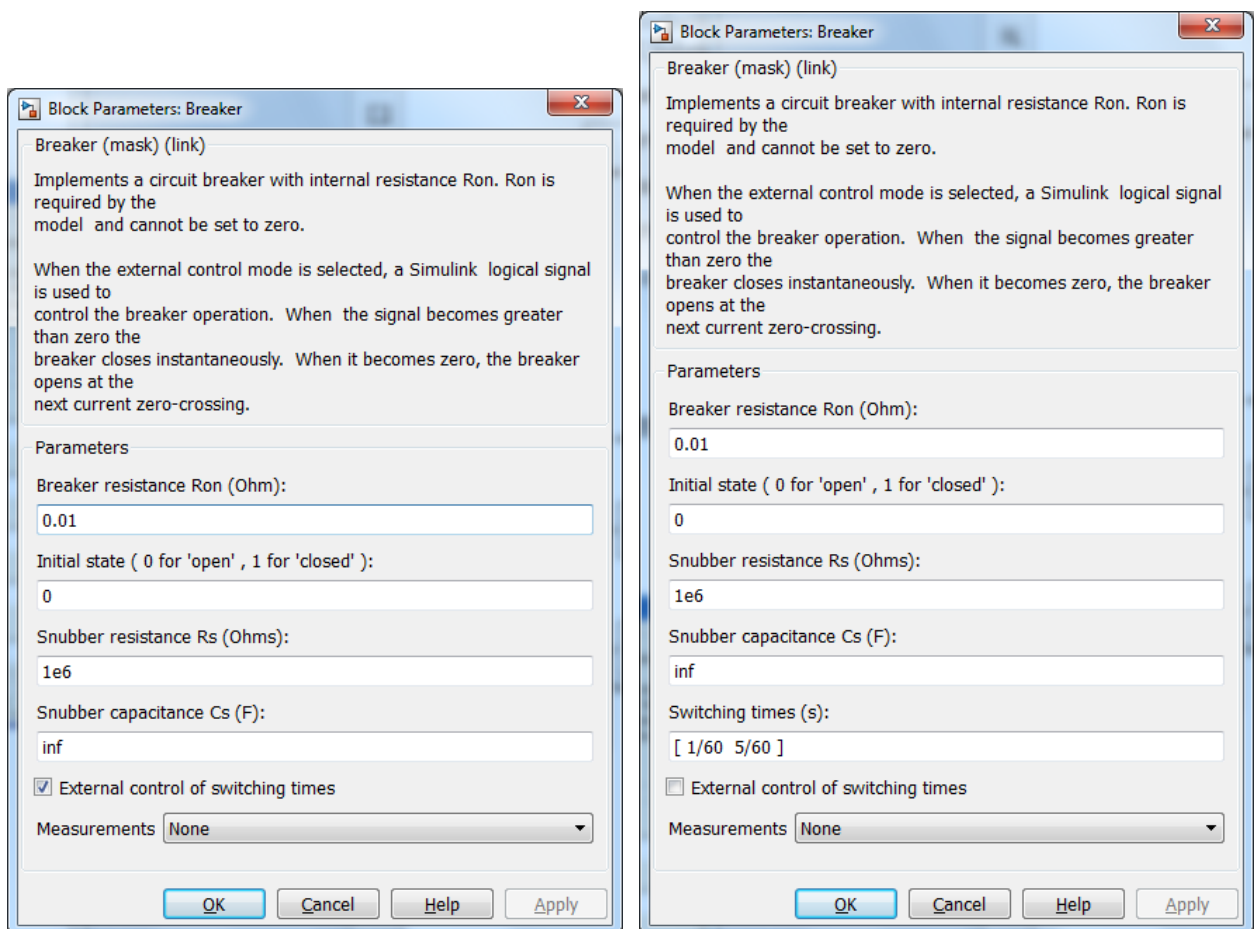


Рис. 8.8. Вікна параметрів блоку *Breaker*

Початковий стан ключа задається параметром *Initial State* (0 for 'open', 1 for 'close'). **Параметр Breaker resistance Ron (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 *Rs-Cs*-гілку (*Snubber*), що здійснює функцію дугогасіння при розмиканні ключа, з параметрами *Snubber resistance Rs (Ohm)* та *Snubber capacitance Cs (F)*. **За замовчанням** параметр *Cs* має значення  $inf(\infty)$ , тобто «снабер» є чисто резистивним. **Встановлення  $Cs = 0$  рівнозначно розмиканню «снабера»**. При цьому змінюється «іконка» блоку (див. рис. 8.9). 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 "c", 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.

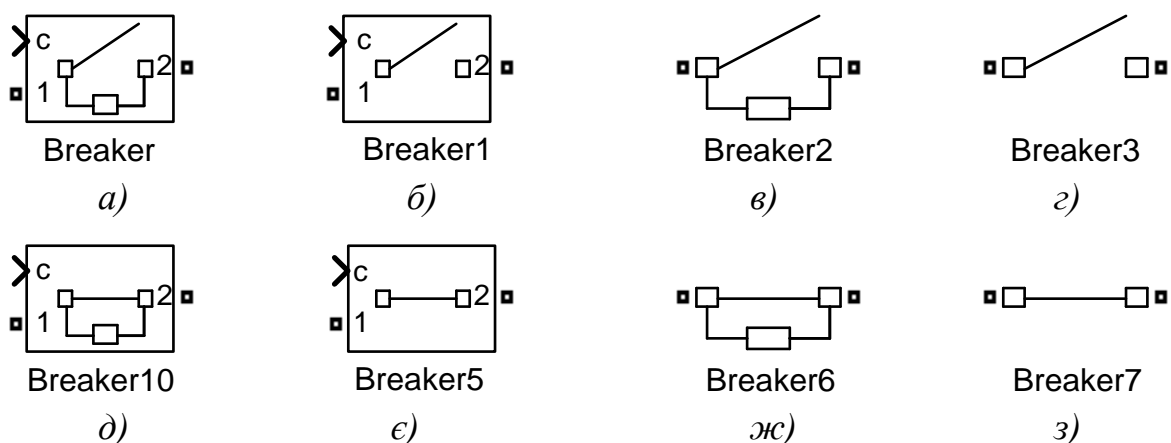


Рис. 8.9. Піктограми блоку *Breaker*:

*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;

$Cs = inf$  – а, в, д, ж;  $Cs = 0$  – 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, має нульовим. Отже, *Breaker* може замкнути, але не може розімкнути коло постійного струму. Його призначення – комутація у колах змінного струму.

При наявності у імітаційній 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. 8.10.

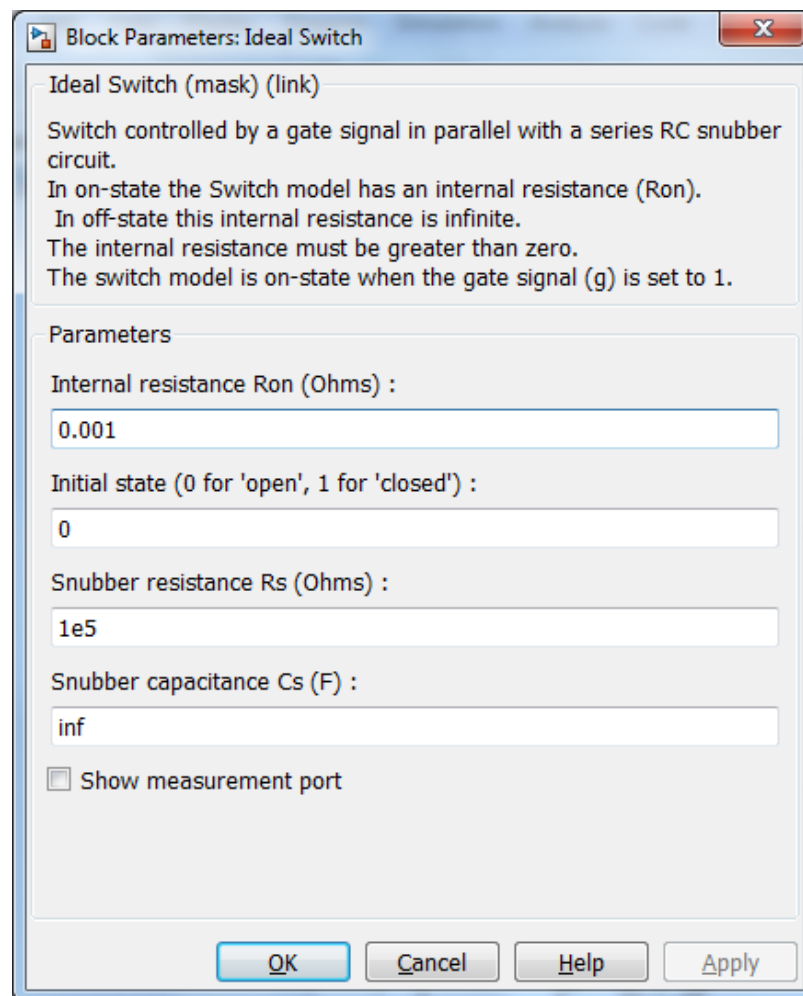


Рис. 8.10. Вікно параметрів блоку *Ideal Switch*

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  $C_s = 0$ , що відображається на вигляді піктограми блок (див. рис. 8.8), 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. 8.8).

### 8.2.5 Графічний інтерфейс користувача *Powergui*

*Powergui (Powerlib Graphical User Interface)* призначений для аналізу імітаційних фізичних моделей електричних кіл і систем.

Блок обов'язково треба розміщати у вікні моделі, у якій присутній хоча б один з блоків бібліотек *SimPowerSystems* (у версіях MATLAB 7 і вище поміщається у вікно моделі автоматично). Блок повинен бути тільки один, і його не можна перейменовувати.

Він дозволяє вирішувати такі задачі:

1) ***Simulation type*** – вибір методу розрахунку динамічних та статичних режимі, зазвичай у вигляді вікна вибору однієї з опцій: *Continuous* (неперервний), *Discretize electrical model* (дискретні, треба вказати період дискретності *Sample Time*), *Phasor simulation* (метод узагальнених векторів, треба вказати основну частоту).

2) **Steady-State Voltages and Currents** – відображення усталених значень перемінних (вікно *steady state values*).

Вікно визначення параметрів цієї операції зображено на рис. 8.11.

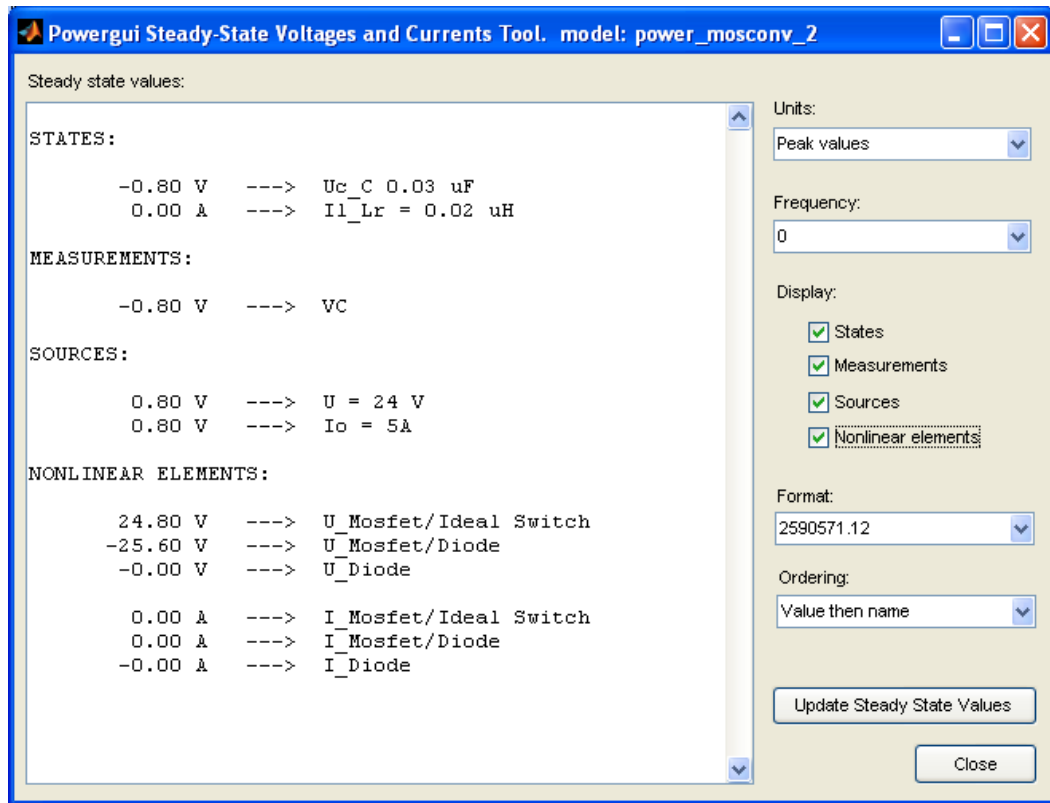


Рис. 8.11. Вікно відображення усталених значень

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, дозволяє виконати ініціалізацію моделі, визначаючи початок моделювання з усталеного режиму або з нульових початкових умов (рис. 8.12).



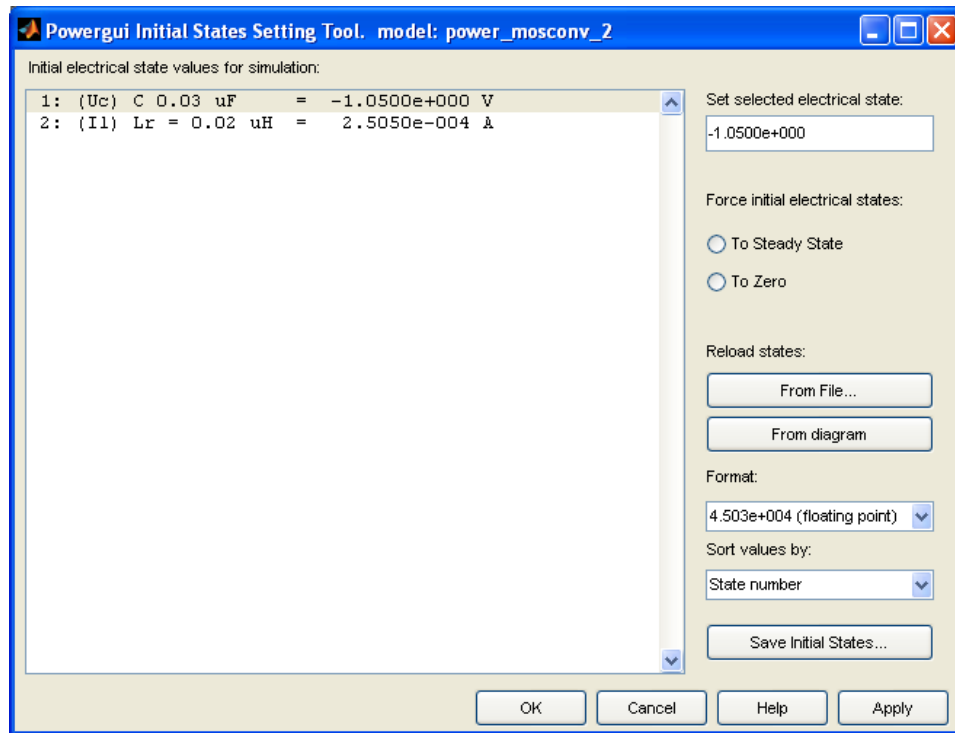


Рис. 8.12. Вікно визначення початкових умов

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. 8.13.

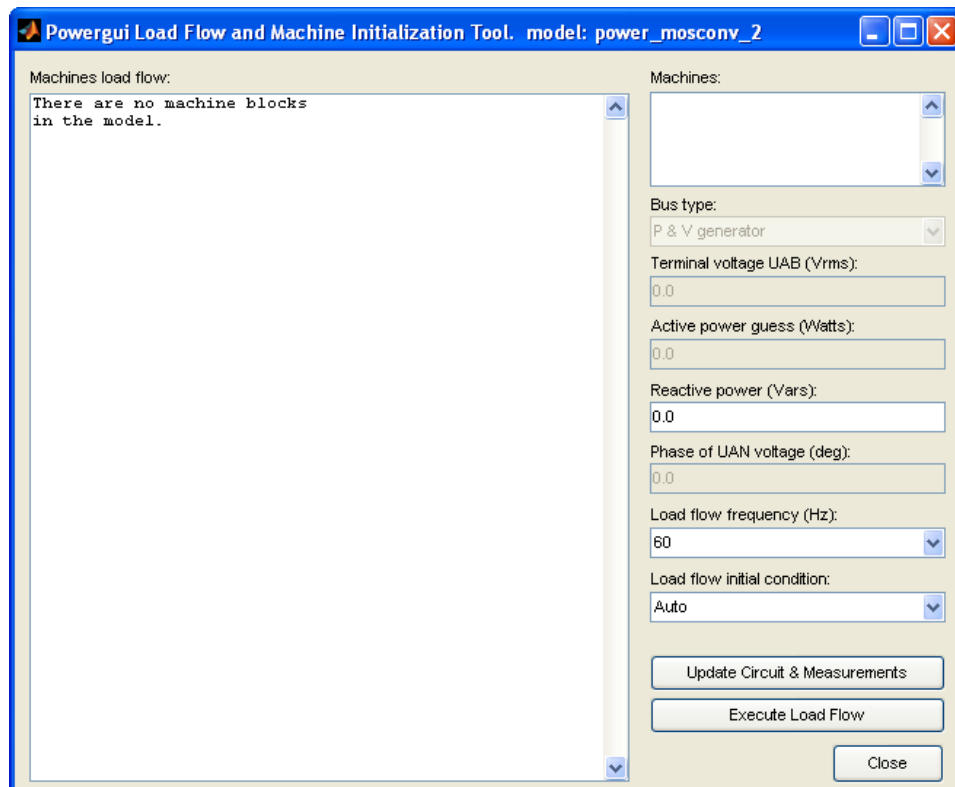


Рис. 8.13. Вікно ініціалізації динамічних трифазних електричних кіл та електричних машин

Більш детально ознайомитися з можливостями цієї операції можна у [7]. Інші функції інтерфейсу користувача будуть розглянуті пізніше у інших дисциплінах або при необхідності самостійно.

### 8.3 Приклад імітаційного фізичного моделювання лінійного розгалуженого електричного кола

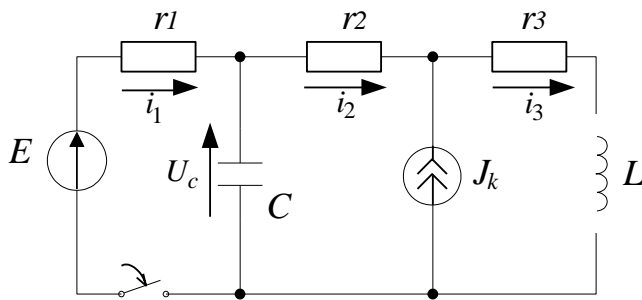


Рис. 8.14. Досліджуване електричне коло

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.

Перевагою імітаційного 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 імітаційної бібліотеки *SimPowerSystems*, що потребує деякого досвіду. *SPS*-модель електричного кола, зображеного на рис. 8.14, подана на рис. 8.15.

*Simulink*-блоки у моделі рис. 8.15 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. 8.16, 8.17.

Як приклад, розглянемо розгалужену схему, зображену на рис. 8.14, для якої треба розрахувати перехідні процеси, тобто знайти  $i_1(t)$ ,  $i_2(t)$ ,  $i_3(t)$ ,  $U_C(t)$

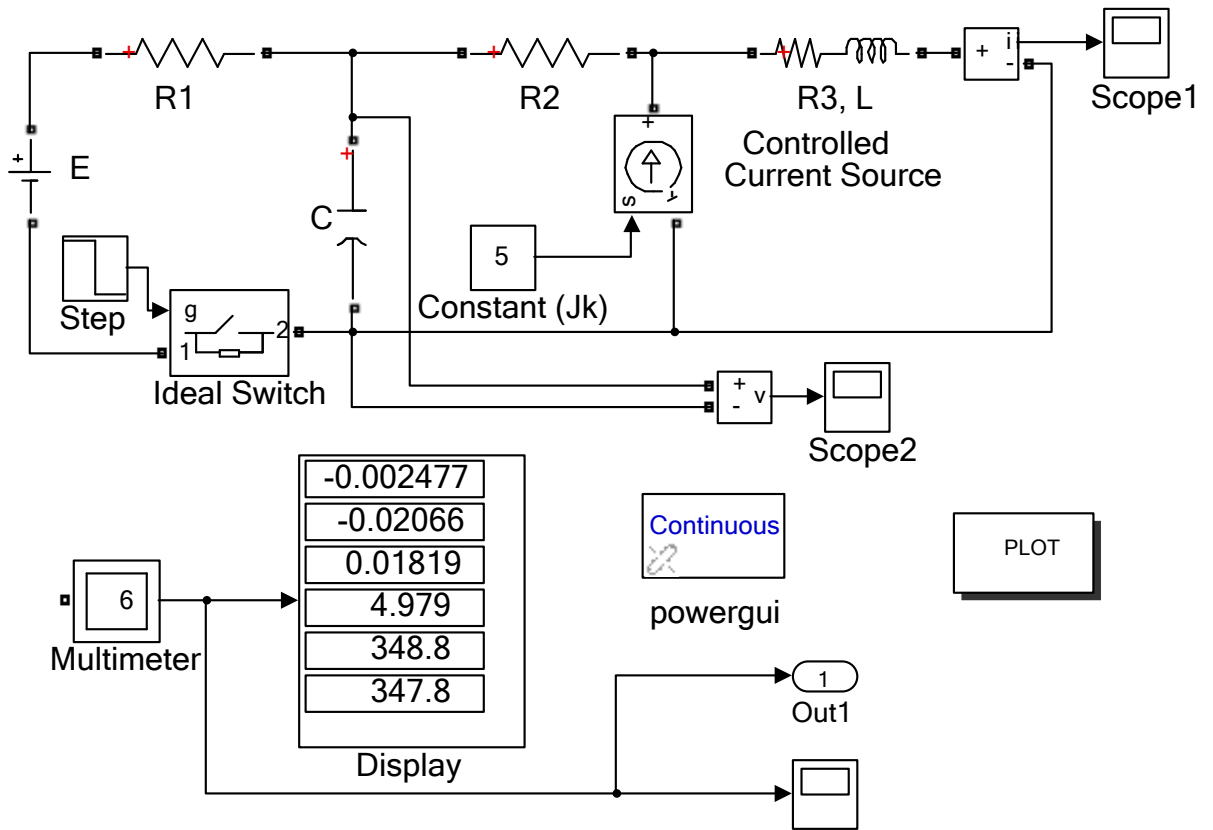


Рис. 8.15. SPS-модель електричного кола, зображеного на рис. 8.16

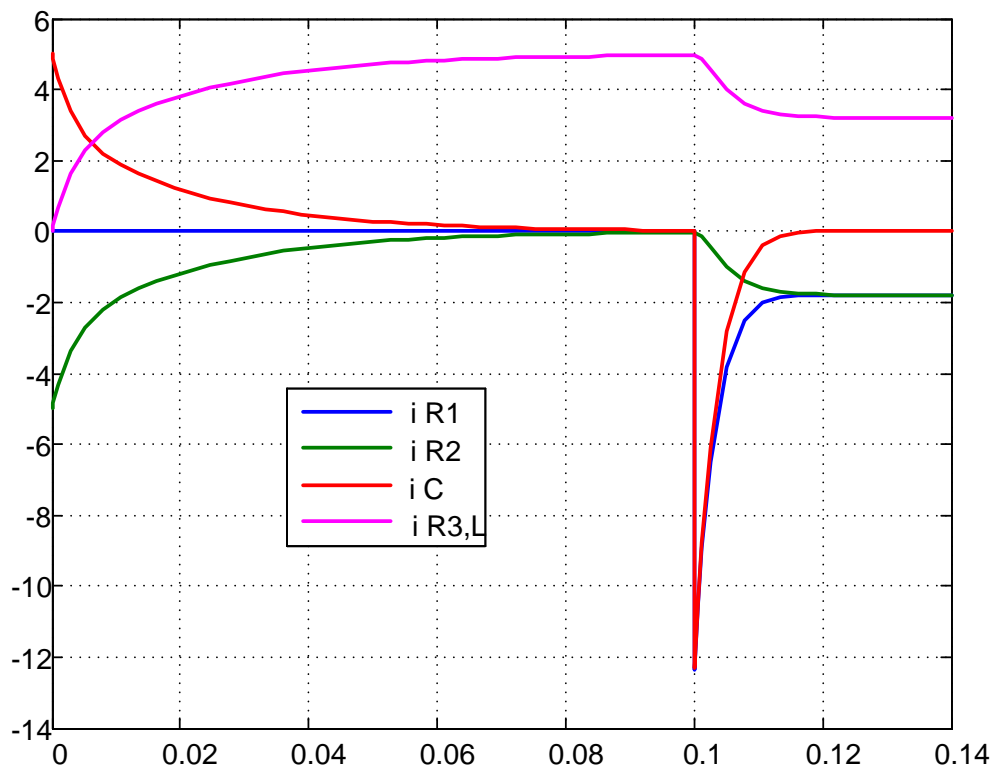


Рис. 8.16. Перехідні процеси струмів моделі рис. 8.15

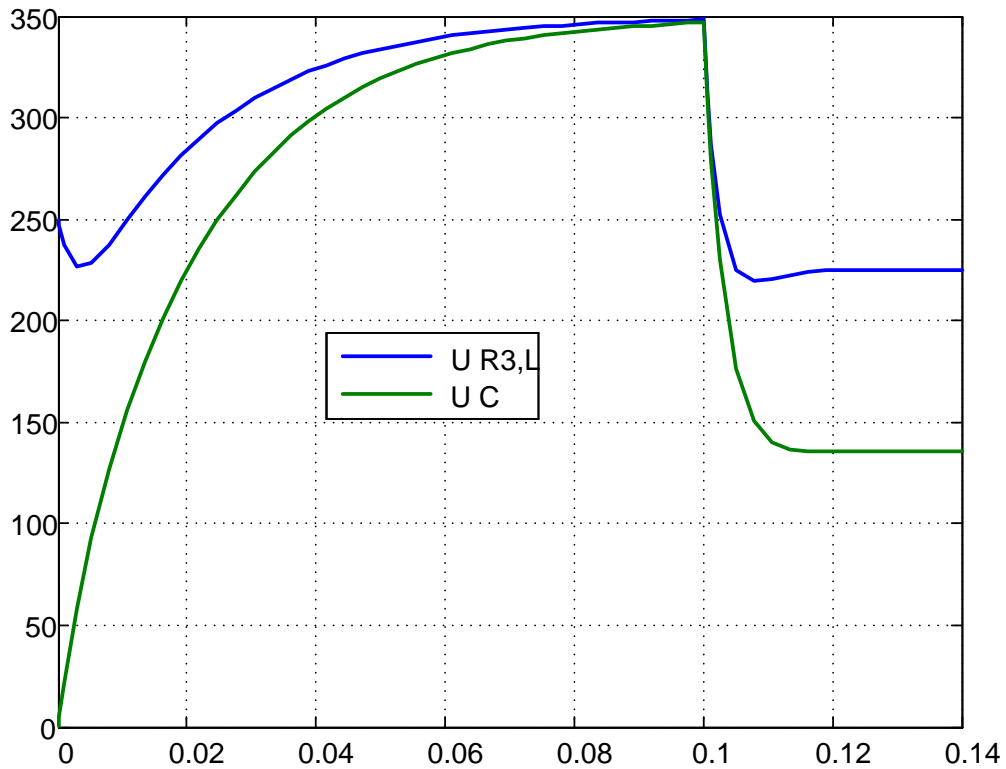


Рис. 8.17. Перехідні процеси напруг моделі рис. 8.15

#### 8.4 Завдання

Отримати графіки перехідних процесів напруги на конденсаторі та струмів у гілках однієї з електричних схем, зображених на рис. 8.18, методом імітаційного фізичного 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. 8.2),  $T=0.1$  с,  $T1=0.07$  с.

За результатами розрахунків побудувати графіки перехідних процесів.

Параметри схем надані у табл. 8.1.

#### 8.5 Методичні рекомендації

1. Для перемикачів у колах постійного струму використовуйте блок *Ideal Switch*, а в колах змінного струму – блок *Breaker*.
2. Для вимірювання струмів та напруг апробуйте блоки *Current Measurement*, *Voltage Measurement* та *Multimeter*.

3. Виведіть в одно графічне вікно всі виміряні напруги, а в інше – всі виміряні струми.

4. Не забудьте, що перехідні процеси в електричних колах протікають дуже швидко і складають соті долі секунди. За результатами симуляції підбирайте час перемикання ключових елементів та час моделювання так, щоб вони трошки перевищували час перехідних процесів.

5. Зовнішнє перемикання ключових елементів здійснюйте блоками *Step* або *Stair Generator*.

6. Не забудьте встановити у блоці *PowerGUI* початкові умови у стан *To Zero*.

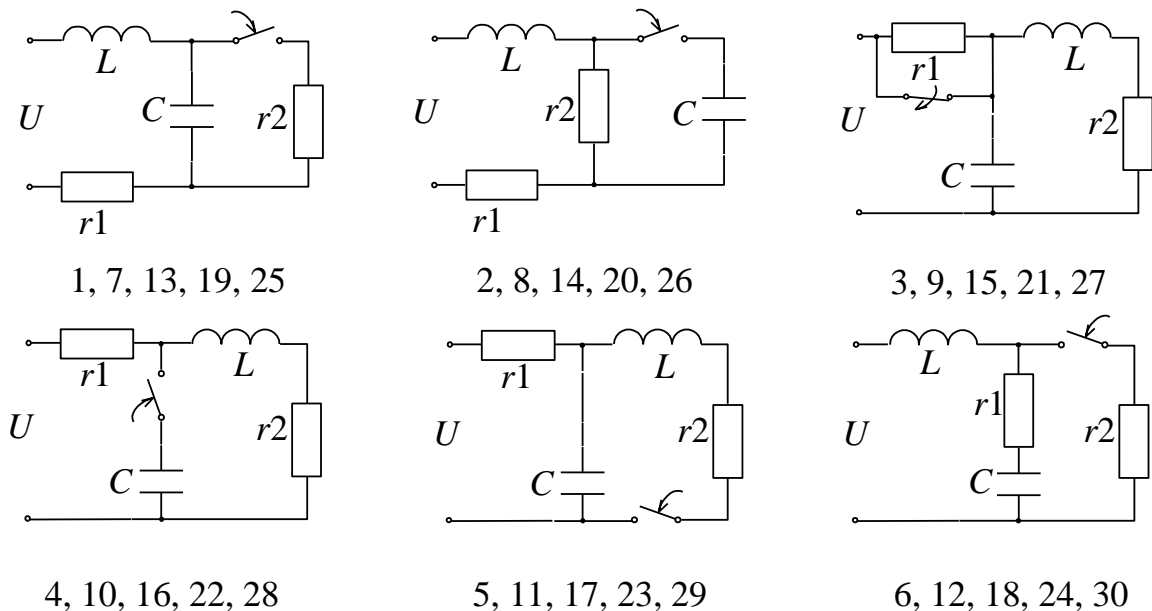


Рис. 8.18. Варіанти розгалужених лінійних електричних кіл

Таблиця 5.1

№ вар.	$U_m$ , В	$r_1$ , Ом	$r_2$ , Ом	$\omega$ , $c^{-1}$	$L$ , 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

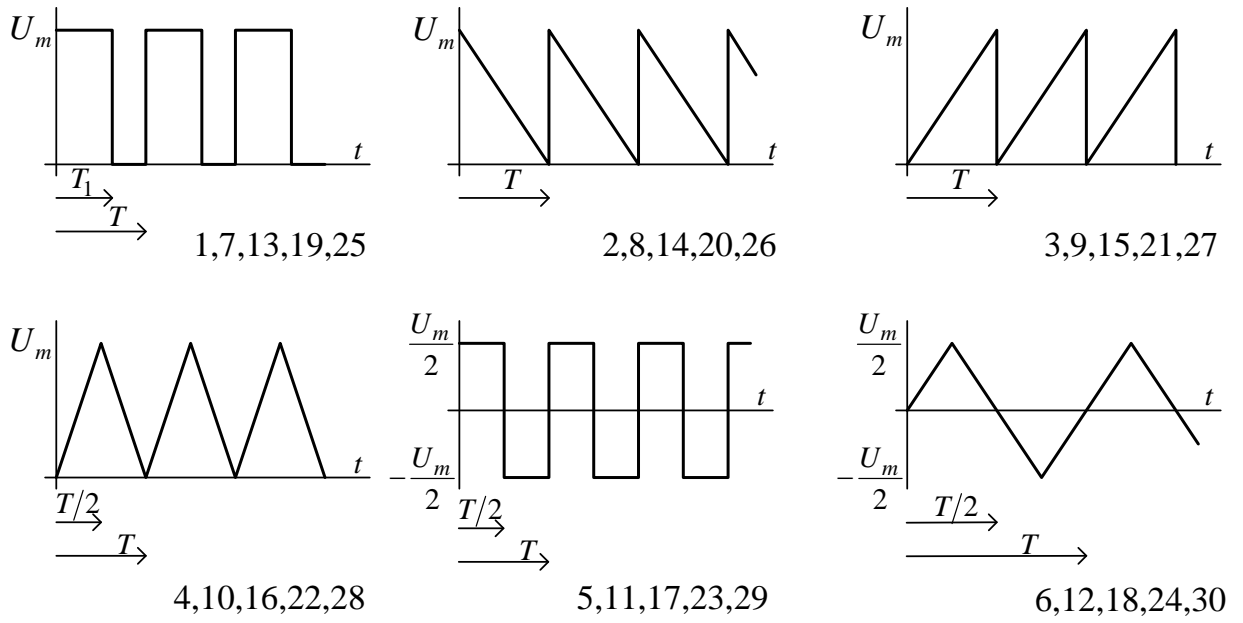


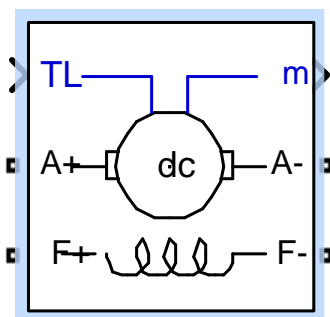
Рис. 8.19. Варіанти вхідних періодичних сигналів

## Лабораторна робота №9

### ІМІТАЦІЙНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ ДВИГУНА ПОСТІЙНОГО СТРУМУ З НЕЗАЛЕЖНИМ ЗБУДЖЕННЯМ

*Мета роботи:* навчитися моделювати двигун постійного струму шляхом імітації фізичного експерименту за допомогою використання блоків бібліотек додатку *SimPowerSystems* пакету *MATLAB*.

#### 9.1 SPS- модель двигуна постійного струму



DC Machine

Рис. 9.1. SPS-блок  
DC Machine

На рис. 9.1 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 [6,9].

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. 9.2 та 9.3 зображені вкладки вікна введення параметрів цього блоку.

Вкладка *Parameters* (рис. 9.2) 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.m<sup>2</sup>) – момент інерції двигуна,  $J$  (кг · м<sup>2</sup>);
- *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.

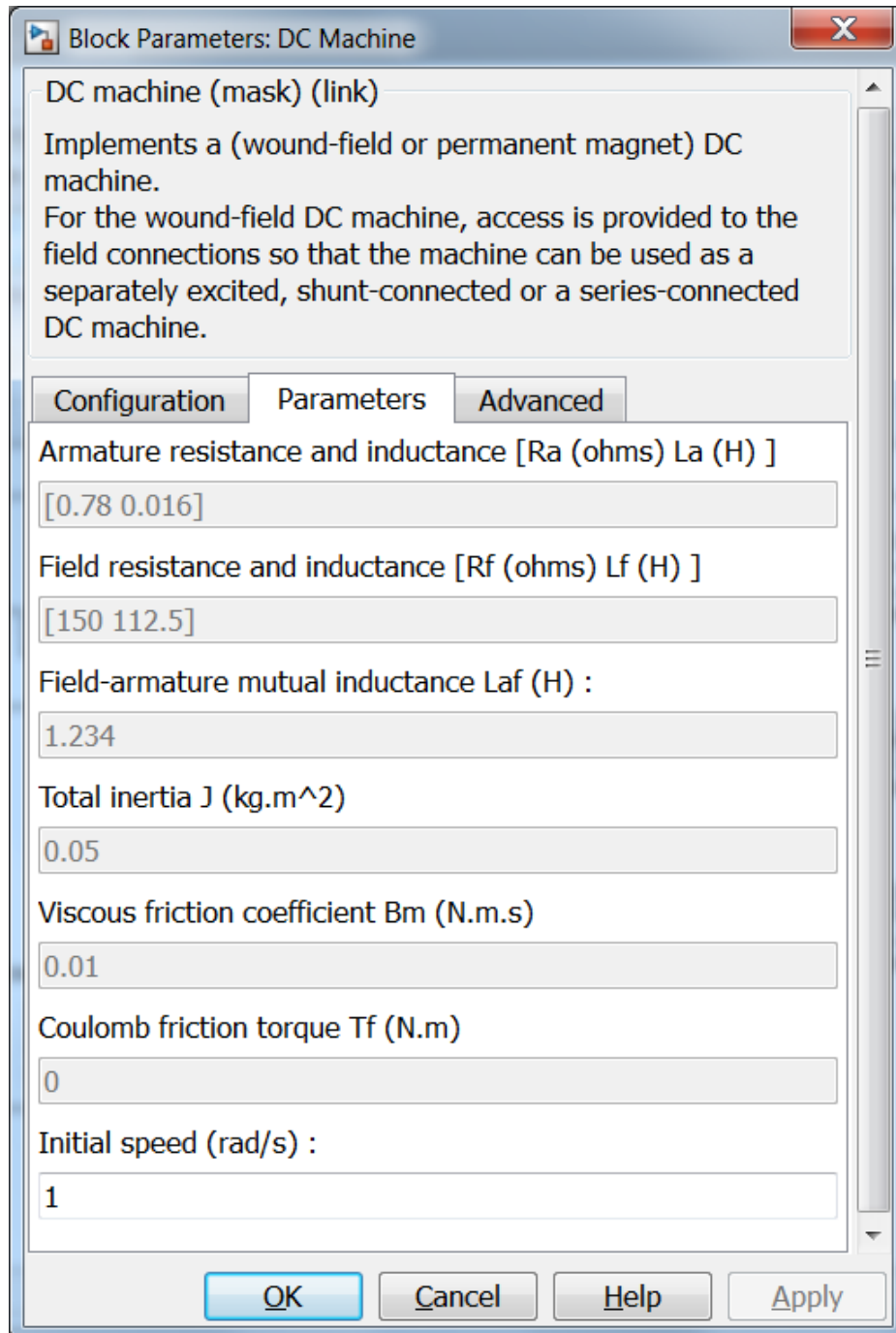


Рис. 9.2. Вкладка *Parameters* вікна встановлення параметрів блоку *DC Machine*



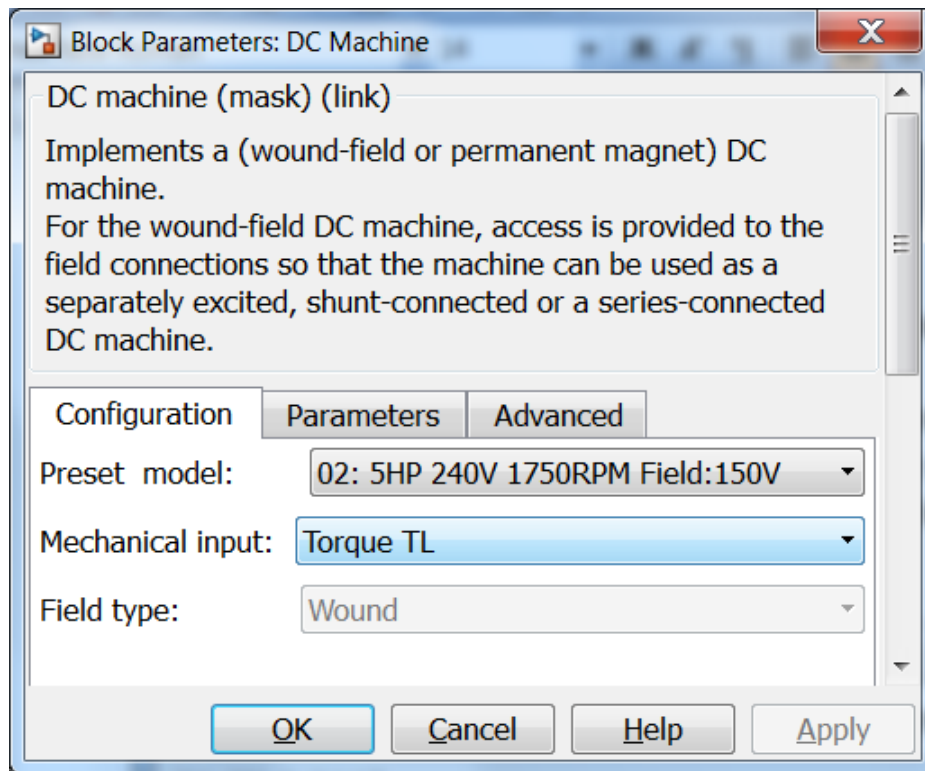


Рис. 9.3. Вкладка *Configuration* вікна встановлення параметрів блоку *DC Machine*

Параметри ДПС та можливість їх встановлення або коригування залежать від стану опції *Preset Model* (попередня ініціалізація моделі даними деякого двигуна) вкладки *Configuration* (рис. 9.3).

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*, при-

зводить 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

$$c = L_{af} i_{fn} = L_{af} \frac{U_{fn}}{R_f} = L_{af} \frac{U_{3H}}{R_3}. \quad (9.1)$$

Модель для дослідження ДПС з незалежним збудженням, у якій використано розглянути вище блок *DC Machine* показано на рис. 9.4.

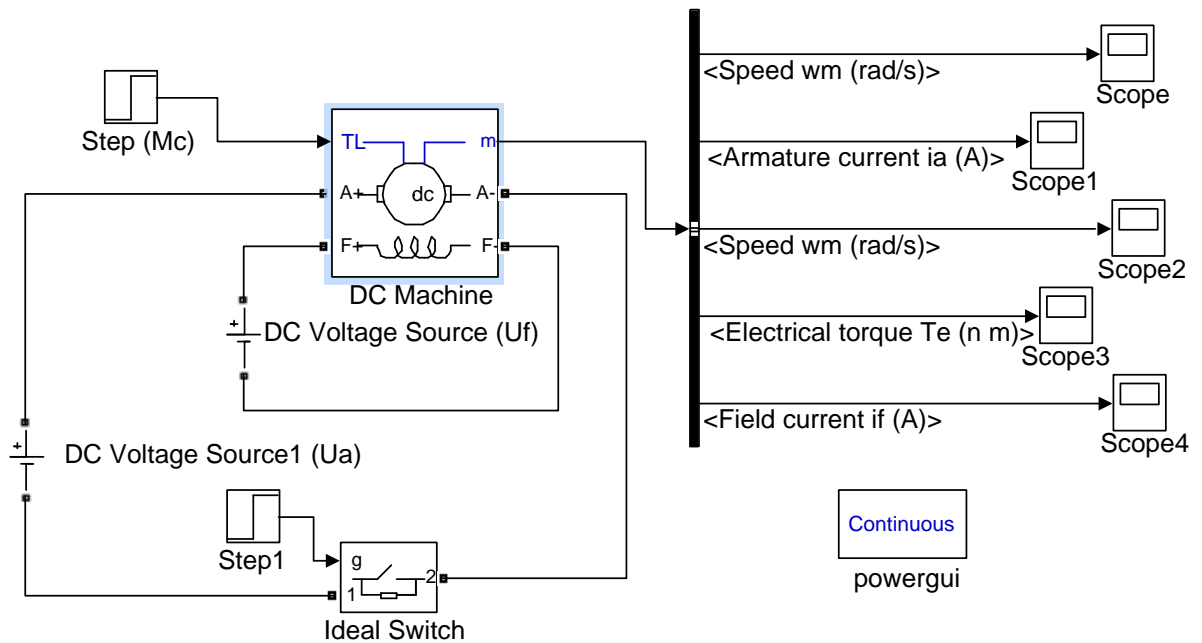


Рис. 9.4 Віртуальна фізична модель двигуна постійного струму з незалежним збудженням

## 9.2 Завдання

Виконати завдання 1, 2 з лабораторної роботи №5 методом віртуального фізичного моделювання. Порівняйте отримані результати.

## 9.3 Методичні рекомендації

1. Переключення режимів здійснюйте блоком *Breaker* або *Ideal Switch*. Уважно слідкуйте за послідовністю розмикання та замикання ключових елементів. Спочатку слід подавати напругу на обмотку збудження, а згодом – на обмотку якоря.

2. При налаштуванні часових параметрів користуйтеся результатами виконання лабораторних робіт №5 і №6.

3. Для формування напруги якоря за лінійним законом скористайтеся послідовно з'єднаними *Simulink*-блоками *Lookup Table* і *Clock* та *SPS*-блоком *Controlled Voltage Source*.

4. Для контролю усталених значень сигналів у абсолютних одиницях використовуйте блоки *Display*, а перехідні процеси виводьте у відносних одиницях.

5. Ініціалізацію моделей та побудову графіків виконайте за допомогою «кнопок», створених з блоків *Subsystem* (див. лабораторну роботу №7).

**Лабораторна робота №10****ІМІТАЦІЙНЕ ФІЗИЧНЕ МОДЕЛЮВАННЯ АСИНХРОННОГО  
ДВИГУНА**

*Мета роботи:* навчитися моделювати асинхронний двигун (АД) шляхом імітації фізичного експерименту за допомогою використання блоків бібліотек додатку *SimPowerSystems* пакету *MATLAB*. Дослідити та порівняти перехідні процеси в АД в режимах прямого пуску і реверсу, плавного пуску та частотного керування. Проаналізувати статичні механічні та швидкісні характеристики АД.

**10.1 Знайомство з блоками асинхронних машин бібліотеки *Machines* другої генерації бібліотеки *SimPowerSystems***

Імітаційні фізичні моделі електричних машин знаходяться в бібліотеці *Machines* додатку *SimPowerSystems*. До складу її входять машини постійного струму, реактивний та кроковий двигуни, асинхронні, синхронні та машини. Електричні машини можуть працювати як генератори і як двигуни. В них передбачені різні варіанти включення обмоток, різні способи збудження синхронних машин, можливість створення систем «електричний вал» та «механічний вал». В багатьох машинах параметри можуть задаватися як в абсолютних одиницях системи СІ (*SI Units*), так і у відносних одиницях (*pu Units*). В даній лабораторній роботі розглянемо блоки асинхронних машин та способи їх використання для дослідження роботи асинхронного двигуна (АД) у різних режимах [6, 9, 10].

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, с, 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. 10.1) визначається типом ротора та типом механічного входу, які встановлюються за допомогою меню параметрів *Rotor type* і *Mechanical input* вкладки *Configuration* (рис. 10.2a).

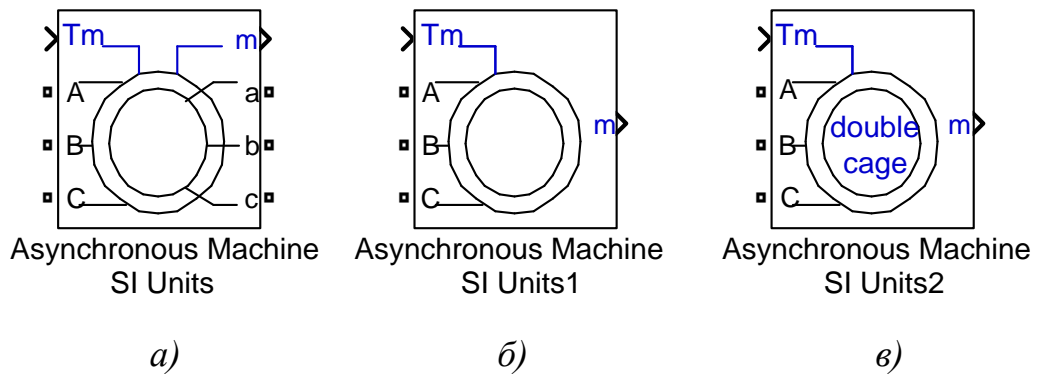


Рис. 10.1. Блоки *Asynchronous Machine* з різними типами ротора (*Rotor Type*):

*a* – з фазним ротором (*Wound*),

*б* – з білячою кліткою на роторі (*Squirrel-cage*),

*в* – з двома білячими клітками на роторі (*Double squirrel-cage*)

Вкладка *Parameters* (рис. 10.6б) діалогового вікна установки параметрів блока 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 ( $l$  – leakage –

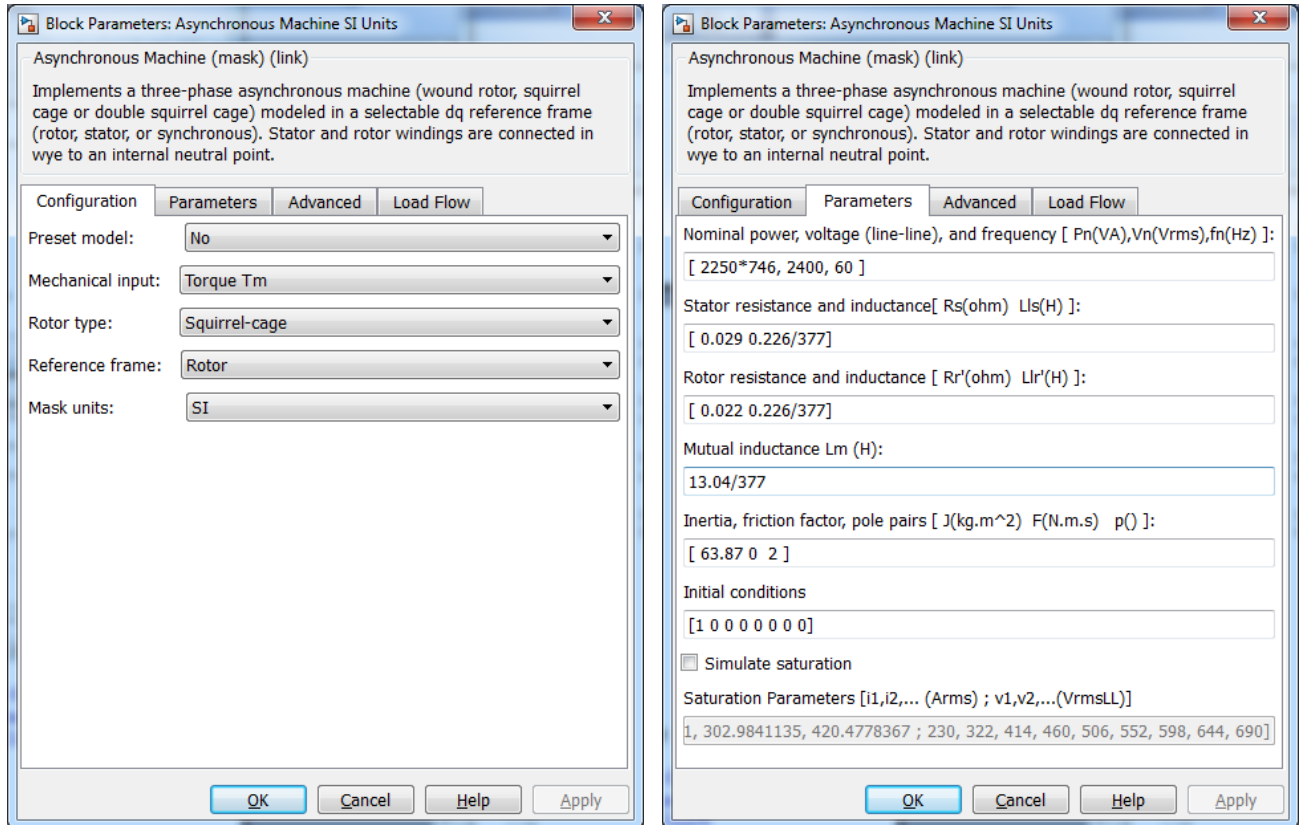
- розсіювання),  $L_{\sigma}$  (Гн);
- *Rotor resistance*  $R_r'$  (Ohm) – активний опір фази ротора, приведений до статора  $R_r$  (Ом);
  - *Rotor inductance*  $L_r'$  – індуктивність розсіювання фази ротора, приведена до статора  $L_{r\sigma}$  (Гн);
  - *Mutual inductance*  $L_m$  (H) – головна взаємна індуктивність,  $L_m$  (Гн);
  - *Inertia*  $J$  (kg.m<sup>2</sup>) – момент інерції ротора,  $J$  (кг · м<sup>2</sup>);
  - *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. 10.2a).

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 **Vrms**, 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.



а)

б)

Рис. 10.2 – Вкладки діалогового вікна встановлення параметрів блоку *Asynchronous Machine SI Units*:  
а – *Configuration*; б – *Parameters*

Слід зазначити, що до параметрів попередньо обраних двигунів не входять 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 пара-

метрів 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. 10.6а) зазвичай можна не змінювати.

## 10.2 Моделювання асинхронного двигуна в режимі прямого пуску

На рис. 10.3 зображена *SPS*-модель, призначена для дослідження прямого пуску АД [19].

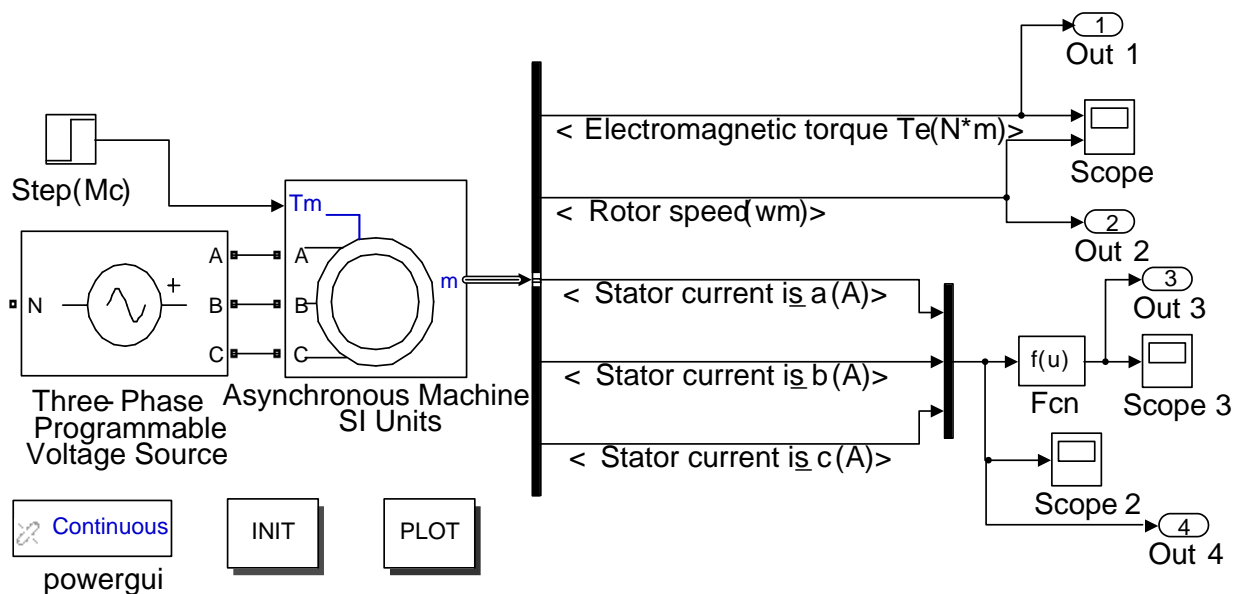


Рис. 10.3. *SPS*-модель прямого пуску асинхронного двигуна

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. 10.4.

Обчислення діючого (ефективного) значення струму статора виконує блок *Fcn* за формулою:

$$I_{se} = \sqrt{(I_A^2 + I_B^2 + I_C^2)/3}. \quad (10.1)$$

Фазні напруги формуються джерелом *Tree-Phase Programmable Voltage Source*, вікно параметрів якого відображено на рис. 10.5.



Замість одного джерела *Tree-Phase Programmable Voltage Source*, можна використати три блоки *AC Voltage Source*, з'єднані зіркою.

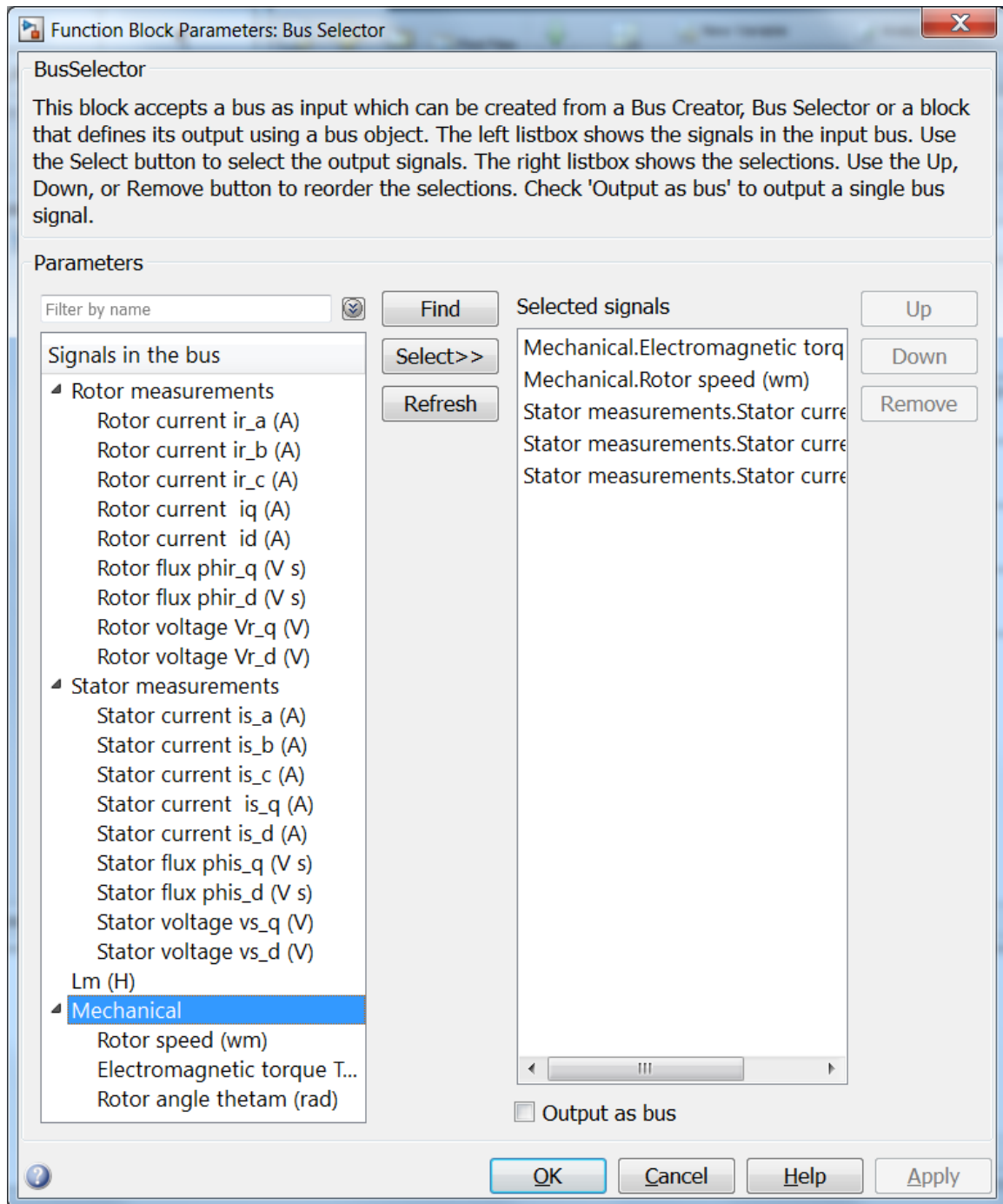


Рис. 10.4. Діалогове вікно блоку *Bus Selector*, підключеного до вихідного інформаційного порту двигуна

При визначенні параметрів джерел необхідно звернути увагу на те, що в параметрах двигуна задано діюче значення лінійної напруги [Voltage (line-line) ( $V_{rms}$ )], а блоки *AC Voltage Source* потребують амплітудних значень фа-

зних 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)].

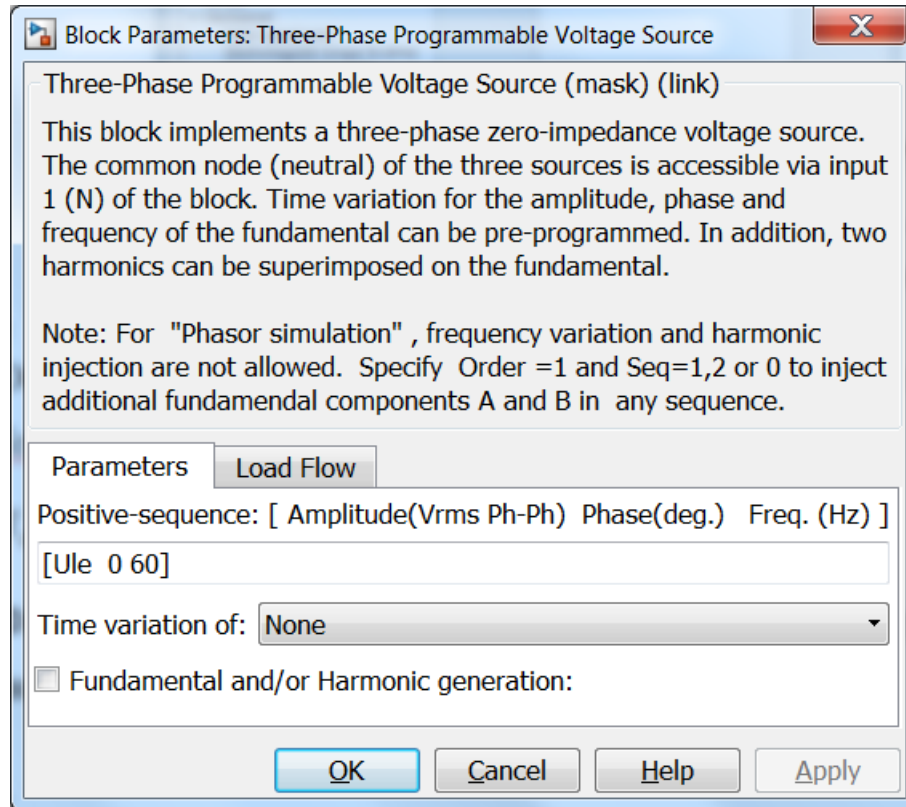


Рис. 10.5. Діалогове вікно блоку *Tree-Phase Programmable Voltage Source*

Перехідні процеси, отримані за результатами симуляції моделі рис. 10.7 подані на рис. 10.6. 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*.

Недоліком прямого пуску є значні стрибки і коливання електромагнітного моменту та струмів АД.

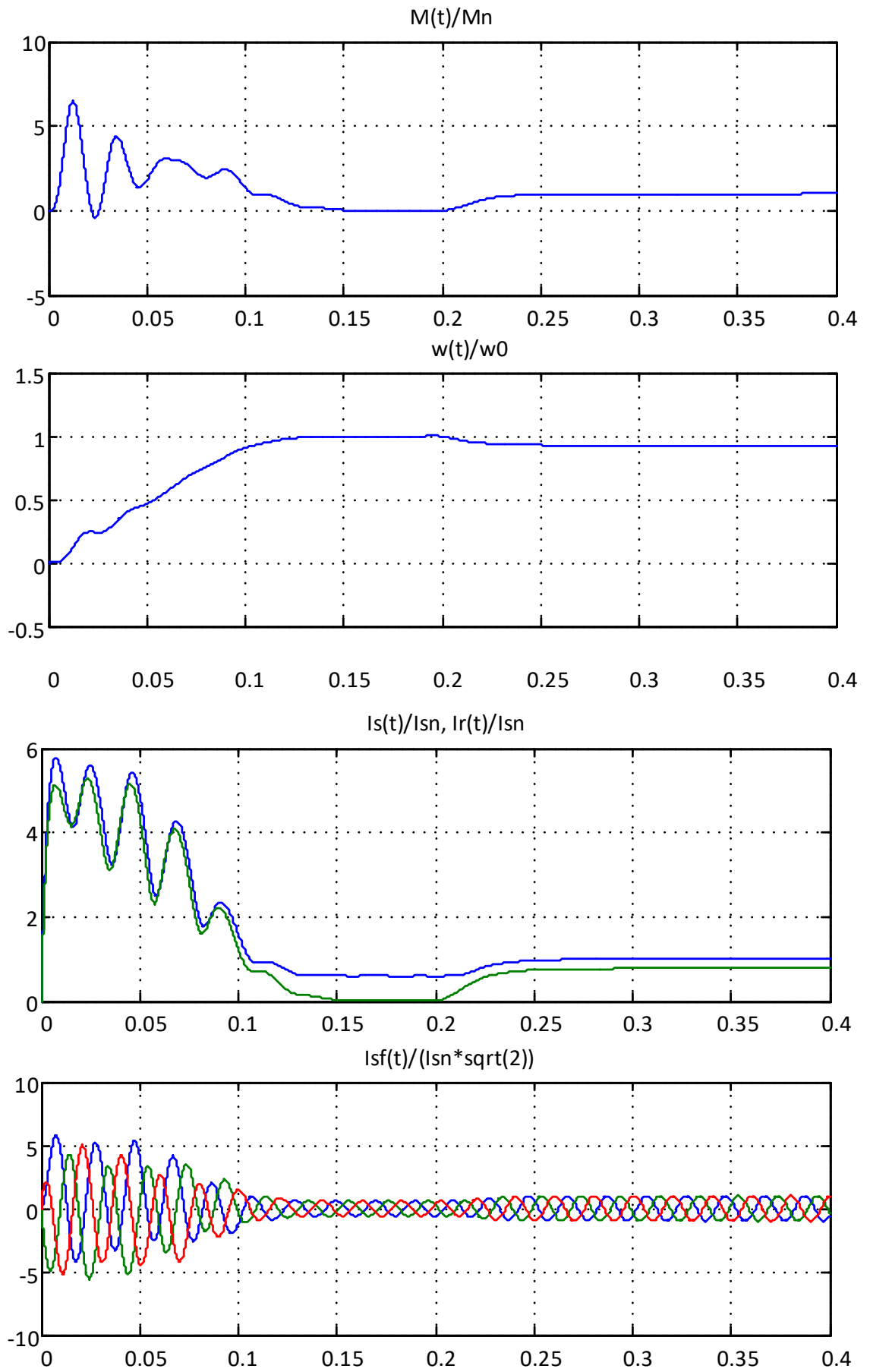


Рис. 10.6. Перехідні процеси при прямому пуску асинхронного двигуна

### 10.3 Моделювання асинхронного двигуна в режимі плавного пуску

Плавний пуску (Soft Start) полягає у плавній зміні амплітуди напруги статора з постійною частотою. Найпростішим варіантом поступової зміни амплітуди є зміна її до номінального значення за лінійним законом. На рис. 10.7 зображена модель, призначена для формування такої напруги за допомогою *Simulink*-блоків за рівняннями

$$\begin{cases} U_{sA}(t) = U_{sfm} \sin(\varphi(t)), \\ U_{sB}(t) = U_{sfm} \sin(\varphi(t) - 2\pi/3), \\ U_{sC}(t) = U_{sfm} \sin(\varphi(t) - 4\pi/3), \end{cases}$$

де

$$\varphi(t) = \int_0^t \omega_s(t) dt = 2\pi \int_0^t f_s(t) dt$$

та подальшого перетворення отриманих фазних напруг в *SPS*-сигнали за допомогою керованих джерел напруги (*Controller Voltage Source*).

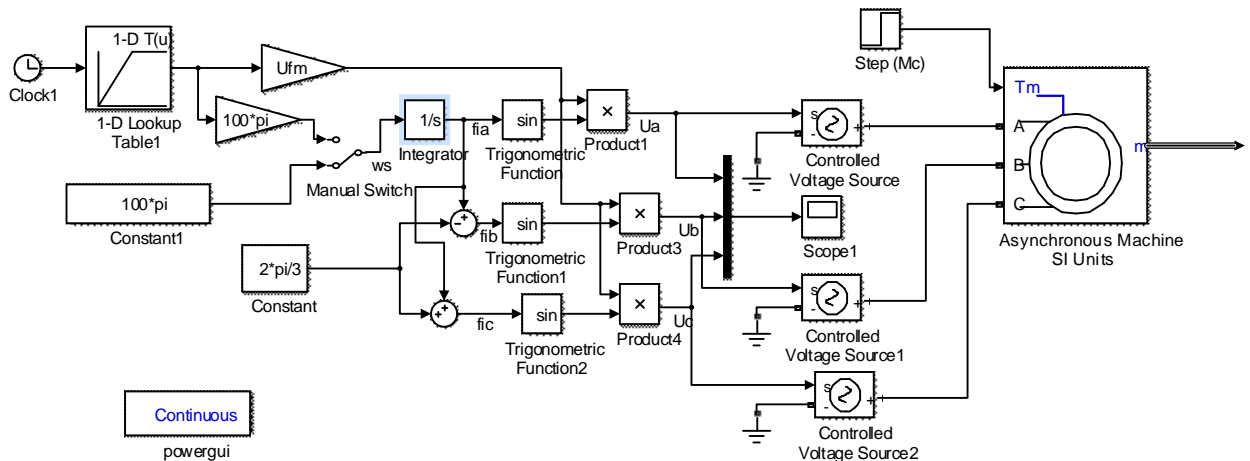


Рис. 10.7. Модель формування напруг статора для плавного пуску

Звісно, що при  $f_s = 50 \text{ Гц} = \text{const}$ ,  $\omega_s = 2\pi f_s = 100\pi = \text{const}$  фазовий кут  $\varphi(t)$  змінюється за лінійним законом  $\varphi(t) = \omega_s t$ . Перехідні процеси в АД при плавному пуску показані на рис. 10.8, а напруги статора – на рис. 10.9.

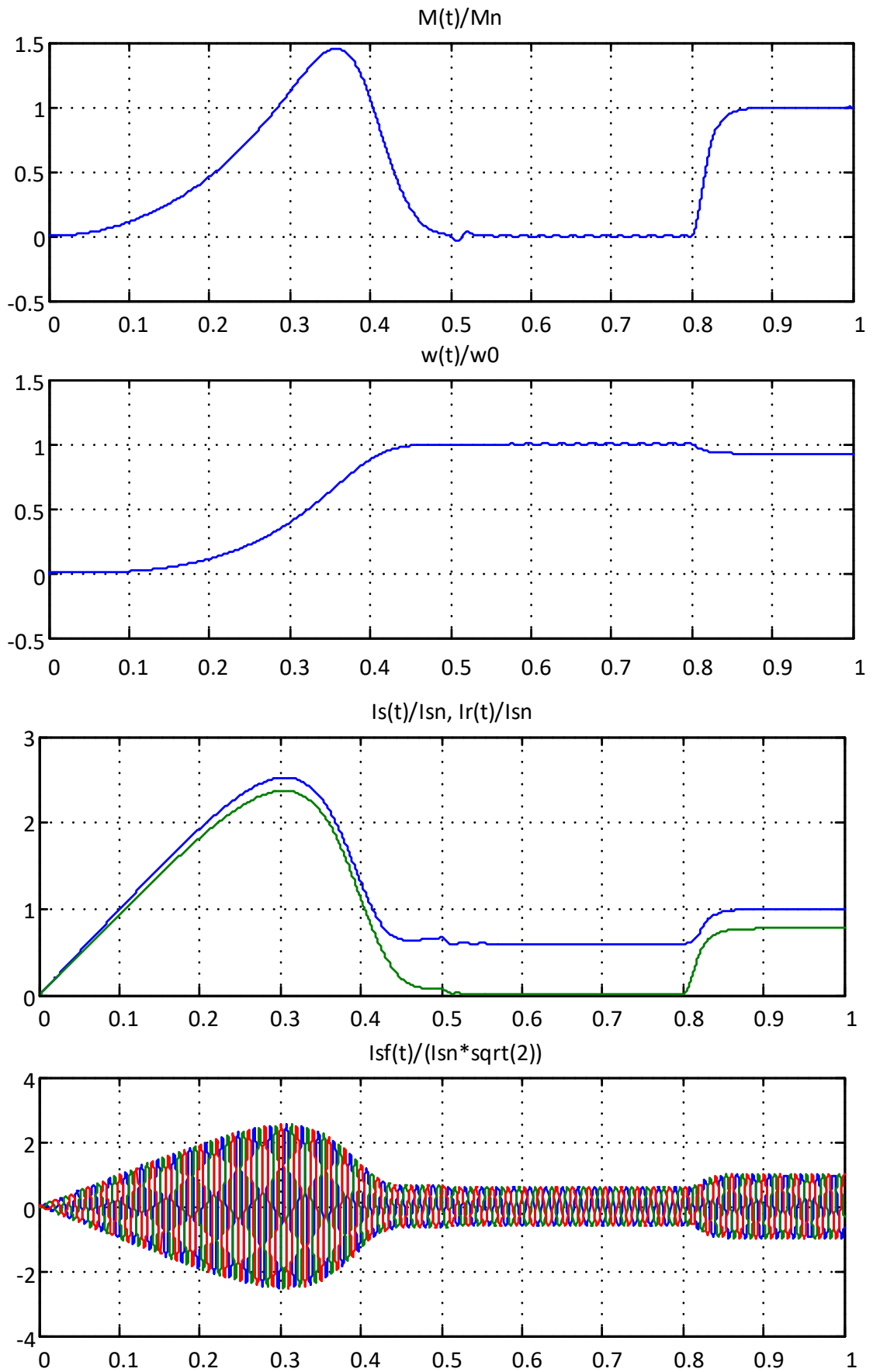


Рис. 10.8. Перехідні процеси при плавному пуску асинхронного двигуна

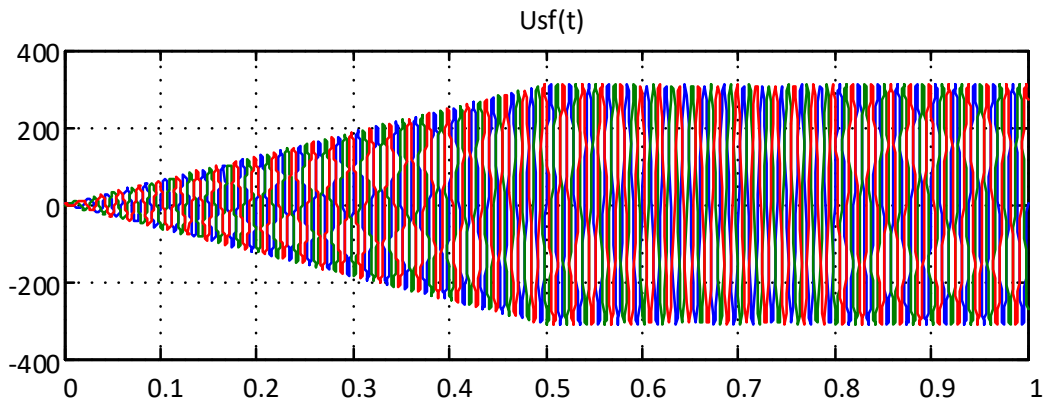


Рис. 10.9. Фазні напруги статора при плавному пуску АД

Плавний пуск АД та регулювання швидкості в невеликому діапазоні здійснюється тиристорними регуляторами напруги (системи ТРН-АД), які у сучасній технічній літературі називають пристроями плавного пуску (*Soft Starter*).

#### 10.4 Моделювання процесів скалярного частотного регулювання швидкості

При частотному регулюванні швидкості двигунів змінного стану застосовують плавну зміну як амплітуди, так і частоти фазних напруг статора [7, 8, 13]. Зазвичай частоту регулюють за лінійним законом, а відношення амплітуди до частоти змінюють за законами  $U/f=const$ ,  $U/f=const$  та  $\sqrt{U}/f=const$ . Для реалізації першого із цих законів достатньо в моделі рис. 10.11 перевести контакт ключа *Manual Switch* у нижню позицію. При цьому перехідні процеси координат АД набудуть вигляду рис. 10.10, 10.11.

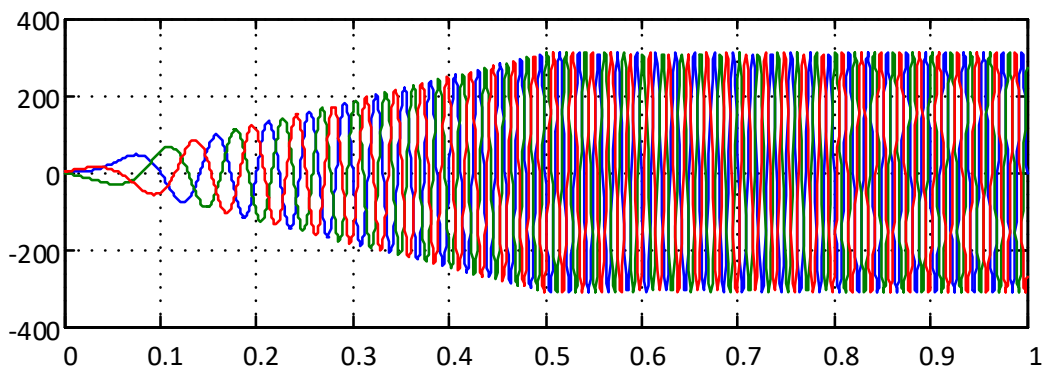


Рис. 10.10. Фазні напруги статора при частотному керуванні  $U/f=const$

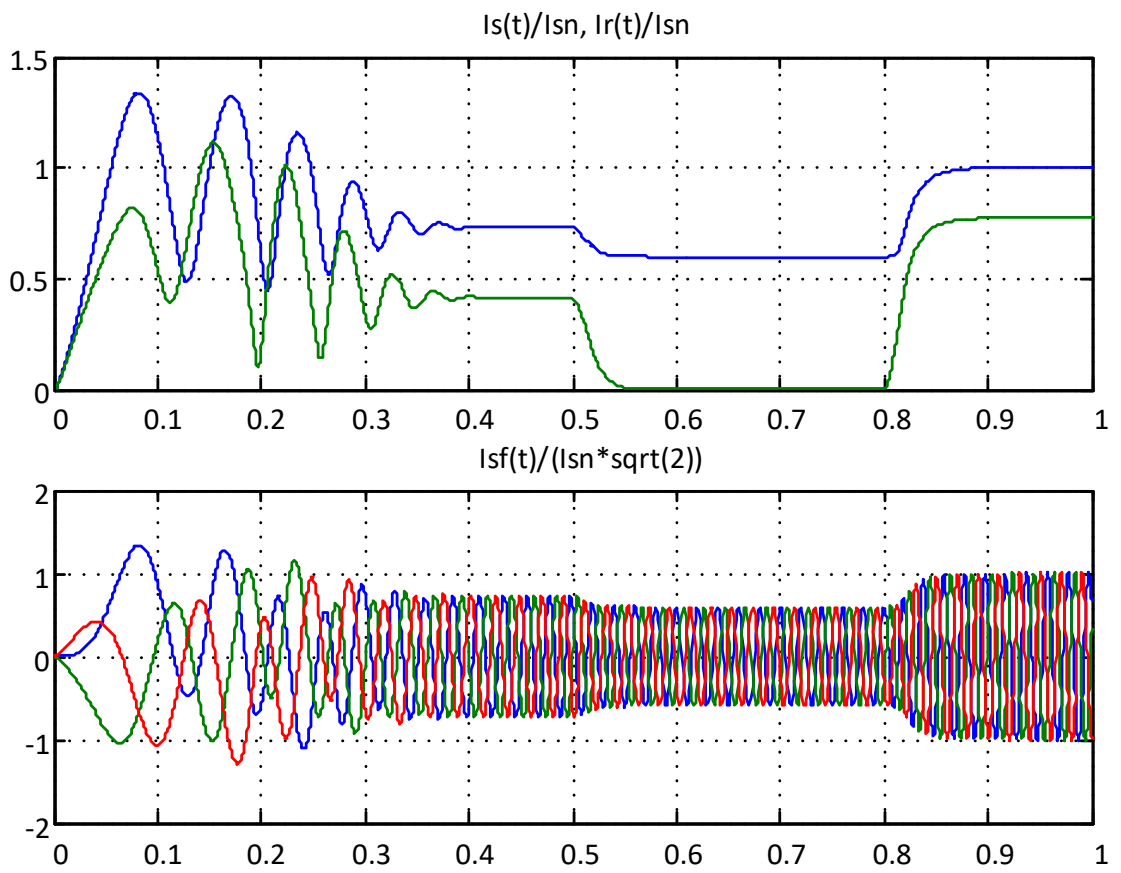


Рис. 10.11. Перехідні процеси в АД при частотному пуску  $U/f = const$

При одночасній зміні амплітуди і частоти напруги живлення статора можна не тільки розганяти двигун, а ще й регулювати його швидкість (див. рис. 10.12, 10.13).

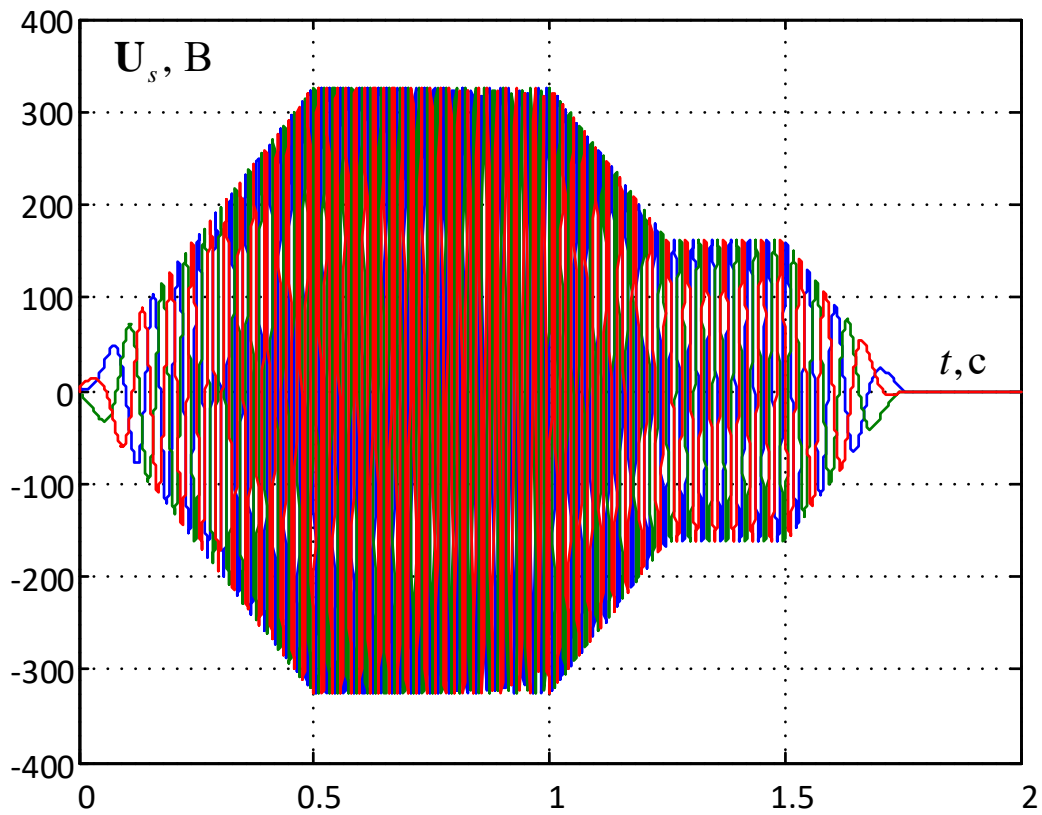


Рис. 10.12. Напряга статора при частотному регулюванні швидкості АД

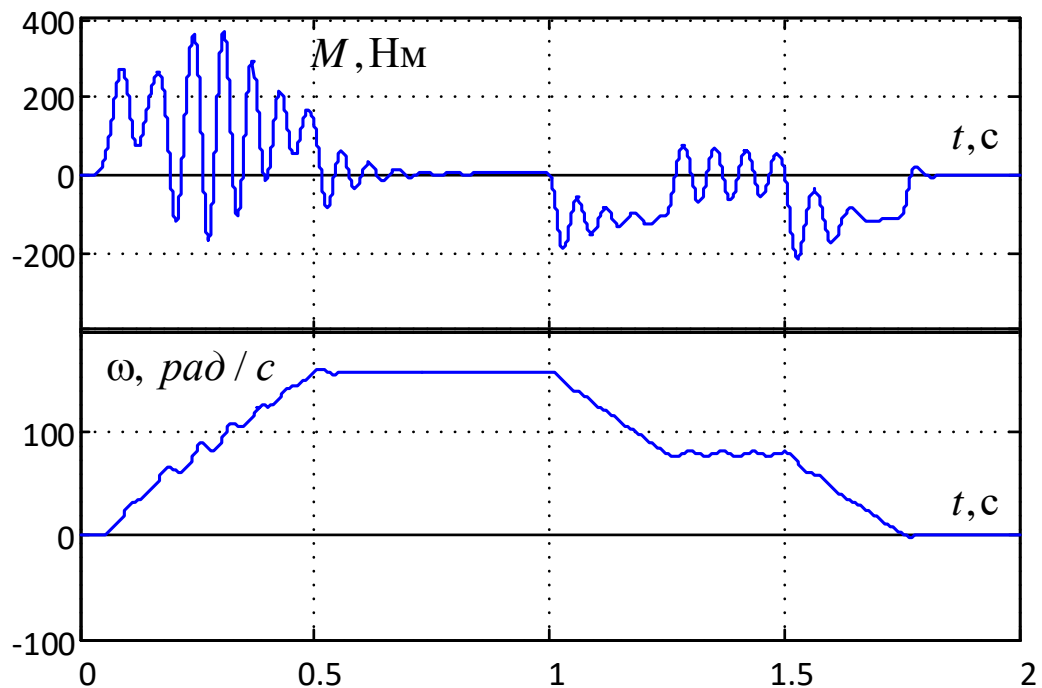


Рис. 10.13. Момент та кутова швидкість АД при скалярному частотному керуванні  $U/f = const$



### 10.5 Побудова статичних характеристик АД

За результатами симуляції розроблених моделей можна побудувати не тільки графіки перехідних процесів, а й статичні механічну  $\omega(M)$  та швидкісну  $I_s(M)$  характеристики АД (див. рис. 10.14, 10.15).

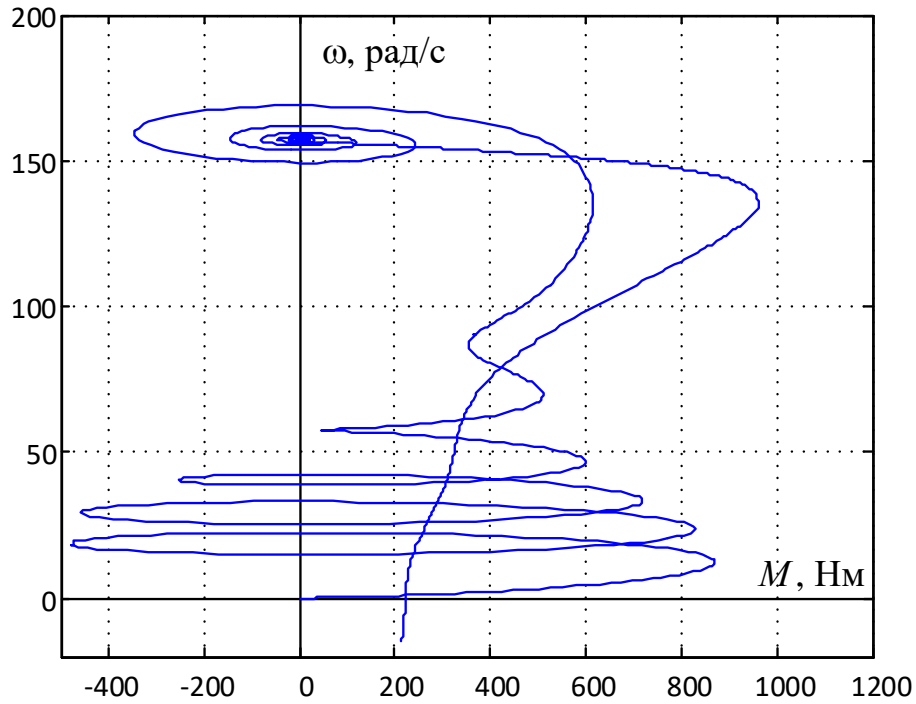


Рис. 10.14. Статична та динамічна механічні характеристики АД

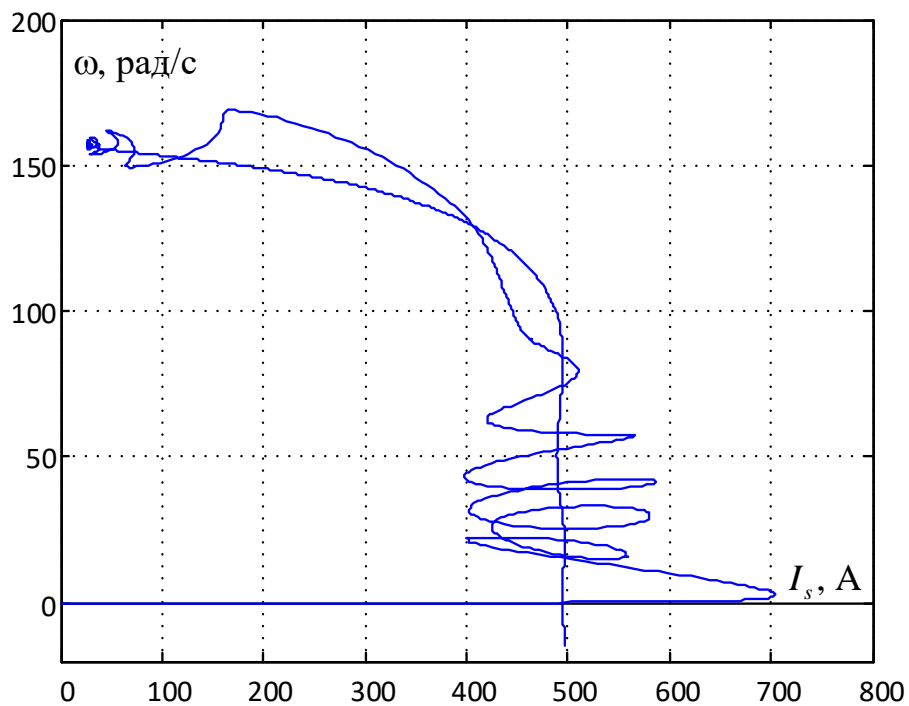


Рис. 10.15. Статична та динамічна швидкісні характеристики АД

Для побудови статичних характеристик достатньо змінювати момент навантаження не стрибком, а поступово, наприклад, лінійно [9, 10]. Динамічні характеристики, які ще називають фазовими портретами отримані в процесі прямого пуску двигуна, а статичні – при повільній зміні Моменту статичного опору за лінійним законом.

### 10.6 Завдання

1. Виконати дослідження процесів прямого, плавного пуску та скалярного розімкненого частотного пуску асинхронного двигуна методом імітаційного фізичного моделювання; після закінчення пуску здійсніть накид номінального навантаження на усталеній швидкості.

2. Виконайте прямий пуск при зниженій на 20% амплітуді живлячої напруги.

3. Виконайте прямий пуск при зниженій на 20% частоті живлячої напруги.

4. Побудуйте статичні та динамічні механічну та швидкісні характеристики АД у двигунному режимі.

Параметри АД з короткозамкненим ротором серії МТК з номінальною лінійною середньоквадратичною напругою статора  $U_{linrmsn} = 380$  В і номінальною частотою живлення  $f_{sn} = 50$  Гц наведені в табл. 10.1.

### 10.7 Методичні рекомендації

1. Для моделювання скористайтеся блоком *Asynchronous Machine SI Units*.

2. Спочатку апробуйте прямий пуск АД з параметрами, обраними через функцію *Preset Model*. Потім використайте параметри з табл. 10.1 згідно зі своїм варіантом. У якості параметрів блоків використовуйте іменами перемінних, значення яких задавайте або розраховуйте у файлі даних або у полі ініціалізації моделі (*File*→*Model Properties*→*Callbacks*→*Init Fn*).

3. Синхронну кутову швидкість АД можна визначити з табл. 10.2, рахованої за формулою

$$n_0 = 60f_{sn} / p = 3000 / p,$$

як найближче значення, більше за номінальне; після цього стає відомою і кількість пар полюсів  $p$ .

Таблиця 10.1

№ 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 <sup>2</sup>
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

Таблиця 10.2

$p$	1	2	3	4	5	6
$n_0$ , об/хв	3000	1500	1000	750	600	500

4. При розрахунку параметрів скористайтеся такими формулами:

$$\omega_s = 2\pi f_s \text{ – кругова частота живлення, рад/с;}$$

$$\omega_n = \frac{\pi n_n}{30} \text{ – номінальна кутова швидкість двигуна, рад/с;}$$

$$M_n = \frac{P_n}{\omega_n} \text{ – номінальний момент, Нм;}$$

$$L_m = X_m / \omega_s, \quad L_{1\sigma} = L_{ls} = X_{1\sigma} / \omega_s, \quad L_{2\sigma} = L_{lr} = X_{2\sigma} / \omega_s;$$

$$U_{f\ peak n} = \sqrt{\frac{2}{3}} U_{linrms n} \text{ – номінальне значення амплітуди фазної напруги}$$

статора;

$$\Psi_{s0} = \frac{U_{f\ peak n}}{\omega_s} \text{ – потокозчеплення холостого ходу статора, Вб;}$$

$$I_{s0} = \frac{\Psi_{s0}}{L_m + L_{1\sigma}} \text{ – струм холостого ходу статора, А;}$$

$$\Psi_{r0} = L_m I_{s0} \text{ – потокозчеплення холостого ходу ротора, Вб.}$$

5. Фіксуйте у графічних вікнах сигнали в долях їх номінальних значень (напруги, струми) або в долях значень в режимі холостого ходу (швидкість, потокозчеплення):

- фазні напруги статора (вікно 1),
- момент та кутову швидкість двигуна (вікно 2),
- фазні струми статора (вікно 3),
- фазні струми ротора (вікно 4),
- діючі значення струмів статора і ротора (вікно 5),
- діючі значення потокозчеплень статора і ротора (вікно 6);

5. Для побудови статичних характеристик лінійне навантаження формуйте блоком *Ramp*, в якому параметр *Start Time* оберіть більшим за час прямого пуску, параметр *Initial output* зробіть нульовим, а параметр *Slope* –  $\approx 3,5M_n / (0,5 \div 1)\text{с}$ . Час моделювання обирайте таким, щоб двигун опрокинувся до швидкості  $\approx -0,2\omega_0$ . Якщо цього складно досягти, то при побудові

графіків статичних та динамічних характеристик користуйтеся функцією `ylim`.

6. Ініціалізацію моделей та побудову графіків виконайте за допомогою «кнопок», створених з блоків *Subsystem* (див. лабораторну роботу 7).

7. Проаналізуйте результати досліджень. Сформулюйте висновки.

### Лабораторна робота №11

## ВИКОРИСТАННЯ SPS-МОДЕЛЕЙ ДВИГУНІВ ДЛЯ ДОСЛІДЖЕННЯ СИСТЕМИ «МЕХАНІЧНИЙ ВАЛ»

*Мета роботи:* придбати навички моделювання багатодвигунних систем, працюючих на спільний вал, з використанням SPS-моделей; впевнитися, що розподілення моментів при накиді навантаження в системі «механічний вал» відбувається пропорційно жорсткості механічних характеристик електричних двигунів.

### 11.1 Особливості застосування SPS-моделей електричних двигунів при дослідженні системи «механічний вал»

Дослідження системи «механічний вал» за допомогою SPS-моделей електричних двигунів стало можливим після відокремлення моделей електромагнітної та механічної частин двигуна, та оснащення блоків механічним портом «швидкість». Модель дводвигунної системи електромеханічної системи «механічний вал» на базі асинхронних двигунів подана на рис. 11.1 [6, 9].

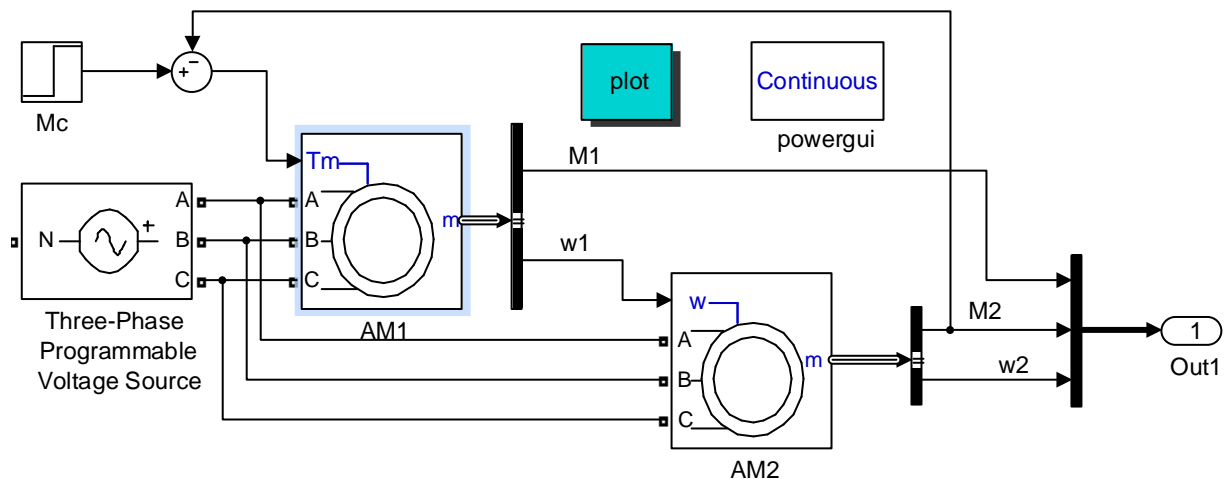


Рис. 11.1. SPS-модель системи «механічний вал» на базі асинхронних двигунів

Вона складається з двох асинхронних двигунів АМ1 та АМ2 з однаковими значеннями номінальної напруги та швидкості, але з різними номінальними моментами. Вони живляться від спільного трифазного *SPS*-джерела *Three-Phase Programmable Voltage Source*. Механічний зв'язок між двигунами здійснюється подачею швидкості  $w_1$  першого двигуна АМ1, на механічний порт  $w$  другого двигуна АМ2.

Спільний момент статичного опору  $M_c$  формується блоком *Step*, але перед подачею його на механічний порт  $T_m$  першого двигуна від нього віднімається електромагнітний момент  $M_2$ , створений другим двигуном. Момент інерції машини зі швидкісним входним портом ігнорується і відноситься до першого двигуна. При чому перша частина цієї операції виконується автоматично, а друга частина повинна бути виконана власноруч користувачем. У такий спосіб моменти обох двигунів підсумовуються і разом діють на спільну механічну масу з моментом інерції  $J_\Sigma = J_1 + J_2$  згідно з рівнянням руху:

$$M_1 + M_2 - M_c = M_j = J_\Sigma \frac{d\omega}{dt}, \quad (11.1)$$

де  $M_1$ ,  $M_2$  – електромагнітні моменти, створені першим АМ1 та другим АМ2 двигунами відповідно.

Дослідження системи «механічний вал» за допомогою моделі рис. 11.1 виконано для двигунів, обраних з використанням функції *Preset Model*: АМ1 – 100 HP, 460 V, 60 Hz, 1780 RPM; АМ2 – 50 HP, 460 V, 60 Hz, 1780 RPM. Перехідні процеси в досліджуваній системі при прямому пуску та при стрибкоподібному накиді навантаження  $M_c = 1000$  Нм в момент часу 0,7 с зображені на рис. 11.2.

Активний опір ротора двигуна АМ2 ( $R_{r2} = 0,058$  Ом) приблизно втричі перевищує відповідний опір двигуна АМ1 ( $R_{r1} = 0,022$  Ом):

$$\frac{R_{r2}}{R_{r1}} = \frac{0,058}{0,022} \approx 2,64$$

Як видно з наведених графіків, відповідно розподіляється і навантаження між двигунами в усталеному режимі [13].

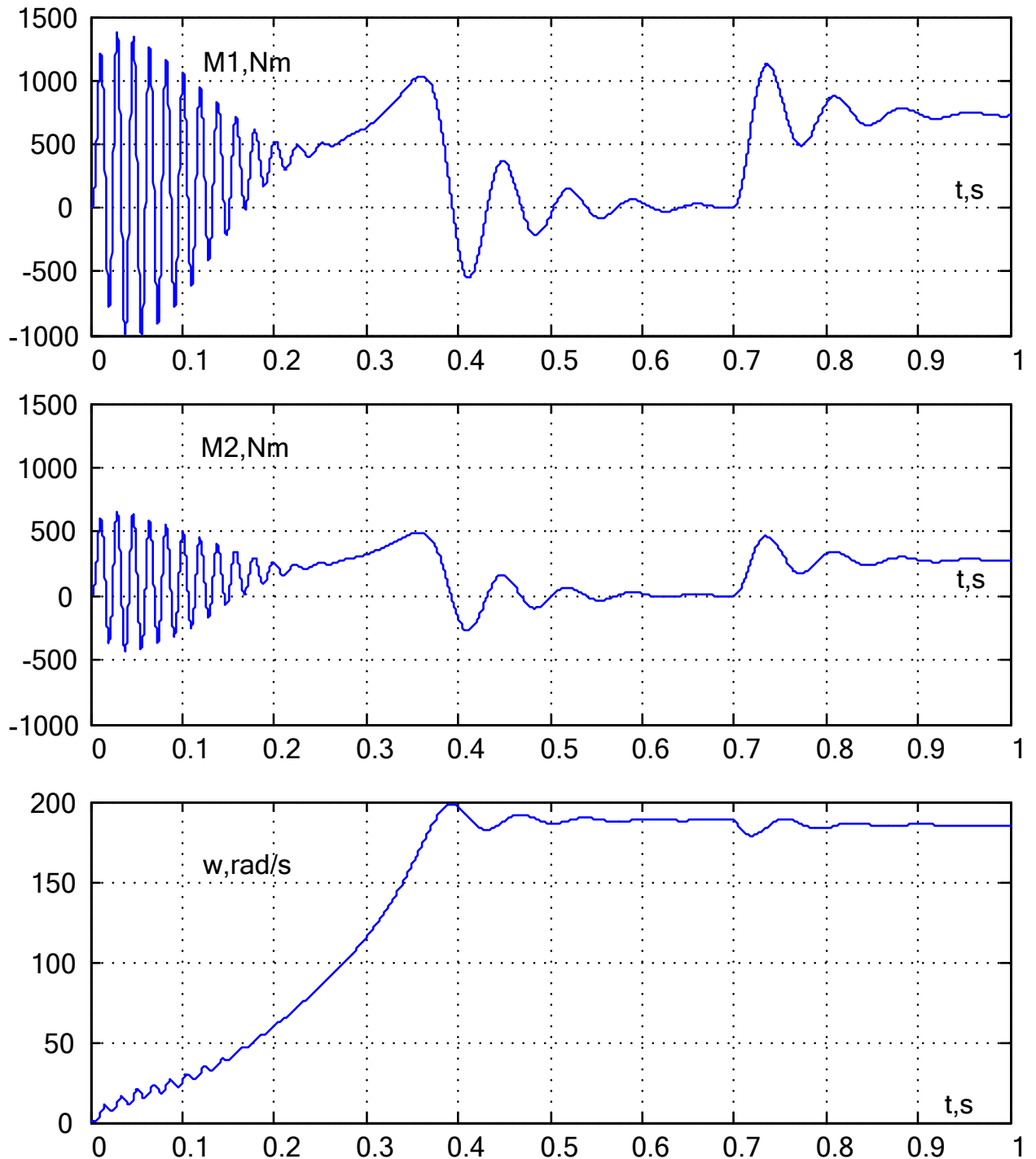


Рис. 11.2. Результати імітаційного моделювання системи «механічний вал» на базі асинхронних двигунів



Аналогічний експеримент можна виконати і для двигунів постійного струму з незалежним збудженням. Модель для цього випадку подана на рис. 11.3.

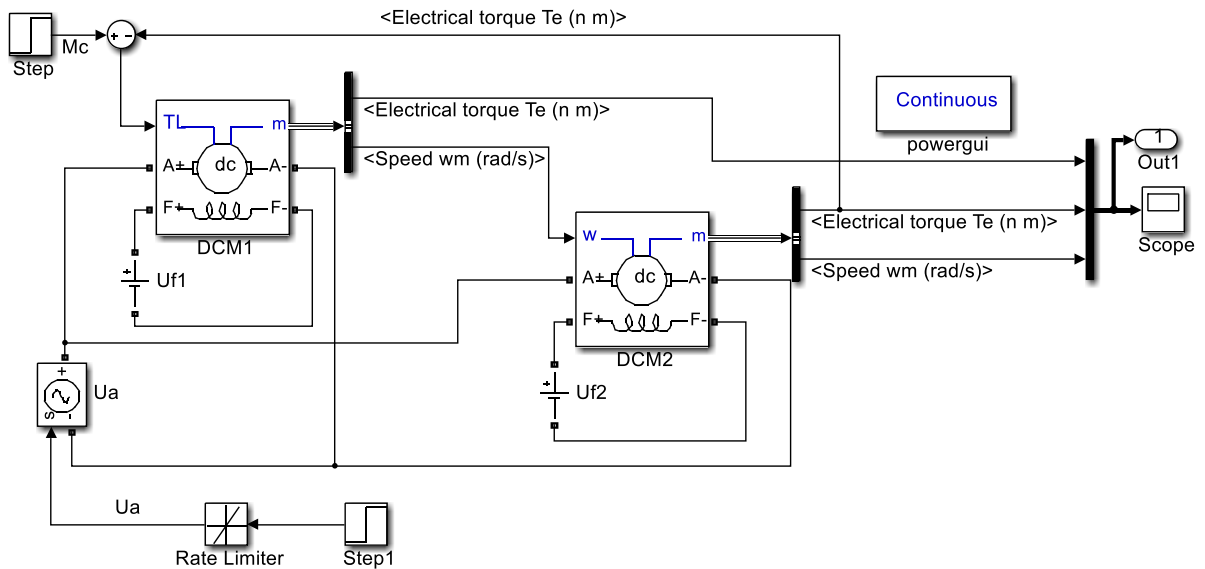


Рис. 11.3. SPS-модель системи «механічний вал» на базі двигунів постійного струму

Вона складається з двох двигунів постійного струму DCM1 та DCM2. Якорі машин, з'єднані паралельно, живляться від спільного керованого джерела ЕРС, а обмотки збудження мають індивідуальні джерела та. Механічний зв'язок між двигунами здійснюється, як і в моделі рис. 11.1, подачею швидкості першого двигуна на механічний порт  $w$  другого двигуна.

В якості досліджуваних двигунів за допомогою функції *Preset Model* обрано 2 однакових двигуна: 200 HP, 500 V, 1750 RPM з напругою збудження 150 V.

Через те, що параметри навіть однакових двигунів можуть виявитися неузгодженими із-за неточностей виготовлення, різних умов експлуатації, в результаті виконання ремонтних робіт тощо, у другому двигуні навмисно змінили активний опір якоря. Отже, експеримент проводили при  $R_{я1} = 0,058 \text{ Ом}$ ,  $R_{я2} = 0,08 \text{ Ом}$ . Момент інерції першого двигуна згідно з методикою, викладеною при складанні попередньої моделі, подвоїли. Графіки перехідних процесів для описаних вище умов, зображені на рис. 11.4.

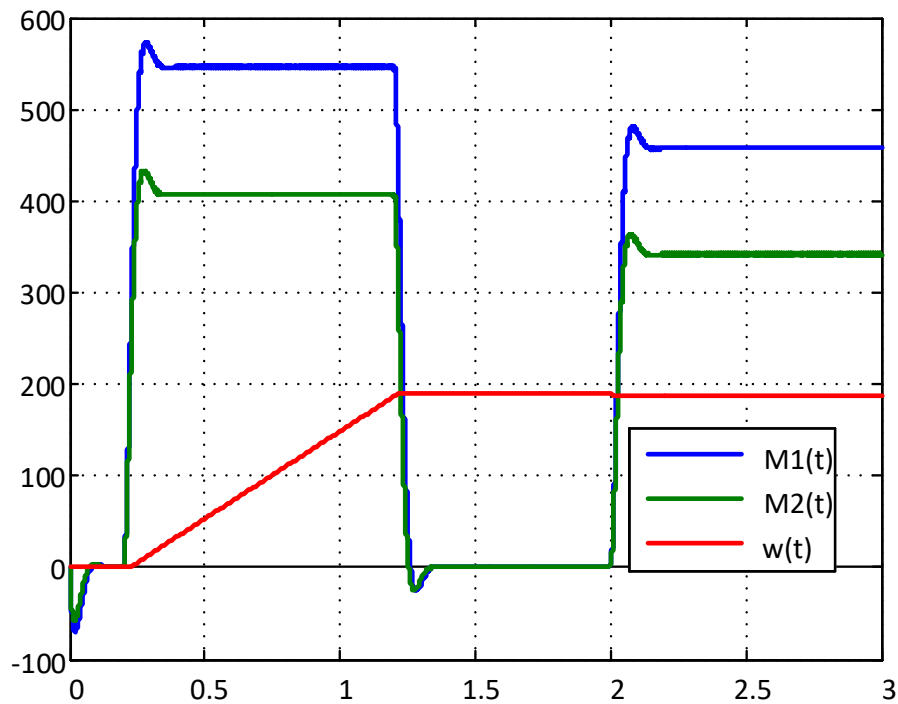


Рис. 11.4. Результати імітаційного моделювання системи «механічний вал» на базі двигунів постійного струму з різними активними опорами якорів

Ще більший вплив на розподілення навантажень чинить розузгодження потоків намагнічування [13]. Наприклад, на рис.11.5 наведені перехідні процеси в досліджуваній системі при зменшенні напруги збудження другого двигуна лише на 1В.

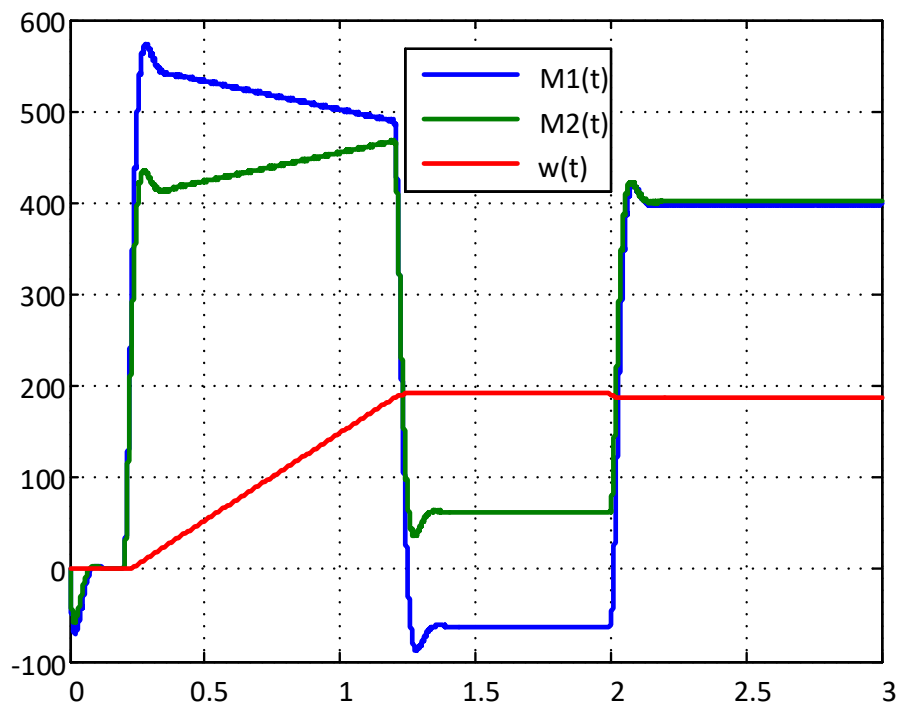


Рис. 11.5. Результати симуляції моделі рис.11.3 з різними активними опорами якорів та зниженою напругою збудження другого двигуна

## 11.2 Завдання

Виконати імітаційне фізичне моделювання систем «механічний вал» на базі асинхронних двигунів та двигунів постійного струму з незалежним збудженням. Дані двигунів взяти з лабораторних робіт №10 та №5 відповідно.

## 11.3 Методичні рекомендації

1. У якості розузгоджуваних параметрів оберіть для асинхронних двигунів активний опір ротора, а для двигуна постійного струму – активний опір якоря.
2. Розузгоджуйте обрані параметри на 10-20%.
3. На графіках фіксуйте моменти двигунів та кутову швидкість спільного валу.

*Лабораторна робота №12***МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ДВОМАСОВИХ  
МЕХАНІЧНИХ СИСТЕМ**

*Мета роботи:* на прикладі моделювання двомасової механічної системи (ДМС) придбати навички математичного опису лінійних механічних об'єктів з одним ступенем свободи, отримати інформацію про статичні та динамічні властивості двомасових механічних систем і їх залежність від сталих часу досліджуваного об'єкту регулювання та коефіцієнтів розподілення мас.

**12.1 Математичний опис двомасових механічних систем**

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:

$$F_{\Pi} = c_{ij} \cdot \Delta s = c_{ij} (s_i - s_j) \quad M_{\Pi} = c_{ij} \cdot \Delta \varphi = c_{ij} (\varphi_i - \varphi_j). \quad (12.1)$$

Пружній деформації перешкоджають момент або зусилля внутрішнього в'язкого тертя, зумовлені силами міжмолекулярної взаємодії деформованого матеріалу і пропорційне різниці між швидкостями протилежних кінцівок передачі:

$$F_B = d_{ij} \cdot \Delta v = d_{ij}(v_i - v_j) \quad M_B = d_{ij} \cdot \Delta \omega = d_{ij}(\omega_i - \omega_j). \quad (12.2)$$

При взаємодії величин (12.49) та (12.50) утворюється пружно-в'язке зусилля або пружно-в'язкий момент

$$F_{ij} = F_{\Pi} + F_B \quad M_{ij} = M_{\Pi} + M_B. \quad (12.3)$$

Для прикладу розглянемо кінематичну схему двомасової пружно-в'язкої механічної системи поступального руху, зображену на рис. 12.1.

Її математичний опис без урахування сухого тертя має вигляд



$$\left\{ \begin{array}{l} F - F_{12} = m_1 \frac{dv_1}{dt}, \\ F_{12} = m_2 \frac{dv_2}{dt}, \\ \frac{ds_1}{dt} = v_1, \\ \frac{ds_2}{dt} = v_2, \\ F_{\Pi} = c_{12}(s_1 - s_2), \\ F_B = d_{12}(v_1 - v_2), \\ F_{12} = F_{\Pi} + F_B. \end{array} \right. \quad (12.4)$$

Рис. 12.1. Кінематична схема двомасової механічної системи поступального руху

Кінематична схема двомасової механічної системи обертального руху зображена на рис. 12.2.

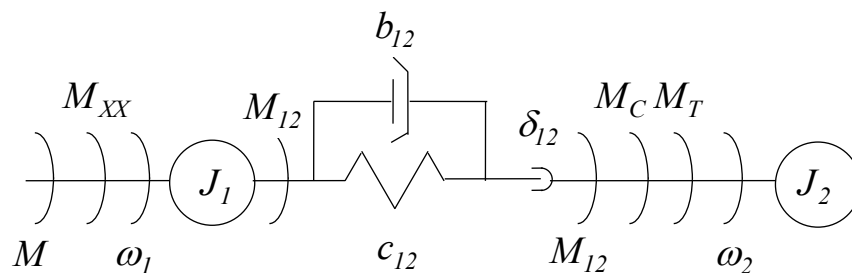


Рис. 12.2. Спрощена кінематична схема двомасової електромеханічної системи

На ній позначено:

- $M_{xx}$ ,  $M_{12}$ ,  $M_C$ ,  $M_T$  – електромагнітний момент двигуна, момент холостого ходу двигуна, момент скручування валу, момент статичного опору від навантаження і момент сухого тертя в механізмі відповідно;

- $\omega_1, \omega_2$  – кутові швидкості двигуна і механізму;
- $c_{12}, b_{12}$  – коефіцієнти жорсткості і в'язкого тертя;
- $\delta_{12}$  – 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]:

$$\left\{ \begin{array}{l} M - M_{12} = J_1 \frac{d\omega_1}{dt}, \\ M_{12} - M_c = J_2 \frac{d\omega_2}{dt}, \\ \frac{d\phi_1}{dt} = \omega_1, \\ \frac{d\phi_2}{dt} = \omega_2, \\ M_{\Pi} = c_{12}(\phi_1 - \phi_2), \\ M_B = b_{12}(\omega_1 - \omega_2), \\ M_{12} = M_{\Pi} + M_B. \end{array} \right. \quad (12.5)$$

## 12.2. Дослідження двомасових механічних систем

Структурна *Simulink*-модель, складена за рівняннями (12.52), зображена на рис. 12.3, а структурна схема двомасової механічної системи обертального руху – на рис. 12.4.

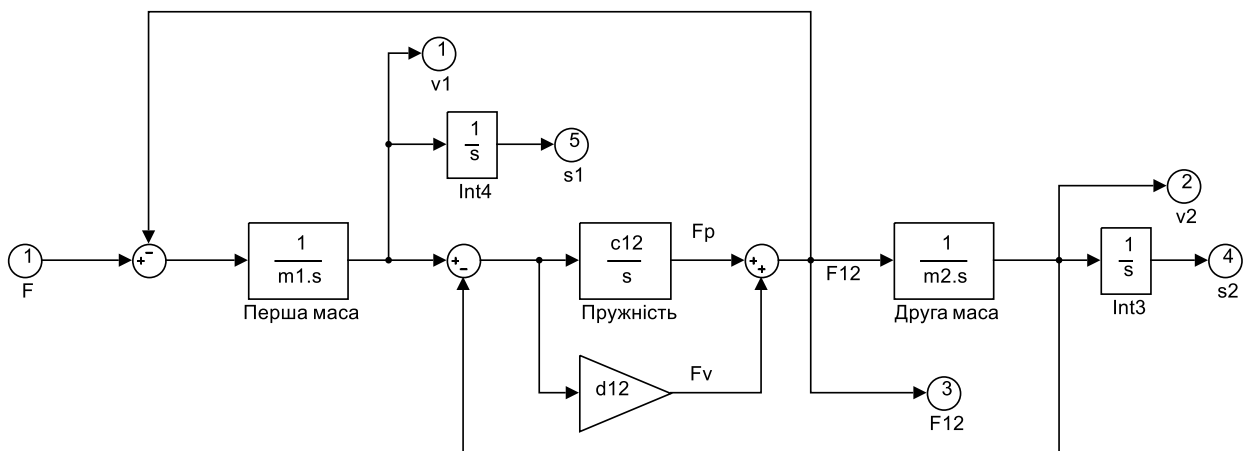


Рис. 12.3. Структурна *Simulink*-модель двомасової механічної системи поступального руху

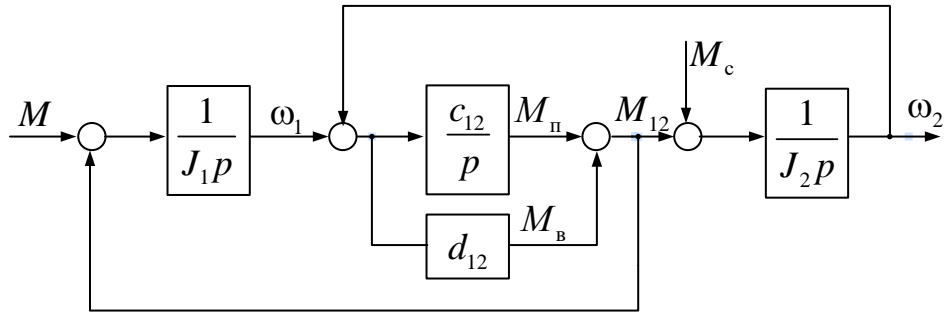


Рис. 12.4. Структурна схема двомасової механічної системи обертального руху

Щоб проаналізувати такий об'єкт, виконаємо еквівалентні структурні перетворення СС рис. 12.4 за керуванням у ланцюжок послідовно з'єднаних ПФ, як це показано на рис. 12.5.

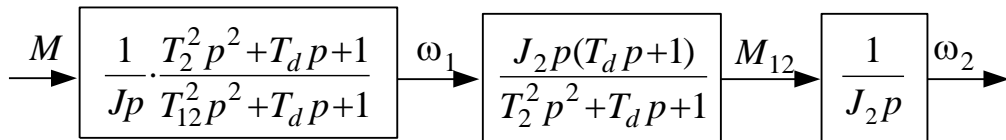


Рис. 12.5. Перетворена структурна схема системи рис. 12.4 без врахування дії статичного моменту

У схемі прийняті такі позначення:

$$J = J_1 + J_2 \quad (12.6)$$

– сумарний момент інерції;

$$T_{12} = \sqrt{\frac{J_1 J_2}{c_{12} J}} = \sqrt{\frac{J_2}{c_{12} \gamma}} \quad (12.7)$$

– стала часу пружних коливань ДМО;

$$\gamma = \frac{J}{J_1} = 1 + \frac{J_2}{J_1} \quad (12.8)$$

– коефіцієнт співвідношення мас;

$$T_2 = T_{12} \sqrt{\gamma} = \sqrt{\frac{J_2}{c_{12}}} \quad (12.9)$$

– стала часу пружних коливань другої маси;

$$T_d = b_{12} / c_{12} \quad (12.10)$$

– стала часу демпфірування пружних коливань;

$$\zeta = \frac{T_d}{2T_{12}} \quad (12.11)$$

– коефіцієнт демпфірування пружних коливань.

Зі СС рис. 12.6 легко знайти такі ПФ:

$$\frac{M_{12}(p)}{M(p)} = \frac{J_2}{J_1} \cdot \frac{T_d p + 1}{T_{12}^2 p^2 + T_d p + 1}, \quad (12.12)$$

$$\frac{\omega_1(p)}{M(p)} = \frac{1}{Jp} \cdot \frac{T_2^2 p^2 + T_d p + 1}{T_{12}^2 p^2 + T_d p + 1}, \quad \frac{\omega_2(p)}{M(p)} = \frac{1}{Jp} \cdot \frac{T_d p + 1}{T_{12}^2 p^2 + T_d p + 1}, \quad (12.13)$$

$$\frac{\omega_2(p)}{\omega_1(p)} = \frac{T_d p + 1}{T_2^2 p^2 + T_d p + 1}. \quad (12.14)$$

При дослідженні багатомасових систем часто нехтують в'язким тертям.

У цьому випадку ПФ (12.12)-(12.14) набувають вигляду:

$$\frac{M_{12}(p)}{M(p)} = \frac{J_2}{J_1} \cdot \frac{1}{T_{12}^2 p^2 + 1}, \quad (12.15)$$

$$\frac{\omega_1(p)}{M(p)} = \frac{1}{Jp} \cdot \frac{T_2^2 p^2 + 1}{T_{12}^2 p^2 + 1}, \quad \frac{\omega_2(p)}{M(p)} = \frac{1}{Jp} \cdot \frac{1}{T_{12}^2 p^2 + 1}, \quad (12.16)$$

$$\frac{\omega_2(p)}{\omega_1(p)} = \frac{1}{T_2^2 p^2 + 1}, \quad \frac{M_{12}(p)}{\omega_1(p)} = \frac{J_2}{T_2^2 p^2 + 1}. \quad (12.17)$$

ПФ (12.15) та (12.17) – консервативні ланки, тобто коливальні ланки з нульовим демпфіруванням, яким відповідають гармонічні перехідні функції без загасання з круговими частотами та періодами коливань

$$\Omega_{12} = 1/T_{12}, \quad \Omega_2 = 1/T_2, \quad T_{k12} = 2\pi T_{12}, \quad T_{k2} = 2\pi T_2. \quad (12.18)$$

Тобто реакція пружного моменту на стрибок електромагнітного моменту та реакція швидкості другої маси та пружного моменту на стрибок швидкості першої маси визначаються рівняннями

$$M_{12}(t) = M \frac{J_2}{J_1} \cdot (1 - \cos(\Omega_{12}t)), \quad (12.19)$$

$$\omega_2(t) = \omega_1 (1 - \cos(\Omega_2 t)), \quad M_{12}(t) = J_2 \frac{d\omega_2(t)}{dt} = J_2 \omega_1 \Omega_2 \sin(\Omega_2 t). \quad (12.20)$$



Два останні рівняння є ключем для перевірки адекватності моделі та відповідності її заданим параметрам. Перехідні процеси, що відповідають цим рівнянням зображені на рис. 12.6-12.7.

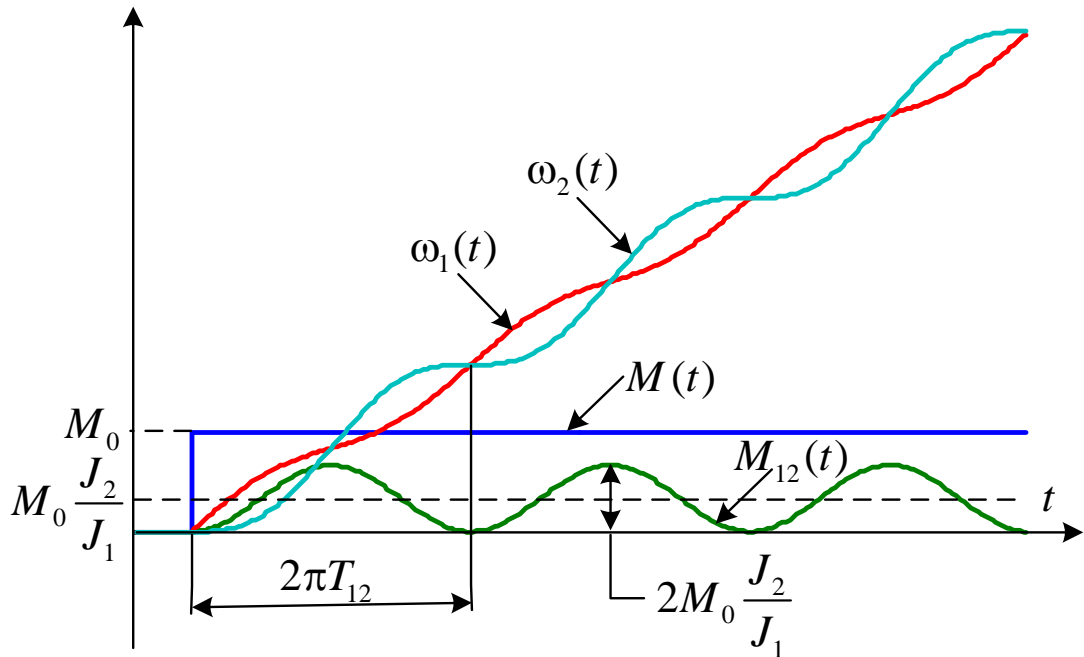


Рис. 12.6. Реакція ДМС на стрибок моменту

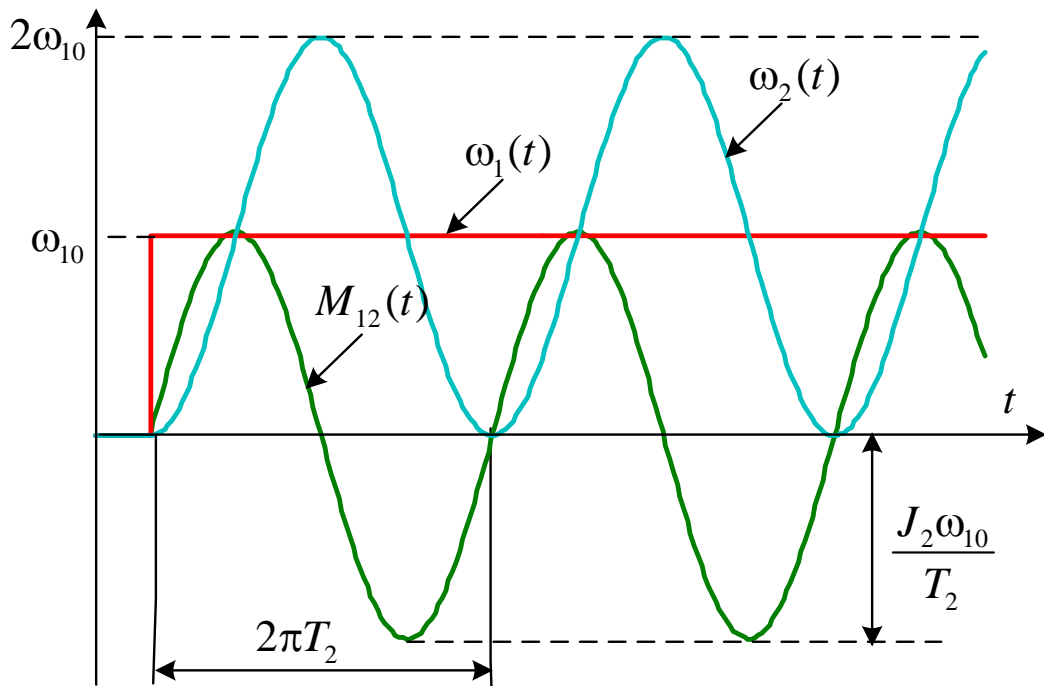


Рис. 12.7. Реакція другої маси на стрибок швидкості першої маси

## 12.2 Завдання

1. Промоделювати ДМС при зусиллі, зображеному на рис. 12.8а, при значенні параметра  $t_0$ , кратному  $T_{12}$  ( $t_0 = kT_{12}$ ,  $k$  – ціле число) і не кратному  $T_{12}$ . Зробити висновки щодо амплітуди коливань досліджуваних сигналів в усталеному режимі.

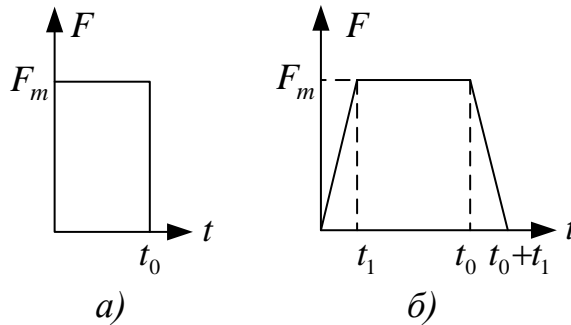


Рис. 12.8. Варіанти вхідних зусиль

2. Промоделювати ДМС при трапецоїдальній формі вхідного зусилля (рис. 12.1б).

3. Повторити експерименти 4 і 5 при 3-х значеннях коефіцієнту в'язкого тертя, заданих у таблиці. Порівняти перехідні процеси.

Параметри систем надано у табл. 12.1.

## 12.3 Методичні рекомендації

1. Для порівняння зберіть *Simulink*-моделі ДМС поступального руху та модель системи з жорстким кінематичним зв'язком між двома масами.

2. Розрахуйте сталі часу, періоди коливань та коефіцієнти демпфірування ДМС при заданих параметрах.

3. До виконання основного завдання перевірте адекватність моделей і відповідність їх заданим параметрам шляхом аналізу перехідних процесів при одиничному стрибку вхідного зусилля та при одиничному стрибку швидкості першої маси, як це показано на рис. 12.6, 12.7.

Таблица 12.1

№ вар.	$m_1$ , кг	$m_2$ , кг	$c_{12}$ , кг/с <sup>2</sup>	$d_{12}$ , кг/с	$F_m$ , Н	$k$	$\alpha$
1,13	5	5	20	0 5 10	16	3 3,2 3,5	0,2
2,14	5	5	20	0 5 10	10	2 2,2 2,5	0,1
3,15	5	5	20	0 5 10	20	4 4,2 4,5	0,3
4,16	5	5	20	0 5 10	15	3 3,2 3,5	0,2
5,17	7,5	2,5	10	0 2 5	8	2 2,2 2,5	0,1
6,18	7,5	2,5	10	0 2 5	18	4 4,2 4,5	0,3
7,19	7,5	2,5	10	0 2 5	16	3 3,2 3,5	0,2
8,20	7,5	2,5	10	0 2 5	10	2 2,2 2,5	0,1
9,21	2,5	7,5	20	0 5 10	20	4 4,2 4,5	0,3
10,22	2,5	7,5	20	0 5 10	15	3 3,2 3,5	0,2
11,23	2,5	7,5	20	0 5 10	8	2 2,2 2,5	0,1
12,24	2,5	7,5	20	0 5 10	18	4 4,2 4,5	0,3

*Лабораторна робота №13*

**МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ КРАНОВОЇ МЕХАНІЧНОЇ СИСТЕМИ ВІЗОК-ВАНТАЖ**

*Мета роботи:* придбати навички складання математичного опису механічних систем з декількома ступіннями свободи за допомогою рівнянь Лагранжа 2-го роду, лінеаризації математичного опису нелінійних механічних систем, дослідження таких систем методом математичного моделювання

**13.1 Математичний опис механічної системи візок-вантаж**

Кранова механічна система візок-вантаж складається з візка, що здійснює поступальний рух вперед-назад і прикріпленого до нього за допомогою гнучкого канату вантажу, який може рухатися подібно до маятника, тобто здійснювати коливання у вертикальній площині.

Схематичне зображення такого механічного об'єкту подано на рис. 13.1, де прийнято такі позначення:  $M$  – маса візка,  $m$  – маса вантажу,  $L$  – довжина канату,  $F_x$  – зусилля, прикладене до візка,  $v$  – горизонтальна швидкість візка,  $s$  – горизонтальне переміщення візка,  $\varphi$  – кут відхилення вантажу від вертикалі,  $\omega$  – кутова швидкість коливань вантажу. Розподілення зусиль в цій системі показано на рис. 13.2.

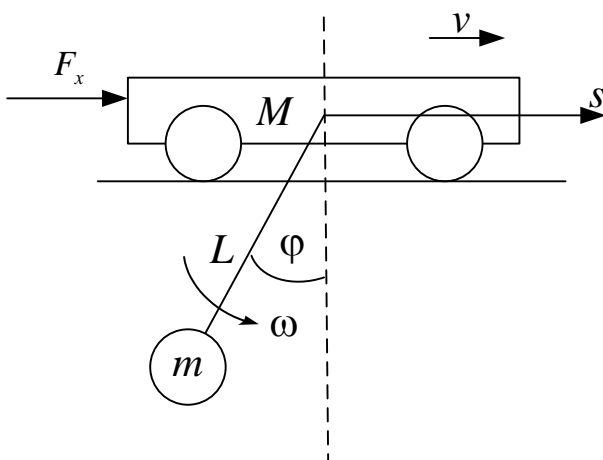


Рис. 13.1. Схематичне зображення візка з підвішеним до нього вантажем

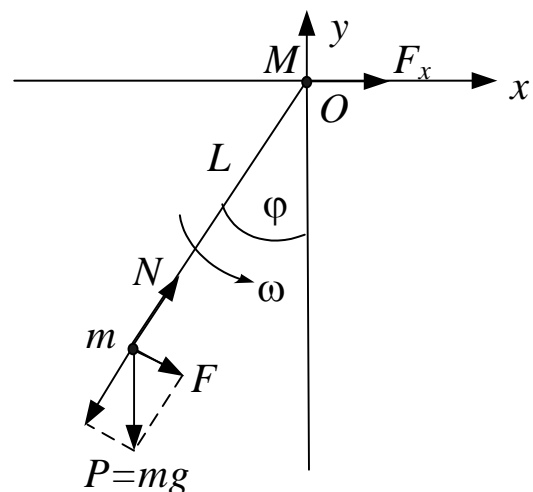


Рис. 13.2. Розподілення зусиль в системі візок-вантаж

Сила тяжіння вантажу розкладається на 2 складових, одна з яких ( $F_N$ ) урівноважує силу натягу канату  $N$ , а друга ( $F$ ) утворює обертальний момент. Візок, як і вантаж, вважатимемо точковими масами, зосередженими у центрах їх тяжіння. Рух візка відбувається у горизонтальному напрямку впродовж осі  $x$  під дією деякої сили  $F_x$  при постійній довжині канату. Канат вважаємо невагомим, і нехтуємо його пружно-в'язкими властивостями.

Для складання математичного опису скористаємося рівняннями Лагранжа другого роду, яке у загальному випадку має такий вигляд:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i, \quad (13.1)$$

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

$$\begin{aligned} \vec{s}_M &= [s_{Mx}(t), 0] = [x(t), 0], \\ \vec{s}_m &= [s_{mx}(t), s_{my}(t)] = [x(t) + L \sin \varphi(t), -L \cos \varphi(t)], \end{aligned} \quad (13.2)$$

у результаті диференціювання яких отримаємо такі вирази для проекцій векторів швидкостей руху візка і вантажу на обрані координатні осі:

$$\begin{aligned} \vec{v}_M &= [v_{Mx}(t), 0] = [v_x(t), 0], \\ \vec{v}_m &= [v_{mx}(t), 0] = [v_x(t) + L \omega(t) \cos \varphi(t), L \omega(t) \sin \varphi(t)]. \end{aligned} \quad (13.3)$$

Для системи візок-вантаж формули потенціальної і кінетичної енергії мають вигляд

$$\begin{aligned} W_k &= \frac{m}{2} \left[ (v_x(t) + L \omega(t) \cos \varphi(t))^2 + (L \omega(t) \sin \varphi(t))^2 \right] + \frac{M}{2} v_x^2(t) = \\ &= \frac{m+M}{2} v_x^2(t) + \frac{mL}{2} \left( L \omega^2(t) + 2v_x(t) \omega(t) \cos \varphi(t) \right), \\ W_p &= -mgL \cos \varphi(t). \end{aligned}$$

Тоді функція Лагранжа для системи «візок-вантаж» матиме вигляд:

$$\mathcal{L} = \frac{M+m}{2} v_x^2(t) + \frac{mL}{2} (2v(t)\omega(t)\cos\varphi(t) + L\omega^2(t) + 2g\cos\varphi(t)). \quad (13.4)$$

Узагальненими координатами і силами для досліджуваної системи являються:

$$\begin{cases} q_1 = x, & \dot{q}_1 = v_x, & Q_1 = F_x; \\ q_2 = \varphi, & \dot{q}_2 = \omega, & Q_2 = 0. \end{cases} \quad (13.5)$$

Визначивши послідовно усі складові рівняння Лагранжа (13.4), одержуємо такі рівняння:

$$\begin{cases} (M+m) \frac{dv_x(t)}{dt} + mL \cos\varphi(t) \cdot \varepsilon(t) - mL\omega^2(t) \cdot \sin\varphi(t) = F(t), \\ L \frac{d\omega(t)}{dt} + \cos\varphi(t) \cdot a_x(t) + g \sin\varphi(t) = 0, \end{cases} \quad (13.6)$$

які доповнюємо рівняннями, що пов'язують узагальнені координати з їхніми похідними:

$$\begin{cases} \frac{dx(t)}{dt} = v_x(t), & \frac{dv_x(t)}{dt} = a_x(t); \\ \frac{d\varphi(t)}{dt} = \omega(t), & \frac{d\omega(t)}{dt} = \varepsilon(t). \end{cases} \quad (13.7)$$

де  $a(t)$  – лінійне прискорення візка;  $\varepsilon(t)$  – кутове прискорення вантажу;

Структурна модель, що відповідає рівнянням (13.11), (13.12), наведена на рис. 13.3.

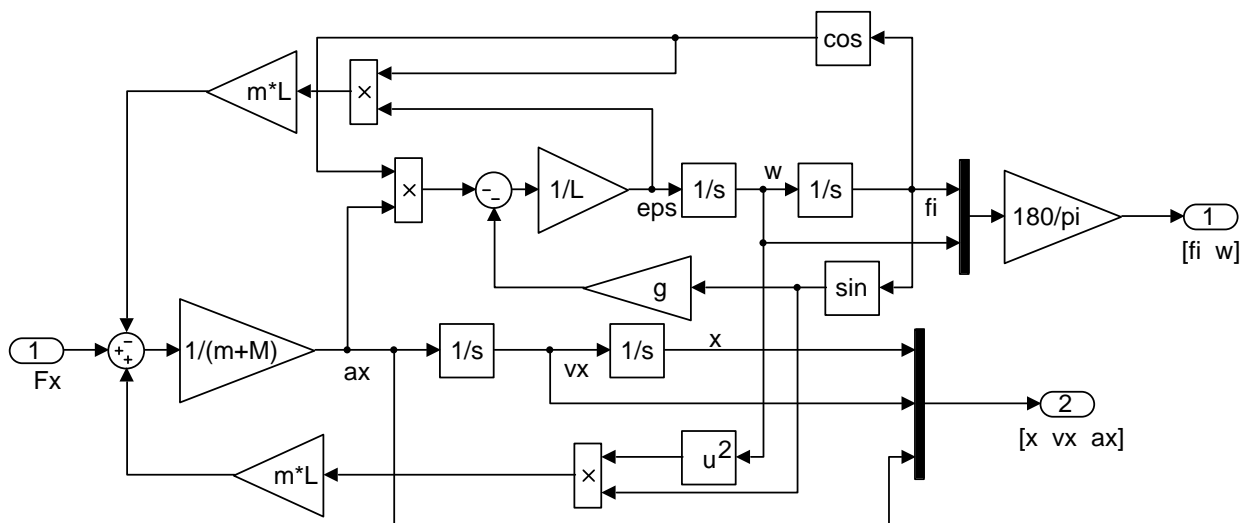


Рис. 13.3. Структурна модель системи візок-вантаж

Перехідні процеси коливань вантажу і візка при стрибкоподібній зміні зусилля, отримані при симуляції структурної моделі рис. 13.3, показані на рис. 13.4.

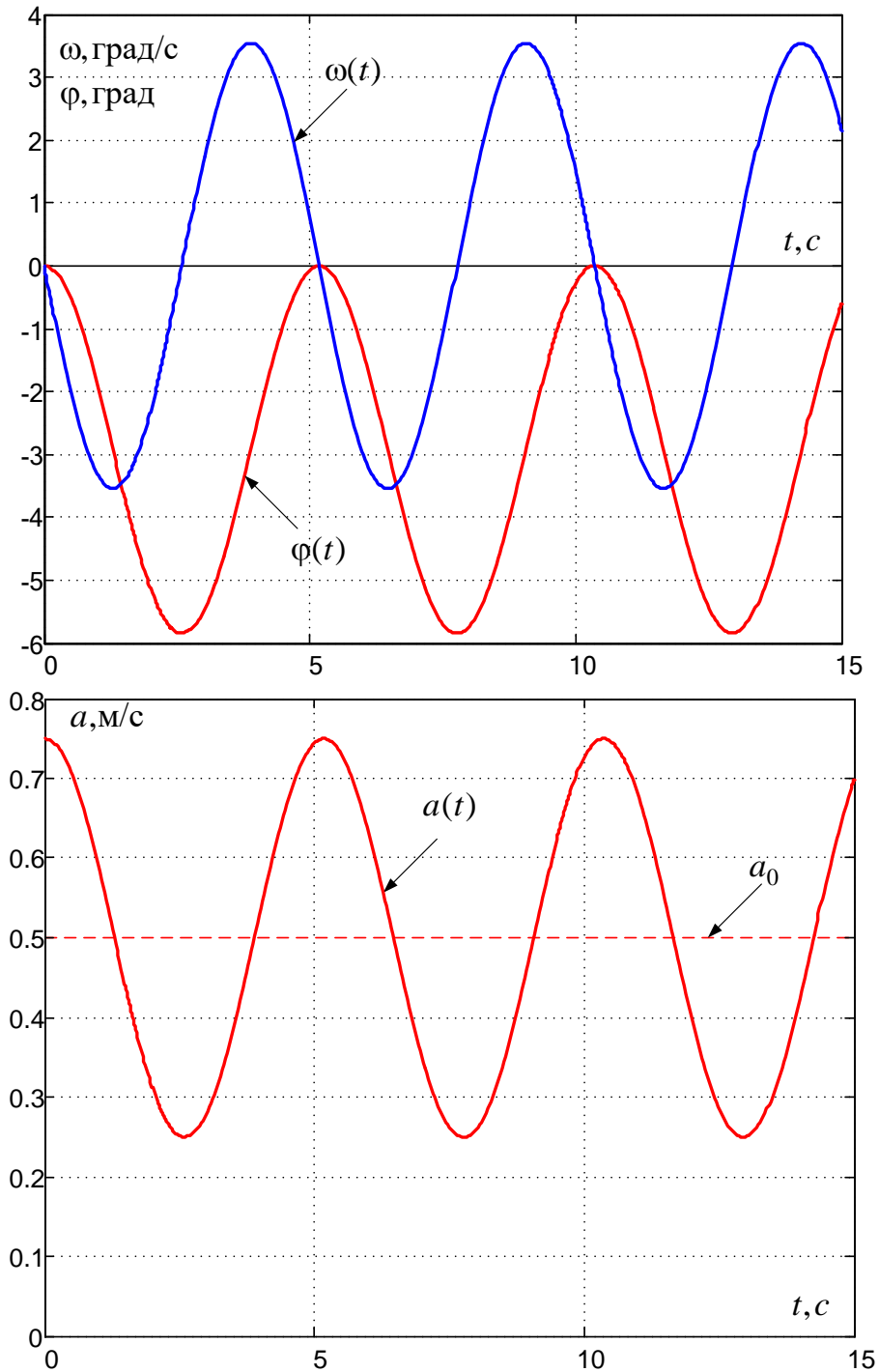


Рис. 13.4. Реакція механічної системи візок-вантаж на стрибок сили

$$F_x(t) = F_0 \cdot 1(t)$$

При моделюванні використано такі параметри: маса візка  $M = 1000$  кг, маса вантажу  $m = 500$  кг, довжина канату  $L = 10$  м, середнє прискорення  $a_0 = 0.5$  м/с<sup>2</sup>,  $F_0 = a_0(M + m) = 750$  Нм.

Для того, щоб оцінити ступінь адекватності розроблених нелінійних моделей, їх треба лінеаризувати. З рис. 13.7 видно, що при пересуванні візка значення кута відхилення канату від вертикалі приймає досить невеликі значення, завдяки чому можна прийняти:

$$\varphi \approx 0, \quad \cos \varphi \approx 1, \quad \sin \varphi \approx \varphi, \quad \omega^2 \approx 0. \quad (13.8)$$

В такому випадку рівняння (13.6) набувають вигляду:

$$\begin{cases} (M + m) \frac{dv(t)}{dt} = F(t) - mL\varepsilon(t), \\ L \frac{d\omega(t)}{dt} = -a(t) - g\varphi(t). \end{cases} \quad (13.9)$$

Такій системі рівнянь відповідає структурна модель, наведена на рис. 13.5.

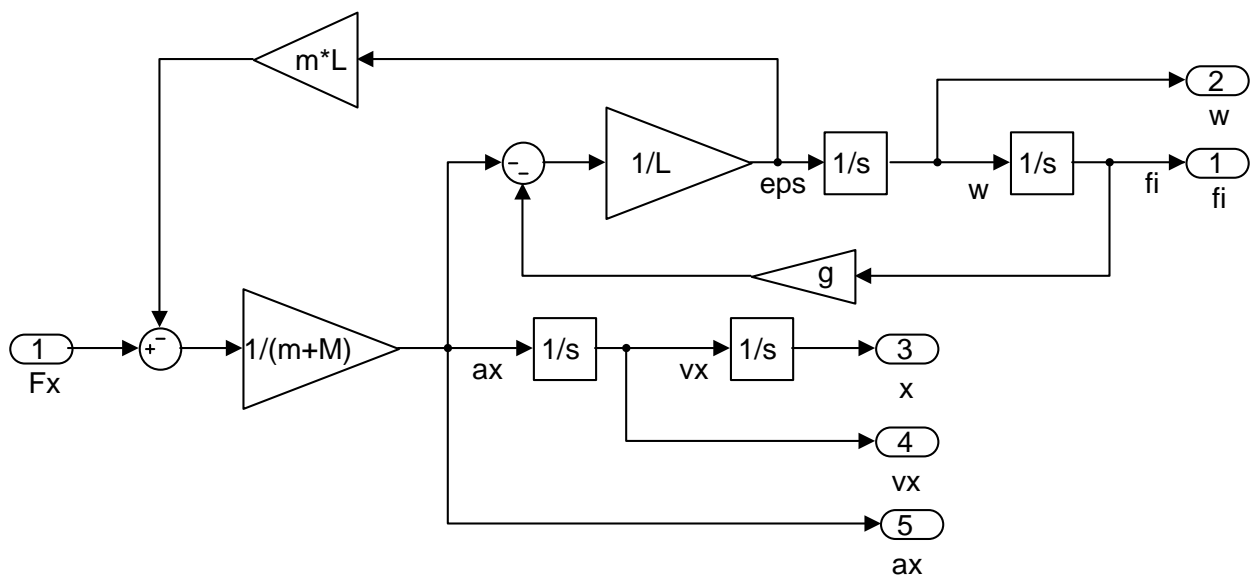


Рис. 13.5. Лінеаризована структурна модель системи візок-вантаж

Із структурної моделі рис. 13.5 знаходимо передавальні функції (ПФ) механічного об'єкта:

$$\frac{\varphi(p)}{a_x(p)} = -\frac{1}{g} \cdot \frac{1}{(T_0^2 p^2 + 1)}, \quad (13.10)$$



$$\frac{\varphi(p)}{F_x(p)} = -\frac{1}{(M+m)g} \cdot \frac{1}{(T_{12}^2 p^2 + 1)}, \quad (13.11)$$

де

$$T_{12} = \sqrt{\frac{L}{g} \cdot \frac{M}{M+m}} = T_0 \sqrt{\frac{M}{M+m}} \quad (13.12)$$

– стала часу коливань двомасової системи візок-вантаж.

Діленням (13.11) на (13.12) отримуємо:

$$\frac{a_x(p)}{F_x(p)} = \frac{1}{(M+m)} \cdot \frac{(T_0^2 p^2 + 1)}{(T_{12}^2 p^2 + 1)} \quad (13.13)$$

Із ПФ (13.11) знаходимо реакцію кутових координат вантажу на стрибок зусилля  $F_x(t) = F_{0x} \cdot 1(t)$  при нульових початкових умовах:

$$\varphi(t) = -\frac{F_{0x}}{(M+m)g} \left( 1 - \cos \frac{t}{T_{12}} \right), \quad (13.14)$$

що відповідає коливанням двомасової системи з періодом  $T_{k12} = 2\pi T_{12}$  навко-

ло похилої осі  $\varphi_{mean} = \frac{F_0}{(M+m)g}$  від  $\varphi_{min} = 0$  до  $\varphi_{max} = -\frac{2F_{0x}}{(M+m)g}$ .

Шляхом диференціювання (13.14) знайдемо закон зміни кутової швидкості:

$$\omega(t) = \frac{d\varphi(t)}{dt} = \frac{F_{0x}}{(M+m)gT_{12}} \sin \frac{t}{T_{12}} = \frac{F_{0x}}{\sqrt{Lg(M+m)M}} \sin \frac{t}{T_{12}}, \quad (13.15)$$

звідкіля маємо

$$\omega_{max} = \frac{F_{0x}}{(M+m)gT_{12}} = \frac{\varphi_{max}}{2T_{12}}. \quad (13.16)$$

З передавальної функції (13.15) 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 (13.16)

$$a(t) = \frac{F_{0x}}{(M+m)} \left[ 1 + \frac{m}{M} \cos \frac{t}{T_{12}} \right]. \quad (13.17),$$

З рівняння (13.17) визначаємо

$$a_{\max} = \frac{F_{0x}}{M}, \quad a_{\min} = \frac{F_{0x}}{M+m} \cdot \frac{M-m}{M} = a_{\max} \frac{M-m}{M+m}. \quad (13.18)$$

Отже, лінійне прискорення коливається навколо середнього значення

$$a_0 = (a_{\max} + a_{\min})/2 = F_{0x}/(M+m) \quad (13.19)$$

з амплітудою

$$A_a = \frac{a_{\max} - a_{\min}}{2} = \frac{F_{0x}m}{2M(M+m)}. \quad (13.20)$$

Розраховуючи характерні параметри графіків рис. 13.5 за наведеними вище формулами, отримуємо такі результати:  $T_{12} = 0,82$  с,  $T_{k12} = 5,18$  с,  $\varphi_{\max} = 5,8^\circ$ ,  $\omega_{\max} = 3,5$  град/с,  $a_{\max} = 0,75$  м/с<sup>2</sup>,  $a_{\min} = 0,25$  м/с<sup>2</sup>.

Результати розрахунків співпадають з параметрами перехідних процесів, що свідчить про адекватність розроблених моделей.

Як бачимо, вантаж з великою масою не тільки розгойдується сам, але й розгойдує візок.

Закінчення розгону зазвичай відбувається при майже стрибкоподібному скиданні тягового зусилля в 0. При цьому подальший вигляд перехідних процесів залежить від того в який саме момент часу здійснюється зміна зусилля. Наприклад, поведінка кута відхилення канату з вантажем від вертикалі визначається рівнянням

$$\varphi(t) = \varphi_0 \cos \frac{t}{T_{12}} + \omega_0 T_{12} \sin \frac{t}{T_{12}}, \quad (13.21)$$

з якого видно, що амплітуда гармонічних коливань в усталеному режимі залежить від початкових значень координат в момент зміни зусилля.

На рис. 13.6, 13.7 показані перехідні процеси в досліджуваній системі під дією зусилля, що стрибкоподібно змінюється з умов забезпечення переміщення одномасової механічної системи за трапецоїдальною тахограмою з усталеною швидкістю  $v_0 = 1,8$  м/с і усталеним прискоренням  $a_0 = 0,5$  м/с<sup>2</sup>.

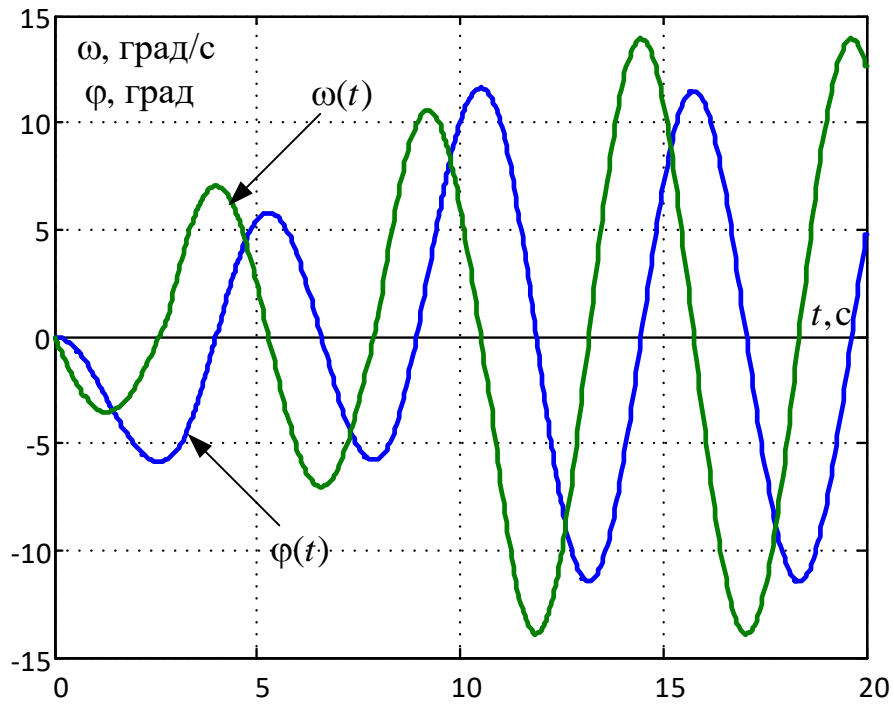


Рис. 13.6. Коливання вантажу при переміщенні візка з стрибкоподібною зміною тягового зусилля при  $t_p = t_r = t_{p0}$

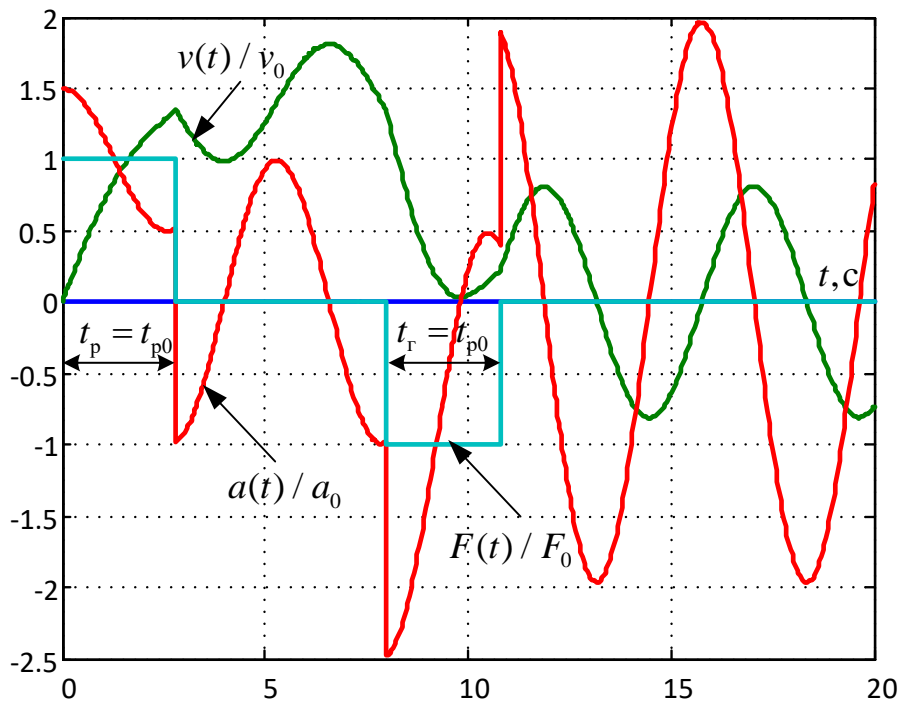


Рис. 13.7. Перехідні процеси зміни тягового зусилля, прискорення та швидкості візка при  $t_p = t_r = t_{p0}$

Час дії зусилля, що визначає час розгону і гальмування візка в останньому експерименті складає  $t_{p0} = v_0 / a_0 = 3,6$  с.

Як видно з наведених рисунків, елементи механічної системи в розглянутому режимі здійснюють незагасаючі коливання. З аналізу рівняння (13.21)

та графіків рис. 13.7 впливає, що позбутися цих коливань можна, якщо зробити час перемикання зусилля кратним періоду коливань двомасової системи. Цей висновок підтверджується графіками перехідних процесів, представлених на рис. 13.8, 13.9

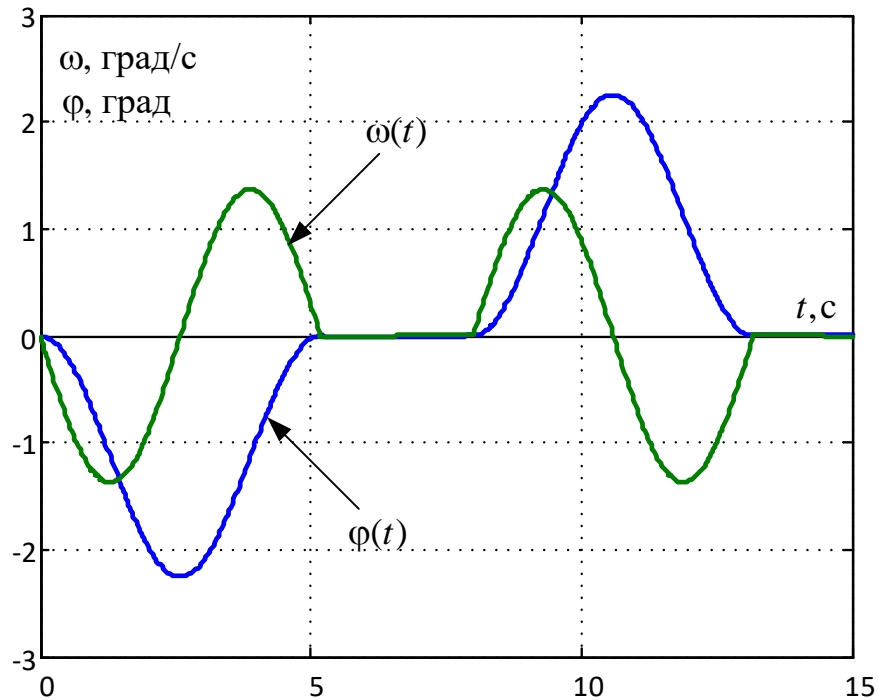


Рис. 13.8. Коливання вантажу при переміщенні візка з стрибкоподібною зміною тягового зусилля при  $t_p = t_r = T_{k12}$

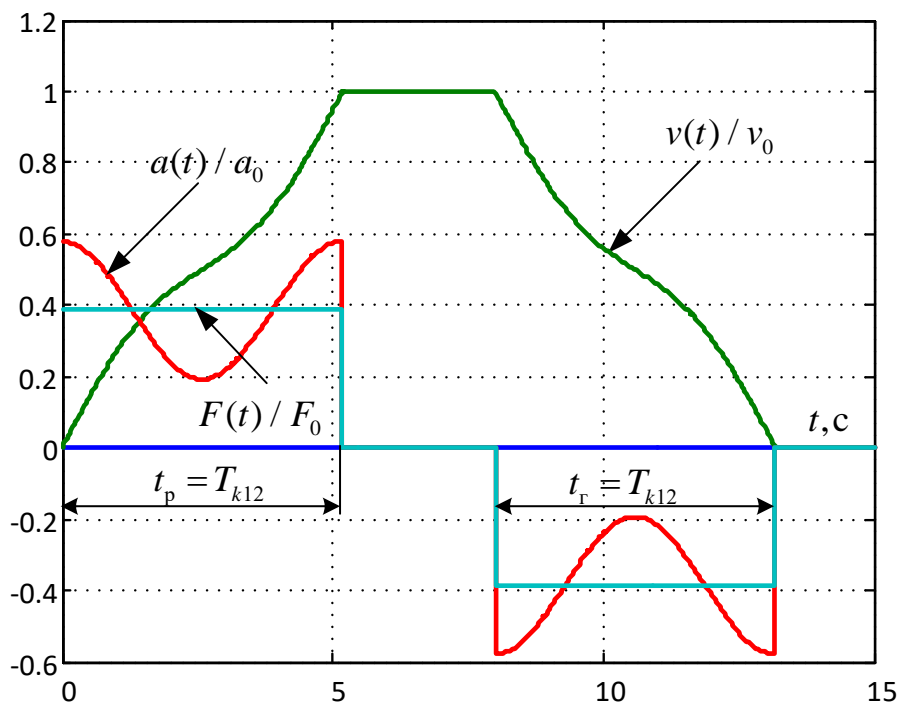


Рис. 13.9. Перехідні процеси зміни тягового зусилля, прискорення та швидкості візка при  $t_p = t_r = T_{k12}$

В останньому випадку система здійснює тільки одне коливання при розгоні і одне при гальмуванні.

### 13.2 Завдання

Виконати математичне моделювання кранової механічної системи візок-монтаж з даними, наведеними в табл. 13.1 в 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 двома-сової системи візок-вантаж.

Таблиця 15.1

№ вар.	$m$ , кг	$M$ , кг	$L$ , м	$v_0$ , м/с	$a_0$ , м/с <sup>2</sup>
1,13	4000	2000	12	0.5	0.5
2,14	4000	3000	15	1	0.5
3,15	4000	4000	10	0.75	0.4
4,16	1200	600	18	1	0.8
5,17	1800	1000	9	1	1
6,18	2000	800	13	0.6	0.5
7,19	3200	2000	8	0.8	0.6
8,20	1000	700	20	0.8	0.8
9,21	2500	800	7	2	2
10,22	3000	1500	11	2	1.6
11,23	3000	2000	15	1.8	1.6
12,24	2400	1000	14	1.5	0.9

### 13.2 Методичні рекомендації

1. Спочатку розрахуйте параметри характерних точок перехідних процесів з аналітичних виразів, наведених у попередньому підрозділі.

2. Користуйтеся розрахованими параметрами для вибору часу моделювання та моментів зміни тягового зусилля.

### Лабораторна робота №14

## ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ ЦИФРОВИХ ІНТЕГРАТОРІВ

*Мета роботи:* зрозуміти різницю між аналоговим та цифровим інтеграторами, процес складання різницевого рівняння цифрових інтеграторів першого порядку, отримання дискретних передавальних функцій цифрових інтеграторів за їх різницевоими рівняннями.

### 14.1 Загальні поняття про дискретну інформацію

Для *дискретної (цифрової) інформації* характерні такі властивості: квантування за часом; квантування за рівнем; запізнення; екстраполяція.

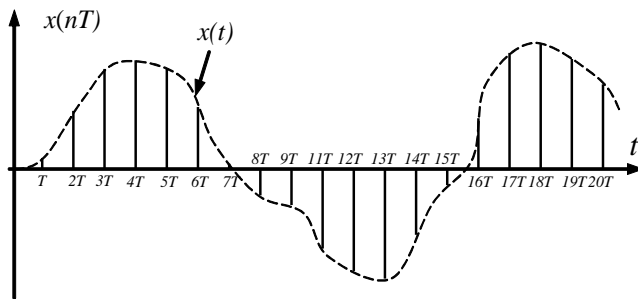


Рис. 14.1. Квантування за часом

#### Квантування за часом

здійснюється шляхом фіксації неперервного вхідного сигналу у дискретні моменти часу  $t_i = nT$ ,  $n = 1, 2, \dots$ , як це показано на рис. 14.1. Як бачимо, вихідний сигнал елемента, що здійснює

таке квантування, має характер послідовності імпульсів різної амплітуди, але постійної частоти. Величина  $T$ , що визначає час між двома сусідніми імпульсами при такому квантуванні, називається *тактом квантування*, або *періодом дискретності*, або *періодом переривання* (в англійській літературі цю величину називають *Sample Time* і позначають  $T_s$ ). Дискретна функція, отримана в результаті такого перетворення називається *решітчастою функцією*. Квантування за часом здійснюється так званими *імпульсними елементами*. Зображення ідеального імпульсного елемента на структурних схемах подано на рис. 14.2.

$$\underbrace{x(t)} \quad \underbrace{T} \quad \underbrace{x(nT)}$$

Рис. 14.2. Структурна схема ідеального імпульсного елемента

Величина періоду квантування залежить від тактової частоти цифрового пристрою та від обмежень, що накладаються на нього датчиками.

**Квантування за рівнем** формується шляхом фіксації неперервного сигналу на наперед визначених (фіксованих) рівнях у довільні моменти часу. Тобто, неперервний сигнал довільної форми замінюється послідовною низкою фіксованих дискретних значень відповідно до статичної характеристики перетворювача неперервного сигналу у дискретний. Рис. 14.3 пояснює процес квантування за рівнем за допомогою перетворювача з багатопозиційною релейною статичною характеристикою.

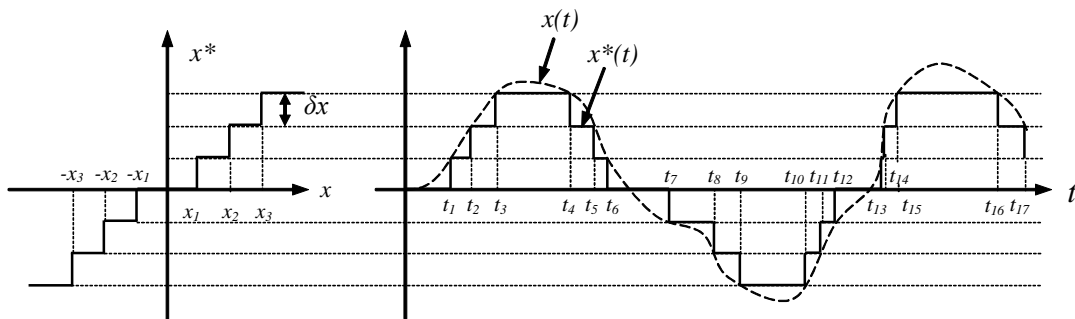


Рис. 13.3. Квантування неперервного сигналу за рівнем

З рис. 14.3 видно, що в результаті квантування за рівнем неперервного сигналу  $x(t)$  довільної форми, отримуємо неперервний сигнал  $x^*(t)$ , що має ступінчастий характер і дорівнює вхідному  $x(t)$  лише в деякі, наперед невідомі моменти часу  $t_0, t_1, \dots$ . Зміна величини вихідного сигналу  $x^*(t)$  здійснюється лише тоді, коли вхідний сигнал  $x(t)$  перетинає якийсь з рівнів релейної характеристики  $x_1, x_2, \dots$ . Зазвичай різниці між двома сусідніми рівнями квантованого сигналу  $x^*(t)$  однакові ( $x_i^* - x_{i-1}^* = \delta x = \text{const}$ ). Параметр  $\delta x$  називають **дискретою за рівнем**. Її величина визначається або кількістю розрядів у слові цифрового або перетворюючого пристрою (при роботі їх в арифметиці з фіксованою крапкою), а саме, ціною одного розряду, або властивостями датчиків вимірюваних сигналів. При роботі цифрових пристроїв у арифметиці з плаваючою крапкою або при використанні багаторозрядних пристроїв ефектом квантування за рівнем звичайно нехтують.

При моделюванні у *Simulink* релейне квантування здійснює блок *Quantizer* бібліотеки *Discontinuities*, як це показано на рис. 14.4.

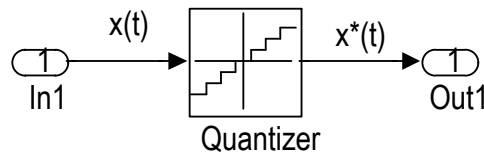


Рис. 14.4. *Simulink*-модель квантування неперервного сигналу за рівнем

Його статична характеристика має зону нечутливості  $[-\delta x/2, \delta x/2]$ . Внаслідок цього він здійснює округлювання чисел до величин, кратних дискреті квантування, заданої параметром *Quantization interval*, за правилами округлювання. Це продемонстровано графіками рис. 14.5 релейного перетворення синусоїди з одиничною амплітудою блоком *Quantizer* з  $\delta x = 0.2$ .

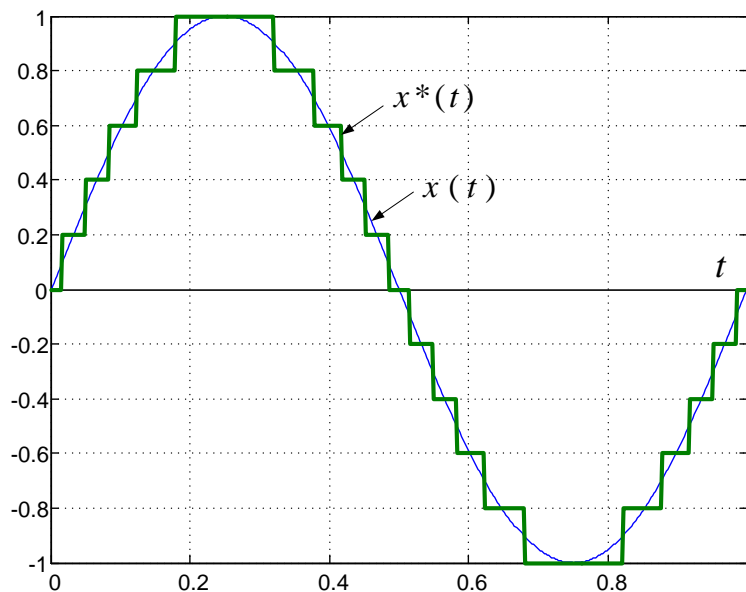


Рис. 14.5. Релейне квантування неперервного сигналу *Simulink*-блоком *Quantizer*

Ефект наявності чистого, або транспортного *запізнювання (Transport Delay)* у цифрових системах зумовлений часом, необхідним цифровому пристрою для виконання алгоритму керування, тобто тривалість цієї затримки визначається швидкодією обчислювального пристрою.

В цифрових системах наявність запізнення приводить до того, що дискретні сигнали змінюють свої значення не в моменти часу, кратні періоду



дискретизації  $nT$ , а в моменти  $nT + \tau$ , тобто функція  $x(nT)$  перетворюється у функцію  $x(nT - \tau)$ . При простих алгоритмах цим явищем нехтують, тобто вважають  $\tau = 0$ .

**Екстраполяцією** називають процес визначення дискретного сигналу у проміжках часу між моментами формування ідеальних імпульсів у вигляді решітчастої функції. У решітчастої функції сигнал між імпульсами має нульове значення. Але, якщо алгоритм керування реалізується будь-яким цифровим пристроєм, то значення сигналу, розраховане у деякий дискретний момент часу, утримується (фіксується) в оперативній пам'яті впродовж усього періоду переривання. Оскільки утримуваний на періоді сигнал є незмінним, тобто таким, що описується степеневим поліномом нульового порядку, то цей спосіб екстраполяції називають фіксацією або екстраполяцією нульового порядку (англ. **Zero Order Hold**, скорочено **ZOH**). Ще раз підкреслимо, що цей спосіб екстраполяції є природним для цифрових пристроїв з запам'ятовуванням. Примусово можна змінити тип екстраполяції, наприклад, застосувати поліноміальну екстраполяцію першого порядку (англ. **First Order Hold**, скорочено **FOH**).

## 14.2 Форми математичного опису лінійних імпульсних систем

Лінійні стаціонарні імпульсні динамічні системи з періодом переривання  $T$  описуються у просторі стану різницевиими матричними рівняннями

$$\mathbf{x}(kT) = \mathbf{A}\mathbf{x}(kT - T) + \mathbf{B}\mathbf{u}(kT - T) \quad (14.1)$$

з початковими умовами

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (14.2)$$

та лінійним матричним рівнянням виходу

$$\mathbf{y}(kT) = \mathbf{C}\mathbf{x}(kT) + \mathbf{D}\mathbf{u}(kT). \quad (14.3)$$

Для переходу до операторної форми сигнали, що запізнюються у часу на період  $T$  помножують на  $z^{-1}$ , де

$$z = \exp(Ts) \quad (14.4)$$

– дискретний оператор. В результаті такого перетворення рівняння (14.1) та (14.2) набувають вигляду:

$$z\mathbf{x}(z) = \mathbf{A}\mathbf{x}(z) + \mathbf{B}\mathbf{u}(z), \quad (14.5)$$

$$\mathbf{y}(z) = \mathbf{C}\mathbf{x}(z) + \mathbf{D}\mathbf{u}(z). \quad (14.6)$$

Формальна аналогія математичного опису неперервних та дискретних систем в операторній формі простору часу дає можливість використовувати для їх перетворення, аналізу та синтезу однаковий математичний апарат.

Зокрема, перехід від рівнянь у просторі стану до дискретних матричних передавальних функцій здійснюється за формулами, аналогічними формулам (4.9) і (4.10) для неперервних систем:

$$\mathbf{W}_x(z) = \frac{\mathbf{x}(z)}{\mathbf{u}(z)} = (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \frac{\text{Adj}(z\mathbf{I} - \mathbf{A})}{\det(z\mathbf{I} - \mathbf{A})}\mathbf{B}, \quad (14.7)$$

$$\mathbf{W}_y(z) = \frac{\mathbf{y}(z)}{\mathbf{u}(z)} = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \mathbf{C} \frac{\text{Adj}(z\mathbf{I} - \mathbf{A})}{\det(z\mathbf{I} - \mathbf{A})}\mathbf{B} + \mathbf{D}. \quad (14.8)$$

Складові дискретних матричних ПФ записують у поліноміальній формі відносно оператора  $z$

$$W_{ij}(z) = \frac{y_i(z)}{u_j(z)} = \frac{H_{ij}(z)}{G(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_1 z + b_0}{a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0}, \quad m \leq n \quad (14.9)$$

або відносно зворотної до нього величини  $z^{-1}$

$$W_{ij}(z^{-1}) = \frac{y_i(z)}{u_j(z)} = \frac{b_0 z^{-m} + b_1 z^{-(m-1)} + \dots + b_{m-1} z^{-1} + b_m z^{-(n-m)}}{a_0 z^{-n} + a_1 z^{-(n-1)} + \dots + a_{n-1} z^{-1} + a_n}. \quad (14.10)$$

Знаменник дискретної ПФ також зветься характеристичним поліномом а чисельник – поліномом впливу

Їх корені утворюють вектор полюсів  $\mathbf{z} = [z_1 \quad z_2 \quad \dots \quad z_n]$  та вектор нулів  $\mathbf{z}_z = [z_{z1} \quad z_{z2} \quad \dots \quad z_{zm}]$ .

Скалярній передавальній функції (14.10) відповідає різницеве рівняння

$$a_n y_i(kT) + a_{n-1} y_i(kT - T) + \dots + a_1 y_i(kT - (n-1)T) + a_0 y_i(kT - nT) =$$

$$= b_m u_j(kT) + b_{m-1} u_j(kT - T) + \dots + b_1 u_j(kT - (m-1)T) + b_0 u_j(kT - mT).$$

Використовуючи розкладення поліномів у чисельнику та знаменнику передавальної функції (14.9) на множники, отримуємо

$$W_{ij}(s) = K \frac{(z - z_{z1})(z - z_{z2}) \dots (z - z_{zm})}{(z - z_1)(z - z_2) \dots (z - z_n)}, \quad K = \frac{b_m}{a_n}. \quad (14.11)$$

### 14.3 Огляд блоків бібліотеки *Discrete*

У цю бібліотеку входять дискретний інтегратор, екстраполятори, ланки запізнювання на 1 або декілька періодів дискретності, дискретні динамічні ланки загального виду з різними способами їхнього математичного опису та ПД-регулятори.

Усі ці блоки мають у діалоговому вікні введення параметрів поле з ім'ям *Sample Time*. У цьому полі може вводитися одне значення, що інтерпретується як період дискретності  $T_s$ , або два значення, перше з яких сприймається як період дискретності  $T_s$ , а друге – як запізнення *offset*. У цьому випадку зміна вихідних сигналів ланок будуть здійснюватися в дискретні моменти часу.

Кожен дискретний блок має вбудований переривач (ідеальний імпульсний елемент) на вході і екстраполятор нульового порядку на виході, так що вхідні сигнали дискретних блоків змінюються тільки в моменти часу  $t_d$ , а вихідні сигнали між двома сусідніми перериваннями утримуються на постійному рівні.

### 14.4 Математичний опис найпростіших цифрових інтеграторів

Передавальна функція цифрового інтегратора залежить від методу чисельного інтегрування. В основу цих методів покладено геометричний сенс визначеного інтеграла, який дорівнює площі фігури, обмеженою кривою підінтегральної функції  $u(t)$ , віссю часу та вертикалями, що виходять з точок границь інтегрування  $t = t_0$  і  $t = t_k$ . Більшість із методів цифрового інтегрування полягає у розбиванні фігури, площу якої треба підрахувати на декілька більш простих фігур, площі яких обчислюються за простими формулами, з

подальшим сумуванням цих площ. Найчастіше інтервал інтегрування поділяють на декілька рівних частин, на кожній з яких виконують локальну інтеполяцію підінтегральної функції степеневими поліномами. В залежності від степені полінома  $n$  утворюються різні методи чисельного інтегрування: при  $n = 0$  – метод прямокутників, при  $n = 1$  – метод трапецій, при  $n = 2$  – метод параболічних трапецій, який зазвичай називають методом Сімпсона. Цей ряд можна продовжувати, але при дискретизації неперервних динамічних ланок підстановочними методами обмежуються заміною аналогових інтеграторів цифровими, синтезованими методами прямокутників та трапецій.

Тому в першу чергу виведемо передавальні функції цифрових інтеграторів за допомогою перелічених вище методів.

На рис. 14.6 наведено ілюстрації до досліджуваних методів чисельного інтегрування.

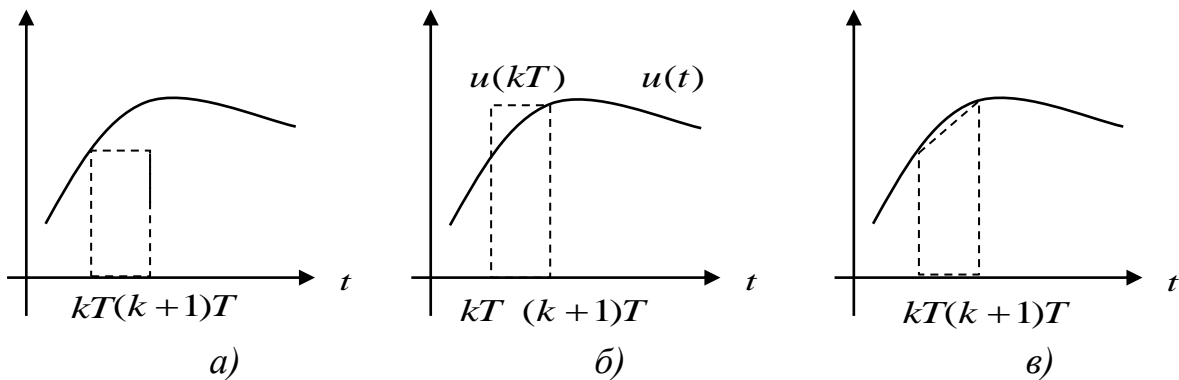


Рис. 12.6. Ілюстрація до методів чисельного інтегрування:  
 а) пряма апроксимація Ейлера; б) зворотна апроксимація Ейлера;  
 в) метод трапецій

Один крок чисельного інтегрування *методом прямої апроксимації Ейлера (Forward Euler)*, який ще називають *методом лівобічних прямокутників*, згідно з рис. 14.6а, описується різницеvim рівнянням

$$y_{FE}(kT + T) = y_{FE}(kT) + Tu(kT), \quad (14.12)$$

яке в операторній формі набуває вигляду

$$zy_{FE}(z) = y_{FE}(z) + Tu(z). \quad (14.13)$$

Після перегрупування (14.13) легко визначаємо передавальну функцію інтегратора:

$$W_{FE}(z) = \frac{y_{FE}(z)}{u(z)} = \frac{T}{z-1} = \frac{Tz^{-1}}{1-z^{-1}}. \quad (14.14)$$

За аналогією синтезуємо цифровий інтегратор *методом зворотної апроксимації Ейлера (Backward Euler)*, який ще називають *методом правобічних прямокутників*, згідно з рис. 14.6б

$$y_{BE}(kT+T) = y_{BE}(kT) + Tu(kT+T), \quad (14.15)$$

$$zy_{BE}(z) = y_{BE}(z) + zTu(z), \quad (14.16)$$

$$W_{BE}(z) = \frac{y_{BE}(z)}{u(z)} = \frac{Tz}{z-1} = \frac{T}{1-z^{-1}}, \quad (14.17)$$

та цифровий інтегратор *методом трапецій (Trapezoidal)*, який іноді називають *методом середніх прямокутників*, згідно з рис. 14.6в

$$y_{tr}(kT+T) = y_{tr}(kT) + T \frac{u(kT) + u(kT+T)}{2}, \quad (14.18)$$

$$zy_{tr}(z) = y_{tr}(z) + T \frac{u(z) + zu(z)}{2}, \quad (14.19)$$

$$W_{tr}(z) = \frac{y_{tr}(z)}{u(z)} = \frac{T}{2} \cdot \frac{z+1}{z-1} = \frac{T}{2} \cdot \frac{1+z^{-1}}{1-z^{-1}}. \quad (14.20)$$

З порівняння передавальних функцій (14.14), (14.17) та (14.20) випливає:

$$y_{BE}(z) = zy_{FE}(z), \quad y_{tr}(z) = \frac{y_{BE}(z) + zy_{FE}(z)}{2}, \quad (14.21)$$

тобто вихідний сигнал інтегратора (14.17) випереджає вихідний сигнал інтегратора (14.14) на один такт квантування, а вихідний сигнал інтегратора (14.20) складається із полусуми вихідних сигналів інтеграторів (14.17) і (14.14).

Взаємозв'язок між розглянутими методами чисельного інтегрування можна продемонструвати за допомогою деталізованої структурної схеми рис. 14.7, складеної за операторними рівняннями (14.13), (14.16) та (14.19).

Як видно з рис. 14.7, основу цифрових інтеграторів складає ланка запізнення на період дискретності  $1/z$ , замкнена додатним зворотним зв'яз-

ком, що здійснює накопичення сум площин прямокутників  $T \cdot u(kT)$  на кожному кроці квантування за часом. Один від одного інтегратори з різними алгоритмами відрізняються точкою знімання вихідного сигналу. Варто звернути увагу на те, що інтегратори з дискретними ПФ (14.17) та (14.20) мають прямий зв'язок входу з виходом. Тому при замиканні їхнього виходу на вхід утвориться алгебраїчна петля. Така ситуація може виникнути при цифровій реалізації задатчика інтенсивності.

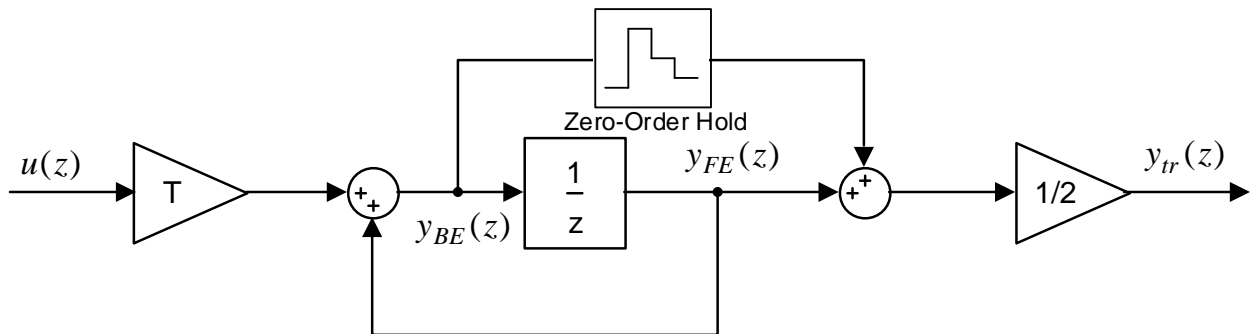


Рис. 14.7. Розгорнута модель дискретного інтегратора з різними алгоритмами чисельного інтегрування

Цікаво порівняти розглянуті вище цифрові інтегратори з інтеграторами, отриманими методами імпульсно-інваріантного, ступінчато-інваріантного та лінійно-інваріантного  $Z$ -перетворень:

$$W_{imp}(z) = T \cdot Z \left\{ \frac{1}{p} \right\} = \frac{Tz}{z-1} = W_{BE}(z),$$

$$W_{zoh}(z) = \frac{z-1}{z} \cdot Z \left\{ \frac{1}{p^2} \right\} = \frac{z-1}{z} \cdot \frac{Tz}{(z-1)^2} = \frac{T}{z-1} = W_{FE}(z),$$

$$W_{foh}(z) = \frac{(z-1)^2}{Tz} \cdot Z \left\{ \frac{1}{p^3} \right\} = \frac{(z-1)^2}{Tz} \cdot \frac{T^2 z(z+1)}{2(z-1)^3} = \frac{T}{2} \cdot \frac{z+1}{z-1} = W_{tr}(z).$$

Отже, при дискретизації неперервного інтегратора переліченими методами  $Z$ -перетворень можна отримати ті ж самі цифрові інтегратори першого порядку, синтезовані за допомогою методів чисельного інтегрування, а саме: імпульсно-інваріантне  $Z$ -перетворення аналогового інтегратора збігається з методом *Backward Euler*, ступінчато-інваріантне – з методом *Forward Euler* і лінійно-інваріантне – з методом *Trapezoidal*.

В *Simulink* усі розглянуті вище алгоритми чисельного інтегрування реалізує блок *Discrete-Time Integrator* (Дискретний Інтегратор).

Вікно параметрів цього блоку показано на рис. 14.8.

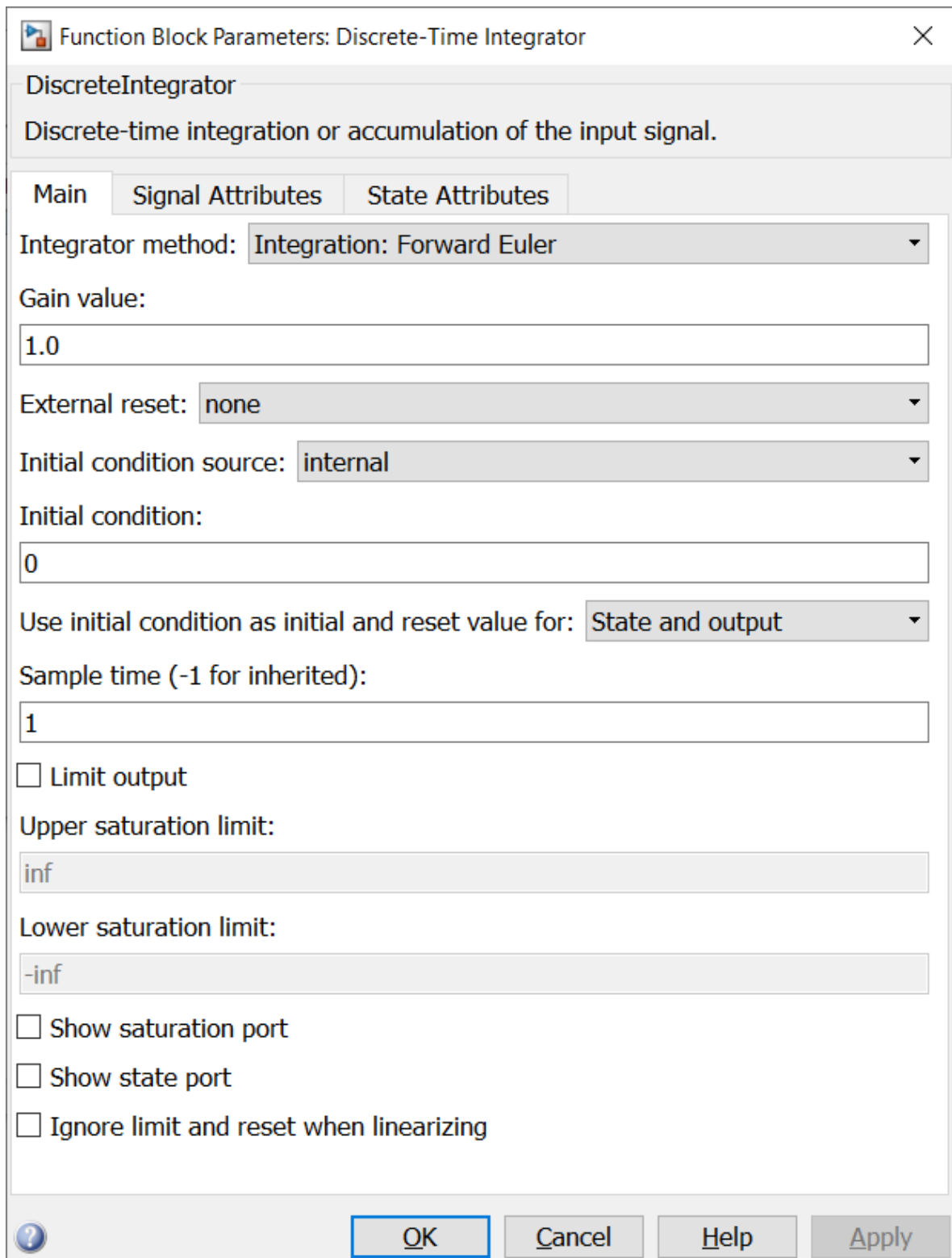


Рис. 14.8. Вікно визначення параметрів та режимів роботи блоку *Discrete-Time Integrator*

Він може не тільки інтегрувати, а ще й накопичувати (акумулявати) вхідний сигнал. Вибір режиму роботи обирається із *pop-up* меню параметру *Integrator method* (див. рис. 14.9).

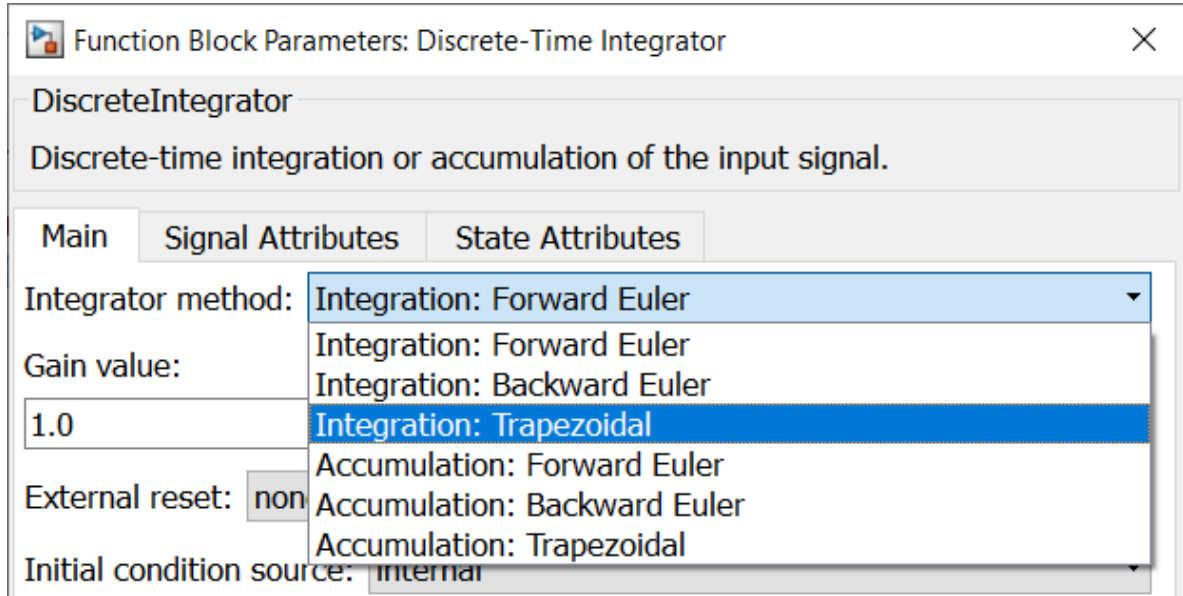


Рис. 14.9. *Pop-up* меню визначення параметру *Integrator method* блоку *Discrete-Time Integrator*

**В режимі акумуляції** інтегратор працює з періодом переривання  $T = 1$ . Отже різниці рівняння при застосуванні різних методів цифрового інтегрування в цьому режимі мають вигляд:

$$y_{AFE}[k+1] = y_{AFE}[k] + u[k], \quad (14.22)$$

$$y_{ABE}[k+1] = y_{ABE}[k] + u[k+1], \quad (14.23)$$

$$y_{Atr}[k+1] = y_{Atr}[k] + \frac{u[k] + u[k+1]}{2}. \quad (14.24)$$

Відповідно змінюються і передавальні функції:

$$W_{AFE}(z) = \frac{1}{z-1} \quad W_{ABE}(z) = \frac{z}{z-1} \quad W_{Atr}(z) = \frac{1}{2} \cdot \frac{z+1}{z-1}. \quad (14.25)$$

При зміні режиму роботи і методу інтегрування автоматично змінюється іконка блоку, в якій відображається його дискретна ПФ.

Додатковим параметром блоку є коефіцієнт підсилення (*Gain Value*). Але зміна цього параметру ніяк не впливає на зображення іконки.



Як і аналоговий інтегратор, дискретний може працювати з внутрішніми (*Internal*) та зовнішніми (*External*) початковими умовами, що обираються з *pop-up* меню параметру *Initial Condition Source*, в режимі обмеження координат (*Limit Output*) і в режимі скидання вихідного сигналу (*External Reset*) за деяких обраних користувачем умов.

Застосування різних методів ЧІ можна продемонструвати за допомогою розгорнутої моделі дискретного інтегратора, зображеної на рис. 14.10, і перехідних функцій, приведених на рис. 14.11. На останньому рисунку для порівняння приведена також перехідна функція неперервного інтегратора.

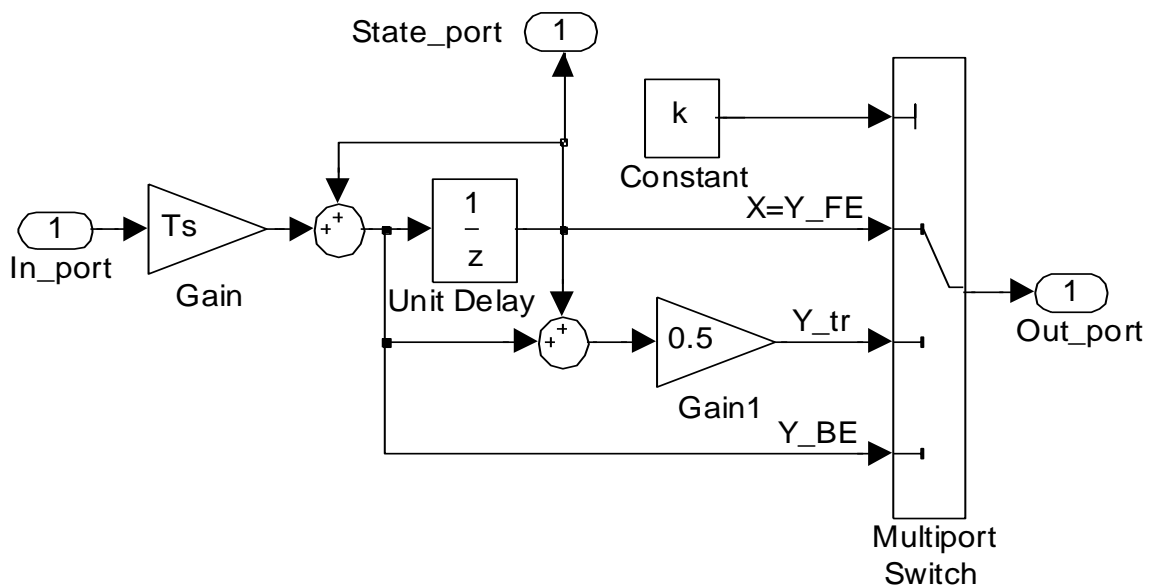
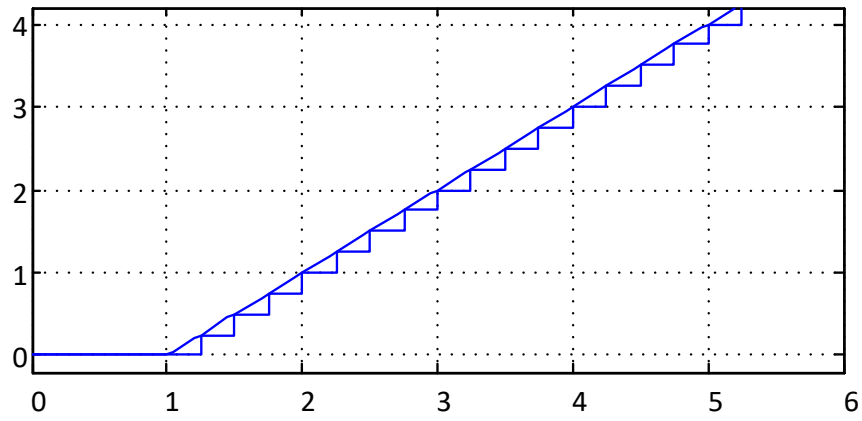


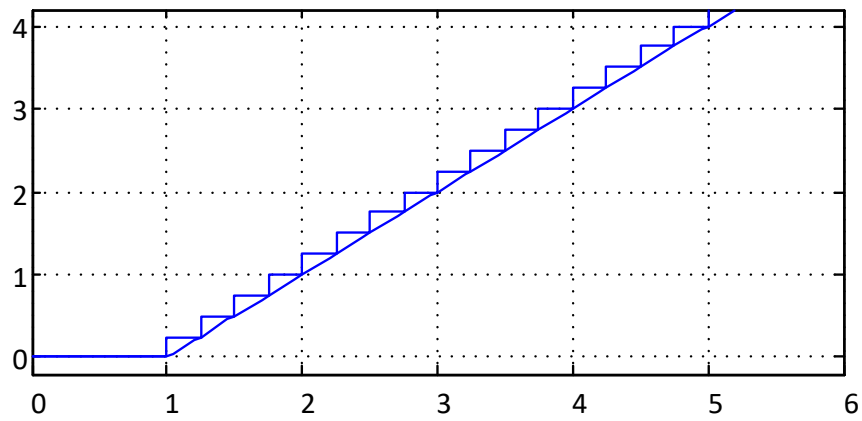
Рис. 14.10. Розгорнута модель дискретного інтегратора з різними алгоритмами чисельного інтегрування

Рис. 14.11 підтверджує, що вихідний сигнал цифрового інтегратора, отриманого методом , на 1 період дискретності запізнюється відносно сигналу, отриманого методом , а сигнал, отриманий методом , уявляє собою середнє арифметичне від двох попередніх. Але розташування вихідних сигналів цифрових інтеграторів відносно вихідного сигналу неперервного інтегратора змінюється в залежності від вигляду вхідного сигналу.

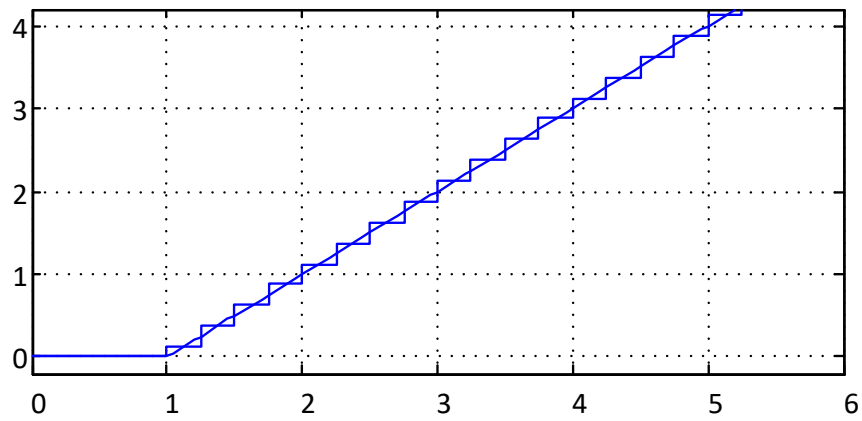
Для прикладу на рис. 14.12 та рис. 14.13 наведені реакції досліджуваних інтеграторів на лінійний та синусоїдальний вхідні сигнали.



a)

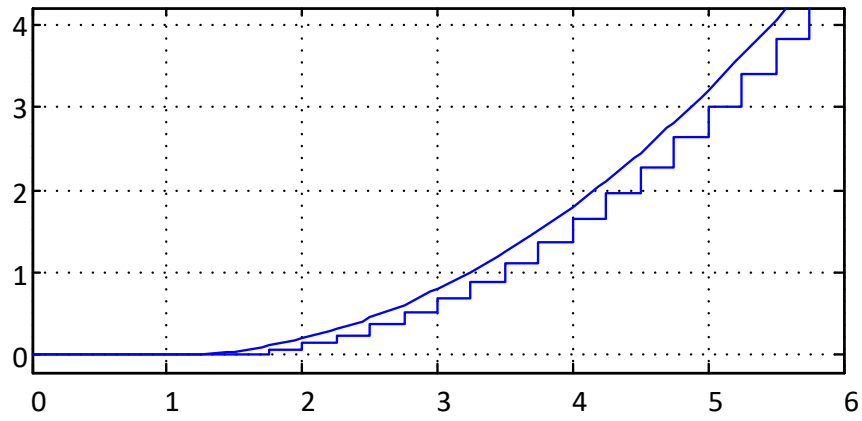


б)

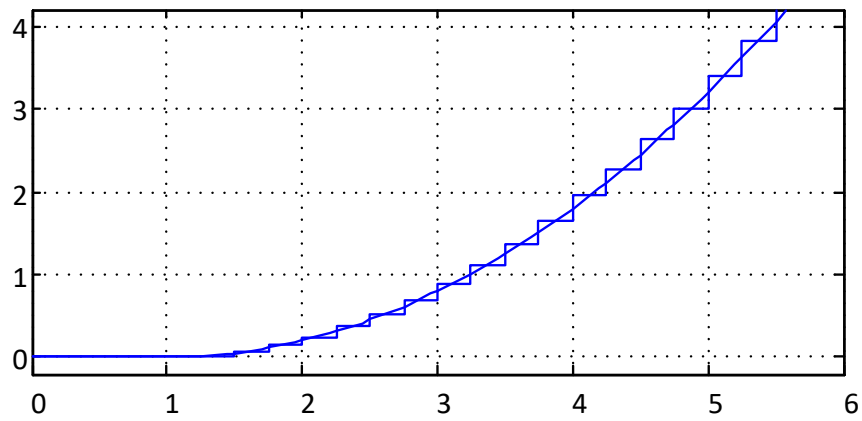


в)

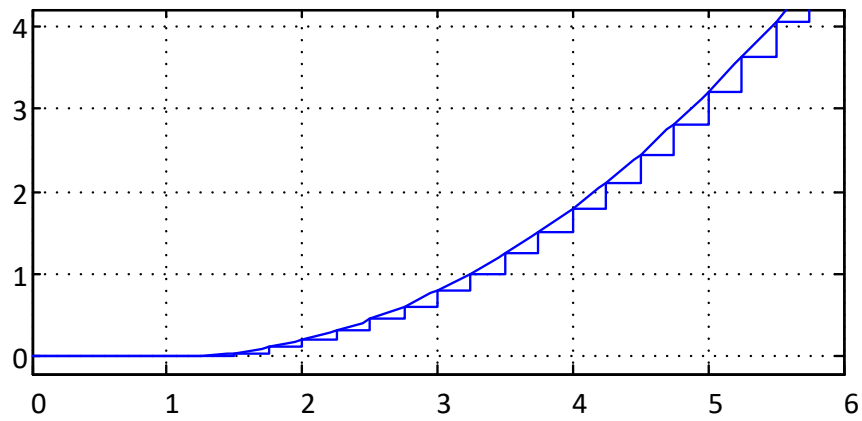
Рис. 14.11. Реакції інтеграторів на стрибкоподібний сигнал:  
а) *Forward Euler*, б) *Backward Euler*, в) *Trapezoidal*



a)

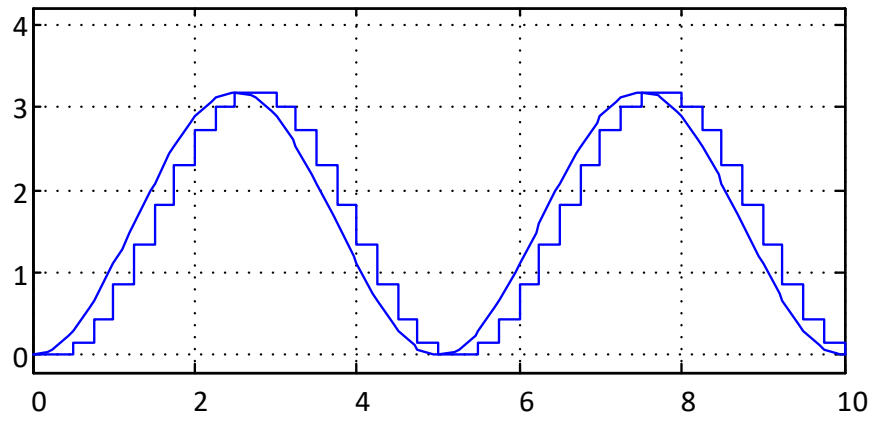


б)

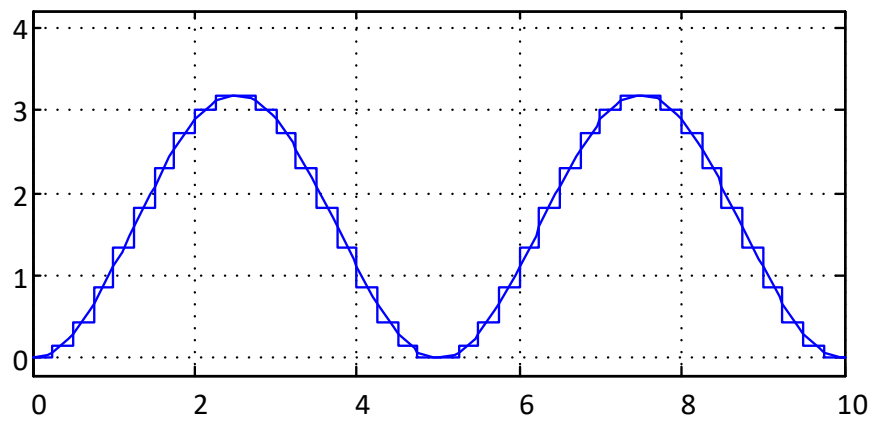


в)

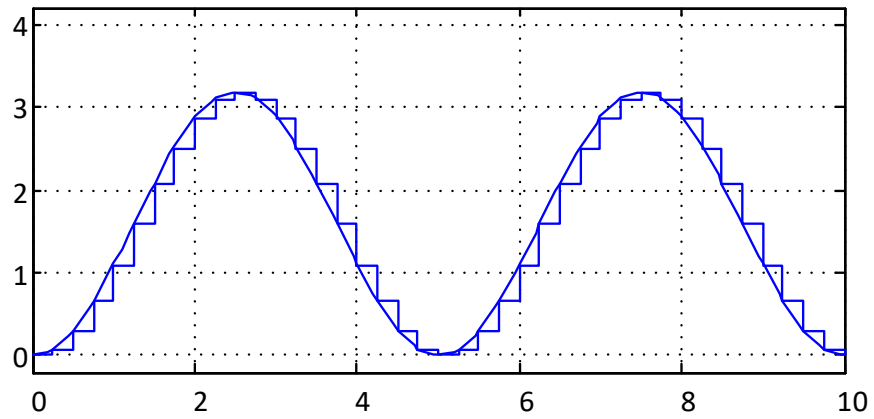
Рис. 14.12. Реакції інтеграторів на лінійний сигнал:  
а) *Forward Euler*, б) *Backward Euler*, в) *Trapeziodal*



a)



б)



в)

Рис. 14.13. Реакції інтеграторів на синусоїду:  
а) *Forward Euler*, б) *Backward Euler*, в) *Trapezoidal*

### 14.5 Завдання

1. Отримати перехідні функції аналогового та цифрових інтеграторів з алгоритмами числового інтегрування *Forward Euler*, *Backward Euler* та *Trapezoidal* для двох значень періоду дискретності.
2. Отримати реакцію на лінійний вплив аналогового та цифрових інтеграторів з такими ж алгоритмами ЧІ для двох значень періоду дискретності.
3. Отримати реакцію на синусоїдальний вплив аналогового та цифрових інтеграторів з такими ж алгоритмами ЧІ для двох значень періоду дискретності.
4. Скласти деталізовану модель цифрового інтегратора з такими ж алгоритмами ЧІ та повторити дослідження для одного із пунктів. Впевнитися у тому, що результати моделювання бібліотечних інтеграторів повністю збігаються з результатами моделювання їх деталізованих схем.

### 14.6 Методичні вказівки

1. Для виконання сформульованих вище завдань зберіть модель, що складається із джерел стрибкоподібного *Step*, лінійного *Ramp* та синусоїдального *Sine Wave* сигналів, приєднаних до неперервного інтегратора *Integrator* і 3-х дискретних інтеграторів *Discrete Time Integrator* з алгоритмами числового інтегрування *Forward Euler*, *Backward Euler* та *Trapezoidal* через багатоканальний ключ *Multiport Switch1*. Вихідні сигнали дискретних інтеграторів приєднайте до ключа *Multiport Switch2*. Організуйте візуалізацію результатів так, щоб в одній системі координат відображалися результати аналогового та цифрового інтегрування. Можливий варіант моделі показано на рис. 14.11-14.13.
2. Розробіть програму, яка виконує симуляцію моделі рис.14.14 для кожного із 3-х вхідних сигналів (зовнішній цикл з варіацією змінної *kinp*) і для кожного цифрового інтегратора (внутрішній цикл з варіацією змінної *kout*) і виводить графіки за зразком, що зображений на рис. 14.2-14.4.

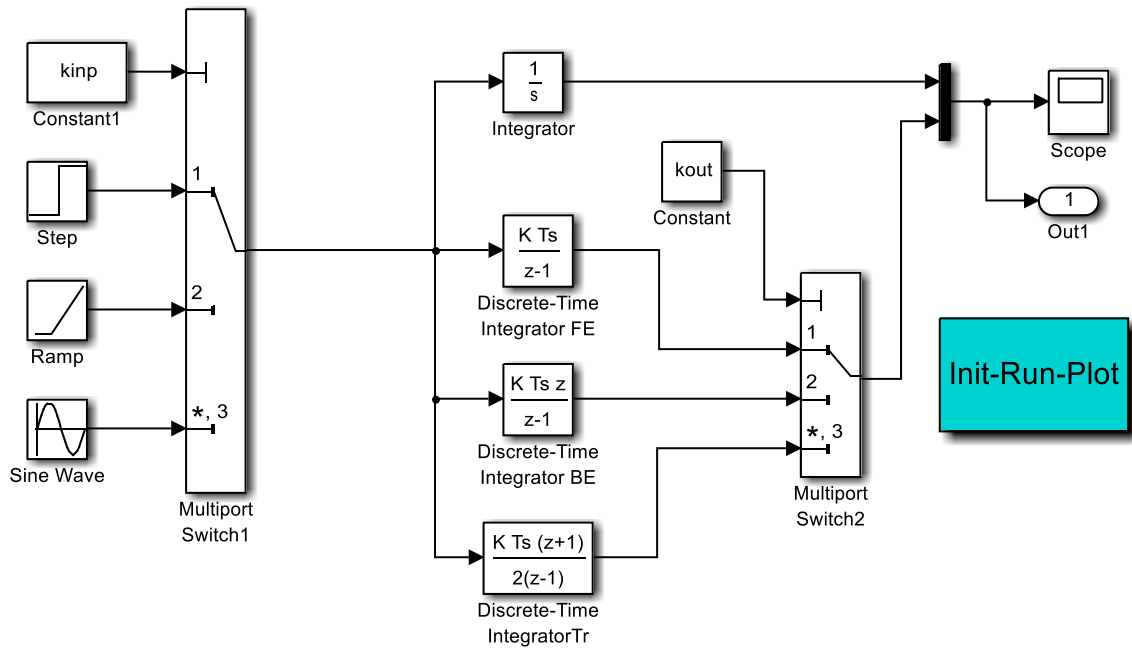


Рис. 14.14. Модель для дослідження властивостей цифрових інтеграторів

3. Проаналізуйте отримані графіки та зробіть висновки щодо точності різних алгоритмів числового інтегрування при різних типах вхідних сигналів.

4. В моделі рис. 14.14 замініть цифрові інтегратори з бібліотеки *Discrete* їх деталізованою структурою, складеною за різницевими рівняннями, як це показано на рис. 14.15, або приєднайте деталізовану модель до блоку *Multiport Switch1* паралельно до бібліотечних інтеграторів.

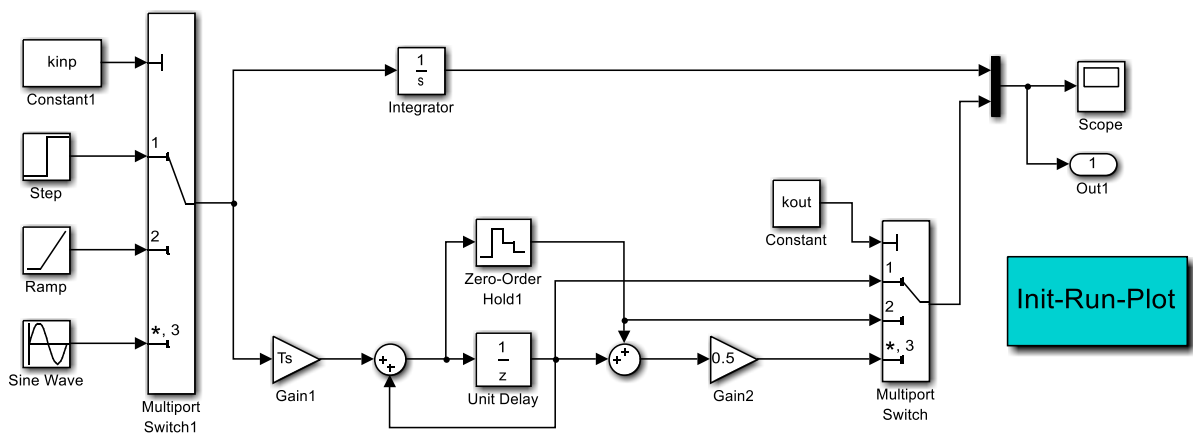


Рис. 14.15. Модель для дослідження властивостей деталізованих моделей цифрових інтеграторів

5. Повторіть один із експериментів, виконаних при дослідженні бібліотечних цифрових інтеграторів, для деталізованих моделей. Впевніться у тому, що моделі рис. 14.14 та 14.15 еквівалентні.

*Лабораторна робота №15***ДОСЛІДЖЕННЯ МЕТОДІВ ДИСКРЕТИЗАЦІЇ НЕПЕРЕРВНИХ  
ДИНАМІЧНИХ ОБ'ЄКТІВ**

*Мета роботи:* навчитися виконувати дискретизацію неперервних динамічних об'єктів точними та приблизними методами, зрозуміти різницю між ними та придбати навички вибору саме того з існуючих методів, що забезпечує найбільшу точність дискретної апроксимації в існуючих умовах.

**15.1 Постановка задачі. Класифікація методів дискретизації**

Як правило, в більшості електромеханічних систем автоматичного керування об'єкт керування має неперервний (аналоговий) характер. Однак, внаслідок відомих переваг цифрових коригувальних пристроїв (цифрових регуляторів) перед аналоговими, для керування неперервними об'єктами досить часто застосовуються саме такі регулятори.

Синтез дискретних пристроїв керування неперервними системами виконують одним з наступних шляхів:

- на основі неперервного об'єкта регулювання синтезують неперервні пристрої керування, а потім перетворюють їх у дискретну форму (метод неперервних моделей);
- будують дискретні моделі об'єкта регулювання і на їх основі синтезують дискретні пристрої керування.

Отже, в обох випадках треба вміти знаходити дискретні апроксимації аналогових передавальних функцій.

Сформулюємо задачу дослідження.

Розглянемо неперервну динамічну систему з вхідним сигналом  $u(t)$  і вихідним сигналом  $y(t)$ , що описується в області змінної Лапласа  $p$  передавальною функцією у поліноміальній формі (*Transfer Function Polynomial*):

$$W(p) = \frac{y(p)}{u(p)} = \frac{H_m(p)}{G_n(p)} = \frac{\beta_m p^m + \beta_{m-1} p^{m-1} + \dots + \beta_1 p + \beta_0}{p^n + \alpha_{n-1} p^{n-1} + \dots + \alpha_1 p + \alpha_0}, \quad m \leq n \quad (15.1)$$



В цій ПФ характеристичний поліном (*denominator*)  $G_n(s)$  є нормованим за коефіцієнтом при старшій степені оператора Лапласа.

Використовуючи розкладення поліномів у чисельнику (*numerator*) та знаменнику передавальної функції (15.1) на множники, отримуємо передавальну функцію у формі *Zero-Pole-Cain*:

$$W(p) = \frac{y(p)}{u(p)} = K \frac{(p - z_1)(p - z_2) \dots (p - z_m)}{(p - p_1)(p - p_2) \dots (p - p_n)}, \quad (15.2)$$

де

$\mathbf{Z} = [z_1 \quad z_2 \quad \dots \quad z_m]$  – вектор нулів (*Zeros*);

$\mathbf{P} = [p_1 \quad p_2 \quad \dots \quad p_n]$  – вектор полюсів (*Poles*);

$K = \beta_m$ .

У просторі стану (*State Space*) такий об'єкт описується матричними рівняннями

$$p\mathbf{X}(p) = \mathbf{A}\mathbf{X}(p) + \mathbf{B}u(p), \quad (15.3)$$

$$y(p) = \mathbf{C}\mathbf{X}(p) + Du(p), \quad (15.4)$$

де

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

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

$\mathbf{B}$  – вектор-стовбець входу розміром  $n \times 1$ ;

$\mathbf{C}$  – вектор-рядок виходу розміром  $1 \times n$ ;

$D$  – коефіцієнт обходу.

Задача полягає у визначенні при заданому періоді квантування  $T$  еквівалентної дискретної передавальної функції (ДПФ)

$$\begin{aligned} W(z) &= \frac{y(z)}{u(z)} = K_d \frac{(z - z_{d1})(z - z_{d2}) \dots (z - z_{dm_d})}{(z - p_{d1})(z - p_{d2}) \dots (z - p_{dn})} = \\ &= \frac{H_{m_d}(z)}{G_n(z)} = \frac{\beta_{m_d}^* z^{m_d} + \beta_{m_d-1}^* z^{m_d-1} + \dots + \beta_1^* z + \beta_0^*}{z^n + \alpha_{n-1}^* z^{n-1} + \dots + \alpha_1^* z + \alpha_0^*} \end{aligned} \quad (15.5)$$

або еквівалентних рівнянь у просторі станів

$$z\mathbf{X}(z) = \mathbf{A}_d\mathbf{X}(z) + \mathbf{B}_d u[z], \quad (15.6)$$

$$y(z) = \mathbf{C}_d\mathbf{X}(z) + D_d u(z), \quad (15.7)$$

тобто у побудові дискретної моделі неперервного об'єкту. Під еквівалентністю в даному випадку розуміють збіг реакцій неперервної системи та її дискретної моделі на будь-яку вхідну дію. Найчастіше під збіжністю реакцій розуміють, що  $y[k] = y(t_k)$  при  $u[k] = u(t_k)$ , де  $t_k = kT$ ,  $k$  – номер кроку квантування.

Зв'язок між рівняннями у просторі стану та передавальними функціями визначається виразами:

$$W(s) = \frac{y(s)}{u(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D = \mathbf{C} \frac{\text{Adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})} \mathbf{B} + D, \quad (15.8)$$

$$W(z) = \frac{y(z)}{u(z)} = \mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + D_d = \mathbf{C}_d \frac{\text{Adj}(z\mathbf{I} - \mathbf{A}_d)}{\det(z\mathbf{I} - \mathbf{A}_d)} \mathbf{B}_d + D_d \quad (15.9)$$

де

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

$\text{Adj}(\mathbf{X})$  – союзна матриця (матриця, складена із алгебраїчних доповнень матриці-аргументу);

$\det(\mathbf{X})$  – визначник матриці.

Поставлену задачу можна розв'язати такими способами [11, 12]:

- 1) за допомогою  $Z$ -перетворень;
- 2) заміною оператора аналогового інтегрування  $1/s$  одним з операторів цифрового інтегрування;
- 3) заміною нулів та полюсів на  $s$ -площині відповідними нулями та полюсами на  $Z$ -площині.

В загальному випадку поставлена задача не має точного розв'язку. Це пов'язано з тим, що при дискретизації вхідного сигналу втрачається інформація про його значення між вузлами квантування. Отже, вихід дискретної моделі від цих значень залежати не може, у той час як реакція неперервної системи залежить від усіх значень вхідного сигналу.

Але все ж таки існують ситуації, у яких дискретна модель може бути точною у розумінні, викладеному вище. Для цього необхідно, щоб значення процесу  $u(t)$  на інтервалі дискретності  $(k-1)T \leq t \leq kT$  однозначно визначались послідовністю  $[u(0), u(T), \dots, u((k-1)T)]$ . Таке має місце для імпульсних систем з амплітудно-імпульсною модуляцією першого роду та для цифрових систем керування, якщо вхідний процес, формується за допомогою ЕОМ. В останньому випадку дискретний вхідний процес перетворюється у неперервний за допомогою екстраполятора. Для таких випадків «точний» розв'язок можливий при застосуванні методів  $Z$ -перетворення

Отже, способи дискретизації, що використовують  $Z$ -перетворення, можна назвати **точними** в тому сенсі, що вони забезпечують збіг неперервного та дискретного вихідних сигналів у моменти часу, кратні періоду переривання, при певному виді вхідного сигналу. У зв'язку з цим розрізняють **імпульсно-інваріантне**, **ступінчасто-інваріантне** та **лінійно-інваріантне**  $Z$ -перетворення.

Методи дискретизації другої групи називають **підстановочними**. Вони за своєю сутністю є **приблизними (неточними)**, тому що вони пов'язані з заміною неперервних інтеграторів у деталізованих структурах цифровими інтеграторами, які синтезуються за допомогою апріорі приблизних методів чисельного інтегрування.

До приблизних методів відноситься і метод відповідності нулів-поліусів. Це пов'язано з тим, що передавальні функції, отримані методами  $Z$ -перетворень, мають у своєму складі не тільки так звані «системні нулі», пов'язані однозначними залежностями з нулями аналогового об'єкту, але й «нулі дискретизації», які можна визначити тільки приблизно.

## 15.2. Дискретизація методами $Z$ -перетворень

З теорії автоматичного керування [11] відомі формули, за якими треба розраховувати ДПФ лінійної неперервної системи з відомою аналоговою ПФ, щоб дискретизована система була імпульсно-інваріантною

$$W_{imp}(z) = T \cdot Z\{W(p)\}. \quad (15.10)$$

ступінчато-інваріантною

$$W_{zoh}(z) = \frac{z-1}{z} \cdot Z\left\{\frac{W(p)}{p}\right\} = (1-z^{-1}) \cdot Z\left\{\frac{W(p)}{p}\right\}. \quad (15.11)$$

або лінійно-інваріантною

$$W_{foh}(z) = \frac{(z-1)^2}{Tz} \cdot Z\left\{\frac{W(p)}{p^2}\right\}. \quad (15.12)$$

При ступінчато-інваріантній дискретизації використовується Z-перетворення з екстраполяцією нульового порядку (*zoh* – *Zero-Order Hold*), а при лінійно-інваріантній дискретизації – перетворення з екстраполяцією першого порядку (*foh* – *First-Order Hold*).

### 15.3 Підстановочні методи дискретизації

Підстановочні методи засновані на взаємних перетвореннях між  $p$  та  $z$ , що є тотожним заміні оператора аналогового інтегрування  $1/p$  одним з операторів цифрового інтегрування.

Передавальна функція цифрового інтегратора залежить від методу чисельного інтегрування. В основу цих методів покладено геометричний сенс визначеного інтеграла, який дорівнює площі фігури, обмеженою кривою підінтегральної функції  $u(t)$ , віссю часу та вертикалями, що виходять з точок границь інтегрування  $t=t_0$  і  $t=t_k$ . Більшість із методів цифрового інтегрування полягає у розбиванні фігури, площу якої треба підрахувати на декілька більш простих фігур, площі яких обчислюються за простими формулами, з подальшим сумуванням цих площ. Найчастіше інтервал інтегрування поділяють на декілька рівних частин, на кожній з яких виконують локальну інтеполяцію підінтегральної функції степеневими поліномами. В залежності від степені полінома  $n$  утворюються різні методи чисельного інтегрування: при  $n=0$  – метод прямокутників, при  $n=1$  – метод трапецій, при  $n=2$  – метод параболічних трапецій, який зазвичай називають методом Сімпсона. При

дискретизації неперервних динамічних ланок підстановочними методами обмежуються заміною аналогових інтеграторів цифровими, синтезованими методами прямокутників та трапецій.

З передавальних функцій цифрових інтеграторів (12.14), (12.17) та (12.20) впливають такі підстановочні формули:

$$p = \frac{z-1}{T} = \frac{1-z^{-1}}{Tz^{-1}}, \quad (15.13)$$

$$p = \frac{z-1}{Tz} = \frac{1-z^{-1}}{T}, \quad (15.14)$$

$$p = \frac{2}{T} \cdot \frac{z-1}{z+1} = \frac{2}{T} \cdot \frac{1-z^{-1}}{1+z^{-1}}. \quad (15.15)$$

Використання підстановки (15.13) часто називають *методом Ейлера*, підстановки (15.14) – *модифікованим методом Ейлера*, а підстановки (15.15) – *білінійним перетворенням* або *методом Тастіна* (*Tustin*).

З аналізу підстановочних перетворень випливає, що найкращим з них є перетворення Тастіна, яке за своїми параметрами і властивостями наближується до  $Z$ -перетворення з екстраполяцією першого порядку. Такий висновок збігається із загальноприйнятою методикою приблизної дискретної апроксимацією неперервних об'єктів., що забезпечує компроміс між простотою і точністю дискретизації.

#### 15.4 Метод відповідності нулів та полюсів

Цей метод відображає неперервні полюси  $p_i$  та нулі  $z_i$  у дискретні, виходячи із взаємозв'язку між оператором Лапласа та дискретним оператором:

$$p_{di} = e^{Tp_i}, \quad z_{di} = e^{Tz_i}. \quad (15.16)$$

В [ ] показано, що застосування співвідношень (15.16) не достатньо для якісної дискретизації у розумінні подібності динамічних та статичних властивостей неперервного об'єкта та його цифрової моделі.

Узагальнюючи, сформулюємо такий висновок: якщо кількість полюсів дискретної ланки перевищує кількість нулів на  $r$ , то реакція цього об'єкта на вхідний сигнал почнеться з  $r$ -ого такту квантування

До того ж передавальні функції, синтезовані шляхом підстановок (15.17) забезпечують різні усталені значення перехідних функцій.

Для ліквідації цих недоліків передавальну функцію дискретної моделі треба

1) доповнити її нулями квантування

$$z_{dkj} = -1 \quad (15.17)$$

так, щоб скоригований у такий спосіб порядок чисельника  $m_d$  був на одиницю менше порядку знаменника:

$$m_d = n - 1, \quad j = m + 1, m + 2, \dots, n - 1 \quad (15.18)$$

або рівним із ним:

$$m_d = n, \quad j = m + 1, m + 2, \dots, n - 1, n; \quad (15.19)$$

2) скоригувати коефіцієнт  $K_d$  у такий спосіб, щоб усталені значення перехідних функцій неперервної та дискретизованої систем були однаковими.

До переваг методу узгодження нулів-полюсів належить висока точність апроксимації, а також те, що дискретні системи, отримані у такий спосіб, не треба перевіряти на стійкість.

### 15.5 Дискретизація неперервних об'єктів у середовищі пакету MATLAB

Для утворення дискретних моделей неперервних систем в MATLAB передбачена функція c2d з таким синтаксисом:

$$\begin{aligned} \text{SysD} &= \text{c2d}(\text{SysC}, T) \\ \text{SysD} &= \text{c2d}(\text{SysC}, T, \text{'метод'}) \\ \text{SysD} &= \text{c2d}(\text{SysC}, T, \text{options}) \end{aligned}$$

Функція  $\text{SysD}=\text{c2d}(\text{SysC},T)$  утворює дискретну модель SysD неперервної системи SysC з періодом дискретності T методом Z-перетворення з використанням екстраполятора нульового порядку.

Функція SysD=c2d(SysC,T,'метод') використовує один з методів екстраполяції, що перераховані у табл. 15.1.

Таблиця 15.1

Метод	Опис методу
zoh	Z-перетворення з екстраполятором нульового порядку на вході (ступінчато-інваріантна дискретизація)
foh	Z-перетворення з екстраполятором першого порядку на вході (лінійно-інваріантна дискретизація)
impulse	Імпульсно-інваріантна дискретизація
tustin	Білінійна апроксимація Тастіна (трапеційна дискретизація)
matched	Метод відповідності нулів та полюсів

### 15.6 Завдання

1. Виконати дискретизацію неперервної передавальної функції двигуна постійного струму (ДПС)

$$\frac{R_a I_a(p)}{U_a(p)} = \frac{\bar{I}_a(p)}{\bar{U}_a(p)} = \frac{\gamma_{кз} T_m p}{T_m T_a p^2 + T_m p + 1}, \quad \bar{I}_a = \frac{I_a}{I_{ан}}, \quad \bar{U}_a = \frac{U_a}{U_{ан}} \quad (15.20)$$

методами ступінчато-інваріантного та лінійно-інваріантного Z-перетворень, методом Тастіна та методом відповідності нулів-полюсів. Порівняти реакції неперервної та цифрових моделей ДПС на стрибкоподібну та лінійну зміну напруги якоря від 0 до номінального значення.

2. Деталізувати ПФ (15.20) і дискретизувати деталізовану структурну модель шляхом заміни аналогових інтеграторів цифровими з різними алгоритмами числового інтегрування.

3. Порівняти реакції неперервної та цифрових моделей ДПС на стрибкоподібну та лінійну зміну напруги якоря від 0 до номінального значення.

Параметри ДПС взяти з лабораторної роботи №5.

## 15.6 Методичні вказівки

1. Ознайомтеся з методами точної та приблизної дискретної апроксимації неперервних динамічних об'єктів і програмними засобами, що визначають параметри дискретних ПФ з бажаним періодом дискретності за параметрами неперервної ПФ.

2. Визначте коефіцієнти чисельників та знаменників дискретних аналогів ПФ (15.20) за допомогою  $m$ -функції  $c2d$  при  $T_s = 0,02$ с.

3. Зберіть модель, що складається із джерел стрибкоподібного *Step* та лінійного (*Clock + Look-Up Table*) сигналів, приєднаних до неперервної ПФ двигуна та її 4-х дискретних апроксимацій через двоканальний ключ *Multiport Switch1*. Вихідні сигнали дискретних ПФ приєднайте до 4-канального ключа *Multiport Switch2*. Організуйте візуалізацію результатів так, щоб в одній системі координат відображалися реакції ДПС на входні сигнали аналогової та цифрових моделей. Можливий варіант *Simulink*-моделі показано на рис. 15.1.

4. Скласти програму для візуалізації результатів симуляції моделі рис. 15.1, які можна розташувати у чотирьох підвікнах, як це показано на рис. 15.2, 15.3.

5. В моделі рис.15.1 замінити блок *Transfer Function* деталізованою структурою, а блоки *Discrete Transfer Function* – структурами, в яких неперервні інтегратори замінено цифровими з різними алгоритмами числового інтегрування, як це показано на рис. 15.4.

6. Організувати візуалізацію результатів моделювання за зразком рис. 15.5.



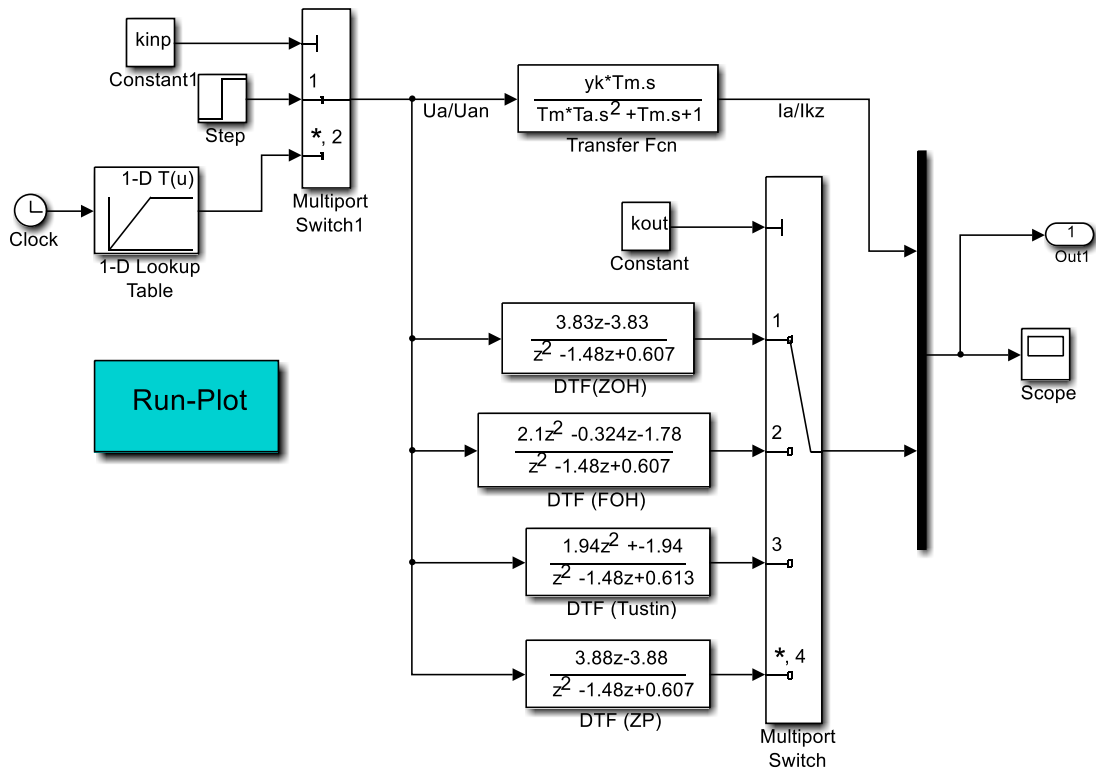


Рис.15.1. Модель для дослідження методів дискретизації ZOH, FOH, *mathed* та *tustin*

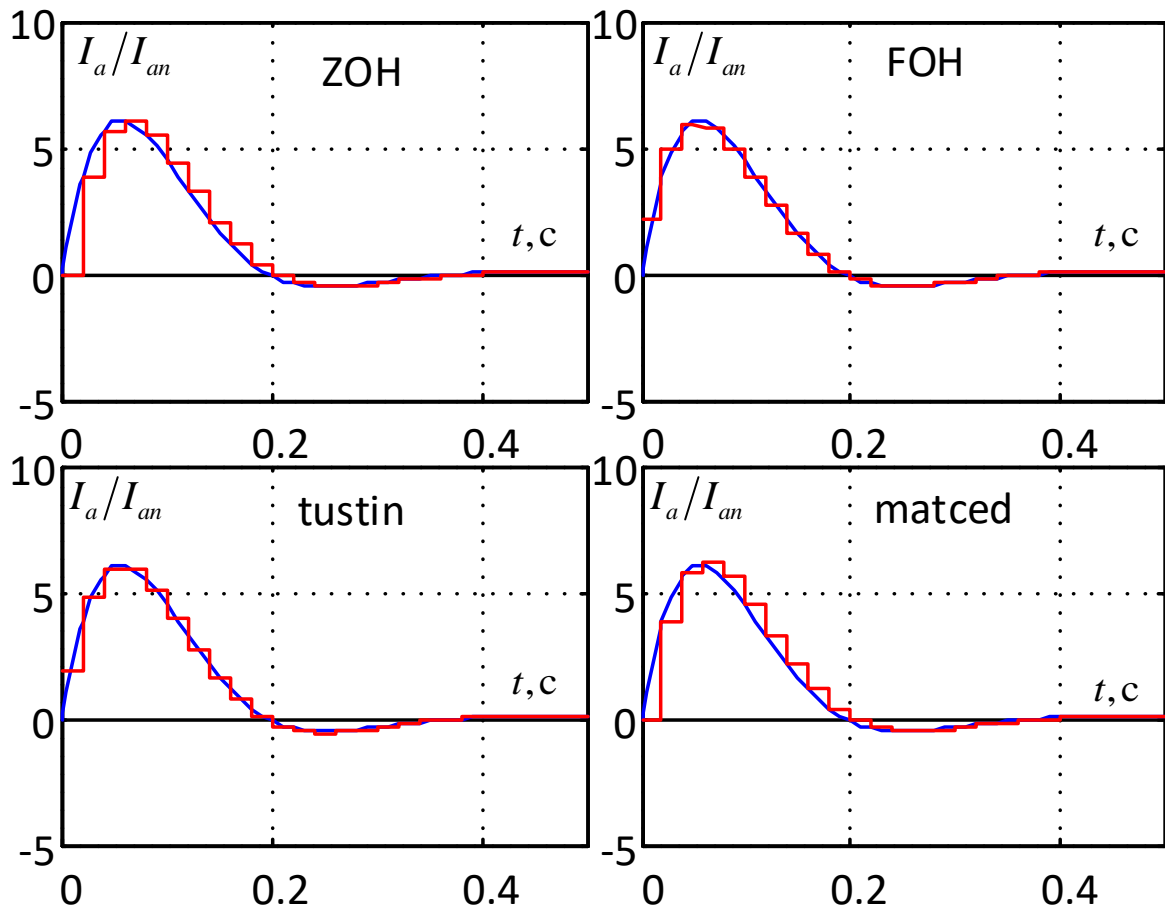


Рис.15.2. Реакція неперервної та дискретних моделей ДПС, отриманих методами ZOH, FOH, *Tustin*, *mathed*, на стрибок напруги якоря

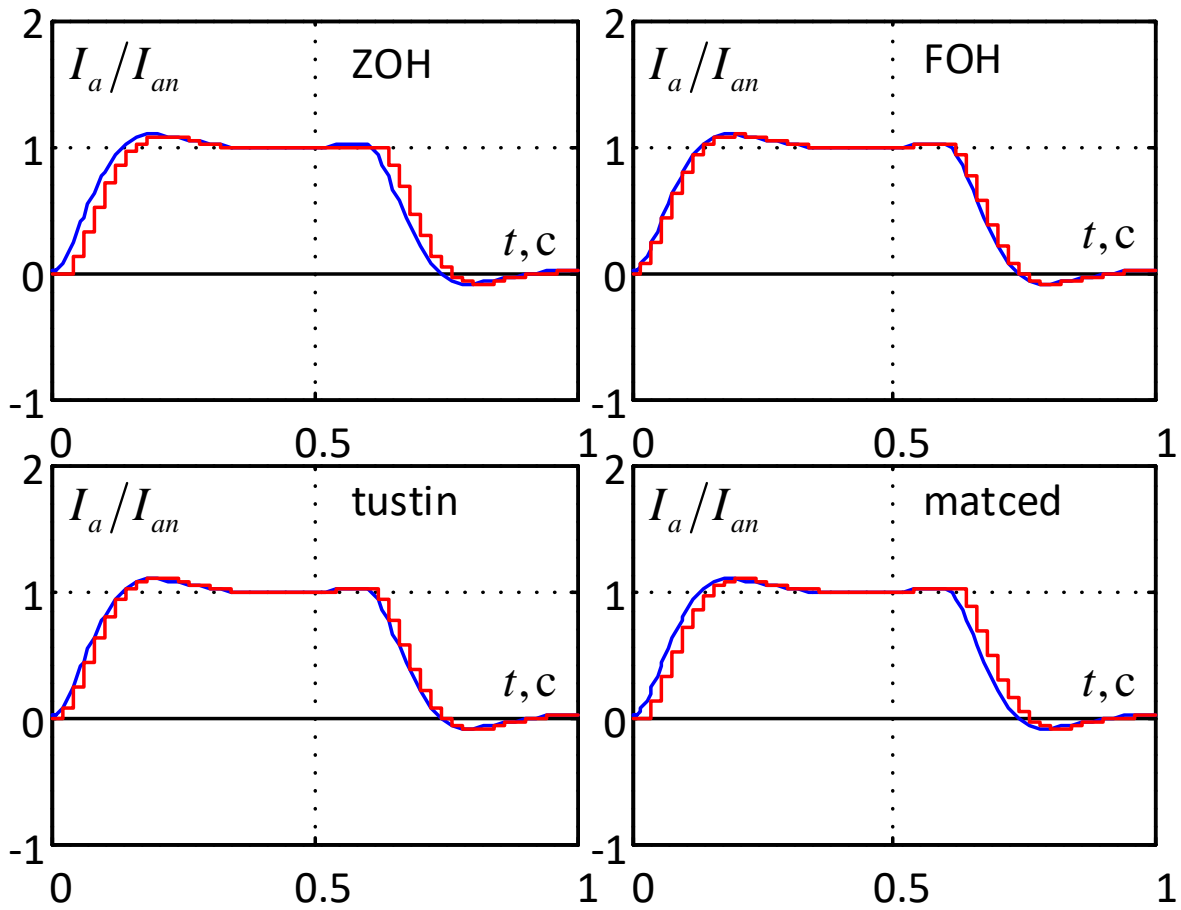


Рис.15.3. Реакція неперервної та дискретних моделей ДПС, отриманих методами ‘ZOH’, ‘FOH’, ‘Tustin’, ‘Matched’, на лінійну зміну напруги якоря

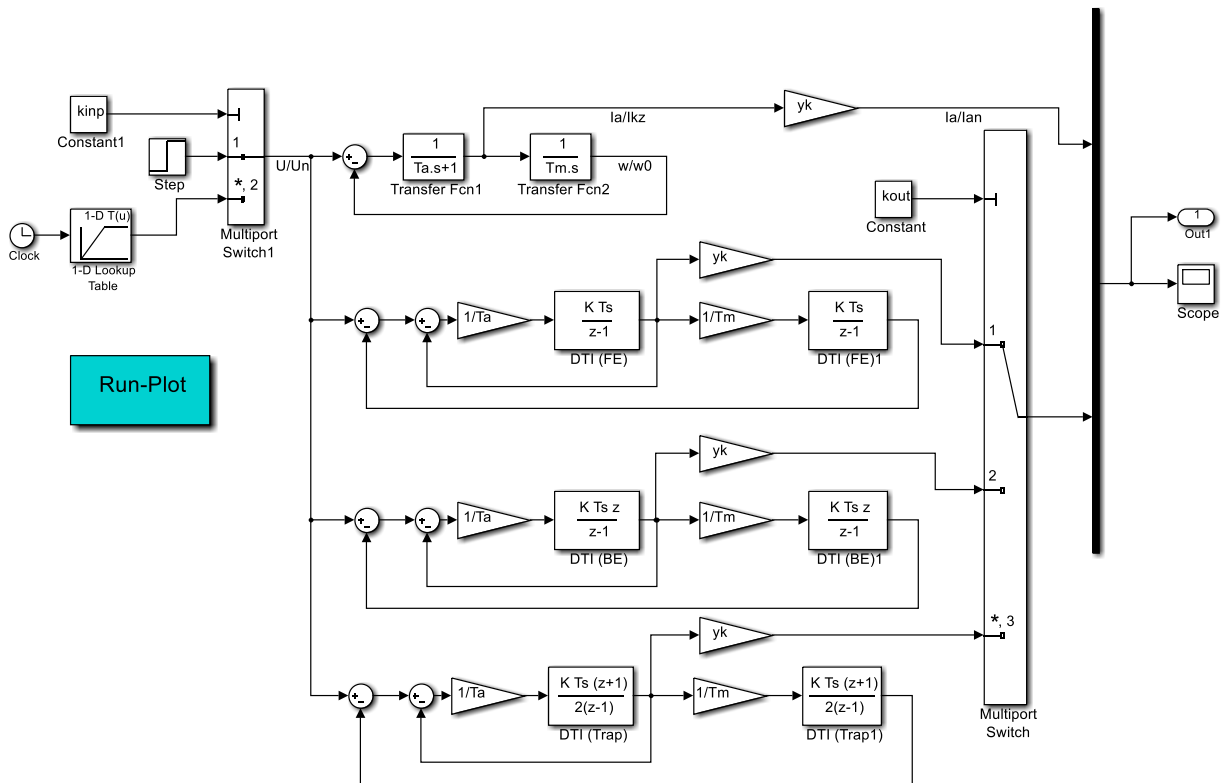


Рис. 15.4. Модель для дослідження підстановочних методів дискретизації

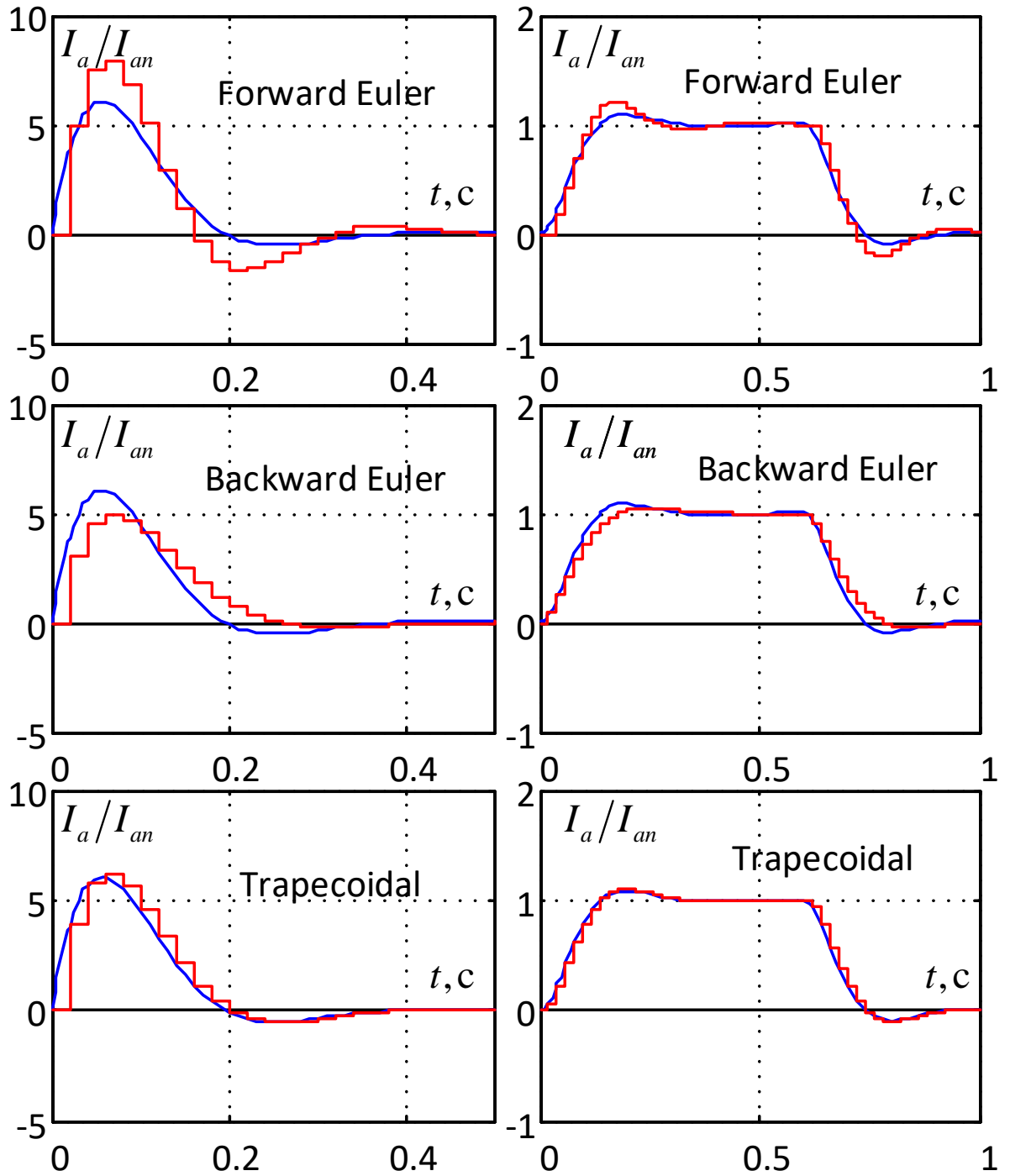


Рис.15.5. Реакція неперервної та дискретних моделей ДПС, отриманих підстановочними методами, на стрибок та на лінійну зміну напруги якоря

## ЛІТЕРАТУРА

1. Островерхов М.Я., Пижов В.М. Моделювання електромеханічних систем в Simulink: Навч. Посібник для студентів вищих навчальних закладів. – К.: ВД «Стилос», 2008. – 528 с.
2. Лазарев Ю.Ф. Начала программирования в среде MatLAB: Навч. посіб. – К.: НТУУ КПІ, 2003. – 424с.
3. Solving control engineering problems with MATLAB (MATLAB curriculum series) / К. Ogata. – Englewood Cliffs, NJ: Prentice Hall, 1994. – 359 p.
4. Dingyü Xue, YangQuan Chen, Derek P. Atherton. Linear Feedback Control Analysis and Design with MATLAB. – Philadelphia: Siam. – 2007, 356 P.
5. Черный А.П. и др. Моделирование электромеханических систем / Учебное пособие. – Кременчуг: КГПИ, 1999. – 204 с.
6. Толочко О.І. Розробка складних електромеханічних систем в середовищі пакета MATLAB з використанням блоків додатку віртуального фізичного моделювання SimScare // Вісник НТУ «ХПІ». Проблеми автоматизованого електроприводу. – Харків: НТУ «ХПІ», 2015, 12. – С. 118-123.
7. Плахтина О.Г. та ін. Частотно-керовані асинхронні та синхронні електроприводи: Навч. посібник. – Львів: Видавництво Національного університету «Львівська політехніка», 2002. – 228 с.
8. Шевченко І.С. , Морозов Д.І. Електромеханічні системи в асинхронному електроприводі: Навч. посібник / Алчевськ: ДонДТУ, 2009. – 349 с.
9. Математичні методи та особливості чисельних розрахунків динаміки електроприводів з асинхронними двигунами: монографія / О.П.Чорний, О.І. Толочко, В.К. Титюк. – Кременчук: ПП Щербатих О.В., 2016. – 300 с.
10. Толочко О.І. Моделювання електромеханічних систем. Математичне моделювання систем асинхронного електроприводу: Навчальний посібник. – Київ: НТУ «КПІ», 2016. – 150 с.

11. Божко С.В., Пересада С.М., Печеник М.В., Толочко О.І.. Цифрове керування електромеханічними системами, 2022. – 149 с. Затверджено Вченою радою КПІ ім. Ігоря Сікорського як підручник для студентів, які навчаються за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка» за освітньою програмою «Електромеханічні системи автоматизації, електропривод та електромобільність».

12. Tolochko O., Palis S., Burmelov O., Kaluhin D. Discrete approximation of continuous objects with MATLAB // International Journal of Science "Applied Aspects of Information Technology", Vol. 4 № 2, 2021. – Pp. 178-191.

13. Зеленов О.Б. Теорія електроприводу. Частина 1: Навч. посіб. – Алчевськ: ДонДТУ, 2005. – 394 с.

14. Лозинський А., Мороз В., Паранчук Я. Розв'язування задач електромеханіки в середовищі пакетів MathCAD і MATLAB: навч. Посібник. – Львів: Видавництво Державного університету «Львівська політехніка», 2000 – 166 с.

15. Проектування систем керування / Г.К. Гудвін, С.Ф. Гребє, М.Е. Сальгадо, 2004. – 911 с.

## ЗМІСТ

Введення .....	3
<i>Лабораторна робота №1. Реєстрація та візуалізація сигналів .....</i>	<i>8</i>
1.1 Основні засоби реєстрації та візуалізації сигналів .....	8
1.1.1 Блоки візуалізації .....	9
1.1.2 Блоки запам'ятовування сигналів .....	18
1.2 Завдання .....	22
1.3 Методичні рекомендації .....	23
<i>Лабораторна робота №2. Формування вхідних сигналів .....</i>	<i>26</i>
2.1 Загальна характеристика блоків бібліотеки <i>Sources</i> .....	26
2.2 Основні засоби формування вхідних сигналів .....	28
2.3 Використання блоку <i>Lookup Table</i> при формуванні кусково-лінійних вхідних сигналів.....	38
2.4 Завдання .....	38
2.5 Методичні рекомендації .....	41
<i>Лабораторна робота №3. Знайомство з бібліотечними нелінійними блоками програми <i>Simulink</i> .....</i>	<i>44</i>
3.1 Класифікація нелінійних блоків .....	44
3.2 Блоки множення та ділення .....	45
3.3 Блоки з нелінійними статичними характеристиками, заданими аналітично .....	49
3.4 Типові нелінійності та їх формування блоками <i>Simulink</i> .....	52
3.5 Моделювання нелінійних функціональних перетворювачів з характеристиками, заданими в табличній формі .....	59
3.6. Моделювання періодичних нелінійностей, заданих таблично або аналітично на періоді .....	68
3.7 Завдання .....	69
3.8 Методичні рекомендації .....	71

<i>Лабораторна робота №4. Знайомство з бібліотечними неперервними ланками програми Simulink</i> .....	75
4.1 Форми математичного опису неперервних лінійних систем.....	75
4.2 Перетворення математичного опису з форми <i>Transfer Function</i> у форму <i>State Space</i> .....	79
4.3 Відомості про блоки бібліотеки <i>Continuous</i> .....	80
4.4 Відомості про лінійні арифметичні блоки бібліотеки <i>Math</i> .....	84
4.5 Завдання .....	85
4.6 Методичні рекомендації .....	85
<i>Лабораторна робота №5. Математичне моделювання двигуна постійного струму з керуванням у колі якоря</i> .....	88
5.1 Математичний опис об'єкту моделювання.....	88
5.2 Нормування структурних схем .....	91
5.3 Визначення та аналіз передавальних функцій .....	95
5.4 Моделювання та аналіз перехідних процесів ДПС в режимі прямого пуску .....	97
5.5 Моделювання ДПС з використанням задатчиків інтенсивності для формування напруги якоря .....	98
5.6 Врахування тертя при моделюванні електричних двигунів .....	106
5.7 Завдання .....	110
5.8 Методичні рекомендації .....	111
<i>Лабораторна робота №6. Математичне моделювання двигуна постійного струму з керуванням у колі якоря та колі збудження</i> .....	113
6.1 Математичний опис об'єкту моделювання.....	113
6.2 Розробка <i>Simulink</i> -моделей для дослідження ДПС з двозонним регулюванням швидкості .....	115
6.3 Аналіз перехідних процесів .....	119
6.4 Завдання .....	123
6.5 Методичні вказівки та рекомендації .....	124

<i>Лабораторна робота №7. Створення підсистем та їх маскування</i> .....	125
7.1. Створення підсистем .....	125
7.2 Маскування підсистем .....	126
7.3 Створення кнопок, що керують виконанням модельного експерименту .....	138
7.4 Завдання .....	140
7.5 Методичні рекомендації .....	141
<i>Лабораторна робота №8. Імітаційне фізичне моделювання електричних кіл з використанням блоків бібліотек SimPowerSystems</i> .....	146
8.1 Основні відомості про бібліотеку SimPowerSystems та правила її застосування .....	146
8.2 Характеристика основних блоків для моделювання кіл постійного струму та однофазних кіл змінного струму .....	151
8.2.1 Джерела електричної енергії .....	153
8.2.2 Електротехнічні елементи .....	154
8.2.3 Основні вимірювальні прилади .....	154
8.2.4 Ключові елементи .....	156
8.2.5 Графічний інтерфейс користувача Powergui .....	159
8.3 Приклад віртуального фізичного моделювання лінійного розгалуженого електричного кола .....	162
8.4 Завдання .....	164
8.5 Методичні рекомендації .....	164
<i>Лабораторна робота №9. Імітаційне фізичне моделювання електричного двигуна постійного струму з незалежним збудженням</i> .....	167
9.1 SPS-модель двигуна постійного струму .....	167
9.2 Завдання .....	171
9.3 Методичні рекомендації .....	171
<i>Лабораторна робота №10. Імітаційне фізичне моделювання асинхронного двигуна</i> .....	172



10.1 Знайомство з блоками асинхронних машин бібліотеки <i>Machines</i> другої генерації бібліотеки <i>SimPowerSystems</i> .....	172
10.2 Моделювання асинхронного двигуна в режимі прямого пуску ...	176
10.3 Моделювання асинхронного двигуна в режимі плавного пуску ..	180
10.4 Моделювання процесів скалярного частотного регулювання швид- кості .....	182
10.5 Побудова статичних характеристик АД .....	185
10.6 Завдання .....	185
10.7 Методичні рекомендації .....	186
<i>Лабораторна робота №11. Використання SPS–моделей електричних двигу- нів для дослідження системи «механічний вал» .....</i>	
11.1 Особливості застосування SPS–моделей електричних двигунів при дослідженні системи «механічний вал» .....	190
11.2 Завдання .....	195
11.3 Методичні рекомендації .....	195
<i>Лабораторна робота №12. Математичне моделювання двомасових механічних систем .....</i>	
12.1 Математичний опис двомасових механічних систем .....	196
12.2 Завдання .....	202
12.3 Методичні рекомендації .....	202
<i>Лабораторна робота №13. Математичне моделювання кранової механічної системи візок-вантаж .....</i>	
13.1 Математичний опис механічної системи візок-вантаж .....	204
13.2 Завдання .....	213
13.3 Методичні рекомендації .....	213
<i>Лабораторна робота №14. Дослідження властивостей цифрових інтеграторів .....</i>	
14.1 Загальні поняття про дискретну інформацію .....	214
14.2 Форми математичного опису лінійних імпульсних систем .....	217
14.3 Огляд блоків бібліотеки <i>Discrete</i> .....	219

14.4 Математичний опис найпростіших цифрових інтеграторів .....	219
14.5 Завдання .....	229
14.6 Методичні рекомендації .....	229
<i>Лабораторна робота №15. Дослідження методів дискретизації</i>	
неперервних динамічних об'єктів .....	232
15.1 Постановка задачі. Класифікація методів дискретизації .....	232
15.2 Дискретизація методами Z-перетворень .....	235
15.3 Підстановочні методи дискретизації .....	236
15.4 Метод відповідності нулів та полюсів .....	237
15.5 Дискретизація неперервних об'єктів у середовищі пакету	
<i>MATLAB</i> .....	238
15.6 Завдання .....	239
15.7 Методичні рекомендації .....	240
Література .....	244