

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторного практикуму та самостійної роботи з дисципліни

“МОДЕЛЮВАННЯ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ”

для студентів спеціальності

F5 «Кібербезпека та захист інформації»,

освітні програми: «Технічний захист інфраструктури об’єктів
інформаційної діяльності» та

«Кіберстійкість цифрових платформ та мережевої інфраструктури»
другого (магістерського) рівня вищої освіти

Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Моделювання систем захисту інформації» для студентів спеціальності F5 «Кібербезпека та захист інформації», освітні програми: «Технічний захист інфраструктури об'єктів інформаційної діяльності» та «Кіберстійкість цифрових платформ та мережевої інфраструктури» другого (магістерського) рівня вищої освіти / Укл.: О. В. Василенко, А. В. Коротун. – Запоріжжя: НУ «Запорізька політехніка», 2025. – 122 с.

Укладачі: Ольга ВАСИЛЕНКО, доц., канд. техн. наук
Андрій КОРОТУН, проф., канд. фіз.-мат. наук

Рецензент: Геннадій СНИЖНОЙ, проф., докт. техн. наук

Відповідальний за випуск: Андрій КОРОТУН, проф., канд. фіз.-мат. наук

Затверджено
на засіданні кафедри ІБтаН
Протокол № 5
від 22. 01. 2025 р.

Рекомендовано до видання
НМК ФІБЕК
Протокол № 6
від “ 29 “ січня 2025 р.

ЗМІСТ

Вступ	5
1 Лабораторна робота №1 “Дослідження підходів до моделювання систем”	6
1.1 Характеристики математичної моделі.	6
1.2 Критерії якості моделей	10
1.3 Вибір платформи для моделювання	12
1.4 Критерії якості симуляції	13
1.5 Особливості платформ ECAD	14
1.6 Критерії вибору та огляд CAE-систем	18
1.7 Етапи виконання роботи	19
1.8 Контрольні запитання	19
2 Лабораторна робота №2 “Моделювання трактів передачі даних ”	20
2.1 SamSim	21
2.2 ANSYS Simplorer (VHDL)	23
2.3 SimApp	24
2.4 Wolfram SystemModeler (MathModelica)	26
2.5 MapleSim	28
2.6 MATLAB / Simulink та Simscape	31
2.7 VisSim	35
2.8 Dymola (Modelica)	39
2.9 20-sim	40
2.10 Тракти передачі даних	44
2.10.1 Втрати в лінії та узгодженість навантаження	44
2.10.2 Модель довгої лінії	45
2.11 Етапи виконання роботи	47
2.12 Контрольні запитання	48
3 Лабораторна робота №3 “Моделювання автоматизованих систем”	49
3.1 Структурні елементи системи управління.	49
3.2 Теоретичні аспекти моделювання систем автоматичного управління	51
3.3 Моделі для елементів САУ в програмі Micro-Cap	53
3.4 Проектування САУ в 20-sim.	55
3.5 Методика моделювання САУ	60
3.6 Контрольні запитання	61

4 Лабораторна робота №4 “Моделювання комп’ютерних мереж”	63
4.1 Імітаційне моделювання систем	63
4.2 Способи моделінгу систем.	65
4.3 Скорочені теоретичні відомості по системі NetCracker . . .	68
4.3.1 Приклад імітаційного моделювання комп’ютерної мережі.	69
4.4 Етапи виконання роботи	75
4.5 Контрольні запитання	76
5 Лабораторна робота №5 “Моделювання нейронних мереж”..	77
5.1 Машинне навчання	77
5.2 Нейронні мережі та функції активації	79
5.3 Вступ до B-Spline мереж	82
5.3.1 Навчання мережі B-Spline	82
5.4 Проектування нейронних мереж в 20-sim	83
5.5 Етапи виконання роботи	88
5.5.1 Тренування B-сплайн мережі	91
5.5.2 Тестування 1-D B-Spline мережі	92
5.5.3 Тренування і розгортання двохвимірної BSpline мережі .	92
5.6 Контрольні запитання	93
6 Лабораторна робота №6 “ Оптимізація технічних засобів захисту інформації ”	94
6.1 Критерії оптимальності систем	94
6.2 Методи оптимізації	99
6.3 Параметрична оптимізація в Micro-Cap	106
6.4 Етапи виконання роботи	108
6.5 Контрольні запитання	110
Перелік рекомендованої літератури	111
Додаток А Перелік питань до модульного контролю і самостійних робіт	112
Додаток Б Математичні процесори для моделювання.	115
Додаток В Загальна структура 20-sim.	116
Додаток Г Класифікація методів симуляції.	117
Додаток Д Структура САЕ.	118
Додаток Е Згорнутий алгоритм моделювання САК	119
Додаток Ж Алгоритм вибору методу інтегрування в 20-sim . .	120
Додаток И Процедури етапу компіляції моделі в САЕ	121
Додаток І Класифікація задач оптимізації.	122

ВСТУП

Моделювання систем захисту інформації – курс, призначений для ознайомлення із методиками та програмами моделювання (моделінгу та симуляції) апаратно-програмних комплексів систем захисту інформації, зокрема комп'ютерних мереж, трактів передачі даних та нейронних мереж для машинного навчання. Предметом курсу є вивчення й застосування сучасних технологій під час моделювання технічних систем захисту інформації (СЗІ).

Основні завдання вивчення дисципліни «Моделювання систем захисту інформації» полягають у:

- **Пізнавальні** – освоєння основних технологій, методик і програм моделювання систем захисту інформації, зокрема із використанням проривних технологій Індустрії 4.0.
- **Практичні** – сформувати навички використання програм для моделювання елементів та систем захисту інформації та використання проривних технологій у своїй професійній діяльності.

У результаті вивчення дисципліни «Моделювання систем захисту інформації» здобувач освіти повинен –

мати уявлення про:

- ❖ принципи та платформи математичного моделювання;
- ❖ моделювання як одну з базових технологій, що забезпечили четверту промислову революцію;
- ❖ методику моделювання на різних рівнях абстракції;
- ❖ принципи застосування штучного інтелекту для перетворення пристроїв захисту інформації «на розумні» речі;

знати:

- ❖ принципи вибору програм для моделювання;
- ❖ критерії якості моделей.

вміти:

- ❖ застосовувати на практиці методики моделінгу систем захисту інформації;
- ❖ обґрунтовано обирати структури, програми та методики моделювання для дослідження та проєктування СЗІ;
- ❖ будувати моделі для проєктування апаратної частини пристроїв захисту інформації та здійснювати симуляцію на платформах CAD та CAE.

1 ЛАБОРАТОРНА РОБОТА №1 «ДОСЛІДЖЕННЯ ПІДХОДІВ ДО МОДЕЛЮВАННЯ СИСТЕМ»

Мета: отримати навички виконання і обробки результатів аналізу систем в середовищі програм автоматизованого моделювання /проектування.

Моделювання є одним з найважливіших етапів проектування будь-якого технічного об'єкта, що дозволяє замінити або значно скоротити етапи налагодження і натурних випробувань. Роль моделювання особливо висока, коли натурні випробування дуже дорогі або небезпечні, як, наприклад, при роботі з космічними апаратами, хімічними і ядерними реакторами та іншими об'єктами.

Термін «моделювання» має дві складові: modeling & simulation:

- «**modeling**» – відноситься, насамперед, до процесу побудови моделей об'єктів і систем;
- «**simulation**» – означає проведення комп'ютерного експерименту з моделлю, з візуалізацією результатів експерименту.

1.1 Характеристики математичної моделі

Модель – створювана людиною подоба досліджуваних об'єктів: макети, зображення, схеми, словесні описи, математичні формули, карти і т.д. Більш строго модель можна визначити як фізичну або математичну конструкцію, що певним чином відображає об'єкт і слугує для його вивчення. Модель є замінником реального об'єкта і володіє, принаймні, двома властивостями:

а) вона відображає ті властивості об'єкта, які істотні для даного дослідження;

б) завжди простіше об'єкта.

Модель повинна бути:

- адекватною;
- надійною;
- простою і зрозумілою користувачу;
- спрямованою на досягнення поставлених цілей дослідження;
- зручною у використанні (ергономічною);
- функціонально повною з точки зору можливостей вирішення головних завдань;

- адаптивною, що дозволяє легко переходити до інших модифікацій або оновлювати дані;
- допускати зміни (в процесі експлуатації вона може ускладнюватися).

Формою графічного представлення інформації про модельовану систему можуть бути:

- операторно-структурні схеми;
- функціональні та принципові схеми різних фізичних пристроїв;
- кінематичні схеми механізмів;
- сигнальні граfi (SFG);
- граfi зв'язків або зв'язані граfi (BG);
- блок-схеми алгоритмів та інші графічні моделі.

Математично технічні системи можуть описуватися різними способами, в залежності від класу системи:

- безперервні системи, описувані диференціальними рівняннями;
- дискретні системи, описувані кінцево-різницевиими рівняннями;
- системи з розподіленими параметрами, які описуються диференціальними рівняннями в приватних похідних;
- системи з післядією, описувані функціонально-диференціальними рівняннями. Такі системи виникають тоді, коли протікання процесу визначається не тільки станом системи в даний момент, але також і передісторією процесу;
- стохастичні системи, описувані ймовірнісними методами.

Будь-яку динамічну систему можна описати однією з форм рівнянь (або їх комбінацією) представлених на рисунку 1.1.

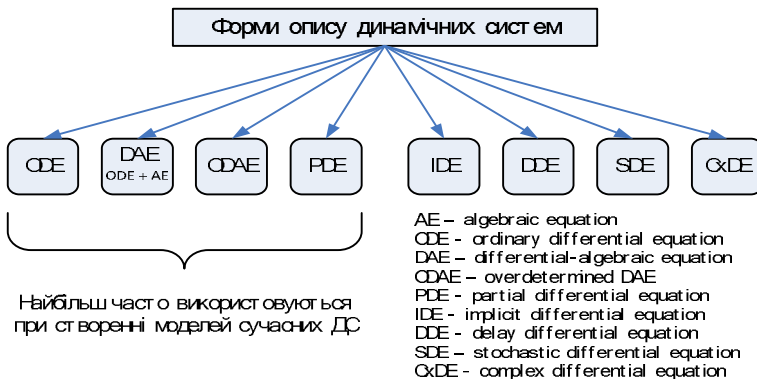


Рисунок 1.1 – Форми опису ДС

Якщо поведінку безперервної системи можна описати відразу лише алгебраїчними рівняннями (АЕ) без попередніх спрощень, то вона не є динамічною (не відображає динаміку). Це властиво лише деяким абстрактним системам.

Процес побудови моделі виявився етапом, що найбільш важко автоматизується. «Ручна» підготовка моделі складного технічного об'єкта пов'язана з великим обсягом перетворень, в яких легко допустити помилку. Використання систем автоматизації моделювання, дозволило істотно підвищити продуктивність праці, знизити число помилок, фахівець в предметній (прикладній) області може самостійно створювати досить складні моделі. У сучасних системах автоматизованого моделювання, виходячи з міркувань зручності сприйняття людиною, використовуються, як правило, графічні засоби завдання вихідної інформації про модель.

Для опису однієї і тієї ж реальної системи може бути використаний різний математичний апарат, в залежності від цілей дослідження і вимог точності та адекватності. Приклад: ММ електричного фільтра другого порядку з резистивним навантаженням та джерелом постійної напруги реалізована в на САЕ-платформі 20-sim трьома способами: блок-схемою (рисунок 1.2), фізичною або принциповою схемою (рисунок 1.3), зв'язаним графом (рисунок 1.4).

Параметри схем однакові: $V_{source}=50$, $L=0.0001$, $C=0.00001$, $R=10$, початкові умови дорівнюють нулю.

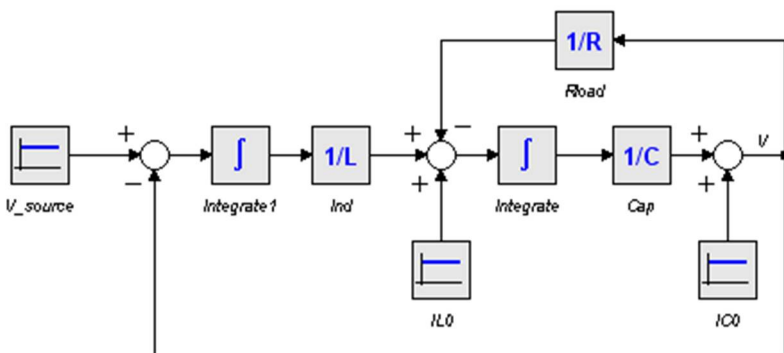


Рисунок 1.2 – Блок-схема (бібліотека «Signal»), що представляє ММ фільтра

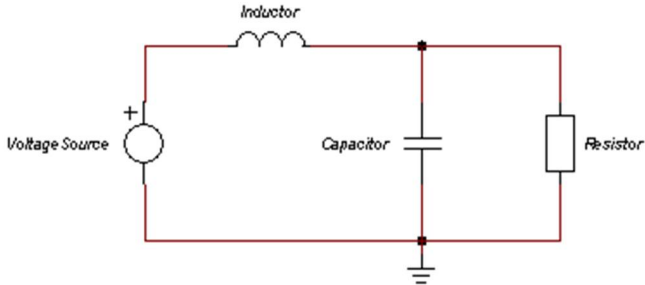


Рисунок 1.3 – Фізична схема (бібліотека «Iconic diagrams») фільтра

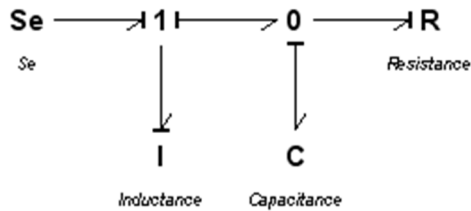


Рисунок 1.4 – Фільтр у вигляді зв'язаного графа (бібліотека «Bond Graph»)

Математична еквівалентність представлених схем підтверджується результатами симуляції графічних моделей, тобто часовими діаграмами на рисунку 1.5.

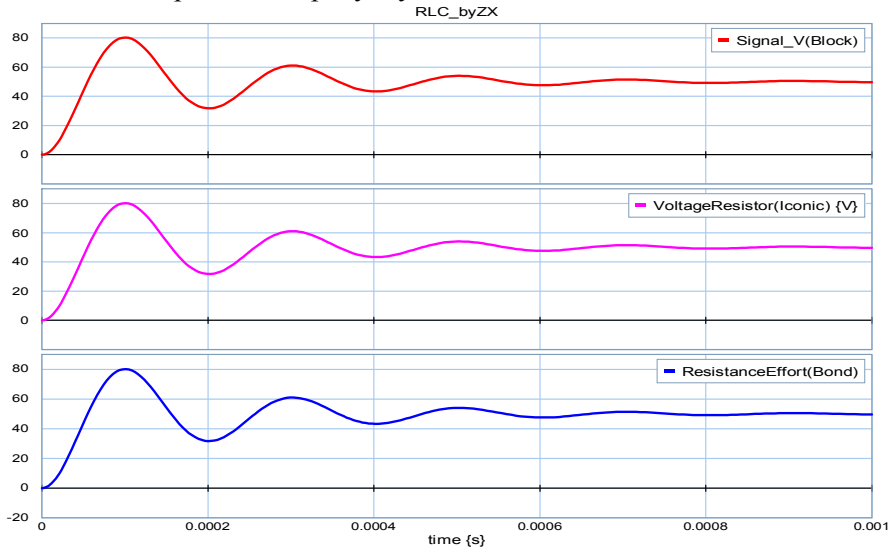


Рисунок 1.5 – Результати симуляції трьох моделей

Можна побачити, що, у загальному випадку, візуально найбільш компактна форма графічного представлення ММ – зв'язаний граф. Фізична схема оперує традиційними компонентами для даного домена (електричного), а блок-схема детально зображує каузальні зв'язки і є графічною аналогією формули, перебудованої відповідним чином, яка описує поведінку системи.

Отже, саме математичний опис буде визначати вибір інструментальних засобів і технологій проєктування систем. При цьому на сучасному етапі особлива увага приділяється проєктуванню технічних систем, характерною особливістю яких є різке підвищення їх логічної складності, жорсткості вимог якості проєктування і зниження часу і вартості розробки. Вирішити зазначені протиріччя, що виникають при проєктуванні технічних систем, особливо засобів і систем управління, на сучасному етапі можливо тільки через розробку і впровадження систем автоматизованого проєктування (САПР). Вміння застосовувати такий інструментальний засіб проєктування, як САПР, ґрунтується на розумінні того, яким чином будується платформа моделювання, її структура та окремі підсистеми, на знанні ММ, методів їх формування, методів і алгоритмів аналізу та синтезу, які покладені в основу цих підсистем.

1.2 Критерії якості моделей

Математична модель завжди відображає тільки частину властивостей реального об'єкта, яка визначається цілями моделювання. При побудові моделі прагнуть як можна більш точно відобразити властивості об'єкта в сенсі, визначеному метою моделювання. З іншого боку, чим простіше математична модель, тим легше її досліджувати і використовувати при вирішенні завдань синтезу. Мистецтво моделінгу полягає в умінні вибрати фактори, суттєві з точки зору мети моделювання, і знехтувати ефектами, які, ускладнюючи математичну модель, не вчиняють помітного впливу на поведінку системи.

Модель **адекватна** оригіналу, якщо вона вірно відображає певні властивості оригіналу і може бути використана для передбачення його поведінки. При цьому адекватність моделі залежить від цілей моделювання і прийнятих критеріїв. Наприклад, модель, адекватна на етапі пошукового проєктування, при деталізації проєкту стає занадто

"грубою". Враховуючи початкову неповноту моделі, можна стверджувати, що ідеально адекватна модель в принципі неможлива.

Точність є мірою адекватності.

При оцінці адекватності моделі може бути використана лише обмежена підмножина всіх можливих значень вхідних параметрів (робочого навантаження і зовнішнього середовища).

Наближеність моделі до дійсного об'єкту можна розглядати в таких аспектах:

- а) з точки зору коректності зв'язку «вхід-вихід» (адекватність);
- б) з точки зору коректності декомпозиції модельного опису стосовно цілей дослідження та використання моделей (автентичність).

В останньому випадку вимагається, щоб усі підмоделі і їх елементи були адекватні відповідним прототипам реального об'єкта. Проблема автентичності значно складніше адекватності і може розглядатися лише при отриманні ММ «з середини». Перша проблема допускає суворий аналіз, однак також є актуальною, складною і далекою від повного вирішення.

Для перевірки моделі можуть використовуватися різні прийоми, такі як:

- а) перевірка фізичного сенсу (дотримання фізичних законів);
- б) перевірка розмірності і знаків;
- в) перевірка меж;
- г) перевірка тренду, тобто тенденції зміни вихідних змінних в залежності від внутрішніх і зовнішніх змінних.

Економічність ММ визначається двома основними факторами:

- витратами машинного часу на прогін моделі (відладку, симуляцію);
- витратами оперативної пам'яті, необхідної для розміщення моделі, що особливо актуально для систем реального часу.

Універсальність моделей визначає область їх можливих застосувань. Зазвичай універсальність досягається тим, що в модель включається велика кількість внутрішніх параметрів, що негативно впливає на економічність. Універсальна ММ адекватна в різних режимах аналізу та функціонування.

Стійкість моделі – це її здатність зберігати адекватність при дослідженні системи на всьому можливому діапазоні робочого навантаження, а також при внесенні змін у конфігурацію системи. Універсальної процедури перевірки стійкості моделі не існує. Часто

перевірка полягає в порівнянні результатів моделювання та результатів вимірювання на системі після внесення до неї змін. Якщо результати моделювання прийнятні, впевненість у стійкості моделі зростає. У загальному випадку можна стверджувати, що чим ближче структура моделі структурі системи і чим вище ступінь деталізації, тим стійкіше модель.

Чутливість. Очевидно, що стійкість є позитивною властивістю моделі. Однак якщо зміна вхідних впливів або параметрів моделі (в деякому заданому діапазоні) не відбивається на значеннях вихідних змінних, то користь від такої моделі мала. У зв'язку з цим виникає задача оцінювання чутливості моделі до змін параметрів системи. Зазвичай таку оцінку проводять по кожному параметру окремо. Дані, отримані при оцінці чутливості моделі, можуть бути використані при плануванні експериментів: більша увага повинна приділятися тим параметрам, за якими модель є більш чутливою.

Як бачимо, критерії якості протирічають один одному, отже задача проектування моделі є оптимізаційною.

1.3 Вибір платформи для моделювання

Класифікацію програм здійснюють за низкою ознак, наприклад, за цільовим призначенням, масштабами (комплексності вирішуваних завдань), характером базової підсистеми – ядра (солвера) програми. Класифікація фіксує місце об'єкта в системі, яка вказує на його властивості. Класифікація створює умови для розробки технічно обґрунтованих норм забезпечення процесу створення, функціонування та стандартизації в області системи автоматизованого моделювання (R&D) та проектування (САПР). За сферою використання найбільш представницькими та широко використовуваними є наступні групи САПР:

- для застосування в галузях загального машинобудування. Їх часто називають машинобудівними САПР або MCAD-системами (Mechanical CAD).

- для радіоелектроніки – ECAD (Electronic CAD) або EDA (Electronic Design Automation) системи.

- в області архітектури і будівництва.

За цільовим призначенням розрізняють САПР для різних аспектів проектування:

- САПР функціонального проектування та моделювання систем на високих рівнях абстракції, інакше САЕ-системи (Computer Aided Engineering).

- конструкторські САПР, часто звані CAD-системами (Computer Aided Design);

- технологічні САПР, інакше звані автоматизованими системами технологічної підготовки виробництва або САМ-системами (Computer Aided Manufacturing).

Також є універсальні математичні процесори, CAS – Computer Algebra System, та інтегровані системи R&D та підготовки виробництва CADAM.

САЕ системи дозволяють досліджувати системи на макрорівні, тобто аналізувати та оптимізувати систему із ціллю прийняття рішення щодо топології (структурний підхід) та функціональних можливостей в цілому (системний підхід). В CAD проєктуються підсистеми різної фізичної природи як на макрорівні (функціональний рівень для змішаних систем), так і на мікрорівні (схемотехнічний рівень). В САМ від ECAD можуть бути передані програми для станків із ЧПУ, роботів та синтезовані технологічні карти.

В системах віртуального виробництва ланка САЕ → CAD → САМ утворює наскрізний цикл під керуванням програми – менеджера даними про виріб (PDM- Product Data Manager).

Для моделювання на нижніх рівнях абстракції використовують об'єктно-орієнтовані програми CAD для моделювання в електронному домені – ECAD). На верхніх рівнях абстракції використовують САЕ. Також на всіх рівнях абстракції використовують універсальні математичні процесори CAS (наприклад, Matlab). (див. Додаток А, Додаток Б).

1.4 Критерії якості симуляції

Характеристики математичного забезпечення справляють істотний, та іноді і визначальний вплив на можливості і показники програм моделювання. Найважливішими з них зазвичай вважаються показники універсальності, алгоритмічної надійності, економічності та точності.

Універсальність, це застосованість математичного забезпечення до широкого класу об'єктів, що проєктуються. Висока ступінь універсальності потрібна для того, щоб певну програму можна

було застосувати для дослідження будь-яких, або більшості проєктованих об'єктів. Ступінь універсальності характеризується заздалегідь обумовленими обмеженнями.

Алгоритмічна надійність, це властивість компонента математичного забезпечення давати при його правильному застосуванні правильні результати. Алгоритмічну надійність можна оцінити ймовірністю отримання правильних результатів при дотриманні заздалегідь встановлених і відомих користувачеві обмежень на застосування методу. Для алгоритмічно надійного методу ця ймовірність близька до одиниці.

Витрати машинного часу, або економічність - це головний обмежуючий фактор при спробах підвищити складність модельованих об'єктів і ретельність їх дослідження. Тому вимога економічності за витратами машинного часу традиційно є однією з основних до математичного забезпечення.

Точність – це головний критерій, що висувається до результатів моделювання, вона є кількісною характеристикою застосованості математичного забезпечення до вирішення конкретних завдань. Точність визначається за ступенем збігу розрахункових і дійсних результатів, зазвичай, із використанням однієї зі стандартних норм. Розрахунок точності може бути непростю задачею, якщо вихідних функцій декілька (сигнал векторний). Зазвичай у таких випадках точність розраховується як зважена сума точності за складовими цього сигналу.

1.5 Особливості платформ ECAD

Серед сучасних програмних комплексів, призначених для аналізу електронних пристроїв, є програми проєктування і програми моделювання. Схему типової ECAD-системи наведено на рисунку 1.6. Блоки, які можуть бути відсутніми у програмах моделювання, позначені пунктиром. Програми проєктування (дозволяють провести наскрізний цикл проєктування, починаючи від введення проєкту у вигляді багатосторінкових принципових схем до передвиробничої підготовки друкованих плат. При цьому виконується цифро-аналогове моделювання пристрою, розробка топології, передача інформації в системи 3D-проєктування, підготовка технологічних файлів і конструкторської документації.

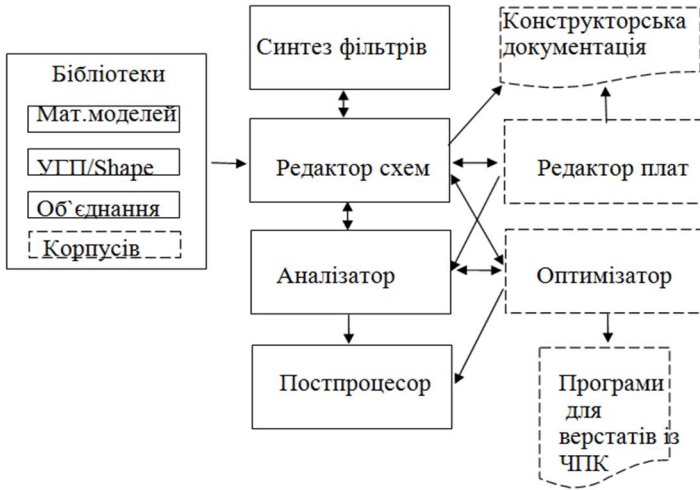


Рисунок 1.6 – Типова структура ECAD

Програми моделювання призначені для схемотехнічного і функціонального моделювання електронних пристроїв. Наприклад, Micro-Cap дозволяє аналізувати аналогові і цифрові прилади, створювати нові макромоделі, що полегшує моделювання без втрати важливої інформації про поведінку системи. Вид аналізу обирають за класом пристрою, який моделюється, і функціями схеми, за якими схема має бути оптимізована.

Перед аналізом необхідно визначити в редакторі *SVE* (*State Variable Editor*), або розрахунком режиму по постійному струму початкові умови: потенціали аналогових вузлів, струми індуктивностей і логічний стан цифрових вузлів, інакше вони будуть обнулені автоматично.

Одноваріантний аналіз дозволяє проаналізувати функції схеми при номінальних значеннях внутрішніх і зовнішніх параметрів. Отже, номінали компонентів, значення параметрів моделі, температура й інші параметри залишаються незмінними впродовж всього сеансу моделювання (для непараметричних схем).

До основних одноваріантних видів аналізу відносять:

- розрахунок перехідних процесів – *Transient*;
- розрахунок частотних характеристик – *AC*;
- розрахунок передатних функцій по постійному струму – *DC*.

Доступні види аналізу в програмі MC12 наведено на рис 1.7.

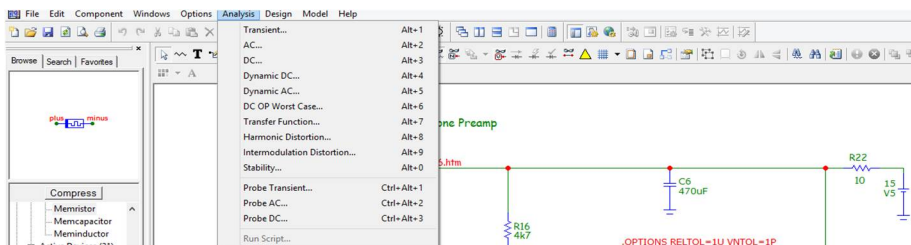


Рисунок 1.7 – Доступні види аналізу в програмі MC12

Аналіз перехідних процесів в ECAD є найскладнішим в плані використаного математичного забезпечення, до якого висувуються найжорсткіші вимоги щодо збіжності та стійкості. Модель для цього виду аналізу майже не спрощується, тому і можливості постпроцесору є найширшими. Розрахунок ведеться із автоматично обраним кроком по часу (хоча можна використовувати і інші аргументи), мінімальне значення якого визначається допустимою відносною похибкою RELTOL. У вікні *Global Setting* RELTOL звичайно = 10^{-3} . Якщо максимальний крок не задано, він визначається автоматично:

$$h_{max} = (T_{max} - T_{min}) / 100.$$

Перед аналізом перехідних процесів автоматично розраховується статичний режим схеми (*DC Operation Point* – робоча точка), якщо початкові умови були обнулені. При аналізі формується система нелінійних алгебро-диференціальних рівнянь, які алгебраїзуються (дискретизуються) методами високого порядку (наприклад, Гіра). Отримана в результаті цього система нелінійних алгебраїчних рівнянь (СНАР) лінеаризуються методом Ньютона. Для розв’язання отриманої системи лінійних алгебраїчних рівнянь (СЛАР) використовують, наприклад, метод LU-перетворення.

Спектральний аналіз виконується після розрахунку перехідних характеристик швидким перетворенням Фур’є із розрахунком коефіцієнта нелінійних спотворень.

Багатоваріантні види аналізу призначені для визначення області безпечних режимів і допусків, та для локальної параметричної оптимізації пристроїв електроніки. Вони можуть бути з детермінованим, або випадковим кроком зміни внутрішніх (W) / зовнішніх (Q) / вхідних (X) параметрів (див. рис. 1.8).

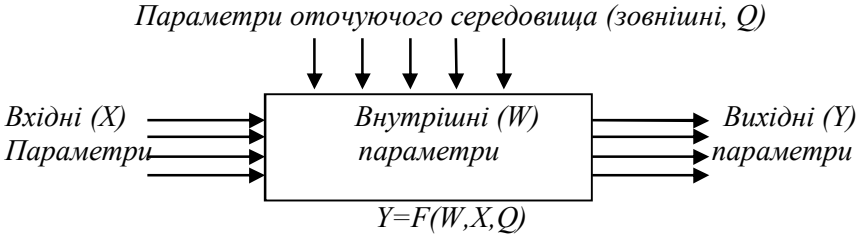


Рисунок 1.8 – Модель системи – «біла скриня»: схема і функціонал

Розрахунок передаточних характеристик по постійному струму можна віднести до багатоваріантних видів аналізу, суть яких полягає в багаторазовому розрахунку пристроїв одноваріантними видами.

Розрахунок сімейства функцій схеми при різних значеннях параметру (значення параметра моделі компонента) називають варіацією параметру (*Stepping/Sweep*). Крок варіації, початкове і кінцеве значення обирає користувач. В якості варіюваного параметру може бути температура, для цього використовують теплові моделі компонентів. Види аналізу із використанням статистичних випробувань: аналіз чутливості (*Sensitivity*), аналіз за методом Монте-Карло (*Monte Carlo*) та на найгірший випадок (*Worst Case*). Для розрахунку чутливості та найгіршого випадку використовують результати аналізу Монте – Карло. Усі види варіації та статистичних випробувань доступні у будь якому виді одноваріантного аналізу.

Алгоритм розв’язання вихідних систем моделі в залежності від виду аналізу показано на рис. 1.9.

Вид аналізу	Аналіз динаміки <i>Transient</i>	Аналіз статички <i>DC</i>	Малосигнальний аналіз <i>AC</i>
аргументи	час	фазові змінні	частота
Форма ММС	СНДР	СНАР	СЛАР
Назва етапу	дискретизація	лінеаризація	розв’язання
Методи розв’язання	Трапецій Гіра (ФДН)	Ньютона-Рафсона	LU-факторизація
	ітераційні методи		прямий метод

Рисунок 1.9 – Алгоритм формування і розв’язання математичних моделей схем

Усі сучасні програми ECAD мають потужні постпроцесори для обробки отриманих результатів: побудови методом лінійної інтерполяції графіків, виділення, панорамування, обробки і допоміжних обчислень над отриманими графіками. Функції постпроцесори відрізняються в залежності від виду аналізу і доступні через контекстне меню, яке активується у вікні завдання функції для аналізу. Майже усі можливості постпроцесора доступні при аналізі перехідних процесів (оскільки в цьому виді аналізу модель системи є найскладнішою, найбільш повною).

1.6 Критерії вибору та огляд САЕ-систем

Сучасні САЕ (структура – в додатку Б) мають наступні риси:

- підтримка ієрархічного проектування як зверху – вниз, так і знизу – вгору, за рахунок реалізації багаторівневого моделювання і методу локальної деталізації моделі;

- компонентне моделювання на основі використання бібліотек, що містять велику кількість графічних і функціональних описів компонентів, причому ці бібліотеки відкриті для додавання в них нових описів користувачем;

- графічний інтерфейс, що поєднує графічні засоби формування візуального образу досліджуваного технічного пристрою з автоматичною генерацією моделі всієї схеми по її структурному опису;

- наявність інтерактивного робочого середовища проектування (керуючої оболонки, монітора), тобто спеціальної програми, з якої можна запускати всі або більшість інших програм пакета, не звертаючись до послуг штатної операційної системи;

- наявність в сучасних САПР і САЕ постпроцесорів моделювання, що дозволяє не тільки переглядати в зручній для користувача формі результати моделювання, але й обробляти ці результати;

- наявність вбудованих засобів чисельного моделювання в режимі реального часу або в режимі масштабування модельного часу;

- реалізація механізмів просування модельного часу, заснованих як на принципі Δt , так і на принципі Δz ;

- інтегрованість з іншими пакетами аналогічного призначення, яка забезпечується відповідними програмами – конверторами, що дозволяють імпортувати і експортувати дані з однієї системи в іншу;

- наявність засобів, що забезпечують формування віртуальних аналогів вимірювально-керуючої апаратури.

Якщо САЕ призначена для вирішення дослідницьких завдань, то до перерахованих характеристик додаються можливості активного обчислювального експерименту, зокрема:

- а) візуалізація результатів під час експерименту;
- б) можливість інтерактивного втручання в хід моделювання;
- в) можливість використання 2D і 3D анімації.

1.7 Етапи виконання роботи

1 Ознайомитися і встановити програмне забезпечення для моделювання (Micro-Cap, 20-sim).

2 Засвоїти правила компонування схем/моделей у редакторі схем ECAD.

3 Дослідити склад меню аналізу і теоретичне підґрунтя одного-багатоваріантних видів аналізу.

4 Дослідити функції постпроцесора.

5 Пояснити у висновках різницю в підходах моделювання систем на прикладах моделей з файлу System1.cir та System2.cir.

Примітка: п.п. 2-4 виконати на прикладі готових бібліотечних файлів з меню DATA MC12 System1.cir та System2.cir.

1.8 Контрольні запитання

- 1 Поняття та приклади динамічних систем.
- 2 Форми представлення моделей (приклади).
- 3 Принципи вибору програми для дослідження систем.
- 4 Способи моделінгу динамічних систем.
- 5 Характеристики моделі.
- 6 Критерії вибору програми моделювання.
- 7 Показники якості моделювання (симуляції) та моделей.
- 8 Мультидоменне моделювання.
- 9 Каузальне моделювання.
- 10 Структура ECAD.
- 11 Чим програми моделювання відрізняються від програм проектування?
- 12 Особливості аналізу перехідних процесів.
- 13 Функції постпроцесорів в ECAD.

2 ЛАБОРАТОРНА РОБОТА №2 “МОДЕЛЮВАННЯ ТРАКТІВ ПЕРЕДАЧІ ДАНИХ”

Мета: дослідити програми моделювання на макрорівні та підходи до моделювання каналів зв'язку/трактів передачі даних.

Для моделювання на макрорівні частіше використовують математичні процесори та системи автоматизованого інжинірингу (CAS та CAE відповідно).

На даний момент існує багато програм рівня CAE, які дозволяють, зокрема, моделювати (*modeling & simulation*) динамічні системи (ДС) різної природи (біологічні, економічні, соціальні, технічні), причому деякі є вузькоспеціалізованими, а інші – більш універсальними: Vensim, IThink, Dynamo, Stella, Powersim, MedModel, Arena, GPSS, Stratum, Scilab, Berkeley Madonna, NI MATRIXx, ACSLx, Modular Modeling System (MMS), Virtual Test Bed (VTB), JModelica.org, MATLAB\Simulink, Simscape, MapleSim, Dynast, Multisim, K2.SimKernel, Jigrein, EASY5, AMESim, Dymola, PSIM, SamSim, SimApp, SimulationX, Simplorer, VisSim, SystemModeler, 20-sim та інші. Багато з них придатні для дослідження систем захисту інформації.

Оскільки моделі на рівні схем можуть бути представлені безпосередньо диференціальними рівняннями, то для прогнозування їх поведінки у часі також можуть бути використані математичні пакети, CAS (*Computer Algebra System*), які можуть розв'язувати диференціальні рівняння (чисельно чи у символічній формі): Maple, MATLAB, Mathcad, Mathematica, Maxima, SymPy, O-Matrix, SciPy, Octave, NumPy, Python(x,y), MuPAD, Sage та інші.

Далі розглядатися будуть технічні динамічні системи, отже увага буде зосереджена саме на CAE-системах для моделювання технічних об'єктів та систем, що володіють засобами графічного опису та представлення моделей.

Нижче розглянуті доступні CAE-програми, де під словом «доступні» розуміються програми, інформацію про які можна відносно легко отримати з всесвітньої мережі чи інших джерел, або ж програми. Більшість з перелічених програм моделювання технічних динамічних систем розвиваються та підтримуються виробниками. Деякі з них, при наявності відповідної інформації, будуть розглянуті за пев-

ним шаблоном, який складається з наступних позицій:

- короткий опис програми (можливості, особливості);
- архітектура (структура) програми;
- можливі форми представлення моделей на етапі моделювання (*modeling*);
- традиційні можливості для дослідження САК;
- перелік методів моделювання (*simulation*), тобто алгоритмів симуляції або інтегрування;
- перелік алгоритмів параметричної оптимізації;
- доступність програми, види ліцензій та редакцій, вартість та наявність безкоштовних версій.

2.1 SamSim

Програма призначена для моделювання лінійних і нелінійних САК. Працює з моделями, які можна представити у формі блок-схем (в програмі 48 типових блоків). За допомогою цієї програми можливо:

- побудова будь-яких схем моделей з елементів, пропонованих бібліотеками;
- завдання параметрів інтегрування і параметрів елементів;
- збереження в файлі і зчитування з файлу моделі (схеми та її параметрів);
- побудова залежностей від часу у будь-яких точках схеми;
- побудова фазових портретів для будь-яких схем;
- побудова частотних характеристик і годографів для будь-яких лінійних схем;
- представлення результатів розрахунку в графічній і табличній формі;
- збереження результатів розрахунку в текстовому файлі, графіків в *bmp* і *jpg*;
- експорт результатів розрахунку в MS Excel;
- виведення на друк схеми моделі та її параметрів, результатів розрахунку.

Програма має всього одну редакцію, яку можна скачати з сайтів автора: samsim.fromru.com, samsim2002.chat.ru, samsim2002.narod.ru. Не потребує встановлення, працює з поточного каталогу. Є файл допомоги і приклади схем. Розроблена в 2002 році, поширюється вільно.

Програма має три алгоритми інтегрування (солвера/розв'язувача): 1-го порядку точності; 2-го порядку точності; 4-го

порядку точності. Тобто, передбачається використання методів Ейлера і Рунге-Кутти. SamSim не має засобів автоматичної структурної або параметричної оптимізації моделі.

Приклад роботи програми можна побачити на рисунку 2.1.

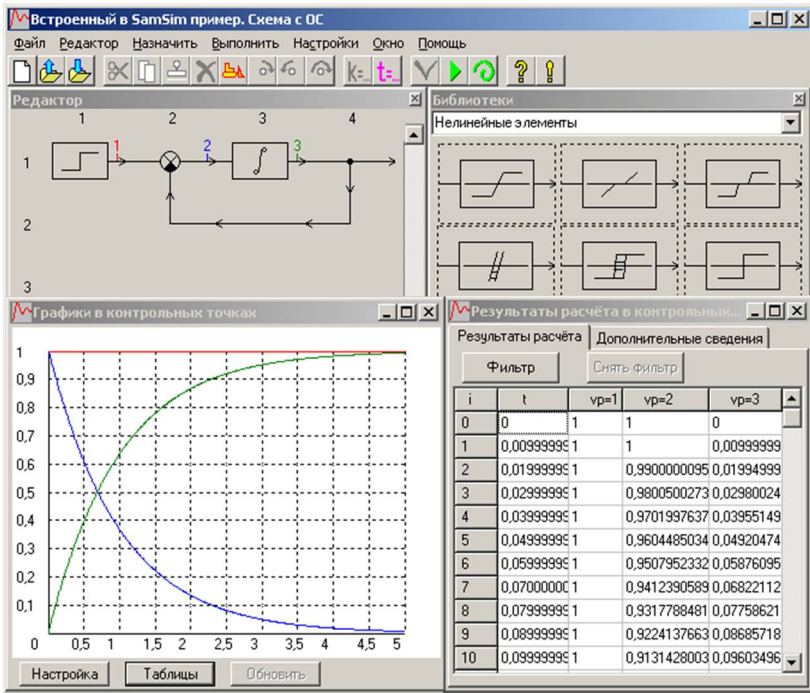


Рисунок 2.1 – Дослідження моделі в програмі SamSim

Програма проста, наочна, може бути використана для попереднього моделювання систем, відпрацювання та дослідження чисельних методів розрахунку. Дозволяє представляти модель системи тільки як блок-схему, типові ланки САК (лінійні) задаються через передаточну функцію (ПФ), в якій можна редагувати коефіцієнти чисельника і знаменника. SamSim може будувати АЧХ, ФЧХ, АФЧХ і ЧХ в логарифмічному масштабі, також може здійснювати моделювання в часовій області.

Однак, є певні недоліки, наприклад, відсутні можливості для розрахунку нулів і полюсів ПФ моделі. Моделювання (simulation) у ча-

совій і частотній областях неможливо без зміни моделі: для отримання частотних характеристик необхідно в якості джерела сигналу ставити генератор коливачої (змінної) частоти ГКЧ. Не має автоматичних засобів визначення показників стійкості та якості САК.

2.2 ANSYS Simplorer (VHDL)

Прикладом мови моделювання дискретних електронних пристроїв на логічному рівні може бути мова VHDL (Very high-speed integrated circuits Hardware Design Language), затверджена у якості міжнародного стандарту IEEE 1076; яка успішно використовується також при синтезі пристроїв. Мова VHDL була модифікована і тепер містить засоби опису аналогових і змішаних моделей. Ці описи разом з базовим варіантом мови VHDL утворюють мову VHDL-AMS (AMS – Analog and Mixed Signal), тобто її можна використовувати для моделювання аналогово-цифрових пристроїв та мультідоменного моделювання. VHDL – не єдина мова проектування дискретних пристроїв САК. Так, для проектування інтегральних схем широко застосовується Verilog, близька до C++ мова SystemC, мова опису ланцюгів EDIF, мови проектування пристроїв на ПЛІС і ін., але для функціонального проектування (вище схемотехнічного) найбільше поширення одержали VHDL і Verilog. В VHDL також є засоби для поведінкових і структурних описів.

ANSYS Simplorer – це програмне забезпечення для моделювання електромеханічних систем, у яких потрібне проведення міждисциплінарних розрахунків, що й створює віртуальну електромеханічну лабораторію на робочому столі комп'ютера. Розробка мультідоменної системи є складним завданням, що вимагає великих зусиль. У таку систему входить безліч нелінійних компонентів з різних областей фізики (доменів): електричної, механічної, термічної і з області управління. ANSYS Simplorer – системний інструмент, що дозволяє паралельно використовувати кілька стандартних технік моделювання (включає принципові схеми, блок-схеми, кінцеві автомати, рівняння) і мов моделювання (VHDL-AMS, SML – Simplorer Modeling Language, C/C++). Програма дозволяє моделювати взаємодію компонентів у різних інженерних областях.

У загальному випадку в Simplorer 8 можливі два методи інтегрування (simulation) та їх варіації (в залежності від технології побудови моделі): метод Ейлера та метод трапецій (неявний). Метод Ейлера

можна встановити явний (explicit) або неявний (implicit), прямий (Forward Euler) або зворотний (Backward Euler); також можливо вибрати адаптивний метод трапецій-Ейлера (Adaptive Trapezoid-Euler).

Після побудови моделі у вигляді принципової схеми (електричної) програма, використовуючи модифікований вузловий підхід (modified nodal approach), будує систему нелінійних диференціальних алгебраїчних рівнянь (у загальному випадку) та використовує методи Ейлера або трапецій для чисельного інтегрування. При цьому нелінійні рівняння лінеаризуються за допомогою ітераційного методу Ньютона-Рафсона (метод дотичних), а потім вже лінеаризована система рівнянь вирішується за допомогою LU-факторизації.

Також можлива параметрична оптимізація, методами: Sequential Nonlinear Programming (метод послідовного нелінійного програмування), Sequential Mixed Integer NonLinear Programming, Quasi Newton (квазі-ньютонівський метод), Pattern Search (пошук з використанням шаблону), Genetic Algorithm (генетичний алгоритм пошуку екстремуму з використанням природного відбору).

Програма дозволяє робити три види аналізу (перехідних процесів, на змінному струмі та на постійному струмі), включаючи побудову діаграм Бode та годографа Найквіста (вже мають відповідні передуповнення). Програма «перевантажена» з точки зору інтерфейсу, має «зайві» модулі, які перешкоджають візуальному сприйняттю та уповільнюють знайомство з програмою.

Маючи ефективну та гнучку VHDL-AMS мову опису моделей, ця програма більше пристосована (підходить) для моделювання на схемотехнічному рівні, що підтверджує наявність у бібліотеці великої кількості моделей напівпровідникових приладів та інструментів для створення апроксимації характеристик компонентів (component characterization tool).

Simplorer дозволяє автоматично вирішувати алгебраїчні петлі (або жорсткі зворотні зв'язки), але алгоритму симуляції потрібно більше ітерацій, а іноді навіть можуть виникати чисельні проблеми з вирішувачем (solver) втрата стійкості та збіжності.

2.3 SimApp

SimApp це програма моделювання поведінки ДС в часовій та частотній областях. Можливе моделювання (modeling & simulation) безперервних і дискретних систем. Найкраще підходить для дослід-

дження систем автоматичного управління. Графічний, блочно-орієнтований інтерфейс з більш ніж 80 основних блоків в бібліотеці, розділених на різні категорії. Можливе створення нових блоків в спеціальному вікні дизайну, збереження часто використовуваних підсистем в бібліотеки для подальшого використання. Результати моделювання представлені у вигляді набору даних в табличній формі, а також у вигляді діаграм: часової, Боде, Нікольса-Блека та Найквіста, які можна експортувати в інші додатки для подальшої обробки. Можливо обчислити власні значення частот (eigenvalues, eigenfrequencies) лінійних систем. Програма підтримує дослідження систем з різними значеннями параметрів і різними зовнішніми збуреннями. SimApp відрізняється від інших подібних продуктів завдяки інтуїтивному управлінню і простотою у використанні, що дозволяє отримати результати протягом декількох хвилин.

Крім того, програма має докладну довідку з функцій та стандартних блоків. SimApp версії 2.6 існує в чотирьох різних редакціях, які відрізняються з точки зору максимального розміру системи (кількість блоків у проекті), функцій і вартості покупки: Workstation (UAH 3,072.84 / EUR 290.00), Light (UAH 1,271.52 / EUR 120.00), Student (UAH 529.80 / EUR 50.00) та Server Edition. Остання має чотири типи ліцензій, їх ціни відрізняються. Пробна версія SimApp 2.6 (Workstation edition 30-day Free Trial) дозволяє розташовувати у проекті всього лише 10 блоків.

Перелік доступних алгоритмів інтегрування в SimApp 2.6:

- Euler (Ейлера);
- Euler-Cauchy (метод Ейлера-Коші для неоднорідних ODE);
- Heun predictor-corrector method (багатокроковий метод прогнозу-корекції за схемою Хойна, з предиктором Ейлера та корекцією методом трапецій);
- Runge-Kutta order 3 (метод Рунге-Кутти третього порядку);
- Runge-Kutta order 4 (як попередній, тільки четвертого порядку точності).

SimApp не має структурної чи програмної оптимізації. Хоча у довідкових додатках до програми і викладається дві методики Зіглера-Нікольса (Ziegler-Nichols) для автоматизованого (табличний розрахунок у Microsoft Excel) тюнінгу ПІД регуляторів, але відсутність параметричної оптимізації робить її неконкурентоспроможною у випадку, якщо потрібен буде автоматичний пошук коефіцієнтів для створення

оптимального регулятора. Також SimApp не може моделювати (simulate) у часовій області алгебраїчні петлі (algebraic loops) без встановлення у коло зворотного зв'язку ланки затримки чи аперіодичної ланки (бо вихідна величина безпосередньо залежить від вхідної та вихідної величини), тобто процес подолання проблеми алгебраїчних петель у кращому випадку автоматизований, але не автоматичний у цій програмі. Зараз вже існують програми, що можуть самостійно (тобто автоматично) вирішувати проблему алгебраїчних петель за рахунок символічних перетворень на етапі створення рівнянь ДС перед процесом симуляції.

2.4 Wolfram SystemModeler (MathModelica)

Раніше цей проєкт мав назву MathModelica, потім його викупила компанія Wolfram. Wolfram SystemModeler, зі слів виробника, є найбільш повним інструментальним засобом для фізичного і чисельного моделювання. На відміну від інших систем, SystemModeler не вимагає придбання додаткових компонентів (як у MapleSim та Simulink) і в повній мірі підтримує стандартну мову фізичного моделювання Modelica. Також SystemModeler передбачає повне з'єднання з системою Mathematica для отримання найбільш інтегрованого робочого процесу фізичного, чисельного моделювання і аналізу. SystemModeler реалізує мову Modelica, яка користується перевагами компонентно-орієнтованого моделювання, де моделюються потоки в компонентах. Це дає суттєві переваги порівняно з блочно-орієнтованим моделюванням. Архітектура програми представлена на рисунку 2.2.

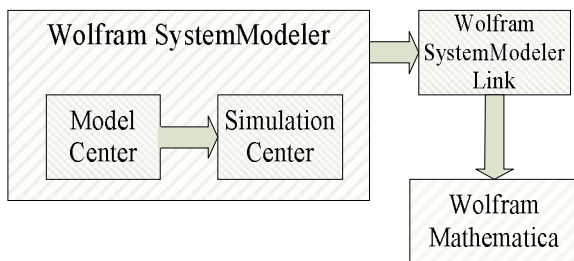


Рисунок 2.2 – Структура програми SystemModeler та зв'язок з Wolfram Mathematica

Серед функціональних можливостей Wolfram SystemModeler можна відзначити:

- використання за основу непатентованої, об'єктно-орієнтованої мови Modelica, що спирається на рівняння;
- графічний інтерфейс для drag-and-drop моделювання;
- текстовий інтерфейс для моделювання на мові Modelica за допомогою рівнянь;
- некаузальне (компонентно-орієнтоване) і каузальне (блочно-орієнтоване) моделювання;
- ієрархічне моделювання (modeling);
- велика кількість вбудованих бібліотек моделей;
- моделювання гібридних систем;
- багатодоменне моделювання, що включає: механіку (в одному і в трьох вимірах), електроніку, гідравліку, термодинаміку, елементи керування, системну біологію;
- створення автономного *.exe файлу для подальшого відтворення результатів моделювання (simulation) без виклику SystemModeler;
- інтеграція з системою Mathematica для аналізу та документування моделей і результатів комп'ютерного симулювання.

Є кілька редакцій цієї програми. Спеціальна редакція для студентів (Annual and Semester Editions – річна та семестрова редакції) коштує \$75 та \$35, академічна редакція – \$350, державна редакція – \$1838 та комерційна редакція – \$2450. Також є безкоштовна пробна версія, що може працювати протягом 30 днів з моменту інсталяції.

SystemModeler має наступні солвери (solvers) або вирішувачі (розв'язувачі):

- DASSL (метод зі змінними шагом та порядком, що використовує формулу зворотного диференціювання);
- CVODES (як і попередній, але ще підтримує передній аналіз чутливості);
- Explicit Euler (явний метод Ейлера);
- Heun's method (алгоритм Хойна);
- Runge-Kutta 4 (метод Рунге-Кутти 4-го порядку);

Три останніх метода мають постійний розмір (довжину) кроку та рекомендуються до використання при симуляції в реальному часі.

Інтерфейс пробної версії SystemModeler 3.0.0 зображений на рисунку 2.3.

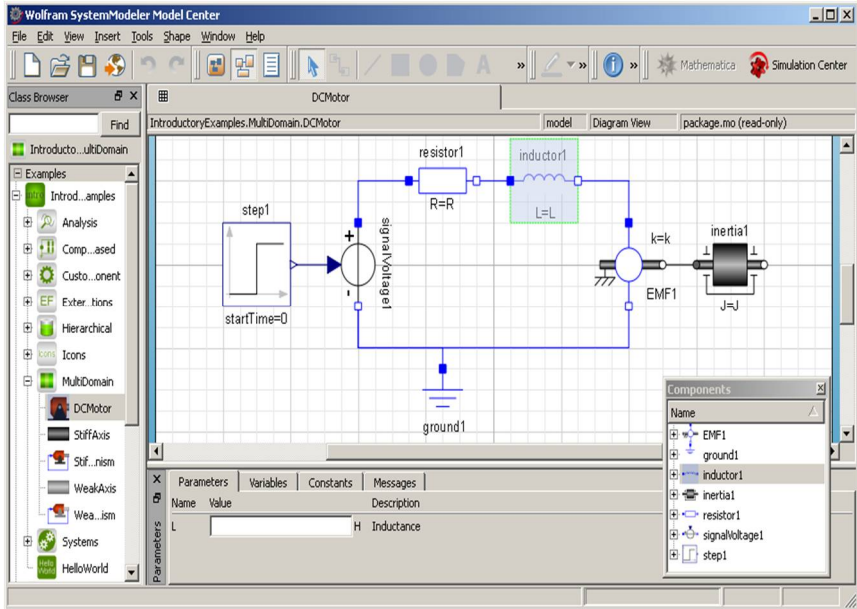


Рисунок 2.3 – Головне вікно Model Center з відкритим прикладом (DCMotor)

SystemModeler не має модулів оптимізації, оскільки є у деякому сенсі дочірньою програмою, тому для виконання параметричної оптимізації треба мати батьківську програму Wolfram Mathematica, яка купується за окремі кошти. Model Center та Simulation Center рознесені як окремі програми та мають різні виконавчі файли .exe, що збільшує час переходу від однієї до іншої та незручні інструменти і підхід (спосіб) до моделювання (modeling), тобто створення моделі. Також програма не має інтерактивної довідкової системи та передумовлених можливостей для відображення та аналізу частотних характеристик (АЧХ, ФЧХ, різні годографи), що є важливим аспектом при дослідженні САК.

2.5 MapleSim

Пакет MapleSim являє собою потужний інструмент імітаційного моделювання для самих різних областей застосування. Реалізовані в ньому технології фізичного моделювання деталей і систем допоможуть конструкторам швидше готувати свої творіння до виробництва.

Пакет дозволяє просто намалювати блок-схему роботи потрібної системи, і MapleSim автоматично згенерує всі рівняння, що описують роботу моделі. На думку авторів, MapleSim відкриває нову еру в конструюванні інженерних систем, скорочуючи тривалість циклу розробки з декількох місяців до декількох тижнів. У моделях пакета MapleSim можна комбінувати фізичні компоненти з традиційними логічними блоками ланцюгів управління. Блок-схема моделі може виглядати точно так само, як описувана нею реальна система. З готових компонентів конструктор може зібрати найскладнішу і багатофакторну модель. Чіткий контроль використовуваних одиниць виміру допомагає уникнути помилок при переході від однієї системи одиниць до іншої. Також в ході конструювання моделі можна одночасно готувати всю супровідну документацію до проектних робіт. Розвинуті алгоритми символічних обчислень дозволяють автоматично спрощувати складні моделі. Крім того, відмова від традиційного чисельного моделювання на користь символічних можливостей платформи Maple допомагає пакету MapleSim домогтися небувало високої точності моделювання і уникнути помилок, пов'язаних з ручним перерахунком формул.

Пакет MapleSim містить більше 500 готових модельних компонентів для 10 прикладних областей, таких, як електричні мережі та пристрої, механіка, системи управління та теплові мережі. Всі ці компоненти систематизовані і розбиті на зручні набори. На даний момент безкоштовна версія пакету MapleSim поширюється у вигляді пілотної демо-версії. Щоб випробувати можливості MapleSim на практиці, потрібно зареєструватися для участі в пілотній програмі на сайті розробника та дочекатися поки з вами зв'яжуться представники компанії для огляду програми у Live-режимі. Вигляд головного вікна програми відповідає рисунку 2.4.

Зі слів авторів програми MapleSim 6, вона володіє наступними ключовими особливостями:

- потужне символічне та числове математичне ядро (Math Engine);
- потужна платформа Modelica (для акаузального моделювання);
- мультидоменне моделювання із вбудованими компонентами;
- середовище фізичного моделювання з технікою Drag-and-Drop;
- підтримка технології багатьох тіл (Multibody Technology);
- генерація оптимізованого коду;
- широкі опції вирішувачів при симуляції;

- доступність рівнянь структурного (системного) рівня;
- керування одиницями вимірювання для кожного об'єкту;
- широкий спектр можливих видів та способів інтерактивного аналізу;
- функції візуалізації у 2D та 3D;
- інтеграція усієї інформації стосовно проекту та зручне її представлення і зберігання у єдиному файлі.

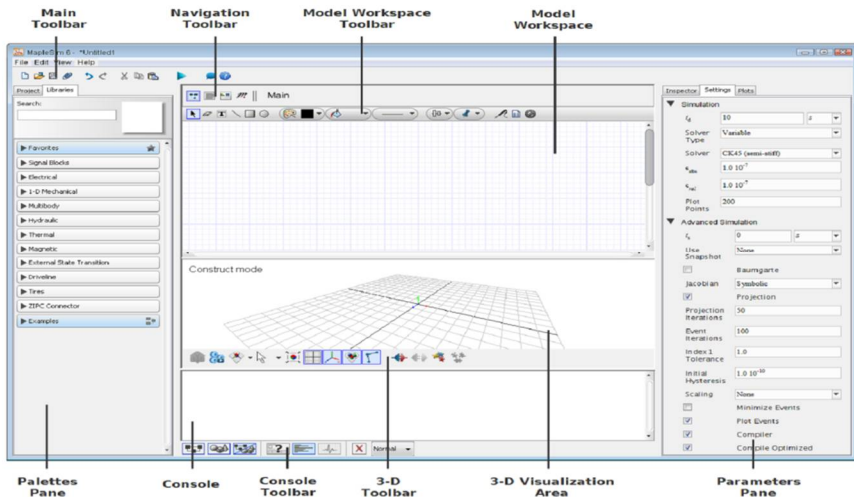


Рисунок 2.4 – Головне вікно програми MapleSim 6

Є кілька редакцій програми MapleSim 6, що різняться функціональністю та ціною: Student Edition (\$124), Commercial (\$6895), Academic (\$1555), Government (\$6550). При чому батьківська програма Maple продається окремо (\$2,845.00 – комерційна версія, \$124 – студентська версія). Єдиною безоплатною редакцією програми є редакція Live Demo, але вона є важкодоступною. Документація доступна з сайту компанії MapleSoft: www.maplesoft.com.

Для етапу симуляції програма пропонує жорсткі, нежорсткі, напівжорсткі адаптивні та неадаптивні чисельні вирішувачі (numerical solvers), до складу яких входять алгоритми інтегрування зі змінним та постійним кроком:

- Forward Euler (простий метод Ейлера з фіксованим кроком);
- Implicit Euler (неявний метод Ейлера з постійним кроком для жорстких систем);

- RK2, RK3, RK4 (методи Рунге-Кутти з фіксованим кроком відповідного порядку);
- RK45 (метод Рунге-Кутти-Фелберга з подвійним порядком точності та змінним кроком для рішення нежорстких систем);
- CK45 (метод Кеша-Карпа для рішення напівжорстких систем);
- Rosenbrock (метод Розенброка для рішення жорстких систем DAE);

З одного боку MapleSim має і багато вбудованих бібліотек, і язык Modelica для описання моделей, а з іншого – повна неможливість роботи програми без батьківського пакету Maple (який купується за окремі кошти). Виходячи з довідкових матеріалів до програми, вона не має вбудованих модулів оптимізації (як і SystemModeler) і дозволяє робити оптимізацію лише у Maple; для досягнення кращих результатів (стосовно оптимізації) пропонується використовувати як доповнення Global Optimization Toolbox, який має додаткові методи оптимізації та спеціалізується на цьому, але його вартість не врахована ні в MapleSim, ні в сам Maple, отже потрібно докуповувати це доповнення окремо.

2.6 MATLAB / Simulink та Simscape

Система математичного моделювання MATLAB / Simulink (компанії MathWorks) є в даний час одним з найбільш популярних інструментів чисельних розрахунків і застосовується в різних областях знань. Головна особливість середовища MATLAB – ретельна опрацьованість і налагодженість всього багатого арсеналу засобів і методів.

Можливості MATLAB перекривають більшу частину потреб розробників систем управління в різних областях техніки. Широкий набір різних чисельних методів у поєднанні з потужними засобами графічної візуалізації робить MATLAB універсальним інструментом інженерних розрахунків і наукових досліджень. Серйозними перевагами середовища MATLAB є її відкритість і розширюваність. Більшість команд і функцій системи реалізовані у вигляді текстових m-файлів (файлів з розширенням .m) і файлів на мові Сі, причому всі файли доступні для модифікації. Це дозволяє користувачеві створювати не тільки окремі файли, але і цілі бібліотеки файлів, формуючи, таким чином, власне предметно-орієнтоване середовище моделювання.

Найважливішою складовою середовища MATLAB є пакет структурного моделювання динамічних систем Simulink. Розробку цього пакету можна розглядати як принципову модернізацію середовища MATLAB, в результаті якої вона придбала всі риси сучасної CAE – компонентне моделювання, графічна форма завдання інформації про об'єкт і т.п. Simulink настільки органічно інтегрований з системою MATLAB, що, розглядаючи сучасні засоби дослідження технічних систем, логічно говорити про середовище візуального компонентного моделювання MATLAB / Simulink.

На базі пакету Simulink розроблені додаткові бібліотеки блоків для різних областей застосування (наприклад, Power System Blockset – моделювання електротехнічних пристроїв, Digital Signal Processing Blockset – набір блоків для розробки цифрових пристроїв, Control System Toolbox – пакет для розробки систем управління і т.д.). Можна очікувати, що така політика буде продовжена і в наступних модифікаціях середовища MATLAB, з'являться нові предметно-орієнтовані бібліотеки.

При моделюванні в Simulink користувач може скористатися досить широкою палітрою методів рішення диференційних рівнянь, а також вибрати спосіб зміни модельного часу (з фіксованим або змінним кроком), що дозволяє проводити моделювання для широкого кола систем, що включають безперервні, дискретні і гібридні системи будь-якої розмірності. Окремий напрямок досліджень пов'язаний з можливим переходом в область лінеаризованих систем, де реалізовано весь арсенал методів лінійного аналізу. Засоби візуалізації дають можливість стежити за процесами, що відбуваються в системі. Для цього використовуються спеціальні пристрої спостереження, що входять до складу бібліотеки Simulink.

Результати моделювання можуть бути представлені у вигляді графіків або таблиць. Базова бібліотека Simulink містить більше 200 блоків, що найчастіше зустрічаються при моделюванні різних систем. Додаткові бібліотеки дозволяють розширити можливості Simulink для застосування в аерокосмічній галузі, обробці сигналів, зв'язку та в інших додатках. Бібліотеки блоків MATLAB / Simulink можуть поповнюватися за рахунок підпрограм, написаних як на мові середовища MATLAB, так і на мовах C++, Fortran і Ada. Крім того, в ньому реалізована процедура інкапсуляції, яка є ще одним зручним для користувача засобом розширення бібліотек.

Недоліки MATLAB / Simulink є загальними для всіх інструментів структурного моделювання. Як у всіх подібних пакетах при створенні складних моделей доводиться будувати досить громіздкі багаторівневі блок-схеми, що не відображають природної структури моделюваної системи. Частково, цей спільний недолік пакетів структурного моделювання усувається використанням додатків, в яких реалізований принцип фізичного мультидоменного моделювання, таких як SimMechanics і SimPowerSystems, що забезпечують можливість побудови моделей механічних та електричних систем відповідно, а також тим, що розробники більшості сучасних пакетів фізичного мультидоменного моделювання передбачають можливість підключення бібліотек Simulink до інших програм.

Програму Simscape і MATLAB \ Simulink пов'язує одна компанія-виробник – MathWorks. Проте, є суттєва відмінність між Simulink та Simscape, остання володіє широким спектром функцій та можливостей фізичного мультидоменного моделювання. Simscape надає середовище для моделювання та симуляції фізичних систем у механічній, електричній, гідравлічній та інших фізичних областях. Вона забезпечує фундаментальними елементами (блоками) з цих областей, які можна зібрати в моделі фізичних компонентів, таких як електричні двигуни, операційні підсилювачі, гідравлічні клапани та інші механізми. Моделі будуть відповідати структурі розроблюваної системи, тому що компоненти Simscape використовують фізичне з'єднання.

Моделі Simscape можуть бути використані для розробки систем управління і тестування продуктивності системного (структурного) рівня. Бібліотеки елементів можуть бути розширені за рахунок використання основаної на MATLAB мови Simscape, яка дозволяє текстове редагування та створення фізичних компонентів, доменів і бібліотек. Є можливість параметризування моделі з використанням змінних та виразів MATLAB, а також проектування моделі об'єкту управління фізичної системи, система управління якої може бути побудована в Simulink. Для розгортання моделі в інших середовищах моделювання Simscape підтримує генерацію Сі-коду.

Для основних продуктів компанії MathWorks (MATLAB, Simulink, Simscape) можливі три типи ліцензій (Commercial, Academic, Student), причому у відкритому доступі цін на продукти з цими ліцензіями немає, а для отримання прайсу потрібно авторизуватися на офіційному сайті та відправити відповідний запит. Для використання

Simscape обов'язково потрібна наявність і Simulink, і MATLAB. Враховуючи, що кожна програма постачається за окрему плату, такий набір буде коштувати чималу суму: MATLAB (USD \$2,150.00) + Simulink (USD \$3,250.00) + Simscape (USD \$2,000.00) = USD \$7,400.00. Причому це тільки мінімальна конфігурація, вже не кажучи про додаткові важливі пакети SimMechanics (USD \$2,000.00), SimHydraulics (USD \$2,000.00), SimElectronics (USD \$2,000.00), SimPowerSystems (USD \$3,000.00), Simulink Control Design (USD \$1,000.00), Simulink Design Optimization (USD \$1,000.00), Control System Toolbox (USD \$1,000.00), Optimization Toolbox (USD \$1,000.00), Global Optimization Toolbox (USD \$1,000.00). Також треба відмітити, що компанія MathWorks не надає безкоштовні пробні версії студентам і приватним особам (Individual License).

Навіть якщо продукти компанії MathWorks і перекривають усі запити споживачів (інженерів, науковців) в багатьох областях науки і техніки стосовно електронних розрахунків, моделювання, аналізу, оптимізації та інших видів процесінгу в віртуальній лабораторії, то ціна на ці продукти відповідна. Є безкоштовна пробна версія (free 30-day trial) програми Simulink (включаючи MATLAB).

В програмі MATLAB широкі можливості щодо рішення ODE; вирішувачі поділяються на методи зі змінним та постійним кроком. Параметрична оптимізація у MATLAB можлива за допомогою додаткового інструментарію Optimization Toolbox – прикладний пакет для розв'язання оптимізаційних задач і систем нелінійних рівнянь. Підтримує основні методи оптимізації функцій ряду змінних:

- безумовна оптимізація нелінійних функцій;
- метод найменших квадратів та нелінійна інтерполяція;
- рішення нелінійних рівнянь;
- лінійне програмування;
- квадратичне програмування;
- умовна мінімізація нелінійних функцій;
- метод Мінімакс;
- багатокритеріальна оптимізація.

Optimization Toolbox реалізує наступні методи гладкої безумовної оптимізації:

- а) метод найшвидшого спуску (Steepest Descent method);
- б) квазі-Ньютонавський BFGS-метод (Broyden-Fletcher-Goldfarb-Shanno);

в) квазі-Ньютонівський DFP-метод (Davidson-Fletcher-Powell);
 г) методи, засновані на побудові довірчих двовимірних областей (trust region).

Перші три з них віднесені до «medium-scale»-методів і призначені для вирішення завдань оптимізації середньої розмірності (не більше 100 змінних), останній – до «large-scale»-методів. За замовчуванням, якщо задається аналітичний вираз для градієнта, запускається метод довірчого інтервалу. В іншому випадку використовується BFGS-метод, але є можливість переключитися і на інші. Метод найшвидшого спуску, як правило, дуже повільний і його не рекомендується використовувати для вирішення практичних завдань.

2.7 VisSim

VisSim – пакет компонентного візуального моделювання фірми Visual Solutions, призначений для розробки і моделювання динаміки безперервних, дискретних та гібридних систем. За своїми функціональними можливостями і способом завдання початкової інформації VisSim близький до пакету MATLAB / Simulink. Simulink більш поширений, тим не менш, VisSim має чимало шанувальників і займає помітну частку ринку інженерних пакетів. VisSim – один з найбільш яскравих представників систем, що реалізують концепцію структурного (каузального) моделювання.

Основним інструментом створення моделей є типові блоки «вхід-стан-вихід». Подібно MATLAB / Simulink кожен типовий блок VisSim реалізує математичну модель або забезпечує візуалізацію того чи іншого явища, процесу або пристрою. Базова бібліотека VisSim містить більше 120 лінійних і нелінійних блоків, згрупованих за функціональною ознакою, що дозволяють моделювати складні системи. Більшість бібліотечних блоків-підпрограм написані на мові C++, однак пакет надає можливість створювати користувацькі бібліотеки практично будь-якою процедурною мовою програмування. Процедура інкапсуляції дає можливість створювати нові блоки (підсистеми), грунтуючись на вже існуючих блоках.

Як і Simulink, VisSim має широкий набір інструментів оцінки якості, стійкості, синтезу, корекції, оптимізації, лінеаризації, налагодження об'єктів в контурі моделі та програмування цифрових сигнальних процесорів. Вирішувач VisSim інтерпретуючого типу, функціонує в динамічному режимі з можливістю online-взаємодії з обладнан-

ням реального часу. У складі пакету вирішувача VisSim реалізовані як явні, так і неявні методи. Можливості моделювання гібридних систем у пакеті VisSim приблизно аналогічні Simulink. VisSim призначений насамперед для моделювання безперервних систем. Загальним принципом гібридного моделювання в VisSim є використання готових гібридних блоків або перемикання заздалегідь заготовлених альтернативних ділянок блок-схем.

Подібно Simulink, пакет VisSim включає в себе ряд додаткових модулів, що розширюють його можливості у конкретних предметних областях, а так само інструменти інтеграції з пакетами MATLAB, Mathcad і транслятор проектів програми Simulink. В якості обмежень пакету VisSim можна відзначити відсутність альтернатив мові блок-схем.

Крім того, VisSim не відповідає вимогам об'єктно-орієнтованого моделювання, не підтримує спадкування, динамічну зміну кількості об'єктів. Тим самим він не дає можливості моделювати систему зі змінною у часі кількістю об'єктів. Але для восьмої версії VisSim існує додатковий модуль State Charts, який дозволяє оперувати UML-сумісними (Unified Modeling Language) графічними діаграмами станів і редагувати та генерувати Сі-код для кінцевих автоматів, створених у цьому середовищі. Більшість додаткових модулів VisSim представлені на рисунку 2.5.

VisSim, окрім інших редакцій, випускається у вигляді академічної (приблизна ціна – 5000грн.) та професійної (28000грн.) редакцій, версія VisSim 3.0 є безкоштовною (вільно розповсюджується) для персонального (академічного) використання з деякими обмеженнями. З сайту розробника ціни на нові версії програми отримати не можливо, поки не зробити відповідне замовлення або запит. Останніх свіжих версій VisSim у вільному доступі немає, але автори програми пропонують зареєструватися на їхньому сайті, щоб мати можливість зробити запит на безкоштовну пробну версію VisSim 8.0.

VisSim дозволяє використовувати декілька методів чисельного інтегрування диференційних рівнянь (методи Рунге-Кутти, Булірша-Стоєра, Адамса-Мултона та ін.) та проводити оптимізацію (методами Пауела, Полака-Рібєрі та Флетчера-Рівза); на рисунку 2.6 представлені доступні алгоритми інтегрування, а рисунок 2.7 відображає можливі методи параметричної оптимізації.

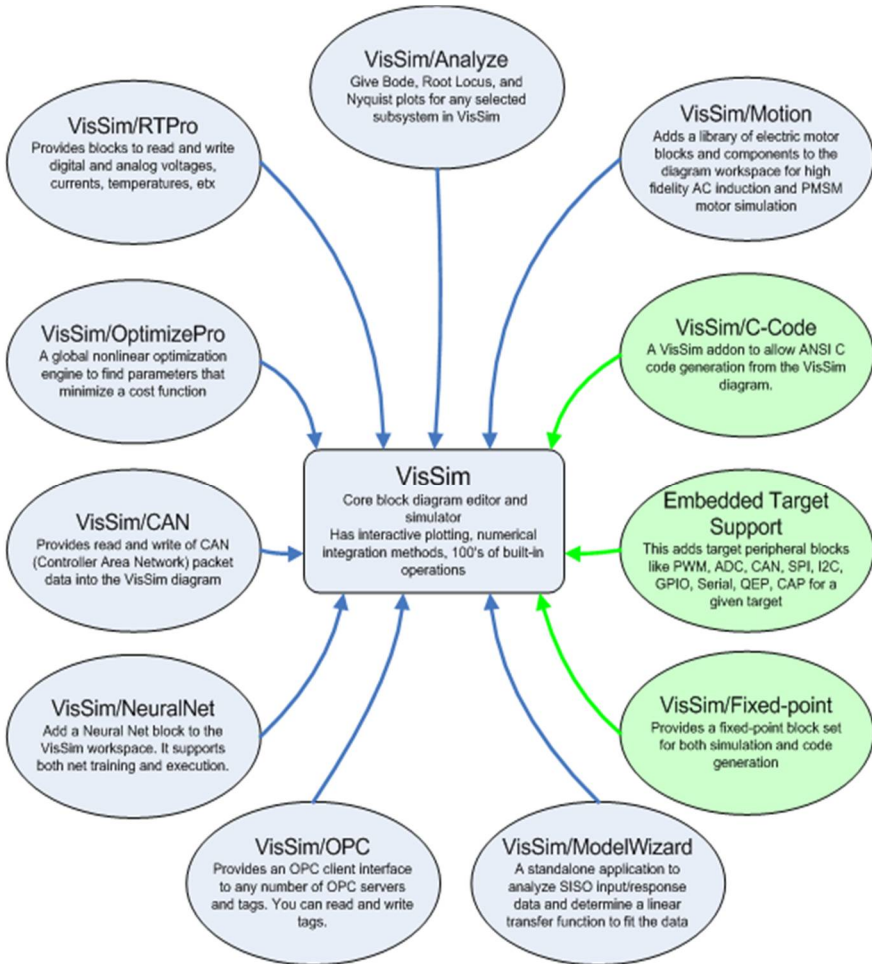


Рисунок 2.5 – VisSim та додаткові модулі

Перевагами програми є спрямованість на створення систем керування (відображення багатьох частотних характеристик, синтез коректуючих ланок), графічне представлення моделей, наявність потужного вирішувача та можливість оптимізації, велика кількість супроводжувачої російськомовної літератури через розповсюдженість та широку вживаність програми, взаємодія з розповсюдженими відомими програмними пакетами (MATLAB, Simulink, Mathcad, Maple, Excel).

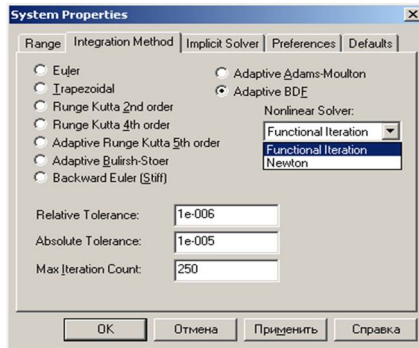


Рисунок 2.6 – Вікно вибору методів інтегрування в VisSim

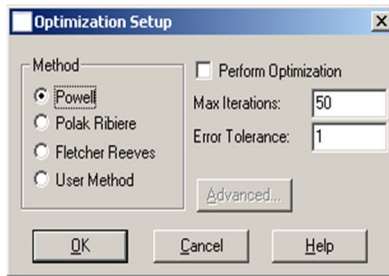


Рисунок 2.7 – Вікно вибору методів параметричної оптимізації в VisSim

Недоліки VisSim: неможливість автоматичного вирішення проблеми алгебраїчної петлі у багатьох випадках (пропорційна ланка, охоплена від'ємним зворотним зв'язком без інерційності у контурі); невдале виконання способу представлення результатів моделювання (графіків функцій часу та частоти) типу «вікно у вікні», незручні засоби аналізу отриманих результатів; проблема розташування ліній зв'язку між елементами та неможливість ручного редагування їх положення, що знижує візуальне сприйняття побудованої моделі та може призвести до помилкової подальшої передачі моделі після її подання у вигляді зображення (бо з'єднання більш ніж двох провідників не супроводжується появою вузла), що показано на рисунку 2.8.

Таким чином, візуальне представлення моделей та графічний інтерфейс користувача відіграють суттєву роль при виборі САЕ-програми, де порушуються не тільки питання ергономіки, але й адек-

ватності подальшого перетворення інформації.

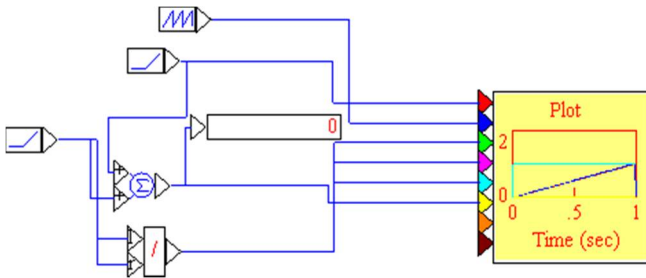


Рисунок 2.8 – Ілюстрація відсутності вузлів з'єднання в VisSim

2.8 Dymola (Modelica)

Dymola являє собою середовище візуального моделювання, що включає універсальну об'єктно-орієнтовану мову Modelica для моделювання складних фізичних систем у таких областях як мехатроніка, автоматика, аерокосмічні дослідження та ін. Мова Modelica, разом з ASCEND, Smile, VHDL-AMS та ін., належить до групи так званих акаузальних мов, тобто мов фізичного моделювання. Коштує Dymola приблизно \$3800.

Стандартна бібліотека Dymola включає елементарні компоненти, що відносяться до електротехніки, механіки, гідравліки і т.д. Спеціалізовані бібліотеки містять моделі пристроїв і явищ для конкретної області. Наприклад, MultiBody Library містить тривимірні механічні компоненти, призначені для моделювання роботів або транспортних засобів. Це моделі твердих тіл, шарнірів, джерел і засобів анімації.

Гнучкість і відкритість пакета Dymola дозволяє користувачам створювати свої власні бібліотеки моделей, описуючи їх на внутрішній мові опису блоків, або модифікувати вже існуючі моделі під свої конкретні потреби. Крім власної мови, Dymola (Modelica) підтримує інтеграцію з такими програмними середовищами, як Fortran, C, SIMULINK (M-files та SimStruct) і деякими іншими. Можливість взаємодії розроблених моделей з системою MATLAB / Simulink дозволяє об'єднати сильні сторони структурного та фізичного моделювання. Dymola має десять алгоритмів інтегрування зі змінним кроком для трьох можливих типів рівнянь, якими описується модель: ODE – зви-

чайні диференційні рівняння, DAE – диференціально-алгебраїчні рівняння, ODAE – перевизначені диференційно-алгебраїчні рівняння (Overdetermined DAE).

Архітектура програми представлена на рисунку 2.9.

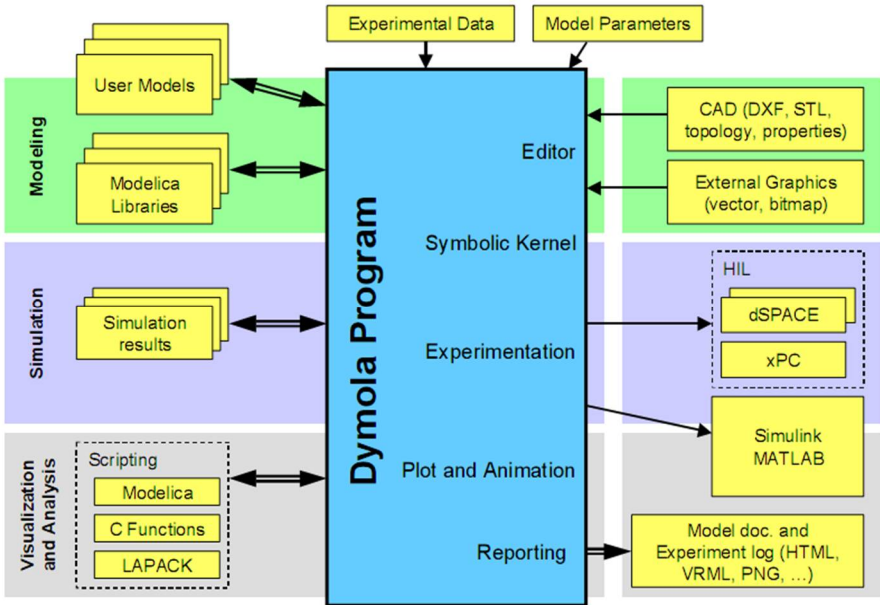


Рисунок 2.9 – Архітектура Dymola

2.9 20-sim

Пакет 20-sim призначений для моделювання динаміки технічних систем – механічних, електричних, гідравлічних, а також складних систем, що містять елементи всіх перелічених. Пакет підтримує візуальне компонентне моделювання – модель вводиться звичайним для пакетів такого типу методом – компоненти вибираються з бібліотеки, переносяться на робочий стіл і з'єднуються зв'язками. 20-sim дозволяє працювати як з орієнтованими блоками, так і з не орієнтованими, що дуже зручно при моделюванні систем управління фізичними об'єктами і установками. Бібліотеки базових компонентів: типових одновимірних направлених ланок (Signal) – джерел, вимірників, перетворювачів, блоків математичних операцій і т.д.; елементів графів зв'язків

(Bond Graph), як найпростіших універсальних енергетичних компонентів; механічних, електричних, гідравлічних, теплових елементів (Iconic Diagram). В якості прикладу на рисунку 2.10 представлена модель системи, де енергетична частина представлена графом зв'язків, а керуюча – блок-схемою, що включає спрямовані ланки.

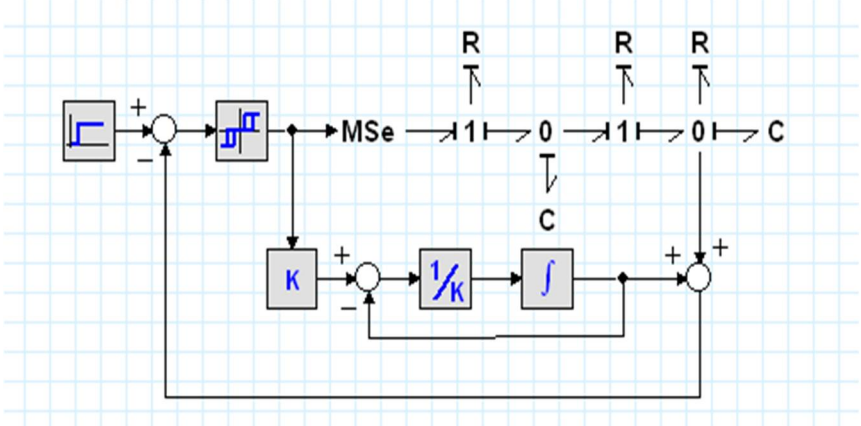


Рисунок 2.10 – Приклад моделі системи в 20-sim

Пакет 20-sim має ряд розширень, одним з яких є Mechatronics Toolbox, що включає в основному моделі різних електромеханічних пристроїв. Важливо відзначити, що більшість моделей, представлених в даному розширенні, відносяться до конкретних мехатронних пристроїв, що підтримуються і супроводжуються виробниками технічних засобів. Вони точні, докладні, закривають всі аспекти функціонування пристроїв і з цієї точки зору дуже корисні проектувальникам, що займаються розробкою МС.

В якості внутрішньої мови опису моделей використовується мова, подібна мові Modelica або мові середовища Maple, але є більш зрозумілою і простою у вивченні. 20-sim має розвинуті засоби анімації, в тому числі 3D, потужний вирішувач, що дозволяє отримувати чисельне рішення як ODE, так і диференціально-алгебраїчних рівнянь (DAE), використовуючи однокрокові і багатокрокові методи різних порядків: Euler, Backward Euler, Adams-Bashford 2, Runge-Kutta 2 та 4 порядків, Runge-Kutta-Dormand-Prince 8, Runge-Kutta-Fehlberg, Vode Adams, Backward Differentiation Formula (BDF), Modified Backward Differentiation Formula (MBDF).

Також програма має 8 алгоритмів оптимізації (2 прямих, 6 градієнтних): Perpendicular Search, Line Climber, Steepest Descent, Continuous Descent, Newton-Raphson, Davidson-Fletcher-Powell, Broydon-Fletcher-Goldfarb-Shanno, Polack-Ribiere.

20-sim має широкі можливості щодо моделювання (modeling) та дослідження електронних систем, також має інтуїтивно зрозумілий і не перевантажений інтерфейс. Привабливою програму також роблять можливість фізичного мультидоменного моделювання та різноманітність форм представлення моделей систем: рівняння у рядковій математичній нотації, орієнтовані блок-схеми, акаузальні традиційні (Iconic) схеми (принципові) та ненаправлені зв'язані графи (графи зв'язків-BG).

Програма має три види версій або редакцій (з ліцензіями Academic та Corporate): Viewer, Standard, Professional. Причому версія Viewer безкоштовна, завантажується з сайту виробника без авторизації (легкодоступна) та не має обмежень (на відміну від Standard), тобто володіє повною функціональністю (як Professional). Мінімальні ціни 20-sim Standard та 20-sim Professional – \$700 та \$1400 відповідно (для академічної ліцензії), в той час як для корпоративної ліцензії вартість цих редакцій збільшена в 6 разів (\$4200 та \$8400 відповідно).

Програма має деякі переваги та особливості:

- 32-бітний розрахунковий машинний код, що дозволяє в кілька раз підвищити швидкість розрахунків при симуляції;
- принцип відкритості, забезпечуваний механізмами підключення зовнішніх програм (Matlab/Simulink) і можливістю вільно переглядати, редагувати існуючі й створювати нові бібліотеки моделей;
- принцип вкладеності або ієрархічності структур (глибина вкладеності не обмежена), що особливо актуально при моделюванні складних ДС;
- значний обсяг бібліотеки моделей, що в багатьох випадках дозволяє сформувати математичну модель динаміки складної технічної системи без необхідності створення нових спеціалізованих ММ;
- дуже проста в освоєнні Modelica-подібна мова опису моделей SIDOPS+, яка використовує математичну нотацію для запису рівнянь;
- присутній потужний інструментарій частотної області, що включає FFT-аналізатор, лінеарізатор моделей і редактор лінійних систем;
- універсальний редактор лінійних систем, який дозволяє реда-

увати систему (модель) як простір станів, передатну функцію, нулі-полюса й перетворювати їх між собою, а також містить опції показу діаграм Боде, годографа Найквіста, графіка Нікольса, діаграм нулів-полюсів, реакції системи на функцію Хевісайда;

- можливість транслювати будь-яку модель у Сі-код для використання в Matlab/Simulink або автономний Сі-код для використання в С и С++ програмах;

- наявність вичерпної контекстної довідкової системи у формі типової довідки WINDOWS для кожної моделі стандартної бібліотеки й елемента інтерфейсу.

20-sim дозволяє використовувати кілька форм представлення ДС при побудові моделей (в тому числі моделювати у змішаній формі):

- форма на основі диференціальних рівнянь (текстова модель);
- блок-схеми (BD);
- традиційні (функціональні) схеми декількох доменів (ID);
- графи зв'язків (BG);
- автомати станів (SM) (за підтримки розвиненої мови програмування SIDOPS+);
- передавальні функції (transfer function, TF).

В 20-sim дозволяється моделювання як безперервних систем, так і дискретних, а також змішаних (або гібридних), які представляють комбінацію двох перших. За замовчуванням, моделі в 20-sim є безперервні; аналіз проводиться за допомогою безперервних (continuous-time) методів інтегрування. Однак, 20-sim автоматично визначає дискретні частини в моделі і проводить симуляцію з фіксованим темпом або кроком (fixed rate). Входи та виходи (з'єднання) дискретних елементів моделі позначаються у головному редакторі зеленим кольором, а самі дискретні елементи ідентифікуються появою спеціальних функцій у їх описі («sample», «hold», «next», «previous», «sampletime»). Для з'єднання безперервних та дискретних елементів застосовуються функції «sample» та «hold» (перша має дискретний вхід та безперервний вихід, а друга - навпаки).

Будь-яка головна модель в 20-sim може бути описана за допомогою композиції підмоделей (submodel) більш низького рівня, які у свою чергу знову можуть складатися з підмоделей. Найнижчий рівень ієрархії завжди представляє собою елементарні підмоделі на

основі рівнянь на мові SIDOPS+. 20-sim не має обмежень на рівень вкладеності моделей і дозволяє інкапсулювати будь-яку частину системи та розгортати її назад за допомогою функцій графічного редактора («implode\explode»).

Нажаль, демо-версія програми не дозволяє зберігати відпрацьований файл для подальшої роботи.

2.10 Тракти передачі даних

Трохи теорії довгої лінії (Transmission line)

Велике правило для лінії передачі полягає в тому, що лінія повинна бути принаймні $1/4$ довжини хвилі, перш ніж вона вважається «довгою». Довжина хвилі обчислюється за формулою $\lambda = v/f$, де « λ » – довжина хвилі, « v » – швидкість поширення, а « f » – частота сигналу. Можна сказати, що довжина хвилі – це відстань, на яку поширюється хвиля за час одного періоду коливань. довжина хвилі визначається як джерелом (через період T), так і властивостями середовища (через швидкість v). При переході хвилі з одного середовища в інше довжина хвилі змінюється, на відміну від частоти, яка залишається сталою.

Хвильовий опір каналу Z_0 є коренем квадратним зі співвідношення паразитної індуктивності до ємності каналу.

2.10.1 Втрати в лінії та узгодженість навантаження:

Зворотні втрати в дротовому або бездротовому передавачі – це відношення переданої потужності до відбитої потужності, коли живлення подається до навантаження. Це критичний параметр, який описує ефективність передачі і може використовуватися як індикатор того, що надмірна потужність відбивається від навантаження назад до джерела. У бездротовому передавачі з низькими зворотними втратами антени високий рівень відбитої потужності може пошкодити підсилювач потужності, який керує антеною.

Зворотні втрати (Return loss) – це міра того, скільки сигналу втрачається, коли він відбивається назад до джерела, тоді як втрати на узгодження (matching loss) – це втрати, що виникають, коли існує велика невідповідність між лінією та навантаженням.

Оптимальна передача енергії в лінії передачі реалізується в режимі «біжучої хвилі», коли лінія передачі навантажена на опір, рівний хвильовому (узгоджене навантаження).

Mismatch loss (втрати, пов'язані із неузгодженістю навантаження) в теорії лінії передачі – це кількість потужності, виражена в децибелах, яка не буде доступна на виході через розбіжності імпедансу та відбиття сигналу.

Лінія передачі, яка закінчується належним чином, тобто має той самий імпеданс, що й характеристичний опір лінії передачі, не матиме відбитків і, отже, не матиме втрат на неузгодженість. Втрата невідповідності відображає кількість енергії, витраченої системою. Це також можна розглядати як кількість отриманої потужності, якщо система була ідеально підібрана.

Узгодження імпедансу є важливою частиною розробки радіочастотної системи; однак на практиці ймовірно буде певний ступінь втрати невідповідності. У реальних системах відносно невеликі втрати викликані втратами невідповідності і часто становлять близько 1 дБ. Невідповідність не призводить до будь-яких втрат ("wasted" сигнал), окрім лінії передачі. Це пояснюється тим, що сигнал, відбитий від навантаження, передається назад до джерела, де він повторно відбивається через реактивний опір, представлений джерелом, назад до навантаження, доки вся потужність сигналу не буде випущена або поглинена навантаженням.

2.10.2 Модель довгої лінії

В більшості програм моделювання використовуються такі способи завдання моделі (далі використано мову PSpice-сумісної ECAD програми Micro-Cap).

Schematic format:

VALUE attribute for **ideal line**

Z0=<value> [TD=<value>] | [F=<value> [NL=<value>]]

Example for ideal line

Z0=50 TD=3.5ns

VALUE attribute for **lossy line**

<physical length> LEN=<len value> R=<rvalue> L=<lvalue>

G=<gvalue> C=<cvalue> Transmission line

VALUE attribute for **lossy line**

<physical length> LEN=<len value> R=<rvalue> L=<lvalue>

G=<gvalue> C=<cvalue>

Examples for lossy line

LEN=1 R=.5 L=.8U C=56PF R=.5 L=.8U C=56PF

Model statement form

.MODEL <model name> TRN ([model parameters])

Examples

.MODEL TIDEAL TRN(Z0=50 TD=10ns)

.MODEL TLOSS TRN(C=23pF L=13nH R=.35 LEN=10)

Model parameters for ideal line

Name Parameter Units Default

Z0 Characteristic impedance Ohms

TD Transmission delay S

F Frequency for NL Hz

NL Relative wavelength 0.25

Model parameters for lossy line

Name Parameter Units Default

R Resistance per unit length Ohms / unit

L Inductance per unit length Henries / unit

G Conductance per unit length Mhos / unit

C Capacitance per unit length Farads / unit

LEN Physical length same as RLG

Параметри LEN, R, L, G, та C мають бути виражені через однакові метрики. Наприклад, якщо C представлено через farads/cm, тоді R має бути також в ohms/cm, тобто LEN має бути в сантиметрах.

Model Equations

Схема заміщення лінії передачі представлена на рисунку 2.11.

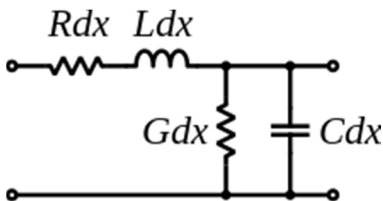


Рисунок 2.11

Принципова відмінність між двома моделями полягає в реалізації затримки. В ідеальній моделі затримка реалізована як пов'язаний список пар даних (час, значення) і точок зупину.

Лінія з втратами використовує метод згортки SPICE3 для представлення лінії, створюючи точне представлення лінії з втратами. Застосований метод полягає в згортці вхідного сигналу з імпульсною характеристикою лінії з втратами для отримання вихідного сигналу. Імпульсна характеристика отримується за попередньо розв'язаною аналітичною формулою. Convolution гарантує, що можна обробляти будь-яку форму хвилі, а не лише кроки та лінійні зміни.

Схема заміщення моделі лінії передачі представлена на рисунку 2.12.

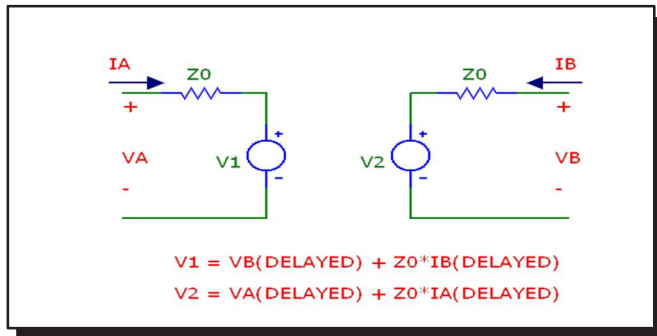


Рисунок 2.12 – Схема заміщення моделі лінії передачі

2.11 Етапи виконання роботи

1 Відкрити файл MC12/Data/Data Communications/Transmission Line Matching.cir (рис.2.13).

2 Отримати графіки для Mismatch and Return Loss (змінні A_r та A_{pr}), вхідної та вихідної потужності (P_{in} та P_{out}). Для цього запустити Transient Analysis і відкрити вкладки Losses та Power в постпроцесорі.

3 Проаналізувати результати у висновках.

4 Для прикладів схем ліній передачі без втрат ознайомтеся із схемами TL1, TL2 і TL3. Всі файли мають розширення *.cir.

5 Для ліній із втратами відкривайте файли LTRA3 та SkinEffect1. Останній використовує залежність опору від частоти для емуляції скін-ефекту. Всі файли мають розширення *.cir.

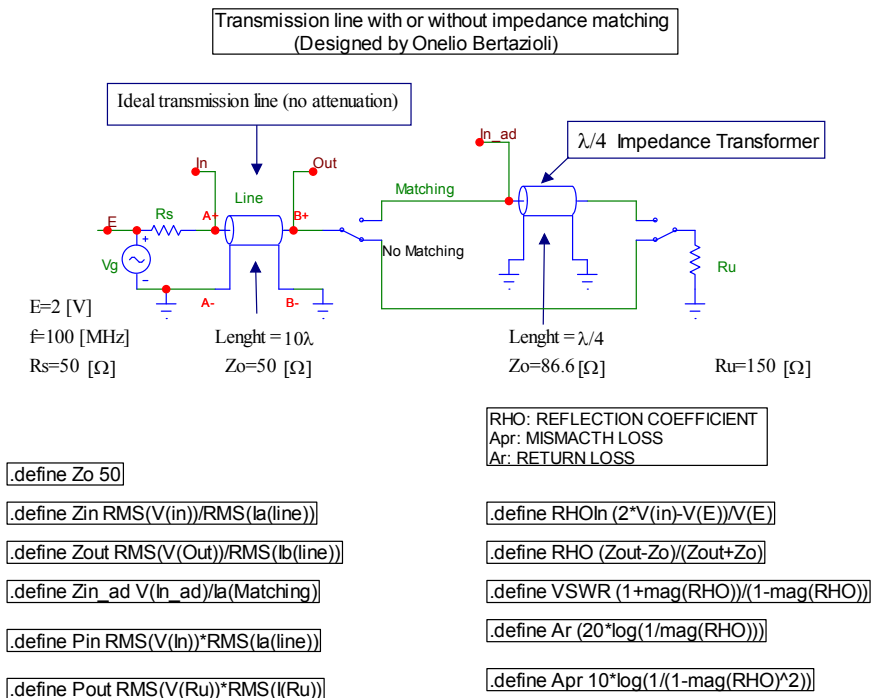


Рисунок 2.13 – Моделі ліній передачі: узгоджена праворуч, неузгоджена ліворуч

2.12 Контрольні запитання

1. Огляд САЕ систем.
2. Огляд CAS систем.
3. Огляд CAD систем.
4. Структура САЕ системи (приклад).
5. Критерії порівняння САЕ систем.
6. В яких системах використовують акаузальний підхід?
7. Від чого залежить ефективність системи моделювання.
8. Чим відрізняються програми моделювання від програм проектування.
9. В яких системах є доступні демо- та студентські версії?
10. Моделі каналів зв'язку.

3 ЛАБОРАТОРНА РОБОТА №3 “ МОДЕЛЮВАННЯ АВТОМАТИЗОВАНИХ СИСТЕМ ”

Мета: дослідити підходи до моделювання систем захисту інформації як різновиду автоматизованих систем.

3.1 Структурні елементи системи управління

Системи автоматизованого управління (САУ) можуть бути складовими систем захисту інформації, або системи захисту інформації можна розглядати як САУ. Модель системи управління можна представити наступним рисунком 3.1.

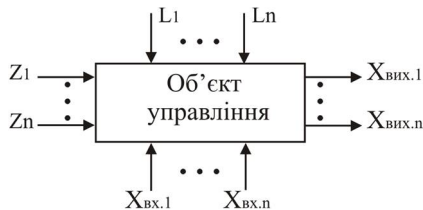


Рисунок 3.1 – Модель системи управління

На кожний об'єкт управління впливають багато зовнішніх факторів, які можна розбити на декілька груп:

- а) $X_{вих}$ – керовані параметри, які характеризують стан процесу;
- б) L – випадкові неконтрольовані збурення;
- в) Z – контрольовані збурення;
- г) $X_{вх}$ – керуючі дії, тобто дії спрямованні на компенсацію дій груп Z та F .

Метою автоматичного регулювання та управління є або додержання регульованого параметра на заданому рівні ($X_{вих}=const$), або змінення цього параметра по деякому закону [$X_{вих}=f(t)$]. В першому випадку говорять про систему автоматичного регулювання (САР), а в другому – про систему автоматичного управління/керування (САУ).

Принцип управління за відхиленням полягає в тому, що вимірюється величина керованого параметра ($X_{вих}$) та порівнюється з заданим значенням ($X_{зад}$), а отримане відхилення перетворюється в керуюче діяння ($X_{вх}$), котре прагне зменшити або усунути це

відхилення шляхом відповідного впливу на об'єкт. Структурна схема системи керування **за відхиленням** наведена на рис.3.2.



Рисунок 3.2 – Структурна схема системи керування за відхиленням

САУ з таким принципом – це **замкнена** САУ, оскільки сигнал з виходу об'єкта подається (замикається) на його вхід (зворотній зв'язок). Тому такий принцип називають ще **принципом зворотного зв'язку** (33).

Принцип управління **за збуренням** полягає в тому, що для зменшення або усунення відхилення керованої величини ($X_{\text{вих}}$) від заданого значення ($X_{\text{зад}}$), яке спричиняється збуренням, вимірюється це збурення і перетворюється в керуюче діяння ($X_{\text{вх}}$), яке компенсує це відхилення. Система з принципом управління за збуренням є **розімкненою** САУ: процес управління не залежить від результату. Структура такої САУ – на рисунку 3.3.

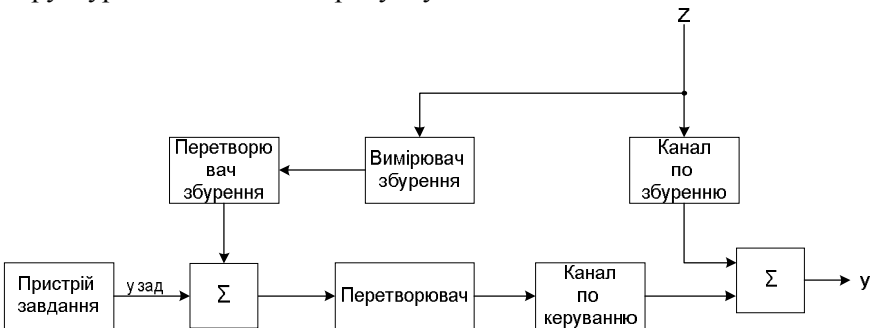


Рисунок 3.3 – Структурна схема САУ з принципом управління за збуренням

Вимірювання збурюючого діяння $L(t)$ здійснюється за допомогою вимірювального елемента (ВЕ), а його перетворення – за допомогою перетворювача (П). Вимірювальний елемент і перетворювач утворюють зв'язок за збуренням. Вихідна дія зв'язку за збуренням (X_L) в суматорі (Σ) складається із задавальною дією ($X_{зад}$). Далі цей сумарний сигнал підсилюється і перетворюється в ПП та подається на виконуючий пристрій (ВП), який через керуючий орган (КО) компенсує дію збурення. Алгоритм управління тут $X_{ex}(t)=f[L(t)]$.

Структурна схема системи керування за **керуючим впливом** наведена на рис.3.4. Якщо дії на об'єкт постійні в часі, то процес йде стабільно і без участі людини. Але, як правило, збурення в часі змінні, що призводить до відхилення вихідних параметрів від заданих значень. Тому об'єкт потребує керування, тобто цілеспрямовану зміну параметрів X_{ex} , щоб компенсувати зовнішні збурення L . Щоб вирішити задачу керування, тобто цілеспрямовано змінювати X_{ex} , необхідно знати залежність між окремими параметрами процесу, тобто функціонал $X_{вих} = f(X_{ex}, L, Z)$, що є математичною моделлю. Необхідно мати також математичні моделі **підсистем** САУ/САР.

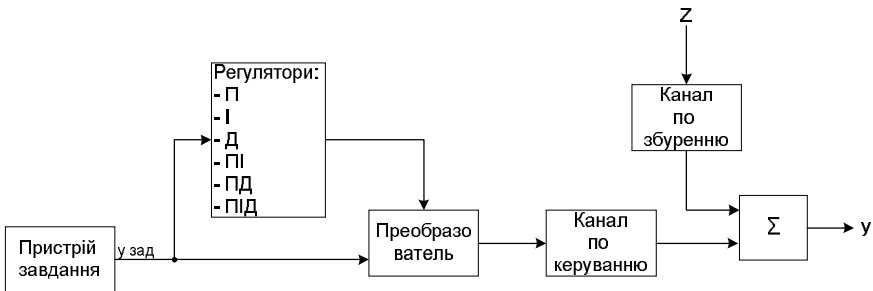


Рисунок 3.4 – Структурна схема системи керування за керуючим впливом

3.2 Теоретичні аспекти моделювання систем автоматичного управління

При моделюванні процесів автоматичного управління дотримуються схеми, за якою реальний об'єкт складається з двох підсистем, які взаємодіють: керованої та керуючої. Класична схема управління з одиничним **негативним зворотним зв'язком** показана на рисунку 3.5.



Рисунок 3.5 – Одноконтурна САР

Призначення регулятора системи полягає в корекції динамічних властивостей об'єкта управління за допомогою сигналу $u(t)$, що управляє, так, щоб реальний вихідний сигнал $y(t)$ якомога менше відрізнявся від бажаного вихідного сигналу $g(t)$. Регулятор виробляє такий сигнал управління, використовуючи помилку регулювання

$$\varepsilon(t) = g(t) - y(t), \text{ щоб виконувалася наближена рівність } \varepsilon \sim 0.$$

При дослідженні систем за їх математичними моделями виникає питання, в якій мірі реальний процес функціонування відповідатиме розрахунковому, оскільки насправді в розрахунках користуються наближеними моделями, коли ряд факторів можуть не враховуватися.

Системи автоматичного управління мають «схильність» до коливання, це викликано тим, що частина енергії з їх виходу може повертатися на вхід, а з появою збурення чи зміни керуючої дії починається перехідний процес.

Стійка система, за умови усталених значень керуючих та збурюючих дій, через певний час, знову повертається до усталеного стану рівноваги, а нестійка система, прийшовши до руху, не повертається до усталеного стану рівноваги, навпаки, її відхилення від стану рівноваги буде або неперервно змінюватися, або змінюватися в формі постійних незгасаючих коливань.

Це питання є предметом **теорії стійкості**. Тому для задовільної роботи системи автоматичного управління перш за все необхідно, щоб вона була стійкою, причому вимога стійкості повинна виконуватися з деяким запасом, який передбачає можливі зміни параметрів системи під час її роботи. Якщо система є стійкою, то представляє інтерес такі її параметри: максимальне відхилення регульованої величини $y(t)$ в перехідному процесі – перерегулювання (відхилення від значення в усталеному режимі у відсотках, час перехідного процесу (час, за який функція змінює своє значення з 10% до 90% від свого значення в усталеному режимі).

Висновки стосовно властивостей систем автоматичного управління можна зробити, виходячи з вигляду диференціальних рівнянь моделі, які наближено описують процеси в системах. Порядок диференціального рівняння та значення його коефіцієнтів повністю визначаються статичними та динамічними параметрами системи та її структури.

Безперервні системи описуються звичайними диференціальними рівняннями та диференціальними рівняннями в частинних похідних, а дискретні системи – диференціально-різницевиими рівняннями. Лінійні та нелінійні неперервні системи автоматичного управління описуються відповідно лінійними та нелінійними диференціальними рівняннями. Лінійні та нелінійні системи (неперервні чи дискретні) можуть бути стаціонарними чи нестаціонарними, із зосередженими чи розподіленими параметрами. Стаціонарні системи описуються диференціальними рівняннями із сталими коефіцієнтами. Якщо система є нестаціонарною, то коефіцієнти відповідних рівнянь є функціями часу.

Така класифікація систем автоматичного управління за способом їх математичного опису є умовною. На практиці в системах автоматичного керування досить зрідка доводиться мати справу з режимами роботи, які значно відрізняються один від одного, частіше існує якийсь один типовий режим для конкретної системи. Саме ця обставина часто використовується для значного спрощення математичної моделі.

Для оцінки динамічних властивостей системи часто розглядається реакція на одиничну ступінчасту дію. Перехідний процес повинен відповідати заданим показникам якості, до яких відносяться час перехідного процесу, перерегулювання. Можуть бути також використані інтегральні оцінки якості перехідного процесу.

3.3 Моделі для елементів САУ в програмі Micro-Cap

В бібліотеці пакету моделювання Micro-CAP наявні макромоделі пристроїв, які необхідні для побудови функціональної моделі САУ/САУ, зокрема, операційні багатополосники, які можуть виконувати операції інтегрування, диференціювання та всі арифметичні дії та чотирьохполосники із складними передатними функціями.

Перелічимо деякі з них:

- ланцюг абсолютного значення *Absolute Value Macro* ;
- ланцюг підсилення *Amplifier Macro*;
- ланцюг затримки *Analog delay Macro*;
- ланцюги суматорів із різним напрямком та кількістю входів для сумування, наприклад, *2-Input Summer Macro*;
- ланцюги для обчислення різності, результатів множення та ділення сигналів *Subtraction Macro*, *Multiplier Macro*, *Divider Macro*;
- ланцюги інтегрування та диференціювання *Integrator Macro*, *Differentiator Macro*;
- ланцюг із зоною нечутливості *Slip Macro*;
- тригер Шмітта (ланцюг гістерезису) *Schmitt Macro*.

Розглянемо умовні графічні позначки (УГП) та схеми заміщення деяких макромоделей. На рисунку 3.6 наведена макромодель елемента, який реалізує затримку сигналу (параметр – затримка в секундах), на рисунку 3.7 – макромодель елемента із передатною функцією, вид якої зображено на УГП цього двополюсника.

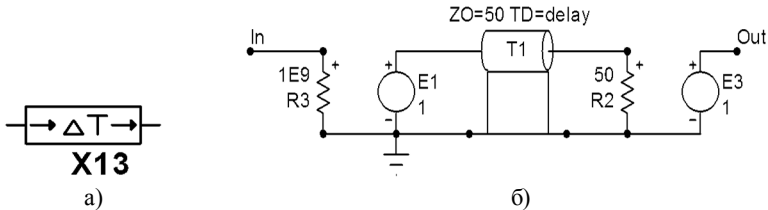


Рисунок 3.6– Ланцюг *Analog delay Macro*: а – УГП; б – схема заміщення

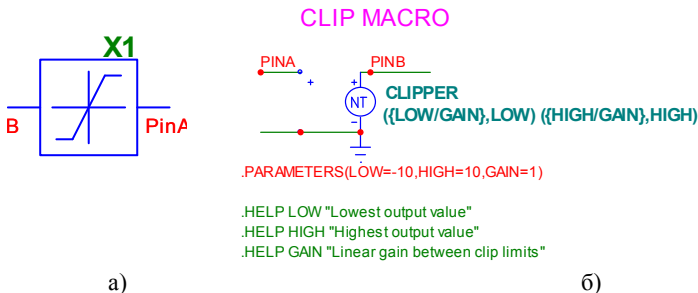


Рисунок 3.7 – Ланцюг *CLIP Macro*: а – УГП; б – схема заміщення із керованим функціональним джерелом NT

3.4 Прокєтування САУ в 20-sim

Прокєтувальник регуляторів (*Controller Design Editor, CDE*) має інтерфейс і всі можливості редактора лінійних систем (*Linear System Editor, LSE*), та додатково опції «Sub System» та «Additional Ports» (інтерфейс – на рисунку 3.8).

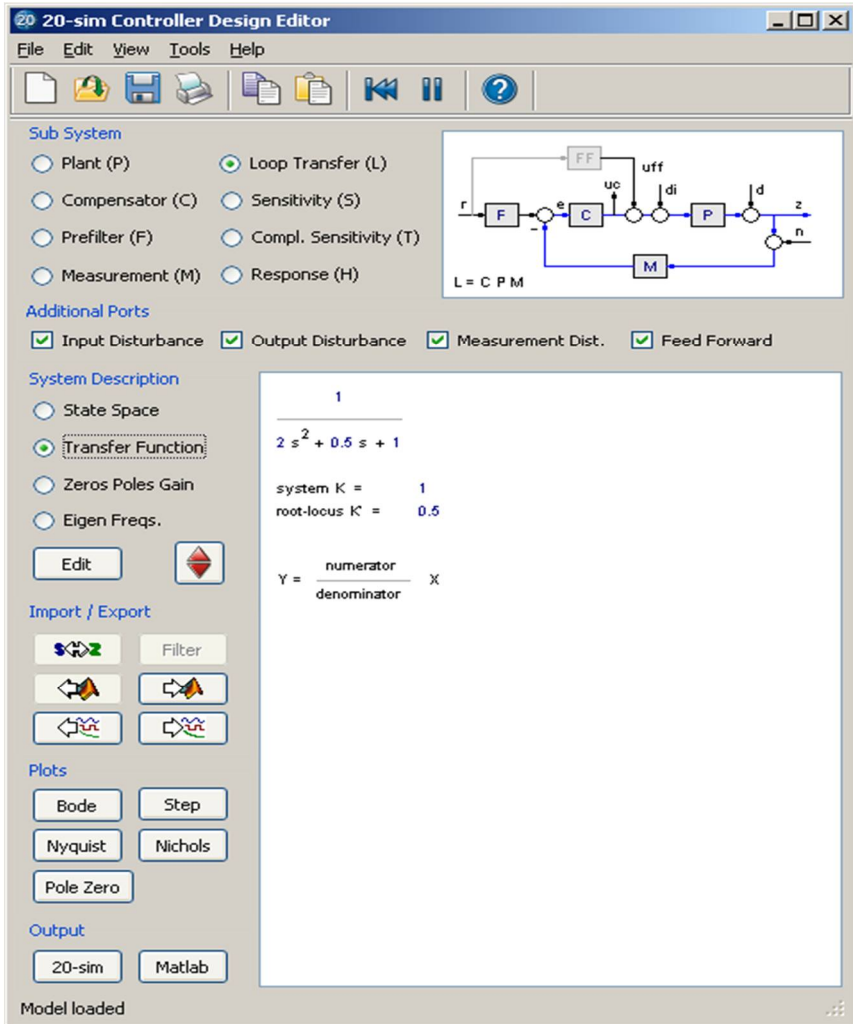


Рисунок 3.8 – Головне вікно CDE

Редактор фільтрів і проєктувальник регуляторів мають важливе значення при створенні САУ та їх аналізі; вони мають зв'язок з редактором лінійних систем LSE, який у свою чергу дає можливість синтезувати динамічні ланки чотирма способами: за допомогою простору станів (*State Space*), передавальної функції (*Transfer Function*), нулів-полюсів ПФ (*Zeros Poles Gain*), власних значень частот (*Eigen Frequencies*).

Редактор лінійних систем (LSE) дозволяє перетворювати дискретну (*discrete-time*) ланку в безперервну (*continuous-time*) та навпаки; також має функції імпортування та експортування редагованої системи в Matlab/Simulink, в текстовий файл в символічній або числовій формах тощо. LSE безпосередньо дозволяє аналізувати і досліджувати розроблену САУ, використовуючи вбудовану можливість відтворення діаграм Бode, Найквіста, Нікольса та відображення карти нулів-полюсів і *h*-функції (*Step Response*).

Рисунок 3.9 показує опції редактора фільтрів для деяких типів регуляторів або фільтрів.

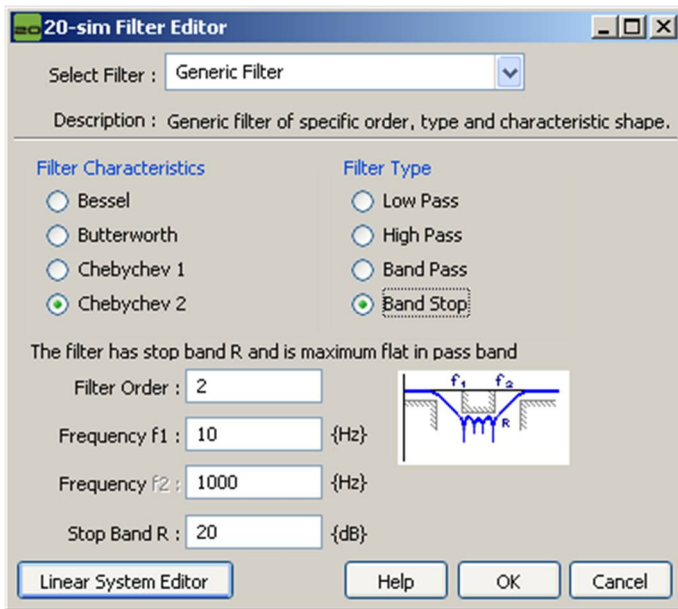


Рисунок 3.9 – Параметри фільтрів загального шаблону

Редактор фільтрів дозволяє створювати різні типи фільтрів (Беселя, Батерворта, Чебишева, ФНЧ та ФВЧ першого і другого порядків, смуговий, режекторний) та регуляторів (П, І, Д, ПІ, ПД, послідовний та паралельний ПД, ПД-компенсатор), використовуючи інформацію про їх бажані частотні та часові характеристики, наприклад: частоти спряження, порядок фільтру, коефіцієнти передачі, добротність, коефіцієнт «слухняності» (*tameness*), постійні часу.

Для опису універсального режекторного фільтру другого порядку, застосовується наступні параметри:

- коефіцієнт підсилення K ;
- резонансні частоти f_z та f_p для нуля (*zero*) і полюса (*pole*) ПФ відповідно;
- коефіцієнти демпфування d_z та d_p .

Вікно для параметризації ПД наведено на рисунку 3.10.

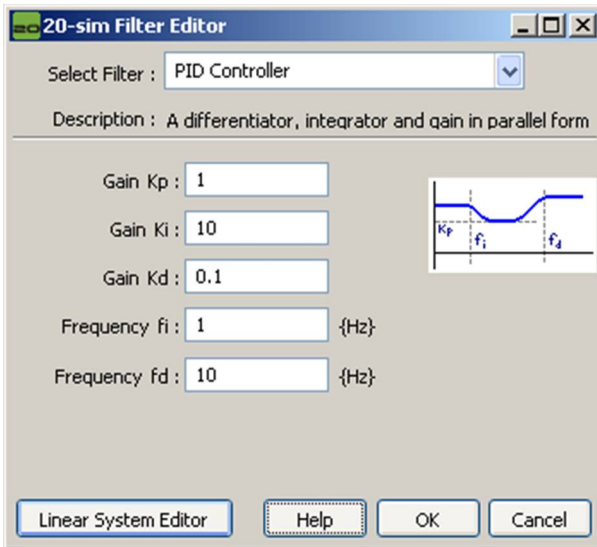


Рисунок 3.10 – Опис паралельного ПД-регулятора

Регулятор описується трьома коефіцієнтами підсилення (для зони НЧ – K_i , ВЧ – K_d та у смузі пропускання K_p) та двома частотами спряження (інтегруючої частини f_i та диференціюючої – f_d). Ілюстрація сенсу коефіцієнтів – на рисунку у правому куті вікна. З цього ж вікна

можна перейти до редактора лінійних систем LSE, для продовження синтезу регулятора (параметризації).

ПІД-компенсатор (рис. 3.11) має у своїй структурі (окрім самого регулятора) фільтр із спадаючою АЧХ в області ВЧ після значення $1/\tau_h$ (параметр «HF roll-off tau h»), всі частоти з більшим значенням відфільтровуються. Такий компенсатор описується постійними часу інтегруючої (*Integration tau i*) τ_i та диференціюючої частини (*Differentiation tau d*) τ_d ; коефіцієнтом підсилення K на інтервалі частот від $1/\tau_i$ до $1/\tau_d$; коефіцієнтом «слухняності» (*tameness*) або крутизни (нахилу) β , що впливає на поведінку диференціювання, тобто на крутизну АЧХ в зоні $1/\tau_d \dots 1/\beta\tau_d$. Виходячи з передавальної функції ПІД-компенсатора, можна сказати, що використовується послідовне з'єднання внутрішніх елементів (П-, І-, Д-частин та фільтра).

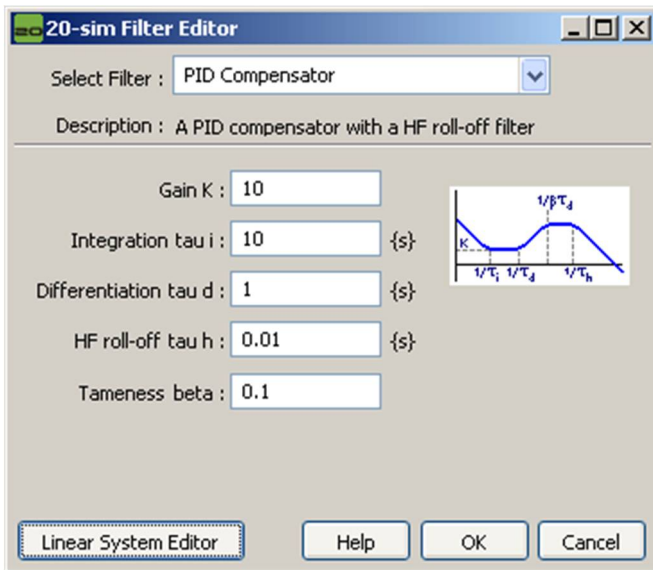


Рисунок 3.11 – Опис ПІД-компенсатора у редакторі фільтрів

Редактор фільтрів пропонує для синтезу декілька типів ПІД-регуляторів з різним набором вхідних параметрів. Проектувальник регуляторів (CDE) дозволяє синтезувати повну структурно завершену САУ з контролем проміжних сигналів, використовуючи будь-яку

форму опису, представлену в *LSE*. Структурно повна система в CDE може бути представлена рисунком 3.12.

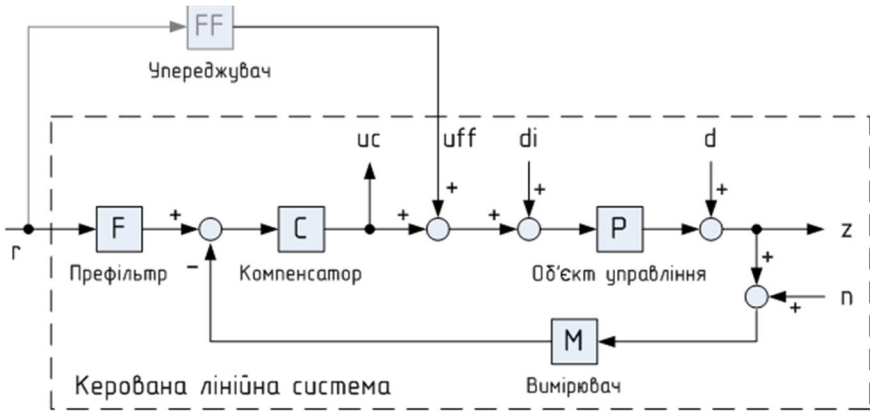


Рисунок 3.12 – Структура САУ, що використовується в CDE

Така система складається з:

об'єкта управління P (*Plant*);
 регулятора або компенсатора C (*Compensator*);
 префільтра F (*Prefilter*);
 вимірювача M (*Measurement*);
 упереджувача FF (*Feed Forward*).

В САУ діють такі вхідні/вихідні сигнали:

r – вхідний вплив на систему або завдання;
 uc – чистий вихідний сигнал компенсатора;
 uff – вхідний сигнал від упереджувача;
 di – вхідне збурення об'єкта управління (input disturbance);
 d – вихідне збурення об'єкта управління (output disturbance);
 z – контрольований результуючий вихідний сигнал системи;
 n – збурюючий вплив з боку вимірювача.

CDE також дозволяє оцінити чутливість системи S (*Sensitivity*), передавальну функцію замкнутого контуру L (*Loop Transfer*),

комплексну чутливість T (*Compl. Sensitivity*), відгук цілої системи H (*Response*).

CDE дозволяє працювати з лінійними системами, а якщо є необхідність розглядати модель нелінійної системи, то її потрібно попередньо лінеаризувати у робочій точці за допомогою інструментів частотної області (FDT).

Також в CDE можна моделювати ланку чистого запізнення (її ПФ: $W(p)=e^{-\tau p}$). Взагалі, моделі з часовими затримками не можуть бути лінеаризовані, тому що немає прямого відношення (зв'язку) між входами і виходами моделі. Однак, для систем з контурами управління корисно було б побачити, як затримки цих контурів неминуче впливають на поведінку моделі. Тому в LSE (та CDE) до моделі можна додати вихідну затримку.

Практично всі CAE, розглянуті вище (окрім, наприклад, SamSim), дозволяють працювати з ланкою чистого запізнення тільки в часовій області, але не можуть відпрацьовувати її у частотному домені. Треба відмітити, що вихідна затримка не зберігається після передачі лінійної системи з LSE до головного редактора. Таким чином, ЧХ моделі з ланкою затримки можна побудувати лише в LSE та CDE, а часові діаграми – також і в симуляторі (після побудови моделі з ланкою чистого запізнення у головному редакторі).

3.5 Методика моделювання САУ

Для дослідження САУ треба спочатку побудувати її модель. Для цього треба, по-перше, ідентифікувати об'єкт, тобто визначити його складові частини, їх параметри, ММ, тощо. Згорнутий алгоритм моделювання САУ наведено в додатку В.

Рекомендується етап побудови моделі (modeling) в програмному середовищі (мова йде, зокрема, про 20-sim) умовно розділяти на кілька частин:

- а) вибір форми представлення моделі;
- б) вибір компонентів моделі;
- в) топологічне з'єднання компонентів моделі;
- г) апріорна корекція моделі;
- д) компонентно-топологічна (апостеріорна) корекція моделі.

Етап симуляції в 20-sim можна розділити на три складові:

- а) вибір алгоритму моделювання та його параметрів;
- б) обирання фазових змінних для моніторингу;

в) запуск симуляції та використання постпроцесора.

Задача вибору методу симуляції (алгоритму моделювання) комплексна, багатокритеріальна, залежить від багатьох факторів.

Етап параметричної оптимізації, не відноситься до етапу симуляції, але залежить від нього: якщо симуляція призводить до неадекватних результатів, то проводити оптимізацію, у загальному випадку, не має сенсу (окрім випадків, коли варіюється параметр моделі, що відповідає за її жорсткість). Оптимізація є невід'ємною складовою сучасних САЕ для моделювання САУ (є необхідність у знаходженні оптимальних коефіцієнтів регуляторів при налагодженні САУ; дослідження алгоритмів оптимізації розглядається на наступних заняттях).

Методика проведення оптимізації в 20-sim:

- а) вибір алгоритму оптимізації та його налаштування;
- б) вибір параметру моделі, що потребує оптимізації;
- в) встановлення критерію оптимізації;
- г) запуск оптимізації та використання постпроцесора.

Для отримання адекватного результату симуляції вибір методу інтегрування в 20-sim можна проводити згідно розробленої методики, яка представлена у вигляді алгоритму в Додатках Г та Ж.

3.6 Контрольні запитання

1. Класифікація САУ.
2. Принципи керування та базові структури автоматизованих систем.
3. Принципи дослідження САУ.
4. Виписати з бібліотеки програми MICRO-CAP макроси, які використовуються для моделінгу САУ, визначити критерії, за якими вони можуть бути класифіковані.
5. Проектування регуляторів, методика.
6. Методика аналізу САУ.
7. Критерії стійкості та методи дослідження стійкості систем.
8. Редактор лінійних систем в 20-sim.
9. Методика підвищення адекватності моделювання САУ.

10. Опишіть обрану систему захисту інформації як автоматизовану систему.
11. Дослідить таблицю 3.1 і підпишіть заголовки для колонок.

Таблиця 3.1 – Класифікація ланок та блоків САУ з бібліотеки програми MC 12

За рівнем абстракції		За функціональним призначенням		
?	?	?	?	?
ABS	Comparator	ABS	Clip	Monostable
AMP	Delay	Dif	Comparator	Noise
Clip	FSK, PSK, VCO	Div	AM	VCO
Delay	Noise	Int	AMP	555
Dif	Schmitt	Int_Res	F	
Int, Int_Res	Relay	Mult	FSK	
Monostable	Motors	Sub	Resonant	
Mult, Div	Resonant	Sum	PSK	
Resonant	Wideband	Sum3	Slip	
Slip	Modulators (FSK,		Wideband	
Sub, F	PSK, PWM etc)			
Sum, Sum3	Limiters			?

4 ЛАБОРАТОРНА РОБОТА №4

“МОДЕЛЮВАННЯ КОМП’ЮТЕРНИХ МЕРЕЖ”

Мета: дослідити підходи до імітаційного моделювання інформаційних систем, зокрема, комп’ютерних мереж.

4.1 Імітаційне моделювання систем

При дослідженні багатьох складних систем з’являється необхідність у рішенні задач, які відносяться до масового обслуговування. Вони найбільш поширені у фізиці часток, організації виробництва, плануванні, автоматичному регулюванні складними системами тощо. Особливістю таких задач є наявність обслуговуючої системи, до якої у будь-який момент часу надходять заявки. Ця система має лінії (канали), які виконують сукупність операцій (обслуговування).

Якщо з точки зору обслуговування всі заявки потоку виявляються рівноправними, тоді сенс має лише факт настання у даний момент часу події – поява заявки. Такі потоки мають назву - потоки *однорідних* подій. Іноді для розв’язання практичних задач необхідно враховувати неоднорідність потоку.

Якщо потік однорідних подій є цілком *детермінованим*, то необхідно задати послідовність t_j одним з можливих способів: перерахувати їх, вказати співвідношення, яке описує t_j як функцію індексу j , або привести рекурентні залежності (дозволяють визначити поточні значення t_j за попередніми).

Однак більш суттєве значення мають випадкові потоки однорідних подій – потоки з обмеженою післядією, якщо випадкові величини S_j є незалежними.

Також великий інтерес представляють *стаціонарні* потоки, для яких імовірнісний режим не залежить від часу. Потік однорідних подій має назву стаціонарного, якщо імовірність $p_k(t, t_0)$ появи k подій за проміжок часу (t_0, t_0+t) не залежить від t_0 , а залежить тільки від t та k .

Потік однорідних подій називається *ординарним*, якщо імовірність $\psi(t_0, t)$ появи двох та більше подій за проміжок часу (t_0, t_0+1) при будь-якому t_0 є нескінченно малою величиною порівняно з t . Випадковий потік однорідних подій з обмеженою післядією може

бути потоком без *післядії*, якщо відсутня імовірнісна залежність подальшого перебігу подій потоку від попереднього. Важливу роль відіграє так званий *найпростіший потік однорідних подій*. Потік називається найпростішим, якщо він є стаціонарним, ординарним і потоком без *післядії*.

Формалізуємо представлення процесів. Нехай система масового обслуговування складається з n ліній, здатних одночасно обслуговувати заявки. Лінія може знаходитися у одному з двох станів – «вільно» або «зайнято». Якщо в момент надходження заявки є вільні лінії, то заявка приймається до обслуговування, якщо всі лінії зайняті - заявка залишається в системі впродовж часу τ_n – час перебування заявки у системі, як претендента на обслуговування. Якщо за час τ_n заявка не буде прийнята на обслуговування, то вона буде втрачена.

У залежності від величини τ_n системи масового обслуговування поділяють на три відмінних класи:

- системи з відмовами. Якщо $\tau_n=0$, то заявка, яка надійшла або негайно обслуговується, або отримує відмову;
- системи з очікуванням. Коли $\tau_n \rightarrow \infty$, заявки, які надійшли до системи не отримують відмову, а очікують коли вони будуть обслуговані;
- системи з обмеженим очікуванням. Якщо $0 < \tau_n < \infty$, заявка застала усі лінії зайнятими, очікує протягом τ_n у черзі, а після закінчення цього часу отримує відмову.

Також для характеристики властивостей обслуговуючої системи необхідно задати τ_s – час обслуговування заявки (час зайнятості лінії).

Реальний процес функціонування системи масового обслуговування слід представляти у вигляді актів (фаз) обслуговування. У приватному випадку обслуговування може бути однофазним.

Дуже розповсюдженим є обслуговування з перевагою (з пріоритетом).

Для систем з відмовами найбільш часто використовується показник якості – *середня доля відмов* $R(t_0, t)$ за проміжком часу (t_0, t_0+1) . Також величина R має сенс імовірності відмов для заявки.

Сутність методу імітаційного моделювання стосовно завдань масового обслуговування полягає в наступному. Будуються алгоритми, за допомогою яких можна виробляти випадкові реалізації

заданих потоків однорідних подій, а також моделювати процеси функціонування обслуговуючих систем. Ці алгоритми використовуються для багаторазового відтворення реалізації випадкового процесу обслуговування при фіксованих умовах завдання. Отримана при цьому інформація про стан процесу піддається статистичній обробці для оцінки величин, що є показниками якості обслуговування.

Цей метод дозволяє досліджувати залежність якості обслуговування від характеристик потоку заявок і параметрів обслуговуючої системи та вивчати перехідні режими.

4.2 Способи моделінгу систем

Математичні моделі детерміністичних об'єктів (при описанні яких вплив випадкових факторів не враховуються), що функціонують у дискретному часі, відносяться до різних типів кінцевих автоматів. Кінцевий автомат характеризується *кінцевими множинами станів z , вхідними сигналами x і вихідними сигналами y* . Автомат можна задати *таблицями переходів та виходів, або графом*.

Стохастичні об'єкти (при описанні яких враховуються випадкові фактори), що функціонують у дискретному часі, можна представляти *ймовірнісними автоматами*. Функція переходів ймовірнісного автомата визначає *розподіл ймовірностей* на множині станів (автомат з випадковими переходами), а функція виходів – *розподіл ймовірностей* на множині вихідних сигналів (автомат з випадковими виходами). Функціонування ймовірнісних автоматів вивчається за допомогою апарату ланцюгів Маркова. Математичними моделями стохастичних об'єктів з безперервним часом служать *системи масового обслуговування* або представники *Марковських випадкових процесів*.

Загальні властивості *динамічних систем*:

- елемент складної системи функціонує в часі (в кожен момент часу 4 він знаходиться в одному з можливих станів z);
- протягом часу елемент переходить з одного стану в інший під дією внутрішніх і зовнішніх причин;
- у процесі функціонування елемент взаємодіє з іншими елементами складної системи і об'єктами зовнішнього середовища (отримує вхідні сигнали x і вихідні сигнали y).

Вплив, представлений набором своїх характеристик, зазвичай називають *сигналом*. Таким чином, взаємодія елементів складної системи, вивчається в рамках *механізму обміну сигналами*.

Сигнали передаються по *каналам зв'язку*, прокладеним між елементами складної системи. Канал, що передає сигнали *миттєво* і *без спотворень*, називається *ідеальним*.

Повністю і правильно формалізована складна система має тільки ідеальні канали зв'язку. Фізичні канали зв'язку не є ідеальними. Тому їх необхідно розглядати як *самостійні елементи системи*, функціонування яких зводиться до відповідних затримок і спотворень сигналів.

При побудові математичної моделі складної системи необхідно враховувати її взаємозв'язок із зовнішнім середовищем.

Взаємозв'язок елементів всередині складної системи з об'єктами зовнішнього середовища включає набори наступних складових:

- процес формування вихідного сигналу елементом, який видає сигнал;
- визначення адреси передачі для кожної характеристики вихідного сигналу;
- проходження сигналів по каналах зв'язку і компоновка вхідних сигналів для елементів, які приймають сигнали;
- реакція елемента, який приймає сигнал, на вхідний сигнал.

Методика описання зв'язків між елементами складної системи за допомогою операторів сполучення поширюється на ті випадки, коли в процесі функціонування системи, структура зв'язків змінюється в часі або підпорядковується зовнішнім командам управління. Для урахування згаданих факторів достатньо розглядати оператори сполучення виду

$$Y_i^{(k)} = R(X_i^{(j)}, t, \alpha),$$

де t – час, α – параметр (сигнал) управління, X – вектор вхідних сигналів, k – оператор зв'язку.

Складні системи, що складаються із багатьох підсистем різної фізичної природи, кожну з яких можна вважати системою, називають *агрегативні системи (АС)*.

З точки зору моделювання агрегат виступає як досить універсальний переробник інформації – він сприймає входні та управляючі сигнали та видає вихідні сигнали. (див. рисунок 4.1).

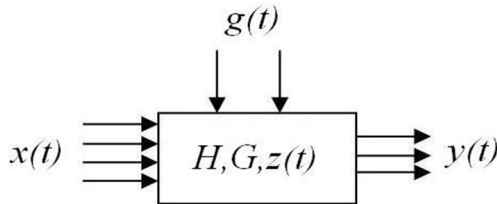


Рисунок 4.1 – Модель системи

У кожний момент часу $t \in (0, T)$ агрегат знаходиться в одному з можливих станів. Стан агрегату є елементом деякої множини Z . Якщо стан $z = (z_1, z_2, \dots, z_i)$ виявляється дійсним вектором, z_i зазвичай називають фазовою координатою. Коли аргумент t проходить свої значення в інтервалі $(0, T)$, стан z змінюється як функція часу $z(t)$. Функції $z(t)$ називають фазовими траєкторіями.

Агрегат має особливі входні контакти, до яких надходять в моменти часу τ_i управляючі сигнали. Управляючий сигнал g є елементом множини G . На виході агрегату утворюються вихідні сигнали. Вихідний сигнал y є елементом деякої множини Y і визначається за станом агрегата $z(t)$ з допомогою оператора G . За кінцевий інтервал часу в агрегат надходить кінцеве число входних і управляючих сигналів, а також агрегат видає кінцеве число вихідних сигналів.

Оператор H зазвичай називають оператором переходів (до нового стану), а оператор G – оператором виходів.

Ціллю моделювання АС є отримання характеристик, які визначаються за допомогою станів системи, яка моделюється.

Принцип побудови моделюючого алгоритму, що дозволяє визначати послідовні стани складної системи через деякі інтервали часу, називають «принципом Δt ». Він є універсальним.

Мови та програми моделювання є проблемно-орієнтованими, тобто вони відображають деяку сукупність понять, характерних для даного класу задач. В будь-якій мові моделювання існує апарат опису статичної та динамічної структур системи.

4.3 Скорочені теоретичні відомості по системі NetCracker

Моделювання систем інформаційної електроніки має свої особливості. Останнім часом, для аналізу таких систем використовують принципи імітаційного моделювання.

NetCracker® – програмний пакет, розроблений компанією NetCracker Technology, дозволяє створювати проекти обчислювальних мереж різної складності/топології та проводити їх аналіз, використовуючи технологію імітаційного моделювання. Цей пакет безкоштовно розповсюджується по мережі Internet, при цьому має достатньо широкі можливості.

Область застосування пакету – створення проекту мережевого рішення, тестування цього рішення і документування остаточного варіанту. База даних устаткування допускає, хоча і з деякими обмеженнями, додавання нового обладнання мережі з характеристиками, що задаються користувачем. Ця можливість, зокрема, в достатній мірі компенсує відсутність новітнього устаткування, моделі яких користувач може створити самостійно. Точність аналізу дозволяє оцінити якісно можливість перевантаження устаткування і каналів передачі даних – знаходити «вузькі місця» проекту, пакет робить можливим ознайомитися з практикою створення найрізноманітніших мережевих рішень майже «живцем» без дорогої тестової лабораторії.

Необхідно враховувати, що вимоги пакету до продуктивності процесора ростуть у міру збільшення числа заданих потоків даних.

NetCracker використовує технологію імітаційного моделювання мережі та дозволяє одержати результати у випадках, коли аналітичні розрахунки громіздкі, у край складні, а нерідко і неможливі.

Нижче приведена універсальна методика проектування комп'ютерних мереж в NetCracker Professional:

1. Вибір розміру мережі (кількість робочих станцій).
2. Вибір топології мережі (кільцева, радіальна, шинна ..., etc).
3. Вибір типу зв'язку (ethernet, token ring, fddi, atm ..., etc).
4. Визначення та розміщення пристроїв та відстаней між ними.
5. Вибір мережевих протоколів (tcp/ip, ipx/spx, appletalk, decnet..., etc).
6. Визначення трафіку передачі даних з урахуванням розподілу розміру транзакцій та часових відстаней між ними.

Задаються розміри блоків даних (транзакції) (англ. Transaction size) і час між пакетами даних (транзакціями) (англ. Time Between Transactions), закони розподілу і відповідні статистичні характеристики (для розміру даних і часу приходу), оскільки потоки даних мають стохастичну природу.

Властивості «обслуговуючого приладу» в NetCracker, на жаль, визначаються не достатньо детально: у вигляді фіксованої затримки обслуговування і абсолютної межі швидкості надходження заявок. Розмір буфера «приладу» задати не можна.

4.3.1 Приклад імітаційного моделювання комп'ютерної мережі

Нижче наведено приклад моделювання мережі, що складається з 20 робочих станцій, розташованих в двох кімнатах, відстань між якими 20 м. У одній з кімнат розташований файловий сервер, який виходить в Internet через супутник і міські мережі. Тип зв'язку – Fast Ethernet 100 Мбіт/с. Структура мережі – зіркова, тобто всі робочі станції сполучені через комутатори.

Створюємо новий проект в програмі NetCracker. У каталозі devices натискаємо на вкладку *buildings* і поміщаємо в робочу область два об'єкта – будівлі (*building*). На одному з них в контекстному меню вибираємо *expand* (розширення). У вікно, що з'явилося, перетягуємо дві кімнати (*room*). Їх теж розширюємо. З'явилися два вікна, що є двома кімнатами в будівлі. Відкриваємо вкладку *lan workstations*. Знизу на панелі зображень з'явиться набір робочих станцій. Перетягуємо в кожну кімнату по 10 звичних станцій (PC) і по одному принтеру з вкладки *peripheral equipment/printers*.

Відкриваємо вкладку *switches/workgroup/ethernet/3com.corp*. З панелі зображень перетягуємо в кожну кімнату OfficeConnect Switch 1600. Це 16-портовий комутатор, що підтримує зв'язок Ethernet. Оснастимо кожну робочу станцію мережевим адаптером Fast Ethernet, що здійснює частоту передачі даних 100 мбіт/с з вхідною і вихідною буферизацією розміром в 100 кбайт (рисунки 4.2).

За допомогою команди *link devices* (рисунки 4.3) сполучаємо кожну робочу станцію і принтери з комутатором, а також комутатори між собою. Відстань до комутатора приймаємо рівним 5м, тип кабелю – вита пара (*twisted pair*).

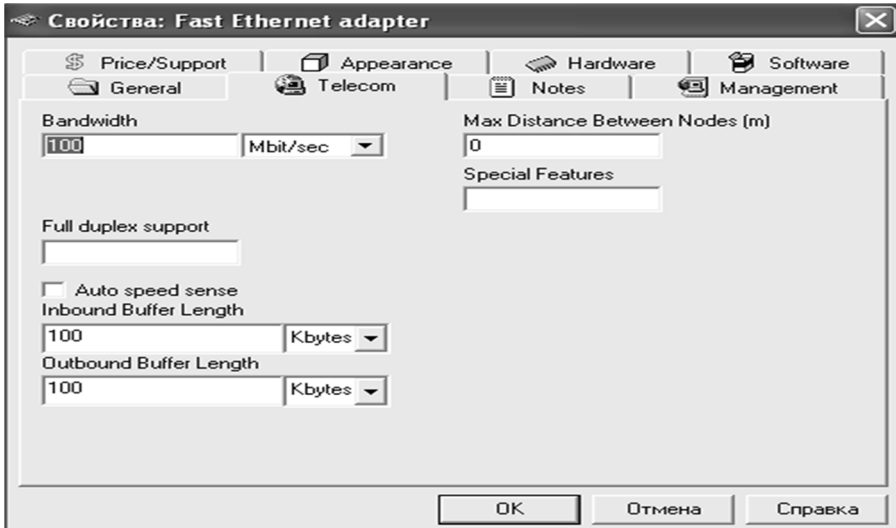


Рисунок 4.2 – Установка частоты передачи и размеров буферов устройства



Рисунок 4.3 – Команда link devices

Отримасмо наступну модель лабораторії (рисунок 4.4):

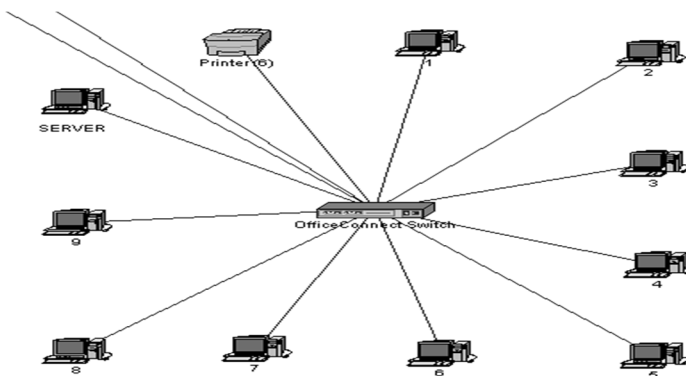


Рисунок 4.4 – Модель мережі кімнати

Вихід в Internet здійснюється через маршрутизатор, сполучений з комутатором і супутниковою антеною. Натискаємо вкладку *routers and bridges* і встановлюємо маршрутизатор *router* у вікно будівлі, з'єднавши його з одним з комутаторів через порт *fast ethernet*. *Router* – маршрутизатор – ізолює трафік, утворюючи логічні сегменти за допомогою явної адресації та здійснює вибір найраціональнішого маршруту з декількох можливих. Існує можливість зв'язувати в єдину

мережу підмережі побудовані з використанням різних мережевих технологій. Кожен пристрій має порти (рис. 4.5), через які з'єднується з іншими пристроями. Залежно від типу порту і типу кабелю зв'язку передача даних здійснюється з різною частотою (рис. 4.6). Необхідно виключити перевантаження каналів зв'язку і оптимізувати час очікування даних для кожного трафіку.



Рисунок 4.5 – Порти пристрою

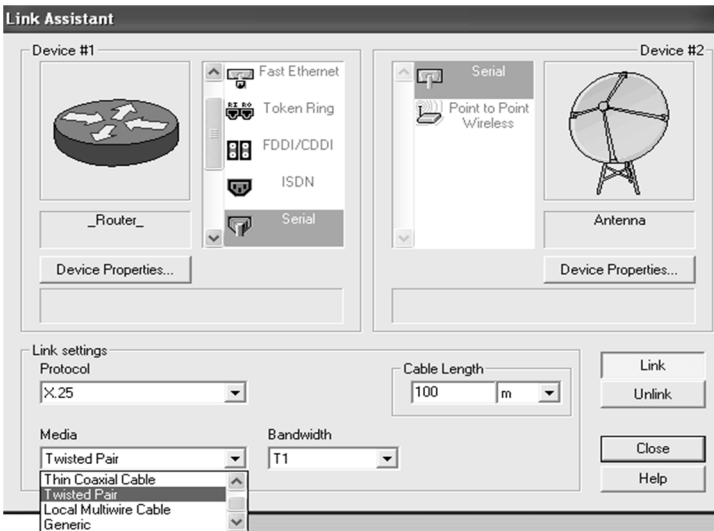


Рисунок 4.6 – Вибір типу зв'язку

Оберемо протокол маршрутизації *Rip* для протоколу *TCP/IP* (рис. 4.7), відкривши в меню *global* вкладку *model settings*:

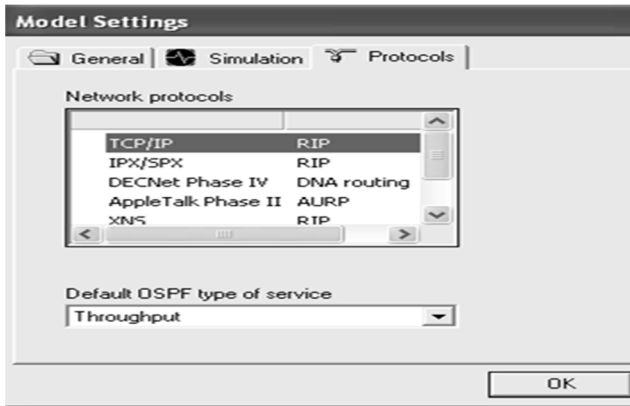


Рисунок 4.7 – Установка протоколів маршрутизації

Встановлюємо тип мережевого протоколу, що використовується кожною робочою станцією – *TCP/IP* (рис. 4.8), в контекстному меню обираємо вкладку *properties*:

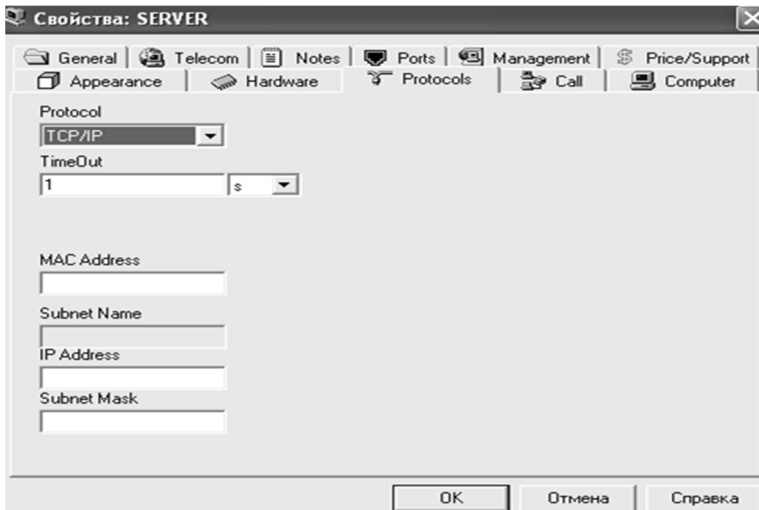


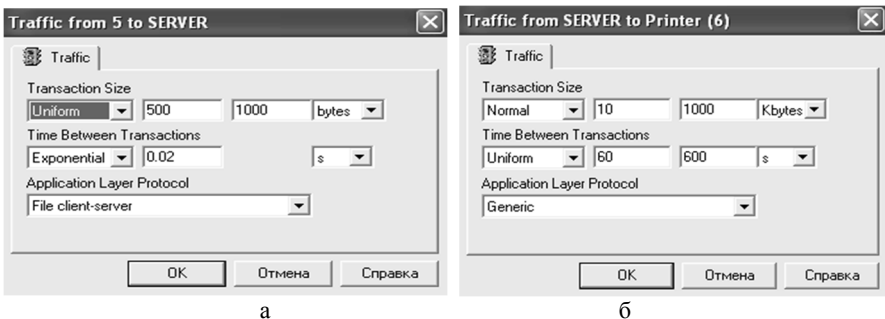
Рисунок 4.8 – Установка типа протоколу мережі

Обираємо команду *set traffic* (рис. 4.9). Задамо трафік між робочими станціями та сервером – *file server's client*, трафік між сервером і другою будівлею – *small office*.



Рисунок 4.9 – Команда set traffic

Задамо трафік для пар клієнт-сервер та принтер-сервер. Виберемо розподіл розміру транзакцій і паузи між ними (рис. 4.10):



а

б

а – для пари клієнт-сервер; б – для пари принтер-сервер

Рисунок 4.10 – Установка параметрів трафіку

Змоделюємо передачу даних через супутник і міську мережу АТМ між двома будівлями (рис.4.11), в одній з яких знаходиться наша мережа. Для цього встановимо на кожену будівлю супутникову антену з вкладки *satellite communication equipment/earth equipment/antennae/generic devices*. З'єднаємо першу антену з маршрутизатором через *serial* порт, а другу – з будівлею. Також з'єднаємо через маршрутизатор будівлі через міську мережу АТМ. Встановимо супутник з вкладки *satellite communication equipment/satellites* і з'єднаємо його з антенами.

ATM (Asynchronous Transfer Mode – асинхронний режим передачі) – технологія для нового покоління мереж з інтеграцією послуг *ISDN*: передача даних у вигляді пакетів, що адресуються;

пакети невеликого фіксованого розміру, що зменшує затримки в мережі.

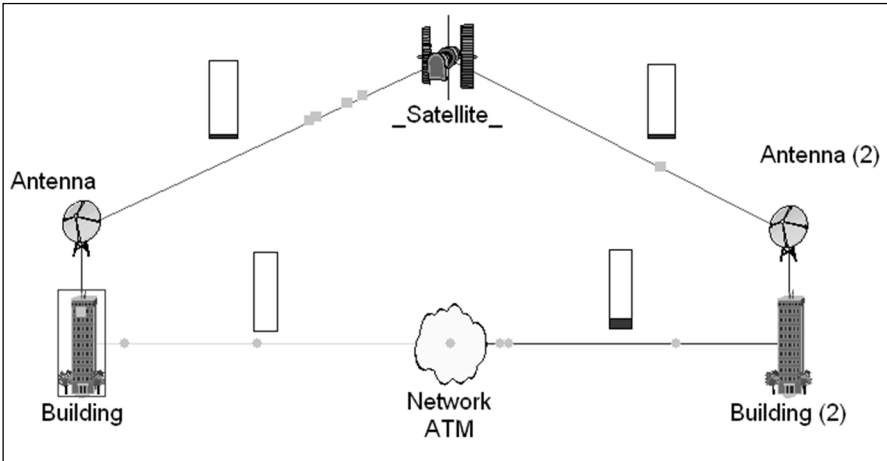


Рисунок 4.11 – Модель мережі: верхній рівень

На рисунку 4.12 спостерігається втрата даних у вигляді червоного спалаху. Причина – час очікування в каналі менший, ніж час проходження пакетів. Час очікування (*timeout*) задається за допомогою мережевого протоколу, в нашому випадку – *TCP/IP*. Задамо більший час очікування (рис. 4.13).

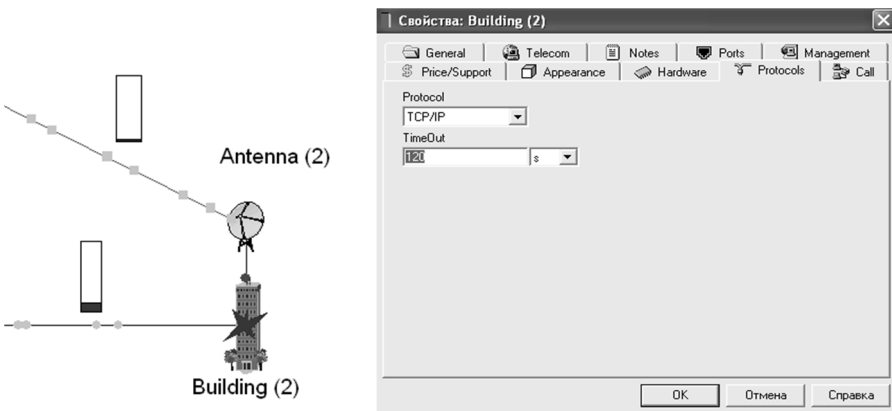


Рисунок 4.12 – Втрата даних

Рисунок 4.13 – Завдання часу очікування

Як бачимо, втрата даних припинилася (рис. 4.14). Модель мережі оптимізовано.

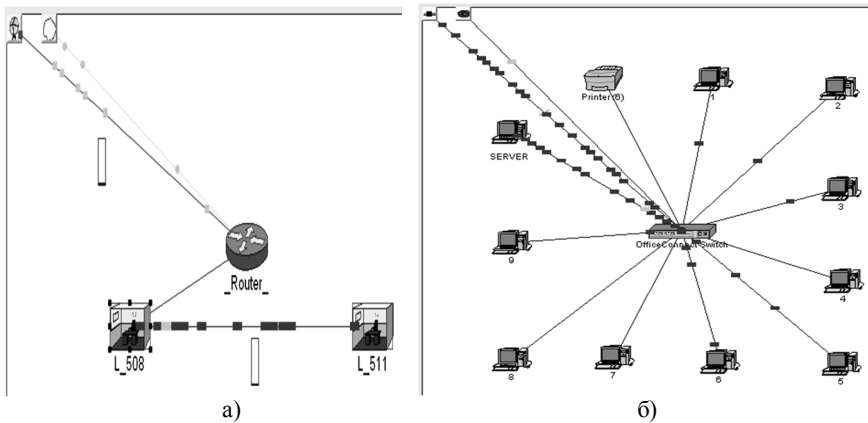


Рисунок 4.14 – Модель мережі: а) – рівень будівлі; б) – рівень кімнати

За допомогою індикаторів статистики, які встановлюються з контекстного меню каналу зв'язку вкладкою *statistics* і установкою галочки *current utilization*, спостерігаємо поточну завантаженість мережевих каналів. Оскільки змодельована нами мережа є невеликою, перевантажень каналів зв'язку не відбувається. Середня завантаженість мережі – в межах 10 %.

4.4 Етапи виконання роботи

1. Встановити програму моделювання комп'ютерних мереж.
2. Коротко опишіть можливості, бібліотеку та інтерфейс програми.
3. Розробіть модель лабораторії (десять робочих станцій, принтер і т.д) із виходом в Інтернет.
4. Варіюючи розподіл густини транзакцій та розміру пакетів промодельуйте різні режими роботи мережі.
5. Отримайте мережу (структуру і параметри), яка дає оптимальне навантаження на елементи. Зафіксуйте, при яких параметрах, це отримано. Результати занести в звіт.

4.5 Контрольні запитання

1. Критерії оптимальності систем.
2. Принципи імітаційного моделювання (виділити особливості, порівняно із моделюванням на схемотехнічному рівні).
3. Алгоритми та програми імітаційного моделювання.
4. Системи масового обслуговування (СМО), приклади.
5. Основні елементи моделей СМО.
6. Закони математичної статистики для моделювання інформаційних систем.
7. Методика і задачі моделювання комп'ютерних мереж в середовищі програми NetCracker, Cisco Packet Trace, etc.
8. Типи моделей транзакцій (функції розподілу).

5 ЛАБОРАТОРНА РОБОТА №5

“ МОДЕЛЮВАННЯ НЕЙРОННИХ МЕРЕЖ ”

Мета: дослідити макромоделі операційних підсилювачів (ОП) макромоделі основних логічних елементів та принципи подійного алгоритму.

5.1 Машинне навчання

Машинне навчання (ML) є підмножиною штучного інтелекту (AI). Воно використовується у випадках, коли потрібно, щоб машини навчалися на великих обсягах даних. Отримані знання застосовуються до нового набору даних. Діаграма Венна (рис.5.1) візуалізує перекриття термінології, пов'язаної зі штучним інтелектом.

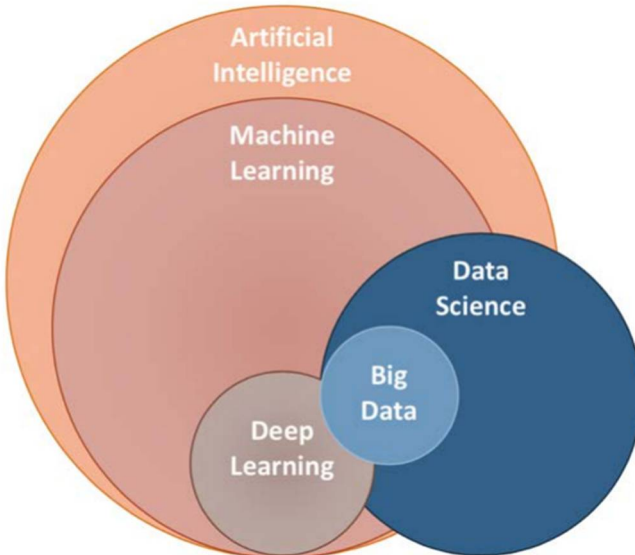


Рисунок 5.1 – Діаграма Венна

Різниця між звичайною автоматизацією і автоматизацією на базі ML показана на рисунку 5.2.

ML надає машині можливість навчатися на нових наборах даних або про них без явних інструкцій. Можна сказати, що ML є реалізацією або поточним застосуванням AI.

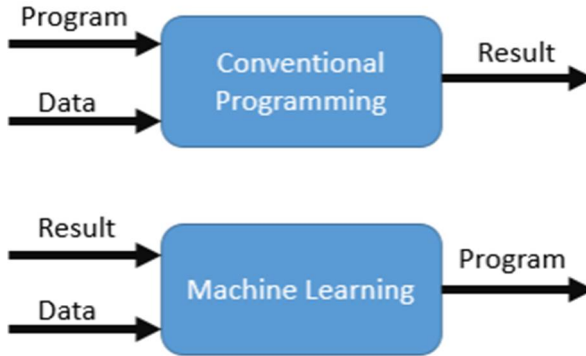


Рисунок 5.2 – Різниця між звичайною автоматизацією і автоматизацією на базі ML

До найпоширеніших методів, які використовуються для "навчання машин", належать:

- Навчання з учителем (*supervised learning*)
- Навчання без учителя (*unsupervised learning*)
- Підкріплювальне навчання (*reinforced machine learning*)

У деяких методах машині заздалегідь повідомляють про незалежні (вхідні) та залежні (вихідні) змінні. Машина вивчає зв'язок між цими двома типами змінних, аналізуючи набір даних, який називається "тренувальним набором даних" (рис.5.3).



Рисунок 5.3 – Процес тренування та тестування мережі

Перед тренуванням моделі даних виконують серію кроків із попередньої обробки даних.

Коли машина достатньо "навчена" або модель ML готова, її застосовують до нового набору даних, який називається "тестовим набором даних". Модель ML переходить у виробничий режим лише після достатнього тестування на надійність і точність.

ML використовує різні алгоритми, такі як проста лінійна регресія, регресія за допомогою дерев рішень, поліноміальна регресія, метод найближчих сусідів (*K-nearest neighbors*) тощо. Ці алгоритми застосовуються до задач регресії, прогнозування, класифікації тощо.

Завдяки значному розвитку бібліотек ML (наприклад, SciKit) за останні кілька років навіть програмісти без статистичної підготовки або формальної освіти в AI можуть використовувати ці бібліотеки для створення, тренування, тестування та розгортання моделей ML. Однак завжди корисно розуміти, як працюють різні алгоритми ML, щоб усвідомлювати свої дії. Останнім десятиліттям штучний інтелект і машинне навчання набули великої популярності для вирішення складних реальних задач. Особливо це стосується проблем, які раніше вважалися дуже складними або навіть неможливими для вирішення комп'ютерами.

5.2 Нейронні мережі та функції активації

Нейронна мережа – це колекція штучних нейронів, чия поведінка заснована на роботі біологічного мозку. Ці штучні нейрони є спрощеннями біологічних нейронів, але засновані на концепціях, які дозволяють їм навчатися. В нейронних мережах є частина нейронів, які називаються функцією активації. Нейромережа – це математична модель, яка імітує здібності людського мозку сприймати та аналізувати інформацію. За типом архітектури існує близько 30 різновидів нейронних мереж: односпрямовані, рекурентні, згорткові, мережі граф-нейронні, спайкові тощо. Структура нейронних мереж:

1. **Вхідний шар**, де мережа приймає запит.
2. **Приховані шари**, де відбуваються обчислення.
3. **Вихідний шар**, коли мережа видає результат.

Нейромережа може виконувати різні типи завдань, залежно від архітектури мережі та навчальних даних, зокрема:

- **Класифікація:** Виявлення спаму, сортування зображень.
- **Регресія:** Прогнозування цін на нерухомість, погоди або оцінка вартості акцій.
- **Генерація:** Створення текстів, зображень, відео та навіть аудіо.
- **Виявлення об'єктів:** Автоматичний контроль якості, системи безпеки.

Нейромережі активно використовуються в таких галузях, як комп'ютерний зір, машинний переклад, автономна навігація, розпізнавання мови, а також:

- **Медицина:** Діагностика захворювань, аналіз медичних зображень, передбачення ефективності лікування.
- **Фінанси:** прогнозування цін на акції, оптимізація портфелів інвестицій.
- **Маркетинг:** Прогнозування вподобань клієнтів, автоматичне створення рекламних матеріалів, аналіз споживчої поведінки.
- **Автоматичний переклад:** Нейромережі можуть навчатися перекладати тексти між різними мовами, забезпечуючи швидкий та точний переклад.
- **Розпізнавання мови:** Розпізнавання та розуміння голосових команд для керування різними пристроями, такими як смартфони та домашня техніка.
- **Автономні транспортні засоби:** Навігація, ухилення від перешкод та безпечне керування автономними автомобілями або дронами.
- **Безпека:** Аналіз відео та зображень для виявлення підозрілої діяльності, захист від кібератак або виявлення шахрайства.
- **Рекомендаційні системи:** Аналіз історії перегляду користувача та рекомендувати продукти, фільми, музику та інше на основі їх вподобань.

Вплив входів на вихід можна описати матрицею коефіцієнтів яка називається тензор. В термінах ML тензор не має геометричних або симетричних законів перетворення, які зазвичай застосовуються в математиці та фізиці. У ML тензор – це узагальнення матриці, також відоме як гіперматриця або p -вимірний масив. Навчання мережі це параметрична оптимізація, цільова функція це мінімізація похибки, для цього цілеспрямовано змінюється параметри в тензорі (вага для кожного сплайна).

Глибоке навчання (DL)

Глибоке навчання можна розглядати як підмножину або продовження ML. Воно стає в нагоді, коли ML не може повністю забезпечити бажані результати. Загалом ML підходить для невеликих наборів даних, тоді як DL – для складніших задач, які вимагають:

- Великої кількості характеристик у даних
- Величезних обсягів даних
- Надзвичайно високої точності

У порівнянні з ML, DL здатне вирішувати складніші задачі, але воно важче у реалізації, вимагає спеціалізованого обладнання (наприклад, GPU) та більше часу для тренування моделі.

DL використовує моделі нейронних мереж для аналізу великих обсягів даних. Siri, Alexa та Google Assistant є прикладами застосування DL для розуміння ваших запитів. Функції Facebook, які розпізнають обличчя на фотографіях, чи рекомендації Netflix — це також робота DL.

Застосування DL активно впроваджується у сферах, таких як:

- Агрегування новин і виявлення фейкових новин
- Автономні автомобілі
- Обробка природної мови (NLP)
- Візуальне розпізнавання
- Віртуальні асистенти

Прориви в DL стимулюють бум у галузі AI.

Можемо виділити три основні тенденції розвитку нейромереж.

1. Поліпшення продуктивності: Очікується, що нейронні мережі будуть продовжувати покращуватися у своїй продуктивності, точності та ефективності. Розвиток нових архітектур, оптимізація навчання та використання потужніших обчислювальних ресурсів дозволять створювати більш потужні та ефективні моделі.

2. Розширення сфери застосування: Нейронні мережі будуть знаходити все більше застосувань у різних галузях, включаючи медицину, фінанси, робототехніку, автомобільну промисловість та багато інших. Вони будуть використовуватися для вирішення складних завдань, автоматизації процесів та прийняття більш точних рішень.

3. Інтеграція з іншими технологіями: Нейронні мережі все частіше інтегруватимуться з іншими технологіями, такими як розподілені обчислення, квантові комп'ютери та інтернет речей. Це створить нові можливості для обробки та аналізу даних у реальному часі.

5.3 Вступ до B-Spline мереж

B-Spline мережі використовуються для відображення вхідних даних на вихід у заданих межах простору вхідних даних. Фактично вони дозволяють отримати передаточну функцію «чорного ящика», який має багато входів і 1 або кілька виходів (рис.5.4).

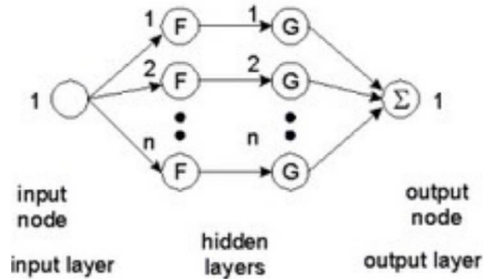


Рисунок 5.4 – Структура мережі

Вони містять приховані шари:

- У першому шарі до вхідних даних застосовуються "базові функції" F . Наприклад використовуються B-сплайни будь-якого порядку, що є частинами поліномів.
- Другий шар множить вхідні дані на вагові коефіцієнти (визначається ступінь впливу на результат).

Ці мережі можуть бути одновимірними (1 вхід) або багатовимірними (n входів), що дозволяє апроксимувати будь-які функції.

5.3.1 Навчання мережі B-Spline

Мережа B-Spline навчається шляхом адаптації ваг. Під час навчання вихід мережі порівнюється з бажаним виходом. Помилка між цими двома сигналами використовується для адаптації ваг. Швидкість цієї адаптації контролюється коефіцієнтом навчання. Високий коефіцієнт навчання змусить мережу швидко адаптувати свої ваги, але це може зробити її нестабільною. Встановлення коефіцієнта навчання на нуль дозволяє мережі зберігати ваги постійними. Навчання мереж B-Spline може виконуватися або після кожного зразка (навчання на кожному зразку), або після серії зразків (навчання після завершення сплайну).

Навчання на кожному зразку

Коли навчання відбувається на кожному зразку, після кожного зразка ваги адаптуються згідно з алгоритмом ітераційної оптимізації ваг. Задано певний вхід (x) і тільки обмежена кількість сплайнів $F_j(x)$ є ненульовими, тому лише кілька ваг адаптуються після кожного зразка.

5.4 Проектування нейронних мереж в 20-sim

САЕ програма 20-sim має засоби для автоматизованого синтезу контролюючих систем на базі нейронних мереж. Редактор нейронних мереж в 20-sim підтримує два відомих типи мереж:

а) адаптивні В-сплайн мережі (*Adaptive B-spline networks*) («В-сплайн» означає «базисний сплайн»);

б) мережі на основі багатосарового перцептрона Румельхарта (*Multi-layer Perceptron networks*), або MLP-мережі.

Основним елементом MLP-мережі, є штучний нейрон. Він виконує просту математичну операцію на входах. Такий нейрон наочно представлено на рисунку 5.5.

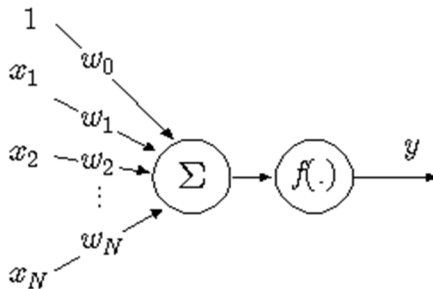


Рисунок 5.5 – Штучний нейрон

Вхід нейрона складається зі змінних $x_1 \dots x_n$ і зсуваючого елементу, відомого як постійний імпульс (*momentum constant*), що дорівнює 1. Кожне з вхідних значень помножується на ваговий коефіцієнт $w_0 \dots w_n$, після чого результати складаються у суматорі Σ . В результаті виконується проста математична функція f , відома як функція активації (*activation function*).

Таким чином, нейрон виконує розрахунки за формулою:

$$y = f(w_0 + x_1 \cdot w_1 + \dots + x_n \cdot w_n)$$

В нейронних мережах з навчанням перцептрона методом зворотного поширення помилки (*backpropagation*) нейрони або «вузли» згруповані в шари (*layers*).

В таких типах нейронних мереж розрізняють три групи шарів:

- вхідний шар, складається з сенсорних елементів;
- один чи кілька проміжних або прихованих (*hidden*) шарів, що складаються з асоціативних елементів;
- вихідний шар, що складається з реагуючих елементів.

Вузли між двома сусідніми шарами з'єднані між собою. Повне підключення мережі настає коли кожен вузол шару з'єднаний з кожним вузлом сусіднього шару. Вхідні сигнали надходять до вхідного шару, єдиною метою якого є поширення інформації в приховані шари, що і виконують фактичну обробку. Оброблена інформація підхоплюється нейронами вихідного шару і транспортується у вигляді вихідних сигналів у зовнішнє середовище. Вузли прихованих шарів оброблюють інформацію приймаючи певний ваговий коефіцієнт для кожного входу (див. рис. 5.5).

Коли мережа знаходиться у режимі навчання, фактор або вага кожного вузла адаптується згідно методу зворотного поширення помилки. А коли мережа знаходиться в режимі експлуатації, вагові коефіцієнти залишаються постійними (якими були виставлені в результаті навчання).

MLP-мережа має здатність до навчання шляхом підбору вагових коефіцієнтів. Під час навчання значення на виході нейронної мережі порівнюється з бажаним результатом, відхилення (похибка) між цими двома сигналами використовується для адаптації вагових коефіцієнтів (*weight*). Ступінь адаптації контролюється показником швидкості навчання (*learning rate*); якщо цей показник великий, то перцептрон буде швидко підбирати коефіцієнти, але стане потенційно нестабільним. Встановлення нульового значення для показника швидкості навчання призведе до утримання перцептроном значень вагових коефіцієнтів на постійному рівні, нейронна мережа перестане навчатися.

MLP-мережі, як і будь-який тип мережі зі зворотним поширенням помилки, може складатися з великої кількості нейронів, що згруповані у шарах. Хоча може бути використана будь-яка кількість прихованих шарів, MLP-мережа з одним прихованим шаром може апроксимувати будь-яку безперервну функцію (в 20-sim MLP-

мережі мають лише один прихований шар). Вікно редактора MLP-мережі показано на рисунку 5.6.

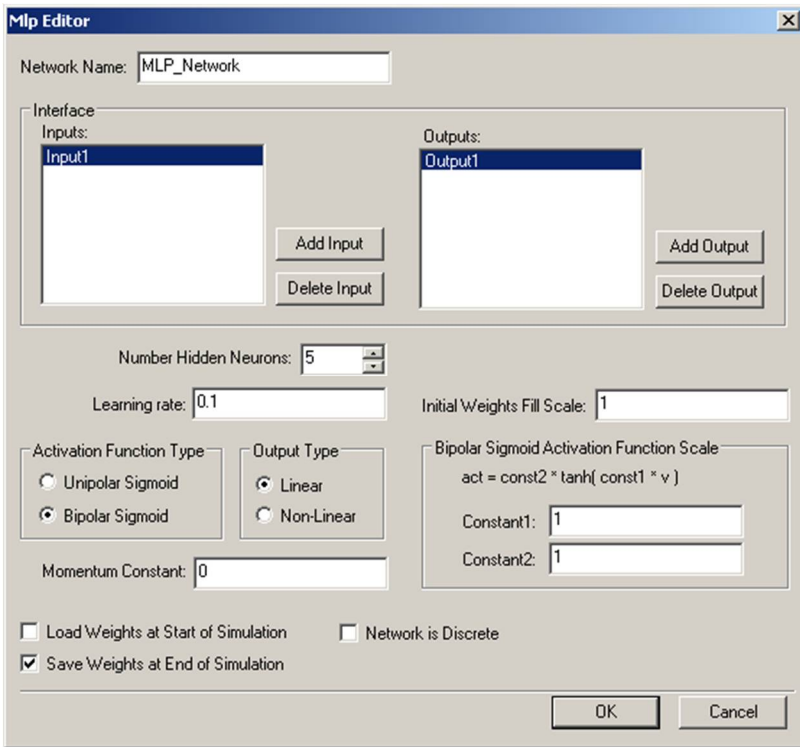


Рисунок 5.6 – Редактор MLP-мережі

Другий тип нейронної мережі, що підтримується в 20-sim – адаптивна В-сплайн мережа. Вікно та настройки редактора В-сплайн мережі представлені на рисунку 5.7.

В-сплайн мережа використовується для співставлення k входів до одного виходу на обмеженій області простору вхідних величин. Кількість входів k вказує на розмірність мережі. При $k=1$ мережа є одновимірною. Перший прихований шар розподіляє входи на n вузлів; кожен вузол цього шару має лише один вхід, до якого застосовується базисна функція F будь-якого порядку.

Другий прихований шар також складається з n вузлів з одним входом, до якого застосовується функція G , яка просто помножує вхідний сигнал in на деякий ваговий коефіцієнт w : $G = w \cdot in$. Вихідний вузол (у вихідному шарі) підсумовує результуючі значення на виході всіх вузлів другого прихованого шару. Коли сплайн-функції різних вузлів правильно розташовані, кожна одновимірна функція може бути апроксимована – сплайни ($F_1 \dots F_n$), поєднані з ваговими коефіцієнтами ($w_1 \dots w_n$), разом утворюють вихідну функцію.

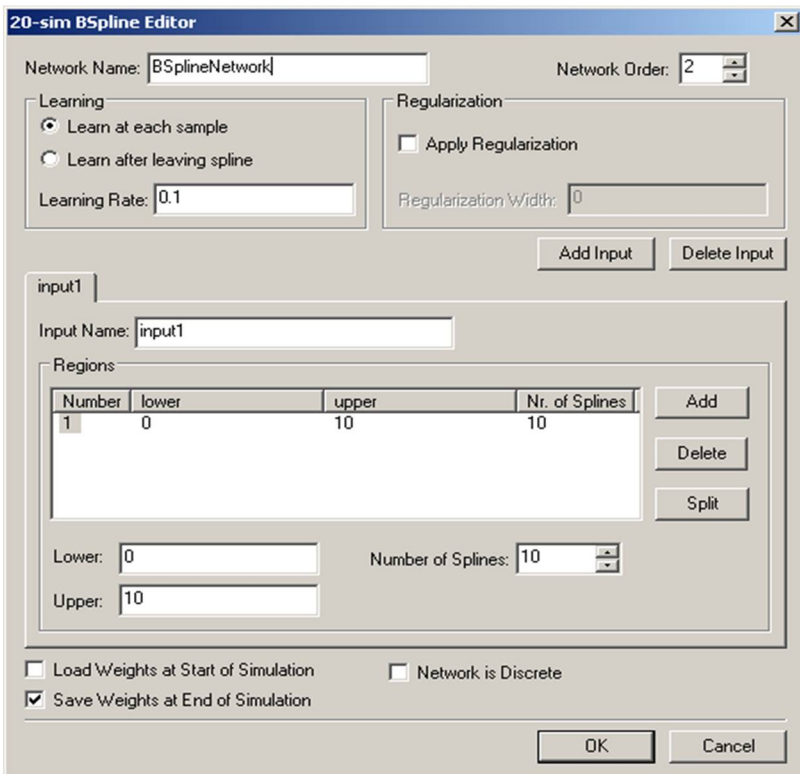


Рисунок 5.7 – Редактор В-сплайн мережі

Налаштування параметрів синтезу мережі в редакторах:

- **Network Name (Назва мережі):** це локальна назва підмоделі, яка представляє мережу B-Spline.

- **Order (Порядок):** введіть порядок сплайнів B-Spline, які повинні бути використані.
- **Learn at Each Sample (Навчання на кожному зразку):** виберіть цю опцію, якщо ви хочете, щоб мережа адаптувала ваги після кожного кроку симуляції.
- **Learn after leaving Spline (Навчання після виходу зі сплайну):** виберіть цю опцію, якщо ви хочете, щоб мережа адаптувала ваги тільки після виходу з області сплайну.
- **Learning Rate (Коефіцієнт навчання):** вкажіть коефіцієнт навчання, який має бути використаний під час навчання. Якщо ви не хочете, щоб мережа навчалася, введіть нульове значення.
- **Apply Regularization (Застосувати регуляризацію):** ця опція поки не підтримується.
- **Load Weights at Start of Simulation (Завантажити ваги на початку симуляції):** виберіть цю опцію, якщо ви хочете використовувати заздалегідь визначені ваги, збережені в файлі (з попереднього запуску). Перед кожним запуском симуляції вам буде запропоновано ввести ім'я файлу з вагами.
- **Save Weights at End of Simulation (Зберегти ваги в кінці симуляції):** виберіть цю опцію, якщо ви хочете зберігати ваги в файл. Після кожного запуску симуляції вам буде запропоновано ввести ім'я файлу для збереження ваг. *Перед запуском моделі на тестуванні (після навчання), обов'язково збережіть ваги.*
- **Network is Discrete (Мережа дискретна):** виберіть цю опцію, якщо мережа B-Spline з'єднана з моделями дискретного часу. Зніміть цю опцію, якщо мережа з'єднана з моделями безперервного часу.

Налаштування входів

Вкладки, показані в редакторі, використовуються для визначення входів мережі. Кожна вкладка показує специфічні налаштування для цього входу.

- **Add Input (Додати вхід):** додати новий вхід до мережі. Додається нова вкладка з налаштуваннями за замовчуванням.
- **Delete Input (Видалити вхід):** видалити вхід, визначений у вибраній вкладці (вкладка, яка знаходиться спереду).

Кожен вхід асоціюється з певною кількістю сплайнів. Щоб дати користувачеві більше гнучкості, кожен вхід можна поділити на певні області, кожна з яких має свою щільність сплайнів.

- **Add (Додати):** додати нову область сплайнів.
- **Delete (Видалити):** видалити вибрану область сплайнів.
- **Split (Розділити):** розділити вибрану область на дві області.

Кожну область можна редагувати. Просто виберіть її зі списку областей та використовуйте наступні опції:

- **Upper / Lower (Верхній / Нижній ліміт):** верхня та нижня межі області.
- **Number of Splines (Кількість сплайнів):** кількість сплайнів. Чим більше сплайнів ви виберете, тим точнішим буде вихід мережі для цієї області входу.

20-sim також підтримує багатовимірні нейронні мережі на основі В-сплайнів; максимальна розмірність – 256, що вказує на максимальну кількість входів мережі. Принцип навчання або тренування В-сплайн мережі такий самий, як і у MLP-мережі (адаптація вагових коефіцієнтів), окрім можливості навчання після кожної окремої виборки (*learn at each sample*) та після серії виборок в кінці тренування сплайна.

Таким чином, нейронні мережі можуть застосовуватись для ідентифікації невідомих функцій та в системах управління, наприклад, у якості нечіткого нейронного регулятора у прямому зв'язку (*neuro fuzzy feedforward controller*) з можливістю навчання і адаптації до збурення з боку об'єкта управління; для розпізнавання образів, прийняття рішень і керування, кластеризації, прогнозування, апроксимації безперервних функцій, стиснення даних та автоасоціативного запам'ятовування.

5.5 Етапи виконання роботи

Вам необхідно послідовно промоделювати файли для одновимірної нейронної мережі (*1DBSplineNetwork Train*, а потім *1DBSplineNetwork Run*) і повторити для двохвимірної (*2DBSplineNetwork Train*, а потім *2DBSplineNetwork Run*).

Місце знаходження файлів для дослідження в бібліотеці 20-sim представлено на рисунку 5.8.

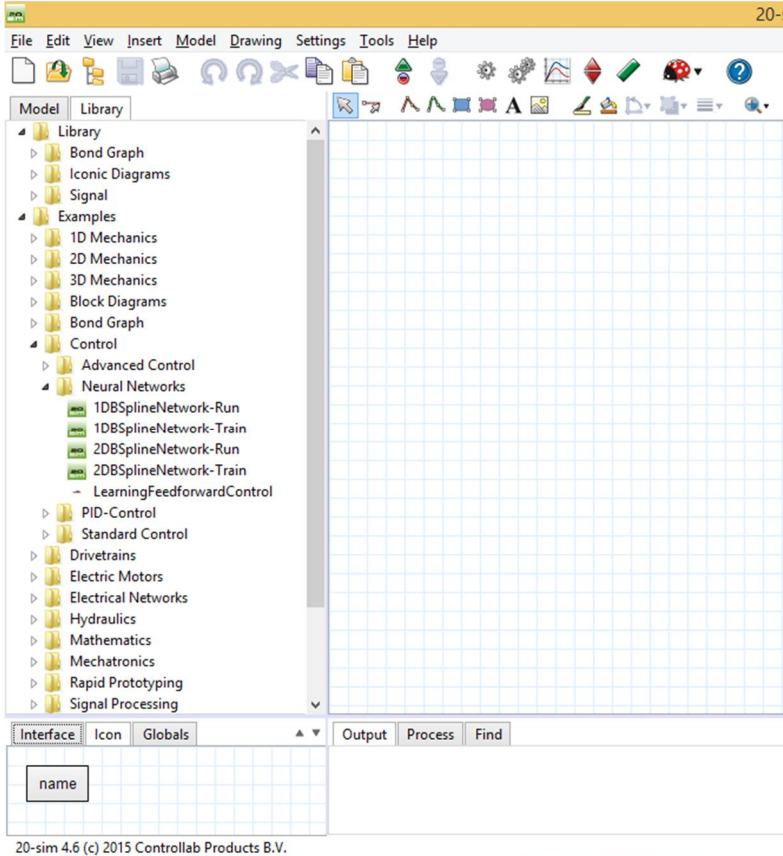


Рисунок 5.8 – Схеми для дослідження в бібліотеці

5.5.1 Тренування В-сплайн мережі

Інструкція до роботи з моделлю 1D B-Spline Network

1. У бібліотеці моделей виберіть: **Examples\Control\Neural Networks**.
2. Оберіть модель **1DBSplineNetwork-Train.emx** (рис.5.9). Після цього попередньо визначена модель буде відкрита.
3. У меню виберіть **Model** і натисніть **Start Simulator**. Відкриється симулятор із завантаженим попередньо налаштованим експериментом.
4. У меню симулятора натисніть **Simulation** і виберіть **Run**.

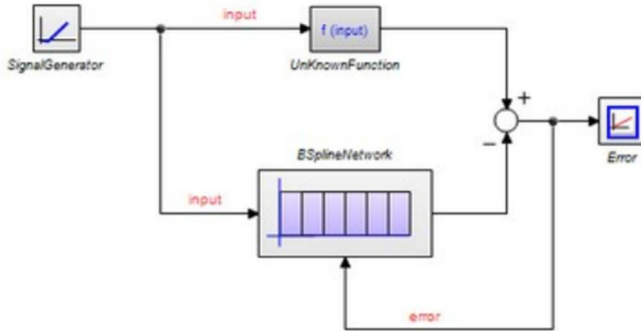


Рисунок 5.9 – Схема тренування одновимірної нейронної мережі

Компоненти системи представлені в таблиці 5.1.

Таблиця 5.1 – Компоненти системи

Компонент	Опис
Мережа В-сплайнів	Перший шар – це набір В-сплайнів, Другий шар – це лінійний шар, який використовує значення ваг В-сплайнів які використовуються для ітераційної апроксимації вхідного сигналу.
Вихідний сигнал $f(x)$	Є прогнозом значення вихідного сигналу, зробленим мережею В-сплайнів.
Блок «Генератор сигналу»	Для створення вхідного сигналу, який діє на Блок із невідомою передаточною функцією і на модель із В-сплайнами. Сигнал може бути будь-якою неперервною функцією, такою як синусоїда або квадратна парабола.
Блок «Невідома функція»	Перетворює вхідний сигнал на вихідний завдяки своїй передаточній функції (заздалегідь невідомій, але ми можемо розкрити параметри блоку і дізнатися)
Блок «Помилка»	Для обчислення динамічного відхилення між результатом роботи (навчання) мережі В-сплайнів та фактичним значенням вихідного сигналу.

На етапі початкового налаштування В-сплайн мережі її ваги є абсолютно не визначеними. Ці ваги можуть бути встановлені

випадковим чином або всі можуть бути встановлені на нуль. Тому симуляція зазвичай починається з навчання мережі для налаштування ваг, щоб забезпечити найкращу апроксимацію нелінійної функції. Після налаштування ваг (тренування, валідація) мережу можна використовувати з будь-яким бажаним вхідним сигналом (тестування верифікація та розгортання).

Виберіть модель *Bspline Network* і натисніть "Go Down" в меню *Model*. Тепер з'явиться редактор B-Spline. З меню редактора виберіть *Model* і натисніть "Check Complete Model". Це застосує зміни налаштувань. Поверніться до симулятора і натисніть "Simulation", а потім "Run". Після запуску симуляції ви побачите результати, які можуть включати графіки, що демонструють процес навчання мережі, апроксимацію нелінійної функції та налаштування ваг B-сплайнів. Після запуску симуляції з'явиться діалогове вікно збереження, яке запитує вас про введення назви файлу ваг. Це файл, що містить усі ваги навченого B-Spline мережі. Ми можемо використовувати ці ваги для звичайних запусків з мережею. Введіть ім'я файлу (не використовуйте заздалегідь визначений файл ваг!), наприклад, test.wgt, і натисніть "Save". Закрийте симулятор. Тепер все готово для виконання звичайного запуску з мережею.

5.5.2 Тестування 1-D B-Spline мережі

У бібліотеці моделей виберіть *Examples\Control\Neural Networks*. Виберіть модель *IDBSplineNetwork-Run.emx*. Тепер буде відкрито попередньо визначену (натреновану) модель. Як ви можете бачити, це саме та сама модель, що використовувалась для навчання, але тепер використовується вхід у вигляді функції *ramp*. З меню виберіть *Model* і натисніть *Start Simulator*. Тепер відкриється симулятор з завантаженою попередньо визначеною матрицею ваг/*weights*. З меню симулятора виберіть *Simulation* і натисніть *Run*. Не зберігайте файл ваг після завершення запуску!

Як ви можете побачити, мережа B-Spline все ще має налаштування для навчання та запитує файл ваг після завершення симуляції. Тепер ми змінимо налаштування мережі B-Spline. Відкрийте редактор B-Spline (виберіть модель, натисніть *Go Down*). Щоб створити звичайний запуск з мережею, ми повинні встановити коефіцієнт навчання на нуль і завантажити файл ваг перед

симуляцією. Змініть налаштування мережі до вигляду, як на зображенні:

Натисніть *OK*, щоб закрити редактор B-Spline. З меню редактора виберіть *Model* і натисніть *Check Complete Model*. Це застосує зміни в налаштуваннях. Поверніться до симулятора і натисніть *Simulation* та *Run*. Перед запуском з'явиться діалогове вікно *Open*, яке запитає вас про введення назви файлу ваг. Це файл, що містить усі ваги навченого B-Spline мережі. Використовуйте файл, який ви зберегли під час навчання, або попередньо визначений файл ваг *1DBSplineNetwork.wgt*. Як ви можете побачити, функція апроксимована мережею B-Spline за допомогою 10 сплайнів першого порядку.

5.5.3 Тренування і розгортання двохвимірної BSpline мережі

Повторіть дослідження для двохвимірної мережі, для цього спочатку натренуйте мережу, представлену у файлі *2DBSplineNetwork Train* (рис.5.10). Не забудьте зберегти матрицю ваг, як це описано вище.

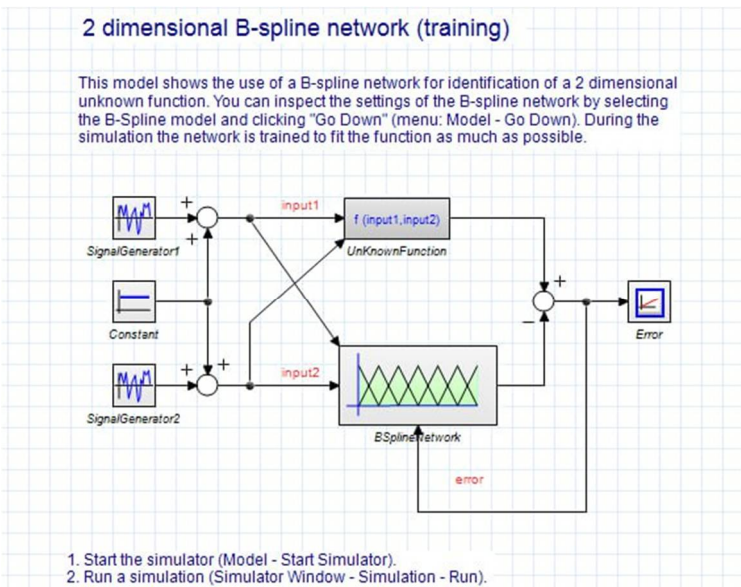


Рисунок 5.10 – Схема тренування двохвимірної нейронної мережі

Потім перевірте роботу натренованої мережі, відкривши файл *2DBSplineNetwork*, який буде вимагати при запуску збережену матрицю ваг.

Дослідить отримані графіки, оцініть якість і швидкість роботи нейронної мережі по відтворенню невідомої функції.

Повторіть експерименти, варіюючи параметр *Learning Rate*, зробіть висновки по його вплив (напряму впливу, характер).

5.6 Контрольні запитання

1 Як в системах захисту інформації використовується штучний інтелект (доповідь, презентація).

2 Дайте приклади, яким чином нейронні мережі можуть працювати в системах захисту інформації (доповідь, презентація).

3 Принципи побудови моделей нейронних мереж (доповідь, презентація).

4 Підготуйте доповідь по використанню глибокого навчання в галузі кібербезпеки.

5 Алгоритм навчання та розгортання нейронних мереж (доповідь, презентація).

6 Класифікація нейронних мереж (доповідь, презентація).

7 Порівняти системи автоматизації рівня 3.0 та рівня 4.0 (на основі AI) (доповідь, презентація).

8 Перспективи штучного інтелекту.

9 Проблеми кібербезпеки в системах на базі AI.

10 Методика синтезу моделей на базі нейронних мереж в CAS/CAE.

6 ЛАБОРАТОРНА РОБОТА №6

«ОПТИМІЗАЦІЯ ТЕХНІЧНИХ ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЇ»

Мета: ознайомитися із принципами, підходами та методиками автоматизованої оптимізації систем на базі математичного моделювання.

6.1 Критерії оптимальності систем

Моделювання систем повинне відповідати на питання: чи є прийняте рішення стосовно структури та функціонування оптимальним.

Базова задача оптимізації ставиться як завдання математичного програмування:

$$\begin{aligned} & \text{extr} F(X), \\ & X \in D_x \\ & D_x = \{X \mid \phi(X) > 0, \psi(X) = 0\}, \end{aligned} \tag{6.1}$$

де $F(X)$ – цільова функція, X – вектор керованих (проектних) параметрів, $\phi(X)$ і $\psi(X)$ – функції обмеження, D_x – допустима область в просторі керованих параметрів.

Складність постановки оптимізаційних проектних завдань обумовлена наявністю у проєктованих об'єктів декількох вихідних параметрів, які можуть бути критеріями оптимальності, але в задачі (6.1) цільова функція повинна бути одна. Іншими словами, проектні завдання є багатокритеріальними, отже, виникає проблема зведення багатокритеріальної задачі до однокритерійної.

Застосовують кілька способів вибору критерію оптимальності.

У *приватному критерії* серед вихідних параметрів один вибирають в якості цільової функції, а умови працездатності інших вихідних параметрів відносять до обмежень задачі (6.1). Ця постановка цілком прийнятна, якщо дійсно можна виділити один найбільш критичний вихідний параметр. Але в більшості випадків такий спосіб має серйозні недоліки.

На рисунку 6.1 представлено двовимірний простір вихідних параметрів y_1 і y_2 , для яких задані умови працездатності $y_1 < T_1$ і $y_2 < T_2$. Крива АВ є кордоном досяжних значень вихідних параметрів.

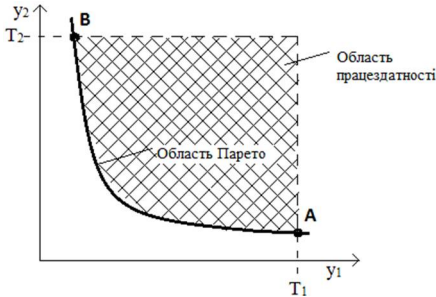


Рисунок 6.1 – Область працездатності

Це обмеження об'єктивне і пов'язано з існуючими фізичними та технологічними умовами виробництва, званими умовами реалізованості. Область, в межах якої виконуються всі умови реалізованості і працездатності, називають *областю працездатності*.

Безліч точок простору вихідних параметрів, з яких неможливо переміщення, що призводить до поліпшення всіх вихідних параметрів, називають областю компромісів (*областю Парето*). Ділянка кривої АВ (рисунок 6.1) відноситься до області Парето.

Якщо в якості цільової функції вибрати параметр y_1 , то результатом оптимізації будуть параметри X , відповідні точці В, яка належить області працездатності, отже, при нестабільності внутрішніх і зовнішніх параметрів велика імовірність виходу за межі області працездатності. Якщо цільовою функцією буде обраний параметр y_2 – оптимізація приведе в точку А.

Адитивний критерій об'єднує (згортає) усі вихідні параметри (приватні критерії) в одну цільову функцію, що представляє собою зважену суму приватних критеріїв:

$$F(X) = \sum_{j=1}^m \omega_j y_j(X) \quad (6.2)$$

де ω_j – ваговий коефіцієнт, m – число вихідних параметрів.

Функція (6.2) підлягає мінімізації, при цьому якщо умова працездатності має вигляд $y_j > T_j$, то $\omega_j < 0$.

Недоліки адитивного критерію – суб'єктивний підхід до вибору вагових коефіцієнтів і неврахування вимог технічного завдання (ТЗ). Дійсно в (6.2) не визначені норми вихідних параметрів.

Аналогічні недоліки властиві й *мультиплікативному критерію*, цільова функція якого має вигляд:

$$F(X) = \prod_{j=1}^m \omega_j y_j(X) \quad (6.3)$$

Неважко бачити, що якщо прологарифмувати (6.3), то мультиплікативний критерій перетворюється на адитивний.

Більш кращим є *максимінний критерій*, в якості цільової функції якого беруть вихідний параметр, найбільш неблагополучний з позицій виконання умов працездатності. Для оцінки ступеня виконання умови працездатності j -го вихідного параметра вводять запас працездатності цього параметра S_j і цей запас можна розглядати як нормований j -й вихідний параметр. Наприклад (передбачається, що всі вихідні параметри приведені до вигляду, при якому умови працездатності стають нерівностями у формі $y_j < T_j$):

$$S_j = (T_j - y_j) / T_j,$$

або

$$S_j = (T_j - y_{номj}) / \delta_j,$$

де $y_{номj}$ – номінальне значення, а δ_j – деяка характеристика розсіяння j -го вихідного параметра, наприклад, трьохсігмовий допуск.

Тоді цільова функція в максимінному критерії є:

$$F(X) = \min_{j \in [1:m]} Z_j(X)$$

Тут запис $[1:m]$ означає безліч цілих чисел в діапазоні від 1 до m . Завдання (6.1) при максимінному критерії конкретизується наступним чином:

$$F(X) = \max_{X \in D_x} \min_{j \in [1:m]} Z_j(X) \quad (6.4)$$

де допустима область D_x визначається тільки прямими обмеженнями на керовані параметри x_i :

$$x_{imin} < x_i < x_{imax}.$$

Для оптимізації систем досить істотне значення має повнота і чіткість опису мети функціонування складної системи, переліку вирішуваних нею завдань. Якщо цілі та завдання системи визначені, постає питання про оцінку якості її функціонування. Якість функціонування складної системи оцінюється за допомогою показників ефективності – числових характеристик системи, які оцінюють ступінь пристосованості системи до виконання поставлених перед нею завдань.

Так як значення показника являють собою дійсні числа, тоді можна говорити про відображення безлічі процесів функціонування системи в безліч дійсних чисел, укладених усередині деякого інтервалу (у межах зміни значень показника ефективності). Таким чином, показник ефективності можна вважати функціоналом, що заданий на множині процесів функціонування системи.

У зв'язку з тим, що складні системи працюють в умовах дії випадкових факторів, значення функціоналів є випадковими величинами. Це створює відомі незручності при використанні їх в якості показників ефективності. Тому при виборі показників ефективності зазвичай користуються середніми значеннями відповідних функціоналів.

Можна визначити функціонали, які характеризують процеси роботи складних систем, що використовуються в якості показників їх ефективності. На цьому шляху можуть бути побудовані (причому різними способами) сукупності функціоналів, що характеризують та інші властивості складних систем: їх надійність, перешкодозахищеність, якість управління і т.д.

Оцінка надійності проводиться за допомогою спеціально обраних функціоналів – показників надійності системи.

Для багатьох складних систем вихід деяких елементів з робочого стану не тільки не призводить до несподіваної втрати працездатності всієї системи в цілому, але іноді навіть є заздалегідь «плановою» подією. До таких випадків відносяться поломки автомобілів в міському транспорті, пориви окремих ліній в телефонній мережі, відмови окремих каналів в багатоканальних системах масового обслуговування і т.д. Тут може йти мова не про повну втрату працездатності системи, а лише про зниження якості її роботи, тобто про зміну ефективності системи.

Тоді в якості показника надійності складної системи може бути обрана абсолютна величина різниці:

$$\Delta\Phi_{\text{надійн}} = \left| \Phi^o - \Phi^*_{\text{надійн}} \right|,$$

що вказує, наскільки знижується ефективність системи внаслідок можливих відмов її елементів в порівнянні з ефективністю ідеальної системи Φ^o , елементи якої абсолютно надійні.

Якщо величина $\Delta\Phi_{\text{надійн}}$ мала, то відмови елементів слабо впливають на ефективність системи. У цьому випадку навряд чи доцільні якісь надзвичайні заходи підвищення надійності - отримані результати можуть не виправдати зроблених витрат. В іншому випадку, коли величину $\Delta\Phi_{\text{надійн}}$ не можна вважати малою, потрібно вжити необхідні заходи: підвищити надійність або резервування елементів, відмови яких мають найбільший вплив на ефективність, провести спеціальну профілактику і т.д.

Для розрахунку показників надійності системи, крім характеристик інтенсивності відмов елементів, необхідно також задати характеристики, що описують витрати часу на відновлення їх працездатності – ремонт або заміну.

Також існують показники, що характеризують й інші властивості складних систем, наприклад, для оцінки *перешкодозахищеності системи*. Нехай як раніше функціонал Φ є показником ефективності складної системи, і нехай його значення $\Phi^*_{\text{перешкод}}$ відповідає функціонуванню системи в умовах дії перешкод із заданими характеристиками, а значення Φ^o відноситься до так званих нормальних умов, коли перешкоди відсутні. Тоді в якості показника перешкодозахищеності системи можна вибрати абсолютну величину різниці:

$$\Delta\Phi_{\text{перешкод}} = \left| \Phi^o - \Phi^*_{\text{перешкод}} \right|,$$

що показує, наскільки змінюється ефективність системи під впливом перешкод із заданими характеристиками.

Деякими особливостями відрізняється оцінка *якості управління* в складній системі. Нехай в даній складній системі управління може бути організовано управління декількома способами (відповідні варіанти позначені буквами A , B , C , ..), а функціонал Φ є показником

ефективності системи, а його значення Φ_A , Φ_B , Φ_C відповідають зазначеним варіантам управління. Тоді абсолютна величина різниці може служити *порівняльною оцінкою* варіантів управління A і B . Для того, щоб виконати *абсолютну оцінку* якості управління (наприклад, для варіанта A), необхідно знати *ідеальний варіант* управління, при якому ефективність системи виявляється найбільшою.

Особливу проблему приділяють показнику стійкості складної системи. Якість функціонування системи доцільно оцінювати за допомогою наборів функціоналів, які є показниками ефективності, надійності, перешкодо-захищеності і т.д., обчислених для заданих умов функціонування. Тільки тоді система володіє необхідними властивостями, коли згадані показники знаходяться в заданих межах. Насправді умови функціонування складної системи не залишаються постійними, вони схильні до збурень різної природи. Істотно знати, чи зберігають при наявності збурень показники якості системи свої значення, а якщо ні, то які можливі відхилення.

Під *стійкістю* функціонування складної системи розуміють *здатність системи зберігати необхідні властивості в умовах дії збурень*. Для того, щоб надати поняттю стійкості більш точний зміст, необхідно конкретно виділити показники, збереження значень яких необхідно досягти, встановити клас допустимих збурень і способи їх вимірювання, а також уточнити суть «збереження необхідних властивостей».

Для системи, *стійкої* щодо функціоналу, що характеризує деяку властивість, *можна вказати такі обмеження, що накладаються на збурення*, при яких даний функціонал *буде зберігати своє значення* в деякому ймовірному сенсі. У разі *нестійкої системи* цього зробити не можна. Більше того, може виявитися, що для обраної властивості системи не *можна підібрати* обмежень на збурення, що забезпечують збереження значень відповідного функціоналу.

6.2 Методи оптимізації

При розробці систем одним з найбільш вживаних є інструментарій часової області, який дозволяє проводити три види аналізу допусків (Монте-Карло, варіаційний (дисперсійний), чутливості) та виконувати параметричну оптимізацію моделі. Параметрична оптимізація може використовуватися для автоматичного підбору параметрів системи, якщо немає можливості

або необхідності використовувати методики синтезу регуляторів: метод масштабування, релейної настройки; методи Зіглера-Нікольса, Коена-Куна, Чіна-Хронеса-Ресвіка (CHR-метод); графо-аналітичний метод Ротача, метод резонансної частоти Широкого, метод домінуючих коренів Соколова, Рейніша, Опельта, метод ВТИ («Всесоюзный технологический институт»), Кеслера (Betrag optimum), Латцеля (Betrag adaptation), Проїса (Betrag optimum за відсутності моделі об'єкта); метод з використанням прогнозуючого нейрокерування з нечіткими регуляторами, як альтернатива звичайного ПІД-регулятора тощо.

На даний момент існує безліч методів або алгоритмів оптимізації (переважно пошуку екстремуму цільової функції ЦФ (objective function)), а також допоміжних технік (використовуваних при оптимізації), що реалізовані на різних мовах програмування (Algol, Fortran, Pascal, Python, Ada, Forth, Lisp, Java, Prolog, C, C++ тощо):

- метод бінарного пошуку або дихотомії, Фібоначчі, золотого перетину, січних, перебору по сітці, метод Мюллера, метод конфігурацій, спряжених напрямків (паралельних дотичних) або метод Пауела (Powell), Нелдера-Міда (Nelder–Mead) або метод амеби чи симплекс-метод швидкісного спуску (деформованого багатогранника), прямого пошуку (Pattern search – пошук за шаблоном) або метод Хука-Дживса (Hooke-Jeeves), евристичний метод Луса-Джаколи (Luus-Jaakola), метод спряжених градієнтів (Conjugate gradient) або метод Флетчера-Рівза (Fletcher-Reeves), субградієнтний метод, квазі-Ньютонівський метод змінної метрики Девідона-Флетчера-Пауела (Davidon-Fletcher-Powell, DFP-метод), метод SR1 (Symmetric Rank 1), метод Гауса-Ньютона (Gauss-Newton), метод Ньютона-Рафсона (Newton-Raphson) з використанням гессіана цільової функції;

- алгоритм імітації відпалу (Simulated annealing) або модельного загартування, альфа-бета відсікання (Alpha-beta pruning) як техніка збільшення ефективності алгоритму «мінімакс» (minimax), алгоритм аукціону (Auction algorithm), метод внутрішніх точок (Interior point methods) з класу бар'єрних методів (Barrier methods) для вирішення лінійних та нелінійних опуклих задач оптимізації, IOSO-технологія (Indirect Optimization on the basis of Self-Organization), ітераційний локальний пошук (Iterated Local Search) як модифікація алгоритму

сходження на вершину (hill climbing) для вирішення задач дискретної оптимізації, метод квантового відпалу (quantum annealing), евристична техніка реактивного пошуку (Reactive search optimization) для локальної оптимізації;

- допоміжні алгоритми QR-розкладання (на базі модифікованого методу Грама-Шмідта, відображень Хаусхолдера чи обертань Гівенса), сингулярного розкладання (singular value decomposition), Кофмана-Грехема (Coffman–Graham), LU-факторизації, спектрального розкладання, LU-скорочення, декомпозиції Холецького, розкладання Джордана, факторизації Такагі, перетворення Карунена-Лоева (Karhunen-Loeve) або метод головних компонент (Principal component analysis) для зменшення розмірності даних. Всі техніки оптимізації можна класифікувати за різними категоріями; пропонується класифікація, що представлена таблицею 6.1.

В практичних задачах оптимізації моделей, якщо передбачається, що цільова функція ($ЦФ$) є гладкою і кількість варійованих параметрів (n) достатньо мала (тобто є можливість зберегти симетричну матрицю $n \times n$), можуть використовуватися наступні класи методів, що подані у вигляді рейтингу (рекомендується обирати алгоритм оптимізації з початку списку):

- методи Ньютона з використанням других похідних;
- методи Ньютона без використання Гессіана;
- квазі-Ньютонівські методи з використанням перших похідних;
- квазі-Ньютонівські методи без використання градієнта (якобіана);
- методи спряжених градієнтів з використанням перших похідних;
- методи спряжених градієнтів без використання перших похідних;
- симплекс-методи (Нелдера-Міда), що використовують тільки значення функції.

Ця ієрархія алгоритмів корисна не тільки при виборі початкового методу рішення, але і вказує на можливі альтернативи, якщо виникають труднощі з першим вибором. Переваги методів Ньютонівського типу відображаються використанням властивостей гессіана функції (тобто матриці других часткових похідних) для поліпшення алгоритмічної ефективності та надання якісної інформації

про обчислений розв'язок; квазі-Ньютонівська апроксимація не гарантує точного представлення істинного Гессіана.

Таблиця 6.1 – Класифікація методів оптимізації

Класифікація	Клас методів	Примітка (підкласи та приклади)
За наявністю обмежень на вектор варійованих параметрів	Безумовні	Без обмежень на вектор параметрів
	Умовні	З обмеженнями типу нерівностей
		З обмеженнями типу рівностей
		З обмеженнями загального вигляду
За розмірністю вектору параметрів	Одновимірні (одно-параметричні)	Варіюється один параметр (методи Фібоначчі, золотого перетину та інші)
	Багато-параметричні	-//- більше одного параметра (метод Нелдера-Міда та інш.)
За характером шуканого рішення	Локальні	Метод Хука-Дживса, Пауела та інш.
	Глобальні	Методи відсікаючої площини, імітації відпалу, гілок і меж
За характером цільової функції (ЦФ)	Стохастичні	ЦФ має випадкові параметри
	Детерміновані	ЦФ апіорно детермінована
За способом досягнення результату	Алгоритмічні	Симплекс-метод та інші
	Ітеративні	Метод Ньютона, внутрішніх точок
	Евристичні	Еволюційні алгоритми
За кількістю врахованих кроків	Однокрокові	Враховується 1 попередній крок
	Багатокрокові	-//- декілька попередніх кроків
За порядком використаних похідних	Нульового порядку (прямі)	Пауела, золотого перетину, Хука-Дживса, Нелдера-Міда

	Першого порядку (градієнтні)	Градiєнтний спуск, BFGS-метод, Левенберга-Марквардта-1, методи DFP та SR1, спряжених градієнтів
	Другого порядку	З використанням гессіана ЦФ: метод Ньютона, Ньютона-Рафсона

Додаткова перевага Ньютонівських методів з'являється при вирішенні специфічних задач (ярова функція, пошук сідлової точки), бо тільки цей клас методів може гарантувати їх вирішення. Для більшості задач надійність і ефективність Ньютонівських методів значно не погіршується, якщо для апроксимації других похідних (Гессіана) використовуються кінцеві різниці (логіка методу взагалі не порушується при їх використанні) або градієнти.

На жаль, ці властивості не зберігаються для квазі-Ньютонівського класу методів, в якому кінцеві різниці значень функції використовуються для апроксимації перших похідних. Навіть при найкращій реалізації квазі-Ньютонівський метод з апроксимацією кінцевими різницями не знайде розв'язок в області, де абсолютне значення градієнта функції невелике, але не нульове. Кінцево-різницеві квазі-Ньютонівські методи відрізняються від таких же методів, з використанням безпосередньо похідних, алгоритмами вибору довжини кроку, а також зміною рекомендованого критерію зупину.

Симплекс-методи (нульового порядку) вимагають найменшу кількість інформації про функцію під час роботи (лише значення самої функції без отримання похідних першого та другого порядків) і є найменш надійними. Цей клас методів слід використовувати, якщо їм немає альтернативи.

Розглянемо декілька методів оптимізації, з яких два перших використовують прямий пошук (нульового порядку), а останні – переважно градієнтний (методи першого або другого порядку):

- метод перпендикулярного пошуку (*Perpendicular Search*) – використовує напрям пошуку, який завжди є перпендикулярним до вісі параметру; одночасно може змінюватися тільки один параметр,

інші утримують свої значення незмінними, а на наступному кроці береться другий параметр (зміна параметрів відбувається по черзі після кожного кроку) і процес продовжується;

- метод «*Line Climber*» – як і метод перпендикулярного пошуку, але спочатку відшукується екстремум цільової функції при зміні одного параметра (другий параметр при цьому залишається незмінним), а потім варіюється лише другий параметр (перший більше не змінюється);

- метод найшвидшого спуску (*Steepest Descent*) – починає пошук у напрямку найкрутішого нахилу цільової функції; цей напрям зберігається для кожного нового кроку до знаходження мінімуму, а потім визначається новий напрям пошуку і процес продовжується;

- метод безперервного спуску (*Continuous Descent*) – як і попередній, але на кожному новому кроці визначається новий напрям пошуку;

- метод Ньютона-Рафсона (*Newton-Raphson*) – використовує не лише градієнт функції, але і похідну другого порядку (створюється гессіан) для визначення напрямку пошуку, який зберігається для кожного нового кроку поки не буде знайдено мінімум; потім визначається новий напрям пошуку і процес продовжується. Цей метод сходиться тільки коли матриця Гессе має позитивні значення, тобто коли початкові умови задані у відносній близькості до мінімуму;

- метод Полака-Райбера (*Polack-Ribiere*) – метод спряжених градієнтів, використовує градієнт функції (та апроксимацію гессіана) для пошуку розв'язку;

- метод Девідона-Флетчера-Пауела (*Davidon-Fletcher-Powell*) – відноситься до класу квазі-Ньютонівських методів змінної метрики, використовує градієнт та апроксимацію гессіана функції (зазвичай використовується другорангове оновлення якобіана для створення апроксимованої матриці Гессе);

- BFGS-метод (*Broyden-Fletcher-Goldfarb-Shanno*) з сімейства Бройденівських методів (квазі-Ньютонівського класу) – узагальнений метод січних, що використовує, окрім похідних першого порядку, подвійне першорангове оновлення градієнта (на кожній ітерації) для створення апроксимованого Гессіана, замість його прямого обчислення (як це робиться в оригінальних Ньютонівських методах).

Саме цей метод найбільш часто використовується для безумовної нелінійної оптимізації.

Переваги та недоліки деяких методів оптимізації зведені у таблицю 6.2.

Таблиця 6.2 – Порівняльна характеристика методів оптимізації

Метод	Переваги	Недоліки
Хука-Дживса	Не потребують вираження ЦФ в аналітичному виді. Прості обчислювальні процедури та реалізація	Не завжди можуть просуватися до мінімуму. Більші витрати часу ніж у градієнтних
Нелдера-Міда		
Резенброка		
Градієнтний	Краща збіжність ніж у попередніх	Втрата точності при обчисленні похідних
Полака-Райбера	Ефективні при застосуванні до ярових ЦФ	Чутливий до помилок, що виникають у процесі обчислень
DFP, BFGS	Найбільша швидкість збіжності при мінімізації ярових функцій	Збіжність залежить від початкового наближення. Накопичення похибки при прямому обчисленні похідних
Левендерга-Марквардта		
Ньютона		
ЛП-пошук	Швидша збіжність і вища точність, ніж у інших евристичних методів	У деяких випадках потребують уточнення результату
Холтона		

Рішення про вибір методу оптимізації треба приймати з урахуванням вигляду цільової функції (гладкість, опуклість, монотонність, комплексність, наявність і кількість локальних мінімумів та їх розташування) та її розмірності (кількості параметрів, які варіюються), доступного часу на відпрацювання ітераційного алгоритму від початку до знаходження рішення (кожен крок супроводжується актом симуляції для знаходження значень функції

тощо), доступної пам'яті для зберігання проміжних результатів (особливо для багатокрокових методів вище нульового порядку).

Методи першого порядку потребують більше кроків (ніж методи нульового порядку) для визначення градієнту і формування якобіана; метод Ньютона-Рафсона окрім цього ще повинен зробити додаткові розрахунки для знаходження похідних другого порядку і сформувати істинний (неапроксимований) Гессіан функції, тому цей метод потребує найбільшу кількість кроків та обсяг пам'яті з усіх інших. Прямі методи (нульового порядку), коли напрям нахилу функції не точно такий, як напрямок пошуку, можуть почати «хитатися», тобто постійно змінювати напрям, роблячи невеликі кроки, отже потенційно нестійкі.

Таким чином, в задачах оптимізації параметрів моделі САУ (в тому числі для автоматичної настройки коефіцієнтів регуляторів) рекомендується використовувати прямі методи для одновимірної оптимізації (з одним варійованим параметром) і *DFP*- та *BFGS*-методи для багатопараметричної оптимізації.

Класифікація задач оптимізації – наведена в Додатку І.

6.3 Параметрична оптимізація в Micro-Cap

Параметрична оптимізація виконується в програмі Micro-Cap методами Пауелла (*Powell*) в будь-якому з видів аналізу: аналіз перехідних процесів, малосигнальний АС-аналіз і розрахунок DC характеристик та виконує оптимізацію параметрів компонентів схеми і її моделей по певних критеріях, які визначають функції схеми (швидкодію, наприклад). При цьому можлива оптимізація по максимальному (*Maximizes*), мініимальному (*Minimizes*) або заданому (*Equates*) значенню вибраного критерію.

Метод змінної метрики (*Девідона-Флетчера-Пауелла*) можна розглядати як результат вдосконалення методу другого порядку (Ньютона), який базується на використанні необхідних умов існування безумовного екстремуму цільової функції: $gradF(x)=0$ та розв'язує означене рівняння ітераційним методом.

Метод Пауелла (квадратичної інтерполяції), ідея якого полягає в побудові інтерполяційного багаточлена 2-й ступеня по трьом точкам, що беруться поблизу мінімуму. Положення мінімуму багаточлена

приймається в якості нового наближення до шуканої точки мінімуму функції.

Метод Пауелла в достатній близькості від мінімуму має квадратичну збіжність, і це є перевагою порівняно із методами, що мають лінійну збіжність (наприклад, із методом золотого січення). Однак, якщо початковий інтервал великий (початкова область збіжності, з якої «стартує» алгоритм), апроксимація функції за допомогою багаточлена другого ступеню може стати надмірно «грубою», що призведе до уповільнення збіжності.

У діалоговому вікні налагодження параметрів процедури оптимізації (визивається натисканням CTRL+F11) і вибору типу цільової функції та її параметрів вводиться наступна інформація:

Знайти (*Find*) — вибір параметрів, що оптимізуються;

Нижній (*Low*) — мінімальне значення параметра, що оптимізується;

Крок (*Step*) — крок зміни параметра, що оптимізується;

Поточний (*Current*) — поточне значення параметра, що оптимізується;

Оптимізувати (*Optimized*) — пошук оптимального значення параметру;

Метод (*Method*) — вибір методу оптимізації:

(*Standard Powell*) — стандартний метод оптимізації Пауелла,

Покроковий *Powell (Stepping Powell)* — параметри змінюються від значення *Low* до значення *High* з кроком *Step*.

Помилка (*Error*) — корінь квадратний з сумарної помилки (різниці між цільовою функцією та її фактичною величиною).

Обмеження (*Constraints*) — обмеження типу нерівностей і рівнянь.

Наприклад: $PD(R1) \leq 100m$, $V(Out) \geq 1.2$,
 $VCE(Q1) * IC(Q1) \leq 200m$.

Оптимізувати (*Optimize*) — початок оптимізації.

Застосувати (*Apply*) — зміна на схемі значення параметрів відповідно до результатів оптимізації.

Формат (*Format*) — вибір форми представлення чисел.

Закрити (*Close*) — завершення режиму оптимізації.

Наведемо скорочено **методику параметричної оптимізації**:

а) Для дослідження пристрою спочатку необхідно створити креслення схеми в редакторі схем Circuit Editor.

б) Вибирається вид аналізу, наприклад, Аналіз – Перехідні процеси... (Alt+1), тобто досліджується початкова схемна надійність проєктованого пристрою в одноваріантному аналізі.

в) Попередній пошук напряму для параметричної оптимізації здійснюється багатоваріантними методами, наприклад, Stepping, окремим розрахунком чутливості для визначення складу варійованих/впливаючих параметрів в цільовій функції.

г) Крім безпосередньої оптимізації в МС є можливість проаналізувати схему при випадковому розкиді параметрів елементів. Для цього необхідно виконати розрахунок *Monte Carlo*, який стає доступним після виконання одноваріантного аналізу, для розрахунку характеристик схем при випадковому розкиді параметрів.

д) Наступний етап – оптимізація схеми. Оптимізація стає доступною тільки після виконання аналізу (наприклад, *Перехідні процеси*).

Після виконання аналізу, запускаємо вікно настройки оптимізації.

Якщо знайдені значення нас влаштовують, можна натискати на кнопку «Застосувати», і знайдені значення будуть перенесені на схему, після чого параметри корегуються у реалізовані (обираються такі, що пропонуються виробниками). Рекомендується запустити повторний аналіз, оскільки, якщо цільова функція мала «гострий» екстремум, нові значення параметрів можуть призвести до виходу аналізованої функції з зони екстремуму. В такому випадку можна використовувати покроковий метод Пауелла, який дозволяє «побачити» всі локальні екстремуми функції.

6.4 Етапи виконання роботи

1. Відкрити папку Optimizer з каталогу Data програми MC12 і відкрити файли opt1.cir, opt2.cir, opt3.cir та opt4.cir.

2. В файлі opt1.cir провести оптимізацію за протоколом, вказаним на робочому полі редактора схем

Optimizer Sample Circuit for Transient DC operating point

Find the value of R2 that makes $V(OUT) = 0.5$

Run Transient analysis (Alt+I).

Select the Optimizer (CTRL + F11).

Click on the Optimize button (F2)

Занести схему, її опис і результати моделювання в звіт.
Описати вікно завдання параметрів оптимізації.

3. Зберегти цей файл під іншим іменем і провести оптимізацію за критерієм максимізації дисипативної потужності на резисторі R2 (див. рис.6.2 та рис.6.3).

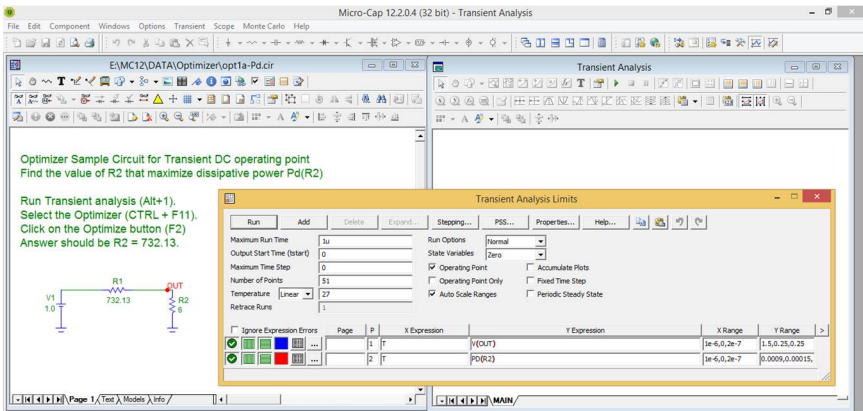


Рисунок 6.2 – Схема і завдання на аналіз при максимізації дисипативної потужності на резисторі R2

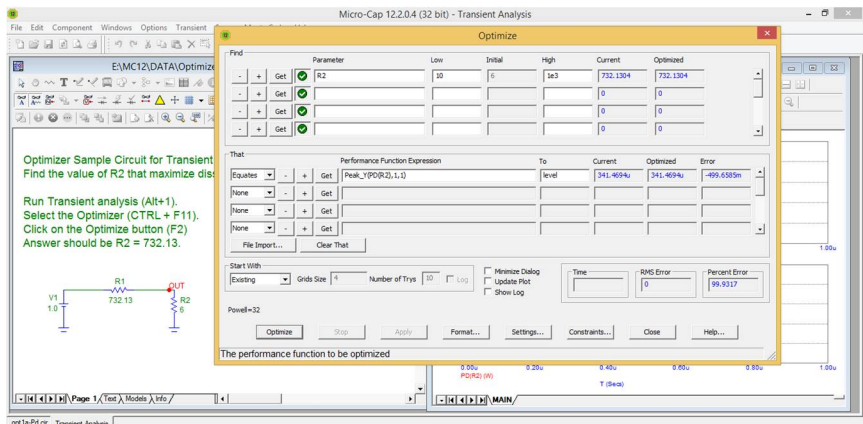


Рисунок 6.3 – Схема і завдання на оптимізацію при максимізації дисипативної потужності на резисторі R2

Порівняти результати з п.2.

- В файлі opt2.cir провести спочатку одноваріантний АС-аналіз, потім, не виходячи з нього, – багатоваріантний аналіз *Stepping*, змінюючи опір в моделі резисторів, як показано на рис.6.4.

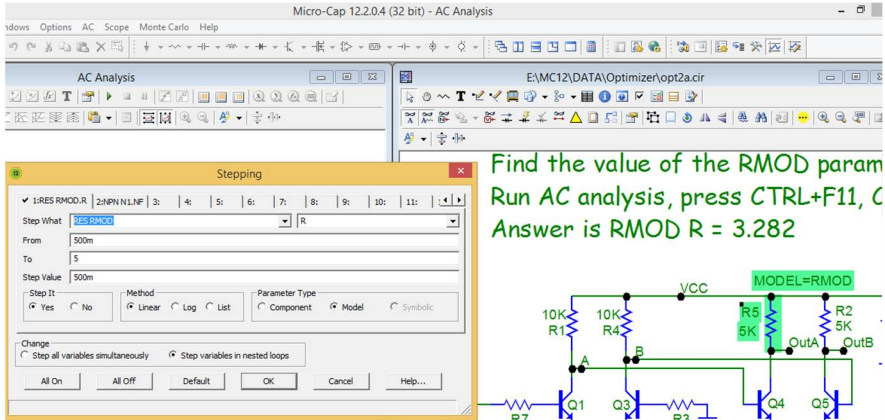


Рисунок 6.4 – Схема і завдання на багатоваріантний аналіз

Занести в звіт схему і результати аналізу.

Потім провести Оптимізацію за наведеною в файлі інструкцією.

Занести в звіт результати.

5. Ознайомитися із можливостями оптимізації, розкритих в файлах opt3.cir та opt4.cir, описати словами, яка задача оптимізації була поставлена і що є результатом оптимізації в даних файлах. Згідно таблиці визначити, який вид оптимізації був проведений в кожному файлі.

6.5 Контрольні запитання

- Методика параметричної оптимізації в ECAD.
- Алгоритми параметричної оптимізації в САЕ.
- Класифікація методів оптимізації.
- Рекомендації по вибору методу оптимізації.
- Методика вибору методу оптимізації.
- Алгоритми пошуку екстремумів.
- Евристична оптимізація.
- Критерії якості систем, протиріччя між ними.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. **Василенко, О.В.** Навчальний посібник з дисципліни «Комп'ютерне моделювання» [Текст]/ О.В. Василенко – Запоріжжя: НУ «Запорізька політехніка, 2020. –175 с.
2. Micro-Cap 12, Electronic Circuit Analysis Program. Reference Manual, Spectrum Software [Electronic resource]. – Access mode: <http://www.spectrum-soft.com/down/rml1.pdf/document> (in English)
3. Simulation Tools [Електронний ресурс]. – Режим доступу: http://www.idsia.ch/~andrea/Andrea_Rizzoli_Home_Page/Sim_Tools.html
4. MathWorks Documentation Center, Solver Pane [Електронний ресурс]. – Режим доступу: <http://www.mathworks.com/help/ simulink/gui/ solver-pane.html>
5. Simulation Tools [Електронний ресурс]. – Режим доступу: http://www.idsia.ch/~andrea/Andrea_Rizzoli_Home_Page/Sim_Tools.html
6. SimApp, Ordering Information [Електронний ресурс]. – Режим доступу: <http://www.simapp.com/order-simapp.php>
7. Differential Algebraic System Solver [Електронний ресурс]. – Режим доступу: <http://acronyms.thefreedictionary.com/DASSL>
8. MapleSim – Key Features [Електронний ресурс]. – Режим доступу: <http://www.adeptsience.co.uk/products/mathsim/maplesim/features.html>
9. Simscape Overview [Електронний ресурс]. – Режим доступу: <http://www.mathworks.com/products/simscape/index.html>
10. MathWorks Documentation Center, Solver Pane [Електронний ресурс]. – Режим доступу: <http://www.mathworks.com/help/simulink/gui/solver-pane.html>
11. VisSim Product Overview [Електронний ресурс]. – Режим доступу: <http://www.vissim.com/products/vissim.html>
12. Dymola Dynamic Modeling Laboratory User's Manual, Version 5.3a [Електронний ресурс]. – Режим доступу: <http://www.inf.ethz.ch/personal/cellier/Lect/MMPS/Refs/Dymola5Manual.pdf>
13. Versions of 20-sim [Електронний ресурс]. – Режим доступу: <http://www.20sim.com/home/versions.html>
14. 20-sim 4.3 Reference Manual / Ir.C. Kleijn, Ir.M.A. Groothuis, Ir.H.G. Differ // Controllab Products B.V., 2012.– 1099 с.

Додаток А

Перелік питань до підсумкового контролю і самостійних робіт

Питання першого рівня складності

- 1 Поняття та приклади динамічних систем.
- 2 Форми представлення моделей (приклади).
- 3 Принципи вибору програми для дослідження систем.
- 4 Характеристики моделі.
- 5 Критерії вибору програми моделювання.
- 6 Показники якості моделювання (симуляції) та моделей.
- 7 Каузальне моделювання.
- 8 Структура ЕСAD.
- 9 Чим програми моделювання відрізняються від програм проектування?
- 10 Особливості аналізу перехідних процесів.
- 11 Функції постпроцесорів в ЕСAD.
- 12 Огляд САЕ систем.
- 13 Огляд CAS систем.
- 14 Огляд CAD систем.
- 15 Структура САЕ системи (приклад).
- 16 Чим відрізняються програми моделювання від програм проектування.
- 17 В яких системах є доступні демо- та студентські версії?
- 18 Моделі каналів зв'язку.
- 19 Класифікація САУ.
- 20 Принципи керування та базові структури автоматизованих систем.
- 21 Принципи дослідження САУ.
- 22 Проектування регуляторів, методика.
- 23 Методика аналізу САУ.
- 24 Редактор лінійних систем в 20-sim.
- 25 Методика підвищення адекватності моделювання САУ.
- 26 Алгоритми та програми імітаційного моделювання.
- 27 Системи масового обслуговування (СМО), приклади.
- 28 Основні елементи моделей СМО.

29 Методика і задачі моделювання комп'ютерних мереж в середовищі програми NetCracker.

30 Методика і задачі моделювання комп'ютерних мереж в середовищі програми Cisco Packet Trace.

31 Типи моделей транзакцій (функції розподілу).

32 Принципи побудови моделей нейронних мереж.

33 Алгоритм навчання та розгортання нейронних мереж.

34 Класифікація нейронних мереж (доповідь, презентація).

35 Методика синтезу моделей на базі нейронних мереж в CAS/CAE.

36 Методика параметричної оптимізації в ECAD.

37 Алгоритми параметричної оптимізації в CAE.

38 Класифікація методів оптимізації.

39 Методика вибору методу оптимізації.

40 Евристична оптимізація.

Питання другого рівня складності

1 Способи моделінгу динамічних систем.

2 Мультидоменне моделювання.

3 Критерії порівняння CAE систем.

4 В яких системах використовують акаузальний підхід?

5 Від чого залежить ефективність системи моделювання.

6 Макроси, які використовуються для моделінгу САУ, визначити критерії, за якими вони можуть бути класифіковані.

7 Опишіть обрану систему захисту інформації як автоматизовану систему.

8 Критерії оптимальності систем.

9 Критерії стійкості та методи дослідження стійкості систем.

10 Закони математичної статистики для моделювання інформаційних систем.

11 Як в системах захисту інформації використовується штучний інтелект

12 Принципи імітаційного моделювання (виділити особливості, порівняно із моделюванням на схемотехнічному рівні).

13 Використання глибокого навчання в галузі кібербезпеки.

14 Порівняти системи автоматизації рівня 3.0 та рівня 4.0 (на основі AI).

15 Дайте приклади, яким чином нейронні мережі можуть працювати в системах захисту інформації (доповідь, презентація).

16 Перспективи штучного інтелекту.

17 Проблеми кібербезпеки в системах на базі AI.

18 Рекомендації по вибору методу оптимізації.

19 Критерії якості систем, протиріччя між ними.

Перелік питань, які пропонуються для самостійної роботи та ДКР для заочного відділення

1 Моделювання систем захисту інформації в універсальних математичних процесорах (CAS).

2 Особливості моделювання систем захисту інформації.

3 Моделі систем: аналіз підходів.

4 Моделювання на системному рівні, програми.

5 Мультидоменне моделювання систем.

6 Програми наскрізного проектування, віртуальне виробництво.

7 Програми автоматизованого інжинірингу (CAE).

8 Моделювання та проектування програмованих IC.

9 Проблеми алгоритмічних збоїв при моделюванні систем.

10 Методика розробки нових моделей/макромоделей.

Додаток Б

Математичні процесори для моделювання

CAE-системи:

- MATLAB\Simulink
- MATLAB\Simscape
- Maple\MapleSim
- EASY5
- AMESim
- Dymola
- PSIM
- SamSim
- SimApp
- SimulationX
- Simplorer
- VisSim
- 20-sim
- ПІК MBTU
- SystemModeler

CAS-системи:

- Maple
- MATLAB
- Mathcad
- Mathematica
- Maxima
- SymPy
- O-Matrix
- Octave
- NumPy
- Python(x,y)
- MuPAD

CAE = Computer Aided Engineering

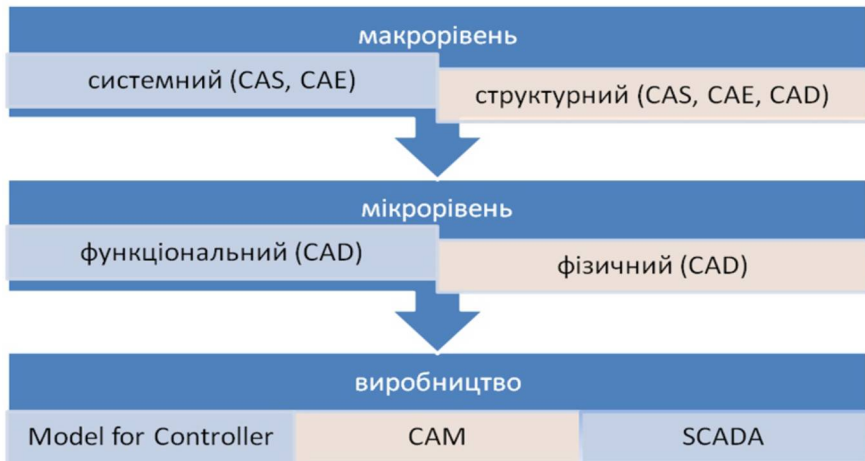
CAS = Computer Algebra System

CAD = Computer Aided Design

CAD-системи:

- MicroCap
- Spice
- Multisim
- Proteus
- Saber
- OrCAD
- PCAD
- TCAD/SYNOPSYS
- CADENCE

НАСКРІЗНЕ ПРОЕКТУВАННЯ/ВІРТУАЛЬНЕ ВИРОБНИЦТВО

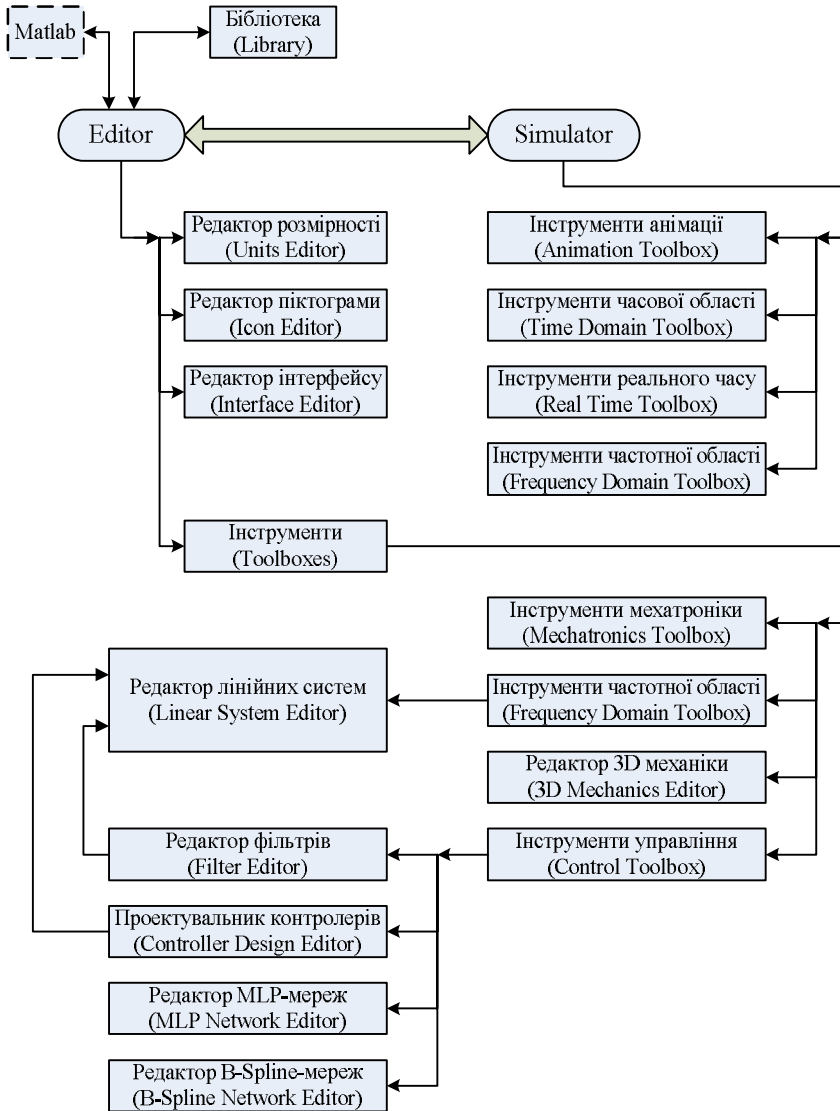


CAM = Computer Aided Manufacturing

SCADA = Supervisory Control and Data Acquisition

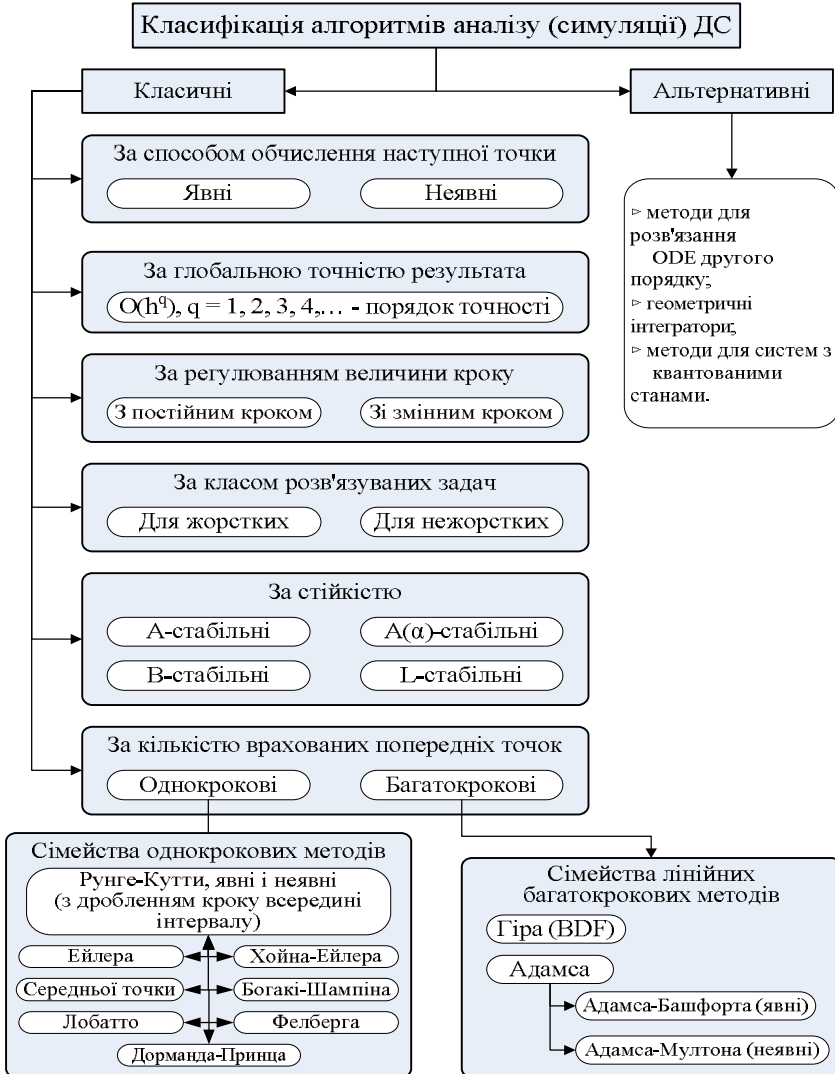
Додаток В

Загальна структура 20-sim



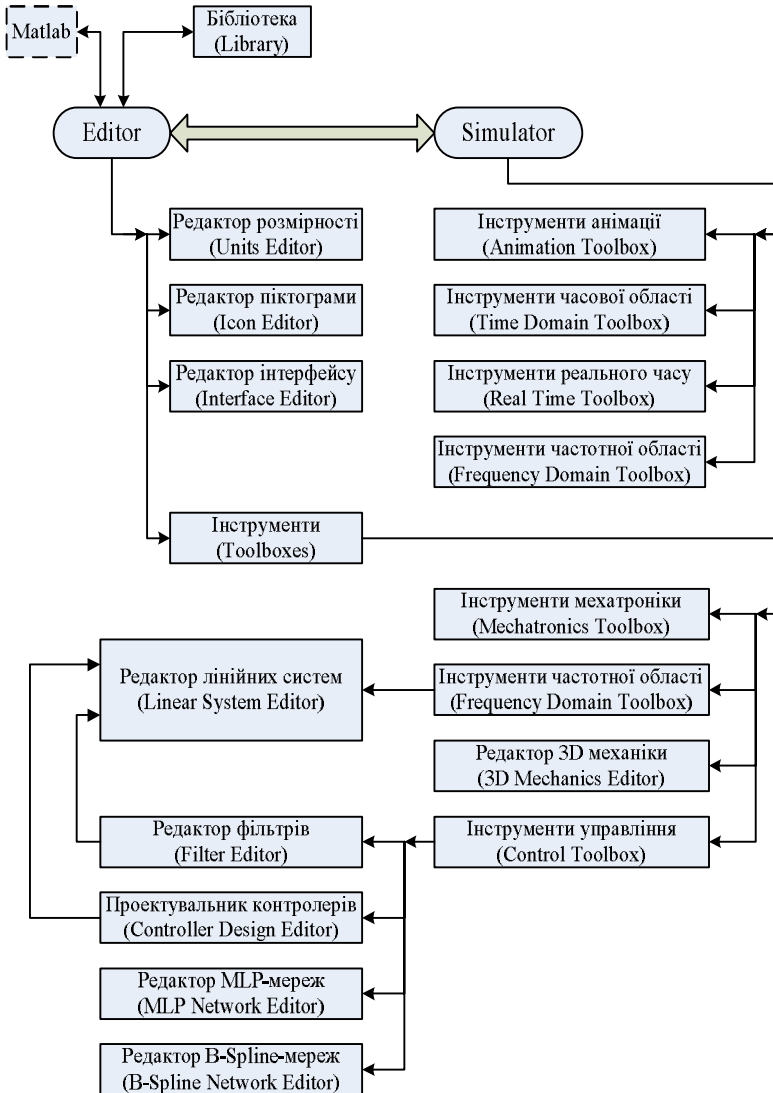
Додаток Г

Класифікація методів симуляції



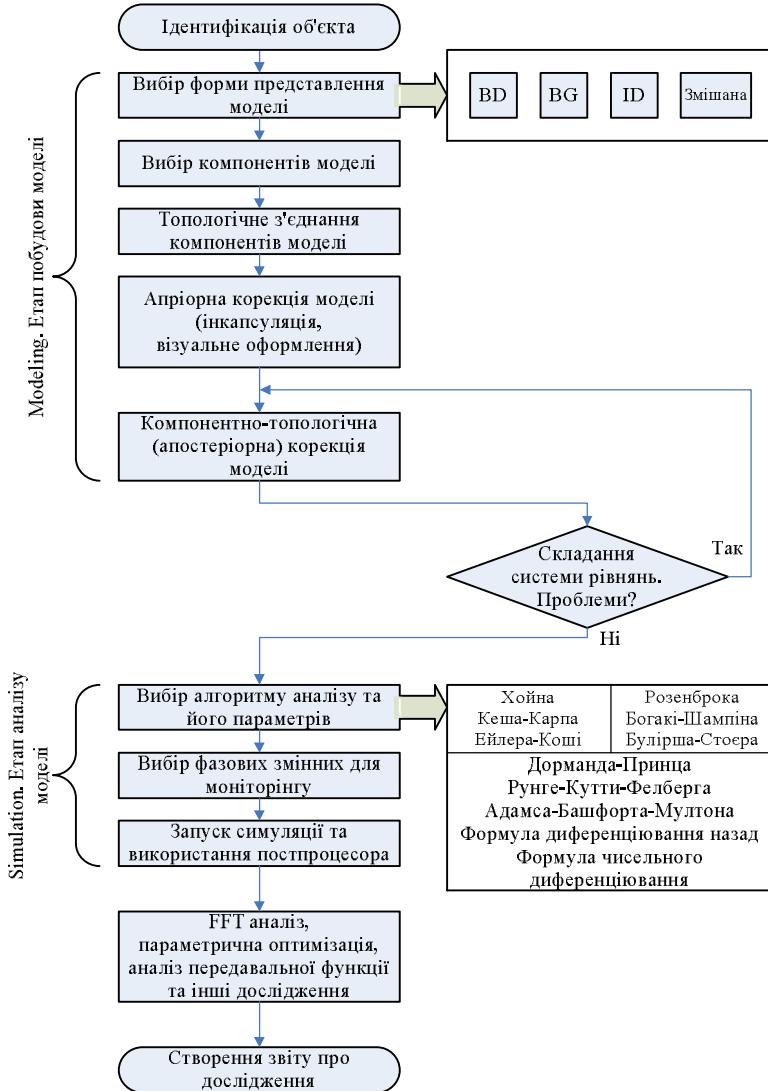
Додаток Д

Структура САЕ



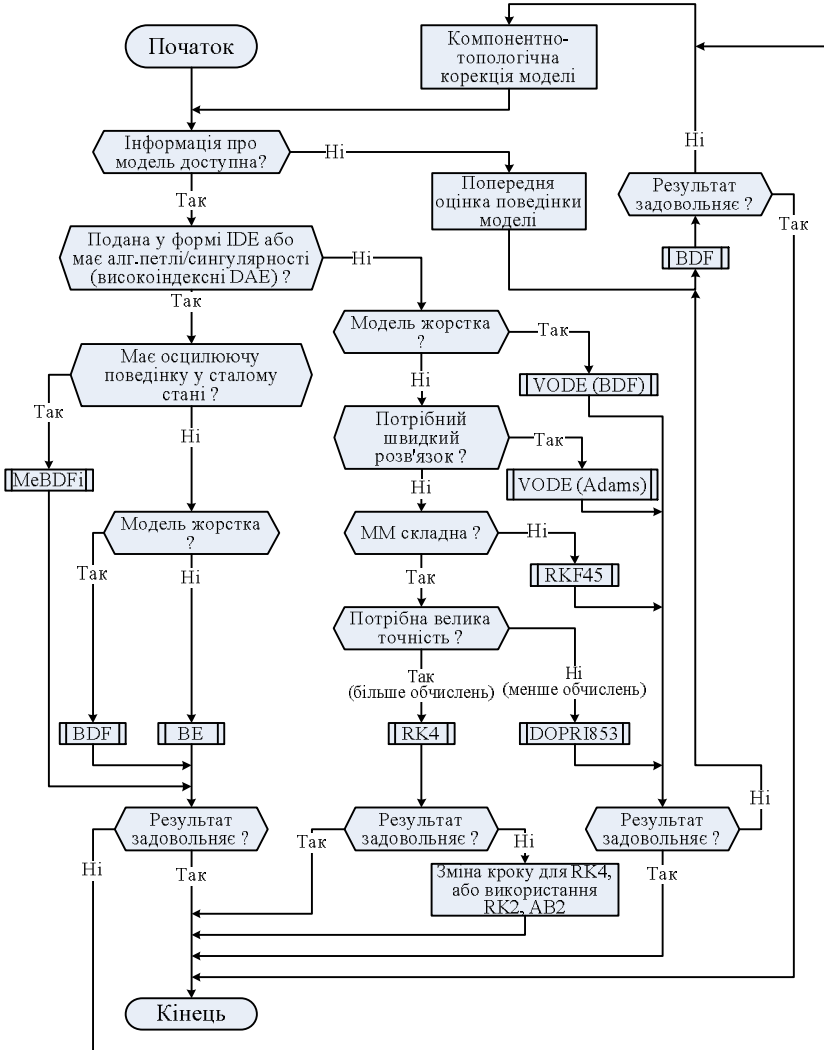
Додаток Е

Згорнутий алгоритм моделювання САК



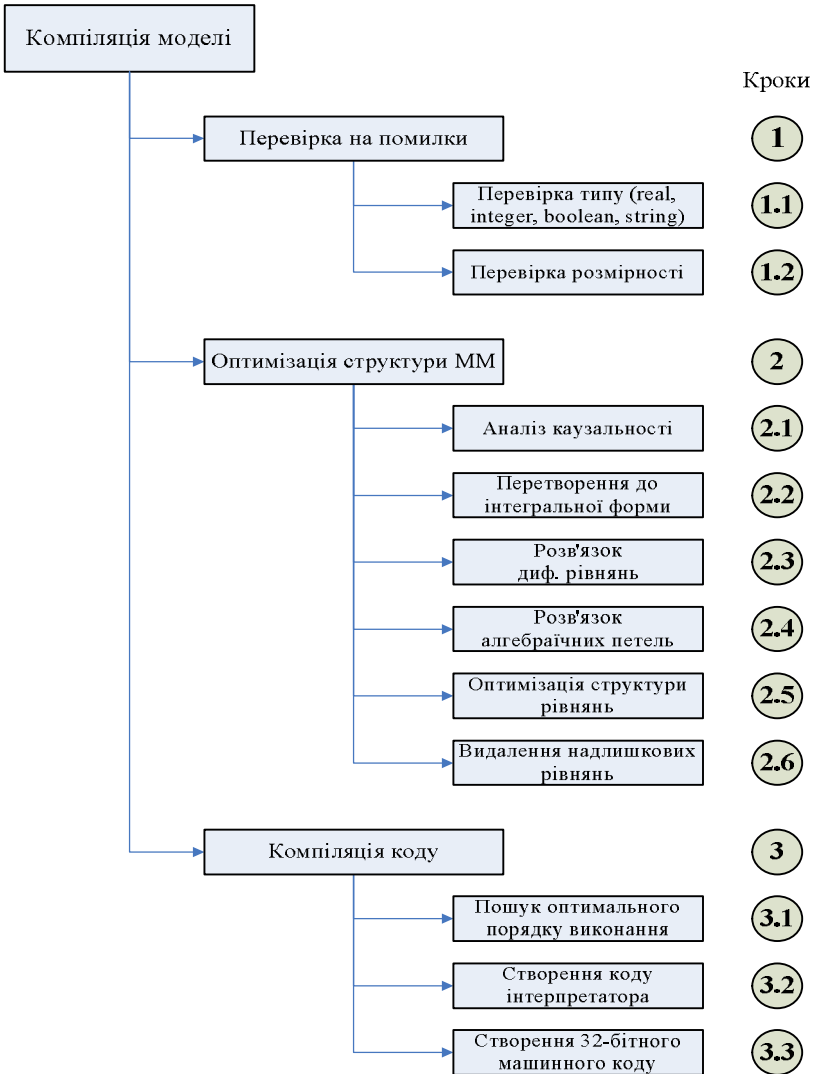
Додаток Ж

Алгоритм вибору методу інтегрування в 20-sim



Додаток И

Процедури етапу компіляції моделі в САЕ



Додаток І
Класифікація задач оптимізації

