

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

КОНСПЕКТ ЛЕКЦІЙ

з дисципліни

«КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ»

для студентів спеціальностей

G6 – «Інформаційно-вимірювальні технології»,

освітня програми: «Інформаційні системи моніторингу і контролю»;

G5 – «Електроніка, електронні комунікації, приладобудування та
радіотехніка»

освітня програма «Мікро- та наноелектронні прилади і пристрої»

Конспект лекцій з дисципліни «Комп'ютерне моделювання» для студентів спеціальності G6 – «Інформаційно-вимірювальні технології», освітня програми: «Інформаційні системи моніторингу і контролю»; та спеціальності G5 – «Електроніка, електронні комунікації, приладобудування та радіотехніка» освітня програма «Мікро- та наноелектронні прилади і пристрої» першого (бакалаврського) рівня вищої освіти / Укл.: Ольга ВАСИЛЕНКО. – Запоріжжя: НУ «Запорізька політехніка», 2026. – 167 с.

Укладач: Ольга ВАСИЛЕНКО, доц., канд. техн. наук

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

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

Затверджено
на засіданні кафедри ІБтаН
Протокол № 4
від 04.02.26 р.

Рекомендовано до видання
НМК ФІБЕК
Протокол № 6
від 04 березня 2025 р.

ЗМІСТ

| | |
|--|----|
| Перелік скорочень та термінів..... | 6 |
| Вступ..... | 8 |
| Тема №1. Підходи до моделювання систем..... | 10 |
| 1.1 Характеристики математичної моделі..... | 14 |
| 1.2 Вибір класу САПР..... | 15 |
| 1.3 Підходи до моделювання систем..... | 16 |
| 1.4 Критерії вибору та огляд САЕ-систем..... | 19 |
| Контрольні запитання..... | 20 |
| Завдання для практичного заняття/самостійної роботи..... | 20 |
| Тема №2. Огляд програм математичного моделювання..... | 21 |
| 2.1 ANSYS Simplorer (VHDL)..... | 22 |
| 2.2 SimApp..... | 24 |
| 2.3 WolframSystemModeler (MathModelica)..... | 25 |
| 2.4 MapleSim..... | 27 |
| 2.5 MATLAB / Simulink та Simscape..... | 30 |
| 2.6 VisSim/Altair Embed..... | 34 |
| 2.7 Dymola (Modelica)..... | 38 |
| 2.8 20-sim..... | 39 |
| Контрольні запитання..... | 43 |
| Завдання для практичного заняття/самостійної роботи..... | 43 |
| Тема №3. Моделювання мехатронних систем..... | 44 |
| 3.1 Особливості мехатронних систем..... | 48 |
| 3.2 Склад мехатронної системи..... | 49 |
| 3.3 Необхідність моделювання мехатронних систем..... | 54 |
| Контрольні запитання..... | 55 |
| Завдання для практичного заняття/самостійної роботи..... | 55 |
| Тема №4. Моделювання САР та САК..... | 56 |
| 4.1 Структурні елементи САК..... | 56 |
| 4.2 Моделі для елементів САК в програмі Micro-CAP..... | 58 |
| 4.3 Приклад побудови САК в програмі Micro-CAP..... | 60 |
| 4.3.1 Теоретичні аспекти моделювання САК..... | 60 |
| 4.3.2 Моделювання регулюючих клапанів..... | 62 |
| 4.4 Моделі ІППН в Micro-Cap..... | 64 |
| 4.4.1 Моделі ШІМ-контролерів для дослідження САК в Spice-сумісних програмах ECAD..... | 66 |

| | |
|--|-----|
| 4.5 Проєктувальник регуляторів та редактор лінійних систем в 20-sim..... | 70 |
| 4.6 Методика моделювання САК..... | 76 |
| Контрольні запитання..... | 77 |
| Завдання для практичного заняття/самостійної роботи..... | 77 |
| Тема №5. Основи симуляції систем..... | 78 |
| 5.1 Перетворення ММ, заданої в графічному інтерфейсі..... | 78 |
| 5.2 Інструментальні засоби моделювання..... | 80 |
| 5.3 Основні поняття алгоритмів моделювання (simulation)..... | 81 |
| 5.4 Класифікація та аналіз існуючих методів інтегрування..... | 85 |
| 5.5 Процедура каузалізації..... | 88 |
| Контрольні запитання..... | 90 |
| Завдання для практичного заняття/самостійної роботи..... | 90 |
| Тема №6. Імітаційне моделювання. | |
| Особливості моделювання комп'ютерних та нейронних мереж..... | 91 |
| 6.1 Критерії оптимальності систем..... | 91 |
| 6.2 Імітаційне моделювання систем..... | 96 |
| 6.3 Способи моделінгу систем..... | 98 |
| 6.4 Скорочені відомості по NetCracker 3.2..... | 101 |
| 6.4.1 Приклад імітаційного моделювання мережі..... | 102 |
| 6.5 Проєктування нейронних мереж..... | 108 |
| Контрольні запитання..... | 112 |
| Завдання для практичного заняття/самостійної роботи..... | 112 |
| Тема № 7. Математичне забезпечення ECAD. | |
| Параметричний синтез та оптимізація..... | 113 |
| 7.1 Математичне забезпечення програм ECAD..... | 113 |
| 7.2 Автоматизований синтез фільтрів..... | 114 |
| 7.3 Синтез параметрів моделей напівпровідникових приладів..... | 118 |
| 7.4 Методи оптимізації..... | 120 |
| 7.5 Параметрична оптимізація в Micro-Cap..... | 126 |
| Контрольні запитання..... | 129 |
| Завдання для практичного заняття/самостійної роботи..... | 129 |
| Тема №8. Методики підвищення ефективності моделінгу та симуляції..... | 130 |
| 8.1 Критерії якості моделювання..... | 130 |
| 8.2 Критерії якості моделей..... | 131 |
| 8.3 Методика усунення алгоритмічних збоїв..... | 133 |

| | |
|--|-----|
| 8.3.1 Рекомендації щодо уникання проблем з алгебраїчними петлями..... | 135 |
| 8.3.2 Способи розв'язання високоіндексних топологічних вироджень ММ..... | 136 |
| 8.3.3 Проблема жорсткості САК та способи її вирішення..... | 139 |
| 8.4 Використання ієрархічного підходу в моделюванні..... | 140 |
| Контрольні запитання..... | 142 |
| Питання до заліку..... | 143 |
| Зміст та об'єм контрольних робіт..... | 145 |
| Перелік джерел посилань..... | 146 |
| Додаток А Питання на вхідний контроль..... | 151 |
| Додаток Б Структура САЕ..... | 152 |
| Додаток В Порівняння можливостей САЕ..... | 153 |
| Додаток Г Умовні зображення блоків для моделювання САК в Micro-Cap та MatLab Simulink..... | 155 |
| Додаток Д Класифікація моделей імпульсних регуляторів в Micro-Cap..... | 157 |
| Додаток Е Згорнутий алгоритм моделювання САК..... | 159 |
| Додаток Ж Алгоритм вибору методу інтегрування в 20-sim..... | 160 |
| Додаток З Процедури етапу компіляції моделі в САЕ..... | 161 |
| Додаток І Класифікація методів симуляції..... | 162 |
| Додаток К Класифікація задач оптимізації..... | 163 |
| Додаток Л Алгоритм вибору методу оптимізації в 20-sim..... | 164 |
| Додаток М Параметри, що оптимізуються в Micro-Cap..... | 165 |

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

- AE* – algebraic equation, алгебраїчне рівняння
- AMS* – Analog and Mixed Signal, аналогові та аналогово-цифрові сигнали
- BD* – block diagram, блок-схема
- BDF*– Backward Differentiation Formula, метод диференціювання назад
- BG* – bond graph, граф зв'язку
- BVP*– boundary value problem, крайова задача
- CAD*– Computer Aided Design, автоматизоване проєктування
- CAE*– Computer Aided Engeneering, автоматизований інжиніринг
- CAM*– Computer Aided Manufacturing, автоматизоване виробництво
- CAS*– Computer Algebra System, універсальний мат. процесор
- CxDE*– Complex Differential Equation, комплексне диф. рівняння
- DAE*– differential-algebraic equation, диф.-алгебраїчне рівняння
- DDE*– delay differential equations, диф.рівняння з затримками
- ECAD*– Electronic CAD, автоматизоване проєктування в електроніці, спеціалізований математичний процесор
- EDA*– Electronic Design Automation, автоматизоване проєктування в електроніці
- FDT*– Frequency Domain Toolbox, інструментарій частотної області
- FF*– feed forward, упереджувач
- FFT*– Fast Fourier Transform, швидке перетворення Фур'є
- ID*– iconic diagram, традиційна (функціональна) схема
- MCAD*– Mechanical CAD, CAD для механіки
- ODAE*– overdetermined DAE, перевизначені DAE
- ODE*– ordinary differential equation, звичайне диференціальне рівняння
- PDE*– partial differential equations, рівняння у приватних похідних
- PWM*– pulse-width modulation, широтно-імпульсна модуляція
- SFG*– signal flow graph, сигнальний граф
- SM* – state machine, автомат станів
- SML*– Simplorer Modeling Language, мова моделінгу Simplorer
- TF*– transfer function, передавальна функція
- UML*– Unified Modeling Language, уніфікована мова моделінгу
- VHDL*– very high-speed integrated circuit design language, мова програмування логічних інтегральних схем

АС – агрегативні системи
АЧХ – амплітудо-частотна характеристика
АФЧХ – амплітудо-фазо-частотна характеристика
ВДЖ – вторинне джерело живлення
ВС – вимірювальна система
ДС – динамічна система
ЕМП – електромеханічний перетворювач
ІВС – інформаційно-вимірювальна система
ІВТ – інформаційно-вимірювальна техніка
ММ – математична модель
МС – мехатронна система
ПІД – пропорційно-інтегрально-диференціальний регулятор
ПФ – передавальна (передатна) функція
РРЧ – режим реального часу
САПП – системи автоматизованого проєктування
САК – система автоматичного керування
САР – система автоматичного регулювання
СНП – силовий напівпровідниковий перетворювач
СП – сенсорні пристрої
ТАК – теорія автоматичного керування
ЦФ – цільова функція
ШИМ – широтно-імпульсна модуляція (сигналу)

ВСТУП

Конспект лекцій призначений для теоретичної та практичної підготовки, самостійних робіт, підготовки до модульного контролю для студентів спеціальності G6 – «Інформаційно-вимірювальні технології», освітня програма: «Інформаційні системи моніторингу і контролю»; та спеціальності G5 – «Електроніка, електронні комунікації, приладобудування та радіотехніка» освітня програма «Мікро- та наноелектронні прилади і пристрої» першого (бакалаврського) рівня вищої освіти.

Дисципліна «Комп'ютерне моделювання» базується на знаннях, отриманих при вивченні дисциплін «Основи алгоритмізації та програмування», «Теорія електронних кіл» та інших дисциплін спеціалізації. Предметом вивчення навчальної дисципліни є: математичні моделі та алгоритми моделювання електронних та мультидоменних систем, зокрема автоматизованих систем.

Програма навчальної дисципліни складається з модулів:

1. моделінг електронних та мультидоменних систем;
2. симуляція електронних та мультидоменних систем.

Мета: підготовка спеціалістів, що зрозуміли і засвоїли основні моделі приладів та пристроїв електроніки та основних алгоритмів моделювання електронних схем. Це дозволить майбутнім спеціалістам приймати оптимальні схемні рішення при автоматизованому проєктуванні пристроїв та систем вимірювальної техніки. В результаті вивчення дисципліни студенти мають навчитися вибирати по критеріям необхідне програмне забезпечення для кожного етапу проєктування ІВС та проводити всебічний аналіз САК та ІВС.

Основне **завдання дисципліни** є освоєння принципів моделювання систем вимірювальної техніки та автоматизованих інформаційних систем.

У результаті вивчення навчальної дисципліни студент повинен отримати **загальні компетентності:**

- здатність спілкуватися державною мовою як усно, так і письмово;
- здатність спілкуватися іноземною мовою;
- навички використання інформаційних і комунікаційних технологій;

- здатність до пошуку, опрацювання та аналізу інформації з різних джерел.

Фахові спеціальні компетентності:

- здатність проектувати засоби інформаційно-вимірювальної техніки та описувати принцип їх роботи;

- здатність, виходячи з вимірювальної задачі, пояснювати та описувати принципи побудови обчислювальних компонент засобів вимірювальної техніки;

- здатність використовувати сучасні інженерні та математичні пакети для створення моделей приладів і систем вимірювань;

- здатність застосовувати стандартні методи розрахунку при конструюванні модулів, деталей та вузлів засобів вимірювальної техніки та їх обчислювальних компонент і модулів;

- здатність ідентифікувати, класифікувати, оцінювати і описувати процеси у мікро- та наносистемній техніці за допомогою побудови і аналізу їх фізичних і математичних моделей;

- здатність застосовувати творчий та інноваційний потенціал в синтезі інженерних рішень і в розробці конструктивних елементів геліоенергетики, приладів фізичного та біомедичного призначення;

- здатність розв’язувати інженерні задачі в галузі мікро- та наносистемної техніки з урахуванням всіх аспектів розробки, проектування, виробництва, експлуатації та модернізації.

Очікувані програмні результати навчання:

- вміння знаходити обґрунтовані рішення при складанні структурної, функціональної та принципової схем засобів інформаційно-вимірювальної техніки;

- знати і розуміти основні поняття метрології, теорії вимірювань, математичного та комп’ютерного моделювання, сучасні методи обробки та оцінювання точності вимірювального експерименту;

- розуміти широкий міждисциплінарний контекст спеціальності, її місце в теорії пізнання і оцінювання об’єктів і явищ;

- вміння використовувати інформаційні технології при розробці програмного забезпечення для опрацювання вимірювальної інформації;

- розуміти застосовуванні методик та методи аналізу, проектування і дослідження, а також обмежень їх використання;

- знати та розуміти сучасні теоретичні та експериментальні методи досліджень з оцінюванням точності отриманих результатів.

ТЕМА №1

ПІДХОДИ ДО МОДЕЛЮВАННЯ СИСТЕМ

Перелік питань:

- Вхідний контроль (запитання у Додатку А).
- Поняття і характеристики математичної моделі (ММ).
- Вибір програми автоматизованого проєктування /моделювання.

У процесі проєктування й експлуатації сучасних технічних об'єктів і систем, зокрема вимірювальних, які досить часто є невід'ємною частиною САК та САР, дослідник постійно змушений мати справу з побудовою та дослідженням моделей цих об'єктів.

Моделювання є науковим інструментом як для чисто теоретичних, так і для практичних досліджень на різних рівнях абстракції. Моделювання є важливою стадією проєктування технічної системи і дозволяє скоротити час та матеріальні засоби при аналізі прийнятих технічних рішень, оптимізації, налагоді та випробуваннях. А коли натурні випробування неможливі за якихось причин, то моделювання стає єдиним засобом для відпрацювання поведінки системи.

Термін «моделювання» достатньо багатогранний. Стосовно технічних систем, в тому числі мехатронних систем (МС), під моделюванням буде розумітися процес, що складається у виявленні основних властивостей досліджуваного об'єкта, побудові моделей та їх застосуванні для прогнозування поведінки об'єкта [1, 2].

САК та САР є динамічними системами (ДС) та досить часто – мультидоменними системами [1, 2]. Моделювання складних ДС, в тому числі, мехатронних, є одним з найбільш потужних інструментів системних досліджень. Основна вимога до результатів моделювання – це точність, оскільки вони мають на меті заміну натурних випробувань. Моделюванням слід користуватися з достатньою мірою обережності та уважності: необхідно не просто отримати результати досліджень, проведених за допомогою тих чи інших моделей, важливо ретельно вивчити досліджувану систему, скласти її опис, підібрати методи відтворення вхідних впливів, побудувати адекватну модель, спланувати і провести модельні експерименти, обробити, візуалізувати та правильно інтерпретувати результати [3].

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 цього експерименту.

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

- безперервні системи, описувані звичайними 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 імовірнісними поняттями і методами.

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 20-sim [4] трьома способами: блок-схемою (рисунок 1.1), фізичною або принциповою схемою (рисунок 1.2), зв'язаним графом (рисунок 1.3).

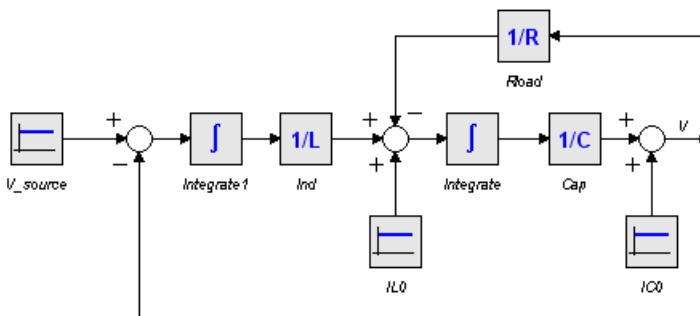


Рисунок 1.1 – Блок-схема (бібліотека «Signal»), що представляє ММ фільтра

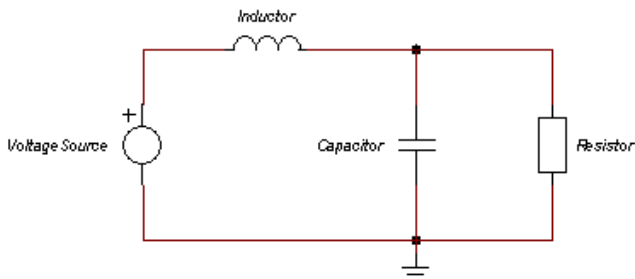


Рисунок 1.2 – Фізична схема (бібліотека «*Iconic diagrams*») фільтра

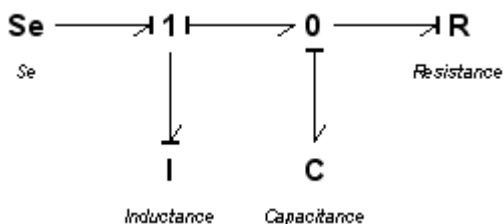


Рисунок 1.3 – Фільтр у вигляді зв'язаного графа (бібліотека «*Bond Graph*»)

Параметри схем однакові: $V_{source}=50$, $L=0.0001$, $C=0.00001$, $R=10$, початкові умови дорівнюють нулю. Математична еквівалентність представлених схем підтверджується результатами симуляції графічних моделей, тобто часовими діаграмами на рисунку 1.4.

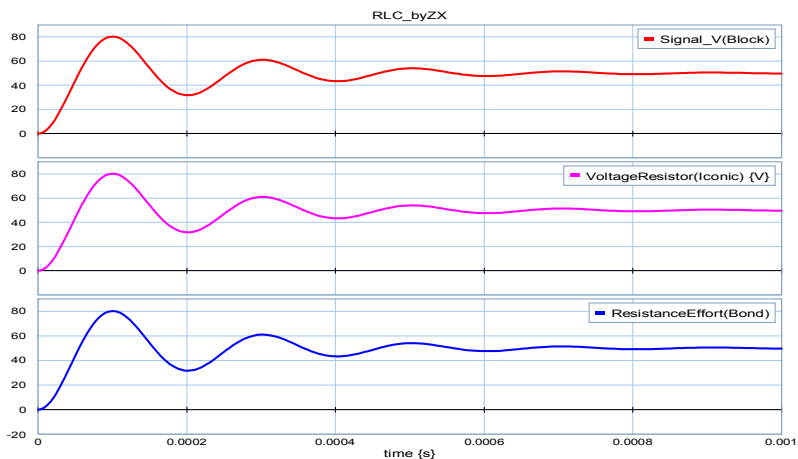


Рисунок 1.4 – Результати симуляції трьох моделей

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.

Будь-яку динамічну систему можна описати однією з форм (або їх комбінацією) представлених на рисунку 1.5.

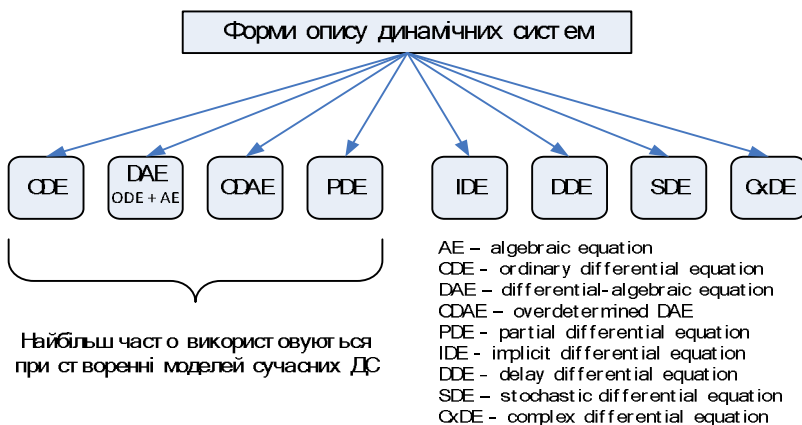


Рисунок 1.5 – Форми опису ДС

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

Математичний опис буде визначати вибір інструментальних засобів і технологій проектування систем, і навпаки: вибір засобу (програми) для дослідження сильно впливає на доступний рівень, підхід та (відповідно) математичний базис до моделювання об'єкту, процесу.

Наразі розвиваються методи та засоби автоматизованого проектуванню технічних систем, характерною особливістю яких є різке підвищення їх логічної складності. При цьому існують досить жорсткі вимоги щодо якості проектування і зниження часу і вартості розробки.

Вирішити зазначені протиріччя, що виникають при проектуванні технічних систем, особливо засобів і систем керування (ЗСУ), на

сучасному етапі можливо тільки через розробку і впровадження систем автоматизованого проектування (САПР). Вміння застосовувати такий інструментальний засіб проектування ЗСУ, як САПР, ґрунтується на розумінні того, яким чином будується САПР, її структура та окремі підсистеми, на знанні ММ, методів їх формування, методів і алгоритмів аналізу та синтезу, які покладені в основу цих підсистем [5].

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 заміником 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;
- що допускає зміни (в процесі експлуатації вона може ускладнюватися).

Формою графічного представлення інформації про модельовану систему можуть бути [6]:

- операторно-структурні схеми, прийняті в ТАУ;
- функціональні та принципові схеми різних фізичних пристроїв;
- кінематичні схеми механізмів;
- сигнальні графи (SFG);

- графи зв'язків або зв'язані графи (BG);
- блок-схеми алгоритмів та інші графічні моделі.

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, наприклад Computer Aided Engineering (CAE), 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. Перехід до зображення структури системи на екрані монітора дозволив виключити етап ручного кодування схеми, однак вимагав розробки нового принципу організації самого процесу чисельних розрахунків.

1.2 Вибір класу САПР

Класифікацію САПР здійснюють за низкою ознак, наприклад, за цільовим призначенням, масштабами (комплексності вирішуваних завдань), характером базової підсистеми – ядра САПР. 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 сфері 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 , інакше CAE-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) [7].

САЕ-системи дозволяють досліджувати системи на макрорівні, тобто аналізувати та оптимізувати систему з метою 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 ЕСАД можуть бути передані програми для верстатів із ЧПУ, роботів та синтезовані технологічні карти.

В системах віртуального виробництва ланка *CAE* → *CAD* → *CAM* утворює наскрізний цикл під керуванням програми – менеджера даними про виріб (*PDM - Product Data Management*) [5, 7].

1.3 Підходи до моделювання систем

Сучасні пакети візуального компонентного моделювання використовують один з двох методів представлення інформації про досліджувану систему – структурне (блочне) моделювання і фізичне мультидоменне моделювання [8]. При цьому за кожним з двох методів можуть стояти певні особливості математичного ядра.

Структурним моделюванням (каузальним) називається техніка моделювання, заснована на використанні моделей у вигляді блоків, для яких визначені входи і виходи. Відповідно, блоки розглядаються як перетворювачі вхідних сигналів у вихідні. При моделюванні лінійних систем зв'язки між вхідними та вихідними сигналами встановлюються за допомогою завдання передавальних функцій. Для нелінійних систем ці зв'язки задаються нелінійними алгебраїчними або диференціальними рівняннями. Оскільки структурні блоки мають виражені входи і виходи, побудовані згідно з цією технікою моделі іноді називають спрямованими (орієнтованими) сигнальними графами. Структурне моделювання використовується, наприклад, в пакеті *Simulink* (рисунк 1.6), *VisSim/Altair Embed*, *SimApp*, *SamSim* тощо.

Фізичним мультидоменним (акаузальним) моделюванням називається техніка моделювання, заснована на використанні бібліотеки

моделей елементів фізичних пристроїв, з яких можна скласти схеми фізичні принципи. Оскільки в енергетичних ланцюгах потік енергії може змінювати напрямок, то для елементів фізичних схем входи і виходи не визначені. Окремим випадком даної техніки моделювання є використання графів зв'язків (bond graph, BG).

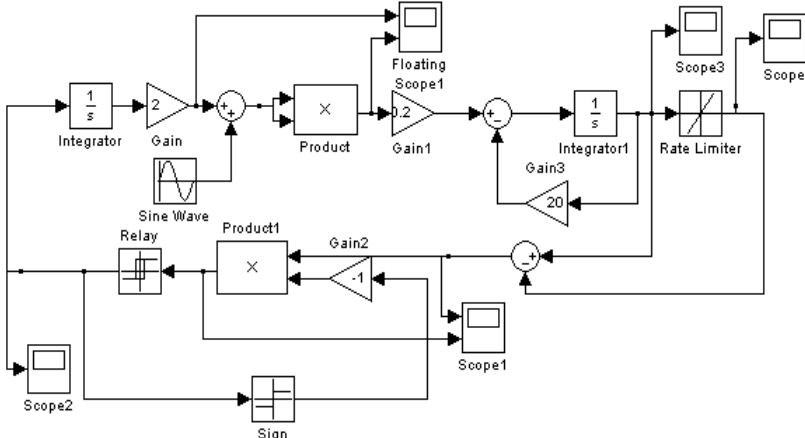


Рисунок 1.6 – Схема для дослідження екстремальної системи в пакеті Simulink

Ідеологія фізичного мультидоменного моделювання полягає в тому, що модель будь-якого технічного пристрою будується як перетворюючий енергію ланцюг (рис. 1.7).

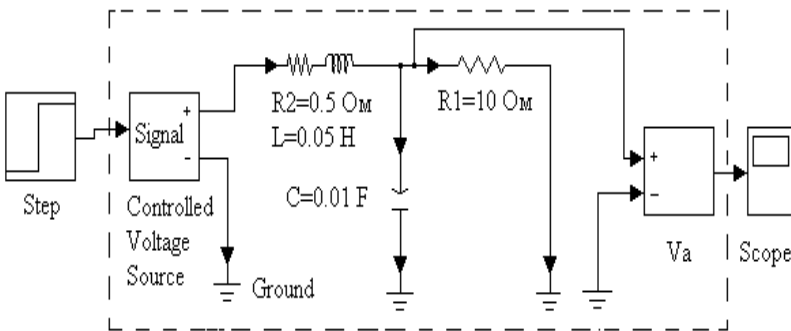


Рисунок 1.7 – Схема для моделювання електричного кола в пакеті SimPowerSystems

У розпорядження користувача програми акаузального моделювання надається бібліотека елементів фізичних пристроїв. В залежності від рівня ієрархії це можуть бути бібліотеки різних енергетичних доменів (електричних, механічних, гідравлічних і т.д.) або більш складних функціональних вузлів, наприклад, механічних ланок і кінематичних пар в механічних моделях, підсилювачів, трансформаторів, перетворювачів, двигунів – у електричних і електромеханічних моделях. Як правило, акаузальний підхід властивий об'єктно-орієнтованим програмам, так званим САД.

Головна відмінність структурного і фізичного мультидоменного моделювання не тільки у формі завдання вихідної інформації, але і в використовуваних методах числового інтегрування диференціальних рівнянь. Зазвичай зі структурним моделюванням пов'язують явний розв'язувач (*solver*), який являє собою бібліотеку класичних підпрограм числового інтегрування, що реалізують явні методи інтегрування.

З фізичним мультидоменним моделюванням зазвичай пов'язують ітераційний розв'язувач. Ітераційний розв'язувач реалізує неявний метод моделювання, при якому на кожному кроці спочатку формується повна нелінійна система алгебраїчних рівнянь, яка далі вирішується ітераційним методом. Важливою особливістю неявних методів (крім усього іншого) є те, що не потрібно штучно розривати систему, щоб організувати потік обчислень. Але явні і неявні методи інтегрування не прив'язані жорстко до структурного та фізичного мультидоменного моделювання.

Деякі сучасні САЕ-системи, незалежно від підходу до моделювання, використовують неявні та явні методи інтегрування [9], а також дозволяють моделювати в часовій області як об'єкти, задані структурними схемами, тобто з використанням спрямованих ланок, так і об'єкти представлені у вигляді фізичних принципів схем (наприклад, *20-sim*, *Maplesim*, *SystemModeler*, *Simplorer*, *Simscape*, *Amesim*).

При завданні вихідної інформації у вигляді принципової схеми в математичне ядро може бути включена процедура перетворення мультидоменної інформації до форми спрямованої структури, а потім можуть бути використані ті ж самі методи явного числового інтегрування, що й у звичайному структурному моделюванні. Саме так зроблено в пакетах *Simmechanics* і *Simpowersystems*, які є підсистемами *Simulink* [10]. У результаті, отриманий деякий проміжний варіант, основним

достойнством якого є форма завдання вихідної інформації. При цьому сам перехід до структури в загальному випадку неоднозначний.

У простих випадках машина може виконувати його самостійно, без звертання до користувача. У більш складних, користувач повинен допомогти програмі сформуванню найбільш раціональну, з погляду витрат на інтегрування, структуру.

1.4 Критерії вибору та огляд САЕ-систем

Сучасні САЕ (структура – в додатку Б) мають наступні відмітні риси [7]:

- 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 числового 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.

Контрольні запитання

1. Поняття та приклади динамічних систем.
2. Форми представлення моделей (приклади).
3. Принципи вибору програми для дослідження електронної системи.
4. Способи моделінгу динамічних систем.
5. Характеристики моделі.
6. Критерії вибору програми моделювання.
7. Показники якості САЕ.
8. Мультидоменне моделювання.
9. Каузальне моделювання.
10. Акаузальне моделювання.
11. Головна відмінність структурного і фізичного мультидоменного моделювання.
12. Поняття САПР.

Завдання для практичного заняття/самостійної роботи

1. За заданою схемою інформаційно-виміральної системи та рівнем абстракції обрати підхід (каузальний, або акаузальний) та клас програм моделювання.
2. Сформувати концептуальну (словесну) модель об'єкта моделювання.
3. Обрати спосіб та рівень моделювання.
4. Обрати математичний базис для опису моделі на етапі моделінгу.
5. Сформувати (синтезувати) модель в обраному базисі (моделінг).

ТЕМА №2

ОГЛЯД ПРОГРАМ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Перелік питань:

- Принципи наскрізного проектування електронних систем (див. конспект).
- Поняття та структура систем віртуального виробництва (див. конспект).
- Огляд програм класу CAE. Каузальне моделювання.

CAD системам приділено значну увагу в [11] та навчально-методичному посібнику [3]. Для моделювання на макрорівні частіше використовують математичні процесори та системи автоматизованого інжинірингу (CAS та CAE відповідно).

На даний момент існує багато програм рівня CAE (або таких, що його охоплюють, наприклад, *Micro-Cap* [12]), які дозволяють, зокрема, моделювати (*modeling & simulation*) 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 MATRILXX*, *ACSLx*, *Modular Modeling System (MMS)*, *Virtual Test Bed (VTB)*, *JModelica.org*, *Yenka (Crocodile Technology)*, *MATLAB\Simulink*, *Simscape*, *MapleSim*, *Dynast*, *Multisim*, *K2.SimKernel*, *Jigrein*, *EASY5*, *AMESim*, *Dymola*, *PSIM*, *SimApp*, *SimulationX*, *Simplorer*, *VisSim/Altair Embed*, *SystemModeler*, *20-sim* та інші, які можуть бути знайдені у [13].

Оскільки ДС 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. Увага буде зосереджена на CAE-системах для моделювання автоматизованих систем (САК, ІВС тощо).

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 програ-

ми. Більшість з перелічених програм моделювання технічних динамічних систем розвиваються та підтримуються виробниками. Результати порівняння можливостей САЕ – в додатку В.

Деякі з них, при наявності відповідної інформації, будуть розглянуті за певним 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;
- ліцензії, вартість та наявність безкоштовних версій.

2.1 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 аналізу аналогово-цифрових пристроїв та мультидоменого 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 [15].

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 мультидоменої 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 керування. *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 числового інтегрування. При цьому нелінійні рівняння лінеаризуються за допомогою ітераційного методу Ньютон-Рафсона (метод дотичних), а потім вже лінеаризована система рівнянь вирішується за допомогою LU-факторизації (варіація метода Гауса).

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 передумовки). Це спрощує процедуру дослідження САК. Але програма «перевантажена» з точки зору інтерфейсу, має «зайві» модулі, які перешкоджають візуальному сприйняттю.

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 розв'язувачем (*solver*) втрата стійкості та збіжності.

2.2 SimApp

SimApp це програма моделювання поведінки ДС в часовій та частотній областях. Можливе моделювання (*modeling & simulation*) безперервних і дискретних систем. Найкраще підходить для дослідження систем автоматичного керування. 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. Можна обчислити власні значення частот (*eigenvalues, eigenfrequencies*) лінійних систем. Програма підтримує дослідження систем з різними значеннями параметрів і різними зовнішніми збуреннями. SimApp відрізняється простотою у використанні.

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 проєкті), 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 проєкті всього лише 10 блоків [16].

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 (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 ПД 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 автоматично вирішувати проблему алгебраїчних петель за рахунок символічних петлеворень на етапі створення рівнянь ДС перед процесом симуляції.

2.3 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 числового 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*. Архітектура програми представлена на рисунку 2.1.

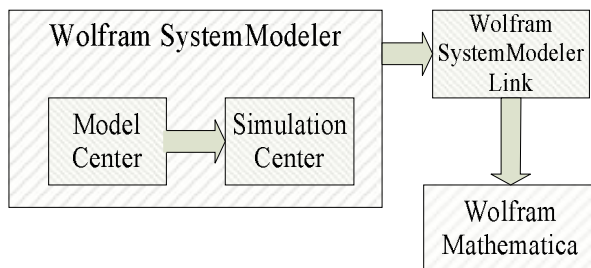


Рисунок 2.1 – Структура програми SystemModeler та зв'язок з Wolfram Mathematica

Також *SystemModeler* передбачає повне з'єднання з системою *Mathematica* для отримання найбільш інтегрованого робочого процесу фізичного, числового 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. 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.

SystemModeler не має модулів оптимізації, оскільки є дочірньою програмою *WolframMathematica*, 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. 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. Інтерфейс пробної версії *SystemModeler 3.0.0* зображений на рисунку 2.2.

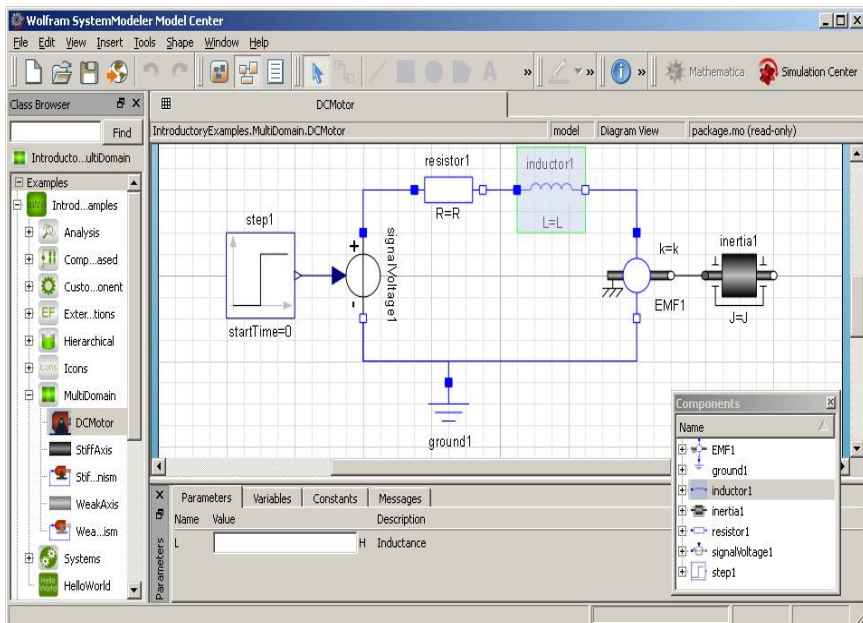


Рисунок 2.2– Головне вікно Model Center з відкритим прикладом (DCMotor)

SystemModeler має наступні солвери (*solvers*):

- *DASSL* (метод зі змінними шагом та порядком, що використовує формулу зворотного диференціювання) [17];
- *CVODES* (як попередній, але є ще аналіз чутливості) [18];
- *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.

Програма не має передвстановлених можливостей для відображення та аналізу частотних характеристик (АЧХ, ФЧХ, різні годографи), що є важливим аспектом при дослідженні САК.

2.4 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.

Вид головного вікна програми відповідає рисунку 2.3.

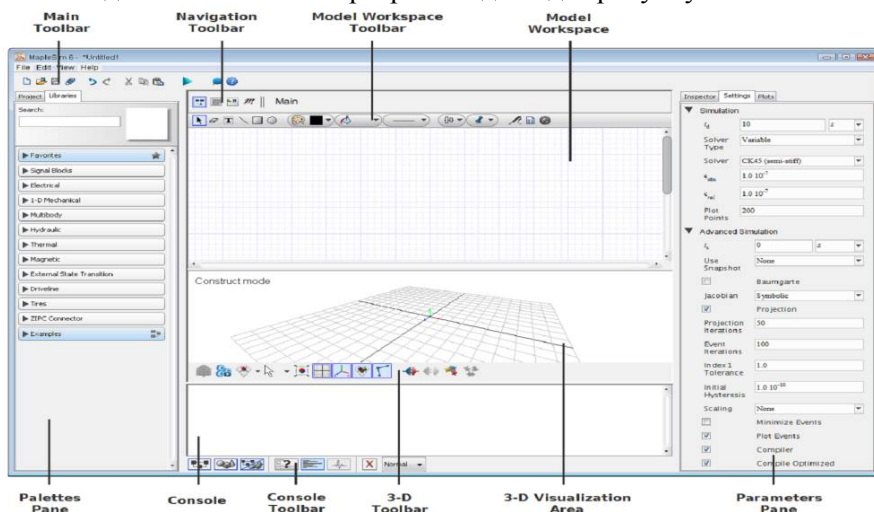


Рисунок 2.3 – Головне вікно програми MapleSim 6

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 керування. Блок-схема моделі може виглядати точно так само, як описувана нею реальна система. З готових компонентів конструктор може зібрати найскладнішу багатофакторну модель, також можна одночасно готувати всю супровідну документацію до проєктних робіт. Розвинуті алгоритми символьних обчислень дозволяють автоматично спрощувати складні моделі. Крім того, відмова від традиційного числового моделювання на користь символьних 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 керування 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.

Зі слів авторів програми *MapleSim 6*, вона володіє наступними ключовими особливостями [19]:

- потужне символічне та числове ядро *MathEngine*;
- 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 розв'язувачів при симуляції;
- доступність рівнянь структурного (системного) рівня;
- керування одиницями вимірювання для кожного об'єкту;
- широкий спектр 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.

Є кілька редакцій програми *MapleSim 6*, що різняться функціональністю та ціною: *Student Edition* (\$124), *Commercial* (\$6895), *Academic* (\$1555), *Government* (\$6550). Причому 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 розв'язувачі (*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);
- *RKF45* (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 ані в *MapleSim*, ані в сам *Maple*, отже потрібно докуповувати це доповнення окремо.

2.5 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 керування 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 *Ci*, 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 САЕ – 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, *PowerSystem Blockset* – моделювання електротехнічних пристроїв, *Digital Signal Processing Blockset* – набір блоків для розробки цифрових пристроїв, *Control System Toolbox* – пакет для розробки систем керування 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 модельованої 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* [20] і *MATLAB \ Simulink* пов'язує одна компанія-виробник – *MathWorks*[21]. 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 керування 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 параметризації моделі з використанням змінних та виразів *MATLAB*, а також проєктування моделі об'єкту керування фізичної системи, система керування якої може бути побудована в *Simulink*. Для розгортання моделі в інших середовищах моделювання *Simscape* підтримує генерацію *Ci*-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), *SimulinkDesign 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) [21]. Також треба відмітити, що компанія *MathWorks* не надає безкоштовні пробні версії студентам і приватним особам (*IndividualLicense*) [22].

Навіть якщо продукти компанії *MathWorks* перебивають 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; розв'язувачі поділяються на методи зі змінним та постійним кроком. Параметрична оптимізація у *MATLAB* можлива за допомогою додаткового інструментарію *OptimizationToolbox* – 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-Ньютонівський 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 [23].

2.6 VisSim/Altair Embed

VisSim/Altair Embed – пакет компонентного візуального моделювання фірми *Visual Solutions*, призначений для розробки і моделювання динаміки безперервних, дискретних та гібридних систем [24]. За своїми функціональними можливостями і способом завдання початкової інформації *Altair Embed* близький до пакету *MATLAB / Simulink*. *Simulink* більш поширений, тим не менш, він має чимало шанувальників і займає помітну частку ринку інженерних пакетів. *Altair Embed* – один з представників систем, що реалізують концепцію структурного (каузального) моделювання. Наразі програма є власністю Siemens.

Основним інструментом створення моделей є типові блоки «вхід-стан-вихід». Подібно до *MATLAB / Simulink* кожен типовий блок *VisSim/Altair Embed* реалізує математичну модель або забезпечує візуалізацію того чи іншого явища, процесу або пристрою. Базова бібліотека *Altair Embed* 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 користувальницькі бібліотеки практично будь-якою процедурною мовою програмування. Процедура інкапсуляції дає можливість створювати нові блоки (підсистеми), ґрунтуючись на вже існуючих блоках.

Як і *Simulink*, *Altair Embed* має широкий набір інструментів оцінки якості, стійкості, синтезу, корекції, оптимізації, лінеаризації, налагодження об'єктів в контурі моделі та програмування цифрових сигнальних процесорів. Вирішувач *Altair Embed* інтерпретуючого типу, функціонує в динамічному режимі з можливістю online-взаємодії з обладнанням реального часу. У складі пакету розв'язувача *Altair Embed* реалізовані як явні, так і неявні методи. Можливості моделювання гібридних систем у пакеті *Altair Embed* приблизно аналогічні *Simulink*. *Altair Embed* призначений насамперед для моделювання безперервних систем. Загальним принципом гібридного моделювання є використання готових гібридних блоків або перемикання заготовлених альтернативних ділянок блок-схем.

Пакет *VisSim/Altair Embed* включає в себе ряд додаткових модулів, що розширюють його можливості у конкретних предметних областях, також є інструменти інтеграції з пакетами *MATLAB*, *Mathcad* і транслятор проєктів програми *Simulink*. В якості обмежень пакету *VisSim/Altair Embed* можна відзначити відсутність альтернатив мові блок-схем. Крім того, *VisSim/Altair Embed* 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.

Додатковий модуль *State Charts* дозволяє оперувати *UML*-сумісними (*Unified Modeling Language*) графічними діаграмами станів і редагувати та генерувати *Ci*-код для кінцевих автоматів, створених у цьому середовищі. Інші особливості програми наведено в [26].

VisSim/Altair Embed, окрім інших редакцій, випускається у вигляді академічної (приблизна ціна – USD \$2000) та професійної (USD \$8000) редакцій [27], версія *VisSim 3.0* є безкоштовною (вільно розповсюджується) для персонального використання з обмеженнями [28].

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 супроводжуючої літератури та широку вживаність програми, взаємодія з поширеними відомими програмними пакетами (*MATLAB*, *Simulink*, *Mathcad*, *Maple*, *Excel*).

Модулі *VisSim/Altair Embed* представлено на рисунку 2.4.

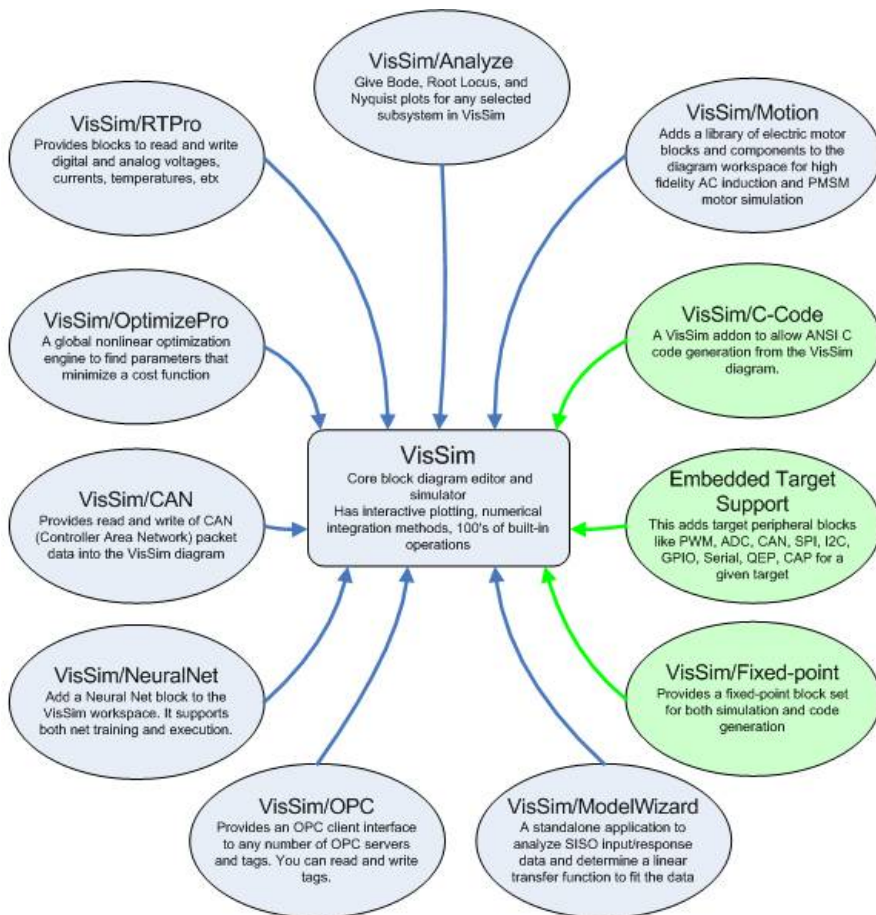


Рисунок 2.4 – VisSim/Altair Embed та додаткові модулі

Altair Embed дозволяє використовувати декілька методів числового інтегрування диференціальних рівнянь (методи Рунге-Кутти, Булірша-Стоєра, Адамса-Мултона та ін.) та проводити оптимізацію (методами Пауелла, Полака-Рібері та Флетчера-Рівза).

На рисунку 2.5 представлені доступні алгоритми інтегрування, а рисунок 2.6 відображає можливі методи параметричної оптимізації.

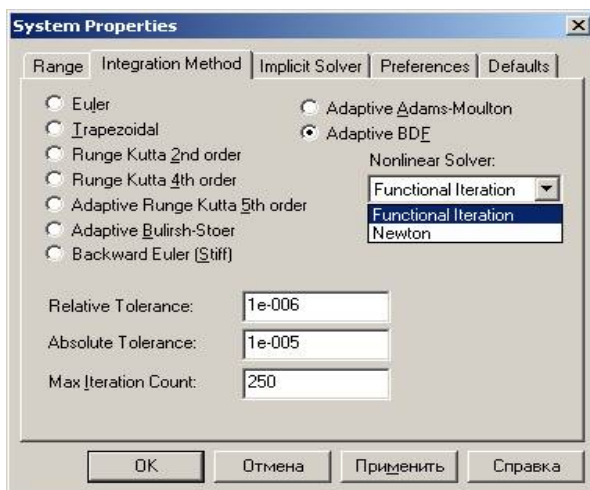


Рисунок 2.5 – Вікно вибору методів інтегрування в VisSim/Altair Embed

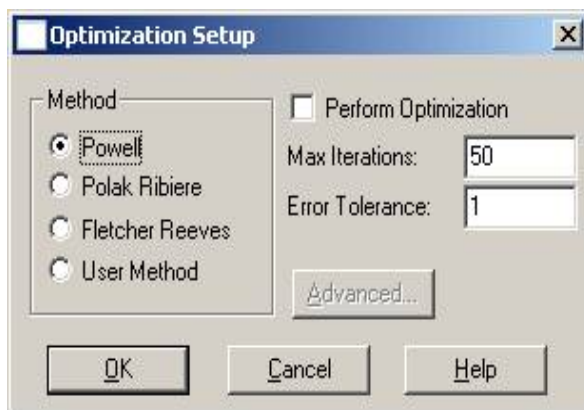


Рисунок 2.6 – Вікно вибору методів параметричної оптимізації в VisSim/Altair Embed

Недоліки *VisSim/Altair Embed*: 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 2.7.

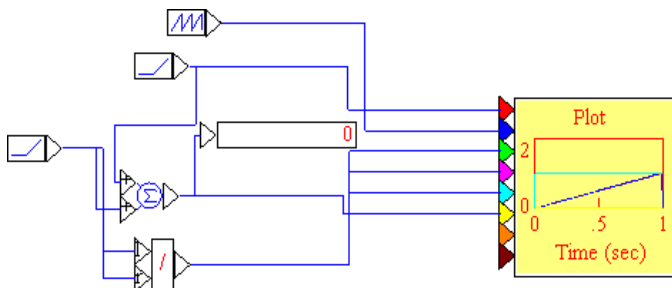


Рисунок 2.7 – Ілюстрація відсутності вузлів з'єднання в VisSim/Altair Embed

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

2.7 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.

Архітектура програми представлена на рисунку 2.8 [29].

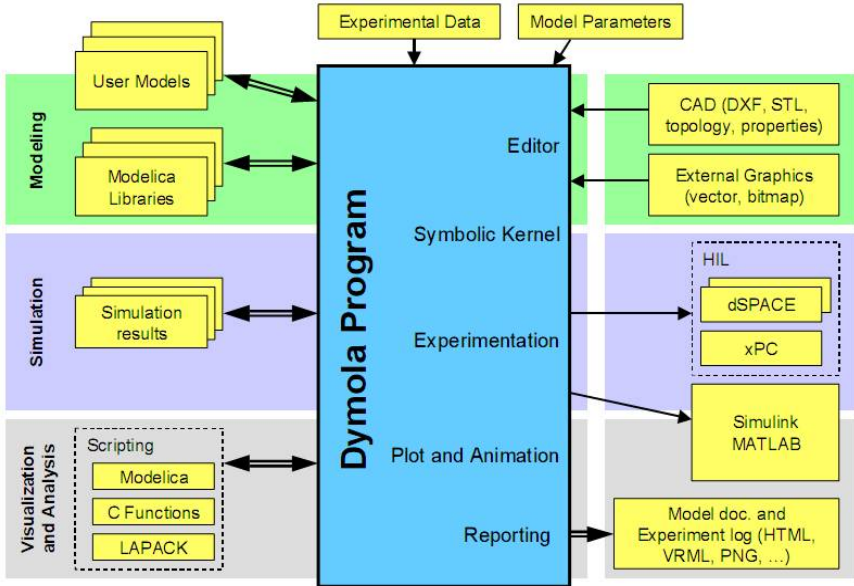


Рисунок 2.8 – Архітектура Dymola

Dymola дозволяє користувачам створювати власні бібліотеки моделей, описуючи їх на внутрішній мові опису блоків, або модифікувати існуючі моделі під свої 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, DAE та ODAE (*Overdetermined DAE*).

2.8 20-sim

Пакет *20-sim* [30] 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 бібліо-

теки, 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 керування 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 (*BondGraph*), як найпростіших універсальних енергетичних компонентів; механічних, електричних, гідравлічних, теплових елементів (*IconicDiagram*). В якості прикладу на рисунку 2.9 представлена модель системи, де енергетична частина представлена графом зв'язків, а керуюча – блок-схемою, що включає спрямовані ланки.

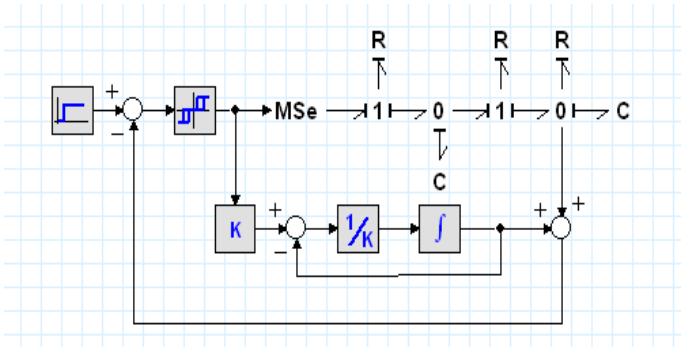


Рисунок 2.9 – Приклад моделі системи в 20-sim

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 проєктувальникам, 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 DAE, використовуючи однокрокові і багаток-

рокові 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-Powel*, *Broydon-Fletcher-Goldfarb-Shanno*, *Polack-Ribiere*.

20-sim має широкі можливості щодо моделювання (*modeling*) та дослідження САК, 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*.

Особливості програми:

- принцип відкритості, забезпечуваний механізмами підключення зовнішніх програм (*Matlab/Simulink*) і можливістю переглядати, редагувати й створювати нові бібліотеки моделей;

- принцип вкладеності або ієрархічності структур (глибина вкладеності не обмежена), що особливо актуально при моделюванні складних ІВС;

- значний обсяг бібліотеки моделей, що в багатьох випадках дозволяє сформувавши математичну модель динаміки складної технічної системи без необхідності створення нових 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 редагувати систему (модель) як простір станів, передатну функцію, нуль-полюса й перетворювати їх між собою, а також містить опції показу діаграм Боде, годографа Найквіста, графіка Нікольса, діаграм нуль-поліосів, реакції системи на функцію Хевісайда;

- можливість транслювати будь-яку модель у *Ci*-код для використання в *Matlab/Simulink* або автономний *Ci*-код для використання в *C* і *C++* 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 розвинутої 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*»).

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

Контрольні запитання

1. Огляд CAE-систем.
2. Огляд CAS систем.
3. Огляд CAD систем.
4. Структура CAE-системи (приклад).
5. Критерії порівняння CAE 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 демо- та студентські версії?
10. Мови моделювання.

Завдання для практичного заняття/самостійної роботи

1. За заданою схемою інформаційно-вимірjuвальної системи та рівнем абстракції обрати підхід та програму моделювання згідно обраних критеріїв.
2. Обрати спосіб представлення моделі згідно інструментально-математичної бази обраної програми.
3. Порівняти програми 20-sim та VisSim/Altair Embed.
4. Дослідити можливість переходу між Matlab та іншими програмами моделювання.
5. Дослідити можливості симуляторів Matlab.

ТЕМА №3

МОДЕЛЮВАННЯ МЕХАТРОННИХ СИСТЕМ

Перелік питань:

- Мехатронна система (МС) як динамічна система (ДС).
- Приклади та структури мехатронних систем.
- Моделювання мехатронної системи (електроприводу).

Система – множина елементів, які знаходяться у зв'язках один з одним, та які створюють певну цілісність, єдність. Елементом системи називають найпростішу складову частину системи, яку умовно розглядають як неподільну. Взагалі існує декілька визначень системи, відмінності між котрими визначається у тому: чи розглядається система як об'єкт існуючий незалежно від спостерігача (визначення наведене вище), чи як метод вивчення й представлення феноменів (приклад: система – комбінація взаємодіючих елементів, організованих для досягнення однієї чи декількох цілей).

Властивості систем: цілісність (діяльність системи підпорядкована певній цілі), синергетичність (гармонійне поєднання і взаємодія всіх елементів і підсистем як єдиного цілого з метою отримання додаткового позитивного ефекту), ієрархічність (кожен компонент системи може розглядатися як система (підсистема) ширшої глобальної системи).

Під динамічною системою розуміють об'єкт або процес, для якого однозначно визначені поняття стану як сукупності величин у даний проміжок часу, та визначено закон, котрий описує зміни (еволюцію) начального стану з перебігом часу. Реальним фізичним системам, модельованим математичним поняттям «Динамічної системи», приписується важлива властивість детермінованості: знаючи стан системи в початковий момент часу, ми можемо однозначно передбачити всю її подальшу поведінку.

Відрізняють динамічні системи з дискретним та неперервним часом. Динамічна система (як з дискретним, так і з неперервним часом) є по суті синонімом автономної системи диференціальних рівнянь, що задана в деякій області і задовольняє там умовам теореми існування і єдності розв'язку диференціального рівняння. Положенням рівноваги динамічної системи відповідають особливі точки дифе-

ренціального рівняння, а замкнуті фазові криві - його періодичним рішенням.

Приклади динамічних систем: LC-контур, механічний маятник, Броунівський рух часток.

В загальному випадку складна система є багаторівневою конструкцією із взаємодіючих елементів, поєднаних у підсистеми різних рівнів.

Зображення модельованого об'єкта у вигляді багаторівневої конструкції із елементів називають структуризацією об'єкта. У структуризованій системі об'єктами матеріального світу є лише елементи та зв'язки.

Основні ознаки складних систем:

- мають багато взаємопов'язаних та взаємодіючих елементів;
- мають складну функцію, яка виконується системою та направлена на досягнення мети функціонування;
- можливість розкладання системи на підсистеми, мета функціонування яких підпорядковуються загальній меті системи;
- наявність керування, розгалуженої інформаційної мережі та інтенсивних потоків інформації;
- наявність взаємодії з зовнішньою середою та функціонування в умовах дії випадкових факторів.

Основні етапи створення складної системи:

- формулювання вимог до системи та складання технічної пропозиції та технічного завдання на проектування ;
- розробка технічного проекту;
- створення дослідного зразка;
- випробування;
- виготовлення та ввід у експлуатацію дослідного зразка системи;
- дослідна експлуатація та доробка зразків;
- організація випуску та ввід у експлуатацію серійних зразків;
- модернізація системи.

Проектування складних систем має дві стадії. Перша відноситься до функціонально-структурних питань та може бути названа макро-проектування м або зовнішнім проектування м системи.

Друга стадія – мікро-проектування або внутрішнє проектування – пов'язана з проектування м елементів складної системи як фізичних одиниць обладнання. Мікро-проектування включає у себе технічні

рішення по основним елементам системи, їх конструкції та параметрам, режимам експлуатації, а також по організації дослідного та серійного виробництва.

Серед задач, що виникають в зв'язку із дослідженням складних систем, можна відокремити два загальних класи:

- задачі аналізу, пов'язані з вивченням властивостей та поведінки системи в залежності від її структури та значень параметрів;
- задачі синтезу, які зводяться до вибору структури та значень параметрів, виходячи з заданих властивостей системи.

На практиці користуються різними неформальними прийомами синтезу складних систем, але всі ці прийоми зводяться до перебору варіантів або «синтезу через аналіз».

Практично будь-яка електронна система може бути описана за допомогою поняття динамічної системи. Динамічна система – математична абстракція, призначена для опису і вивчення систем; являє собою математичну модель деякого об'єкта, процесу або явища; це така система, параметри якої явно чи неявно залежать від часу. В одному з можливих визначень під динамічною системою розуміється структура, в яку в певні моменти часу вводиться щось (речовина, енергія або інформація) і з якої в якісь моменти часу виводиться щось. Вона служить для опису причинно-наслідкових зв'язків з минулого в майбутнє. Таким чином, динамічна система характеризується своїм початковим станом і законом, за яким система переходить з початкового стану в інший.

Структура моделей ДС часто відповідає структурі досліджуваного об'єкта (системи). В основі таких моделей лежить елемент (блок), з прихованою від зовнішнього спостерігача внутрішньою структурою. Блок інкапсулює свої властивості. Структурно складна модель ДС складається з багатьох блоків, взаємодіючих між собою через функціональні зв'язки між видимими ззовні змінними. Структура такої системи зазвичай є ієрархічною. Елементи системи можуть, в загальному випадку, змінюватися в процесі функціонування системи. Як правило, елементи складної системи характеризуються різними фізичними принципами дії, що, врешті-решт, не настільки помітно в підсумковій ММ, але надзвичайно важливо на етапі побудови моделі. Види та зв'язок ДС, САК та МС (останні складаються з пристрою керування УУ та об'єкта керування ОУ) показане на рисунку 3.1.

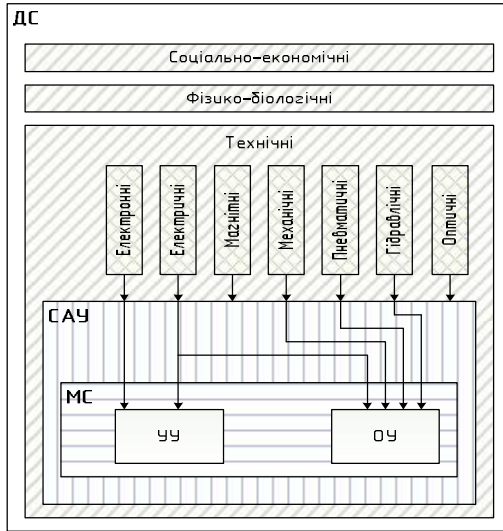


Рисунок 3.1 – Зв'язок ДС, САУ та МС

Прикладами складних ДС можуть бути:

- в галузі технічних систем: електромеханічна слідкуюча система відтворення кута повороту; система стабілізації температури у відсіку космічної орбітальної станції; цифрова система керування рухом літака в режимі заходу на посадку та інші;

- у галузі фізико-біологічних систем: система стабілізації кров'яного тиску людини; система контролю за поширенням інфекційних захворювань; система відтворення запасів риби в деякому районі риболовецького промислу; система підтримки теплового балансу Землі та інші;

- в галузі соціально-економічних систем: система керування галузю промисловості; система прогнозування купівельного попиту на деяку групу товарів та інші.

Надалі в якості динамічних систем будуть розглядатися тільки технічні системи. Системи керування технічних об'єктів є окремим випадком ДС.

Цілісність, структура, елемент, нескінченність та ієрархічність складають ядро системоутворюючих понять загальної теорії систем і є основою системного уявлення об'єктів і формування концепцій системних досліджень.

3.1 Особливості мехатронних систем

Слово «Мехатроніка» складається з двох частин – механіка і електроніка, це нова область науки і техніки, присвячена створенню і експлуатації машин і систем з комп'ютерним керуванням рухом, яка базується на знаннях в галузі механіки, електроніки і мікропроцесорної техніки, інформатики і комп'ютерного керування рухом машин і агрегатів.

З аналізу даного визначення випливає:

- МС призначені, для реалізації заданого руху й основу будь-якої МС становить деякий виконавчий механізм;
- необхідною частиною МС є привод – електромеханічний, гідравлічний або якийсь інший;
- важливим компонентом МС є керуючий пристрій, завдання якого – забезпечення складних координованих рухів механічної частини.

Це дозволяє визначити особливості моделі мехатронних пристроїв, розуміючи під ними об'єкти, для дослідження та проектування яких використовуються математичні моделі, що відображають взаємний вплив процесів різної фізичної природи – механічних, електричних, інформаційних тощо [31].

МС, що включають в свій склад електромеханічні перетворювачі з електронною комутацією (актуатори), різного виду датчики (сенсори), силові напівпровідникові перетворювачі, мікроконтролери та персональні комп'ютери, широко застосовуються та інтенсивно розвиваються.

В даний час мехатронні пристрої широко використовуються в медицині (штучні органи), в комп'ютерній техніці (приводи дисків, принтери, сканери), в автомобілях (гальмівні системи, системи керування двигуном), в роботах і маніпуляторах, у верстатах з числовим програмним керуванням (ЧПК), в аудіо- і відео-техніці, у спеціальних пристроях космічної та військової техніки.

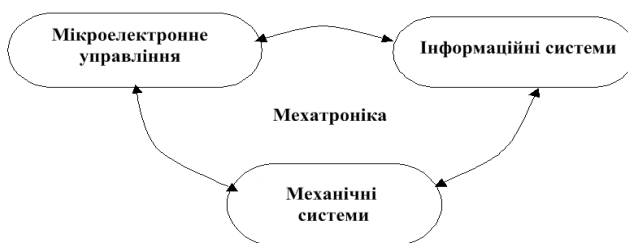
Застосування мехатронного підходу при створенні машин з комп'ютерним керуванням визначає їх основні переваги в порівнянні з традиційними засобами автоматизації:

- відносно низьку вартість завдяки високому ступеню інтеграції, уніфікації й стандартизації всіх елементів і інтерфейсів;

- високу якість реалізації складних і точних рухів внаслідок застосування методів інтелектуального керування;
- високу надійність, довговічність і перешкодозахищеність;
- конструктивну компактність модулів (до мініатюризації в мікромашинах);
- поліпшені масогабаритні й динамічні характеристики машин внаслідок спрощення кінематичних ланцюгів;
- можливість комплексування функціональних модулів у складні МС під конкретні завдання замовника [32, 33].

3.2 Склад мехатронної системи

У мехатронних системах прийнято виділяти три складові частини – механічну, електронну й комп'ютерну, об'єднання яких за допомогою інтерфейсів і утворює систему в цілому. Суть мехатронного підходу полягає в тісному взаємозв'язку зазначених компонентів на всіх етапах життєвого циклу виробу, починаючи зі стадії його проектування і маркетингу та закінчуючи виробництвом і експлуатацією замовником. Як нова область техніки, мехатроніка перебуває на стику мікроелектронного керування, інформаційних систем і механічних систем (рисуюнок 3.2) [33].



Рисуюнок 3.2 – Мехатроніка як міждисциплінарна наука

Мехатронний пристрій має наступні характерні ознаки [32]:

1. Наявність вихідної механічної ланки (ВМЛ), що виконує зовнішні функції мехатронного пристрою; силового електромеханічного привода вихідної ланки; пристрою програмного керування приводом; інформаційної системи, що контролює стан навколишнього середовища й стан самого мехатронного пристрою.

2. Мінімум перетворень інформації й енергії (наприклад, використання високомоментних або лінійних двигунів без редукторів) – принцип мінімуму перетворень.

3. Використання того самого елемента мехатронного пристрою для реалізації декількох функцій – принцип сполучення функцій.

4. Об'єднання корпусів мехатронного пристрою – принцип сполучення корпусів.

5. Застосування надщільного монтажу елементів.

Мехатронний пристрій може бути або машиною, або вузлом (функціональним елементом, модулем) машини. Прикладами сучасних мехатронних пристроїв є модулі верстатів і промислових роботів, принтери, побутова техніка й т.п. Зв'язки між функціональними перетворювачами (ФП) в МС наведено на рис. 3.3.

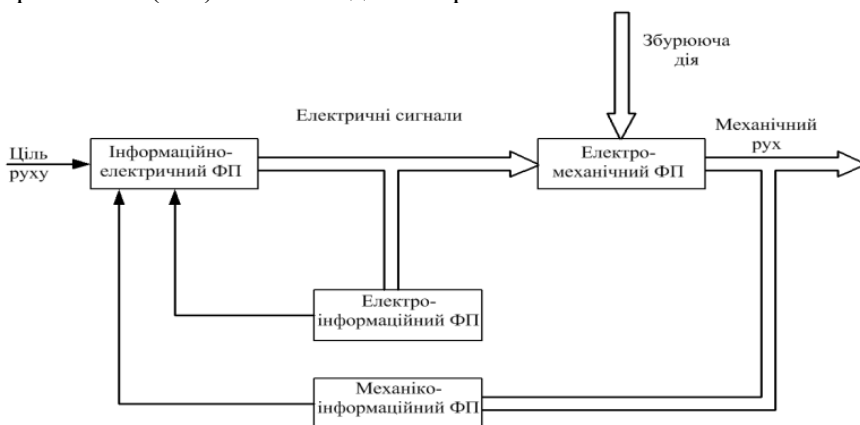


Рисунок 3.3–Інформаційні й енергетичні потоки в мехатронній системі

Тут використано скорочення ФП – функціональний перетворювач; жирними стрілками показані енергетичні потоки, тонкими – інформаційні потоки.

Концептуальну модель робота представлено як результат взаємодії під-систем МС з відповідними складовими кожної підсистеми та областями взаємодії в середовищі Виробництво–Вимоги ринку–Менеджмент (рисунок 3.4). Як бачимо, перетин кожних двох підсистем утворює низку питань для досліджень. Функціональна схема МС (рисунок 3.5) в загальному вигляді включає три підсистеми: інформаційну, енергоелектронну і електромеханічну [33].

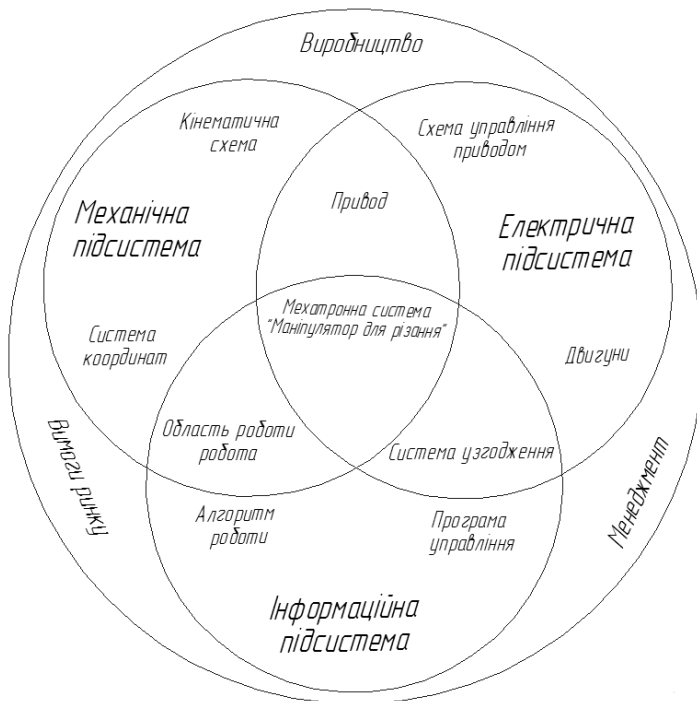


Рисунок 3.4 – Модель мехатронної системи робота-маніпулятора

Електромеханічна підсистема містить об'єкт керування (ОК) і електромеханічний перетворювач (ЕМП). Енергоелектронна підсистема включає силовий напівпровідниковий перетворювач (СНП) та вторинне джерело живлення (ВДЖ). Інформаційна підсистема містить систему керування і діагностики (СКД) і блок сенсорних пристроїв (СП). При комплексному дослідженні МС необхідно розглядати динамічні, статичні, енергетичні, спектральні характеристики тощо.

Прикладом мехатронної системи є металорізальний верстат із числовим програмним керуванням (ЧПК), топологія підсистем та кінематична схема якого наведена на рисунку 3.6. Суппорт верстата встановлений на фундаменті. На ньому є два механічні елементи, які за допомогою електроприводів переміщуються по ступенях рухливості на величини x , y , z , φ . Керування електроприводами ведеться від керуючих пристроїв, вартість яких досягає 40 % від вартості верстата.

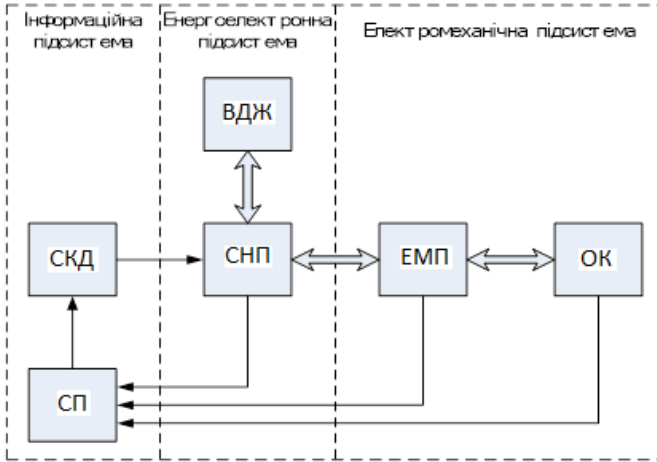


Рисунок 3.5 – Функціональна схема мехатронної системи

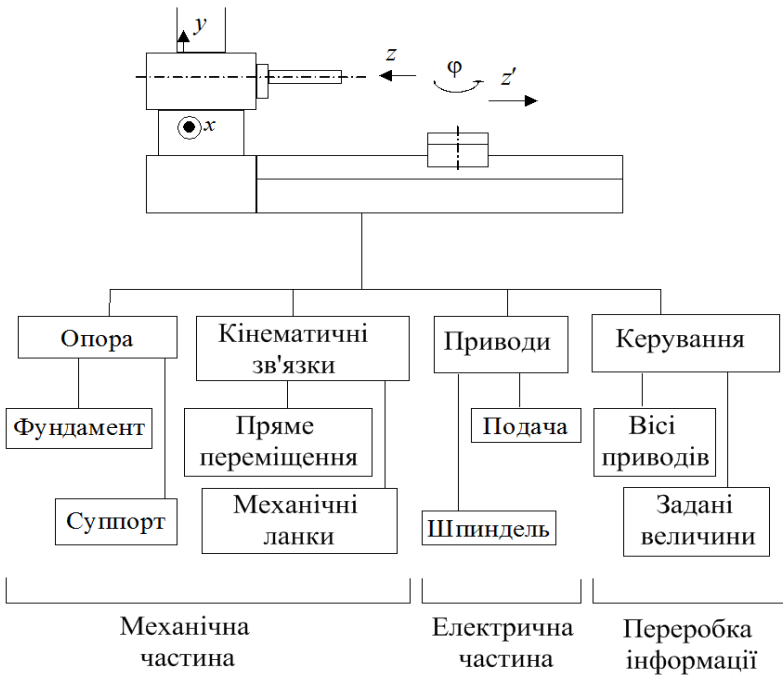


Рисунок 3.6 - Металорізальний верстат як мехатронна система

Слід підкреслити основні особливості МС:

а) об'єкти керування МС безпосередньо не пов'язані з виробництвом. Як правило, вони замінюють людину в умовах, де її можливості обмежені або там, де її здоров'ю та життю загрожує небезпека;

б) об'єкти керування (ОУ) мехатронних систем, як правило, мають змінні параметри. Часто не вдається створити математичну модель об'єкта керування на основі фізичних закономірностей його роботи. У цьому випадку для математичного опису ОУ доводиться вдаватися до методів ідентифікації;

в) електромеханічні перетворювачі ЕМП часто є конструктивною ланкою об'єкта керування;

г) керування потоком енергії від ВДЖ до ЕМП здійснюється силовим напівпровідниковим перетворювачем СНП. Об'єднання ВДЖ, СНП та ЕМП створює мехатронну енергетичну підсистему, в якій виявляються нові властивості (принцип синергії);

д) мехатронна енергетична підсистема, як правило, є нелінійною, імпульсною підсистемою з дискретно змінними параметрами. Тому її аналіз, а також синтез системи керування вимагає застосування методів, що базуються на сучасних комп'ютерних технологіях.

Розрахунки й проєктування МС включають:

- побудову спільної мультидоменної моделі;
- дослідження динамічних характеристик безперервної й імпульсної частини;
- синтез регулятора;
- дослідження динамічних, статичних характеристик системи.

У мехатроніці об'єктами керування доцільно вважати самі електромеханічні системи різного ступеня складності, зокрема нею може бути електричний привод [33].

Головною особливістю сучасного етапу розвитку мехатронних модулів є інтелектуалізація процесів керування їх функціональними рухами. Технічна реалізація інтелектуальних мехатронних модулів руху (ІММР) стала можливою завдяки бурхливому розвитку в останні роки мікропроцесорних систем, орієнтованих на завдання керування руху.

Постійне вдосконалювання виробничих технологій веде до стабільного зниження вартості апаратних засобів, що зробило їх до теперішнього часу рентабельними для практичного впровадження.

3.3 Необхідність моделювання мехатронних систем

Будь-яка ДС (зокрема, її об'єкт керування ОУ), що вимагає керування, повинна бути ідентифікована для побудови її моделі з метою створення необхідного керуючого пристрою для неї. За поведінкою моделі певної ДС можна, у деякому наближенні, судити про поведінку реальної системи, а через складність або неможливість проведення випробувань на реальній системі має сенс моделювати поведінку системи з використанням її моделі. Таким чином, моделювання мехатронних і електромеханічних систем є актуальним завданням.

Аналогічно тому, що МС це синергетичне об'єднання механічної, електричної та комп'ютерної (електронної, цифрової) частин, засоби моделювання повинні допускати спільне моделювання цих частин на єдиній методологічній основі, даючи можливість будувати і досліджувати багатоаспектні моделі.

Реалізувати це можливо двома способами. По-перше, можна перейти до єдиної системи диференціальних рівнянь. У цьому випадку всі фізичні особливості окремих частин системи будуть загублені. Варіантом такого підходу є структурне моделювання, де всі змінні є скалярними сигналами і їх можна з'єднувати (як у структурній схемі). Недолік підходу – великий обсяг попередніх перетворень (у випадку системи рівнянь), або одержання схеми, що мало нагадує реальну систему, що є загальним недоліком каузального підходу у моделюванні.

Інший варіант – використання систем моделювання, які здатні на єдиній методологічній основі моделювати механічні, електричні й інформаційні компоненти, тобто поєднувати їх у єдину схему, зберігаючи при цьому звичні для фахівців у предметних областях способи завдання вихідної інформації, тобто надавати можливість мультидоменного моделювання. Цей підхід відрізняється від прийнятого в ТАУ тим, що в ММ використовуються не абстрактні сигнали, а величини, що безпосередньо характеризують фізичний стан об'єкта (струми, потенціали, тиски, сили й т.п.) і вони зв'язані компонентними рівняннями. Саме цей підхід стає домінуючим останнім часом для дослідження МС. При цьому основу моделювання МС становить моделювання механічних конструкцій (для цього призначені програми класу MCAD). Саме компоненти механічних конструкцій описуються найбільш складними математичними моделями. Вони найбільше багато-

вимірні й пред'являють найбільш жорсткі вимоги до засобів моделювання [31, 34].

Оскільки кожна мехатронна система має об'єкт керування та пристрій керування, тобто є конкретизованою САК, то система для моделювання повинна володіти широкими можливостями щодо синтезу, аналізу та оптимізації САК. Представниками таких САЕ є *SimApp*, *MATLAB / Simulink*, *VisSim/Altair Embed*, *Dymola*, *20-sim*. Рекомендовано для моделювання мехатронних систем застосовувати програми фізичного мультидоменного моделювання, як найбільш придатних для цих цілей з точки зору представлення моделі (*modeling*) та стійкості й адекватності аналізу поведінки у часі (*simulation*). Найбільш потужним та пристосованим для дослідження САК є *MATLAB / Simulink*, недоліками є неповна підтримка фізичного мультидоменного моделювання, відсутність пробної версії та необхідний набір додатків, що в сумі з батьківським пакетом коштують більше \$15000.

Контрольні запитання

1. Поняття системи, ознаки ДС, МС. МС як вид ДС.
2. Особливості мехатронних систем.
3. Склад мехатронної системи.
4. Функціональна схема мехатронної системи.
5. Розрахунки й проектування МС.
6. Необхідність моделювання мехатронних систем.
7. Особливості сучасного етапу розвитку МС.
8. Поняття синергії.
9. Програми та підходи моделювання МС.
10. Функціональна схема мехатронної системи.

Завдання для практичного заняття/самостійної роботи

1. Представити задану схемою інформаційно-вимірювальної системи як мехатронну систему, визначити її підсистеми, показати інформаційні зв'язки.
2. Поставити задачу / завдання для моделювання ІВС з урахуванням мультидоменності системи.
3. Обрати і описати ІВС із системою контролю як мультидоменну, мехатронну систему, визначити структурні блоки.

ТЕМА №4

МОДЕЛЮВАННЯ САР ТА САК

Перелік питань:

- Структурні елементи САК (самостійно, по курсу ТАУ).
- Макромоделі елементів САК.
- Моделювання автоматизованої системи керування(ІППН) (див. конспект).

4.1 Структурні елементи САК

Модель системи керування можна представити рисунком 4.1.

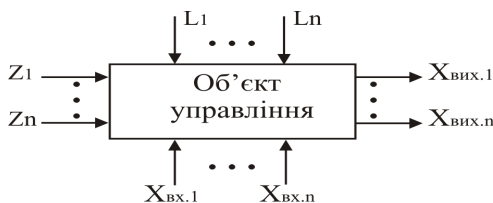


Рисунок 4.1 – Модель системи керування

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

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

Метою автоматичного регулювання та керування 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 керування (САК). Принцип керування 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 САК за відхиленням – на рис.4.2.

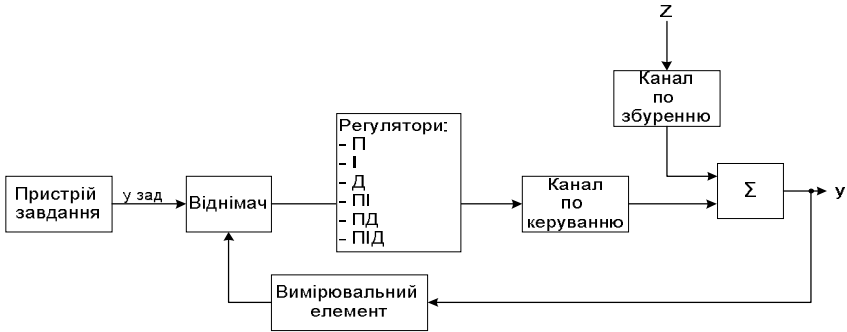


Рисунок 4.2 – Структурна схема системи керування за відхиленням

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

Принцип керування за збуренням полягає в тому, що для зменшення або усунення відхилення керованої величини ($X_{вих}$) від заданого значення ($X_{зад}$), яке спричиняється збуренням, вимірюється це збурення і перетворюється в керуюче діяння ($X_{вх}$), яке компенсує це відхилення. Структура такої САК – на рисунку 4.3.



Рисунок 4.3 – Структурна схема САК з принципом керування за збуренням

Вимірювання збурюючого діяння $L(t)$ здійснюється за допомогою вимірювального елемента (BE), а його перетворення – за допомогою перетворювача ($П$). Вимірювальний елемент і перетворювач утворюють зв'язок за збуренням. Вихідна дія зв'язку за збуренням (XL) 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 керування тут $X_{ex}(t)=f[L(t)]$. Система з принципом керування за збуренням є розімкненою САК: процес керування не залежить від результату.

Структурна схема системи керування за керуючим впливом наведена на рисунку 4.4.

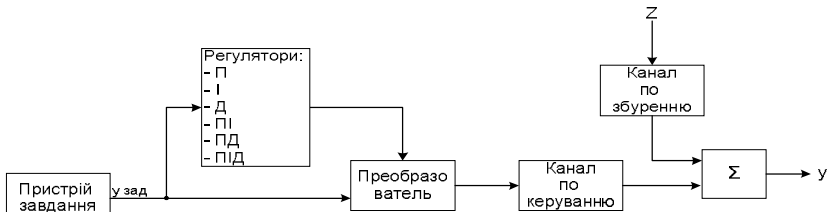


Рисунок 4.4 – Структурна схема системи керування за керуючим впливом

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 та мати моделі підсистем (елементів, блоків) САК/САР.

4.2 Моделі для елементів САК в програмі Micro-SAP

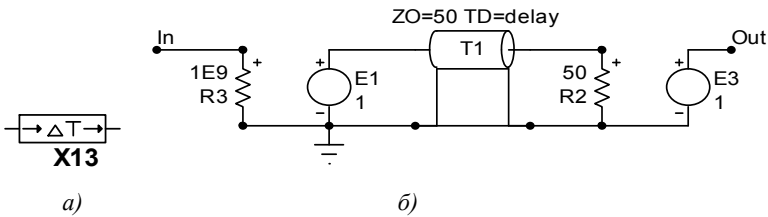
В бібліотеці пакету моделювання Micro-SAP наявні макромоделі пристроїв, які необхідні для побудови функціональної моделі САР/САК, зокрема, операційні багатополіусники, які можуть виконувати операції інтегрування, диференціювання та всі арифметичні дії та чотирихполіусники із складними передатними функціями (див. додаток Г) [12]. Перелічимо деякі з них:

- ланцюг абсолютного значення - *AbsoluteValueMacro*;
- ланцюг підсилення - *Amplifier Macro*;
- ланцюг затримки - *Analog delay Macro*;
- ланцюги суматорів із різним напрямком та кількістю входів,

наприклад, *2-Input Summer Macro*;

- ланцюги для обчислення різниці, 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*.

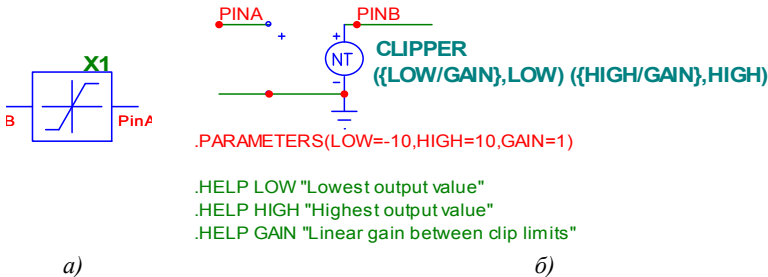
Розглянемо умовні графічні позначки (УГП) та схеми заміщення деяких макромоделей. На рисунку 4.5 наведена макромодел ь елемента, який реалізує затримку сигналу (параметр – затримка в секундах), на рисунку 4.6 – макромодел ь елемента із передатною функцією, вид якої зображено на його УГП.



а – УГП; б – схема заміщення

Рисунок 4.5– Ланцюг Analog delay Macro

CLIP MACRO



а – УГП; б – схема заміщення

Рисунок 4.6– Ланцюг CLIP Macro

4.3 Приклад побудови САК в програмі Micro-Cap

4.3.1 Теоретичні аспекти моделювання САК

При моделюванні процесів автоматичного керування дотримуються схеми, за якою реальний об'єкт складається з двох підсистем, які взаємодіють: керованої та керуючої.

Класична схема керування з одиничним негативним зворотним зв'язком (одноконтурна САК) показана на рисунку 4.7.



Рисунок 4.7 – Одноконтурна САК

Призначення регулятора системи полягає в корекції динамічних властивостей об'єкта керування 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 керування, використовуючи помилку регулювання $\varepsilon(t) = g(t) - y(t)$, 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 керування 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 [35].

Для задовільної роботи системи автоматичного керування 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 її параметри: максимальне відхилення регульованої величини $y(t)$ в перехідному процесі – перерегулювання (відхилення від значення в ustalеному режимі у відсотках, час перехідного процесу (час, за який функція змінює своє значення з 10% до 90% від свого значення в ustalеному режимі).

Висновки стосовно властивостей систем автоматичного керування можна зробити, виходячи з виду 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 структурою.

Безперервні системи описуються звичайними диференціальними рівняннями та диференціальними рівняннями в частинних похідних, а дискретні системи – диференціально-різницевиими рівняннями. Лінійні та нелінійні неперервні системи автоматичного керування 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 керування 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. Частіше за все використовують локальну лінеаризацію моделі системи, що дозволяє при моделюванні переходити до функцій комплексної частоти (через перетворення Лапласа, Фур'є тощо).

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 [35].

4.3.2 Моделювання регулюючих клапанів

Для регулювання швидкості подачі рідин і газів в САР застосовують різні конструкції клапанів, принцип дії яких базується на зміні геометричних розмірів каналу проходження потоку, наприклад електро-механічний моторизований регулятор. Зміна положення штоку такого моторизованого клапана здійснюється системою регулювання, тобто цей пристрій є мехатронною системою.

Синтезуємо структурну схему моделі клапана на базі кінематичної схеми клапана (рисунок 4.8). Спрощено путь передачі механічної енергії по кінематичній схемі регулятора такий: електропривод – редуктор – обертально-поступальна передача – переміщення затвора клапана.

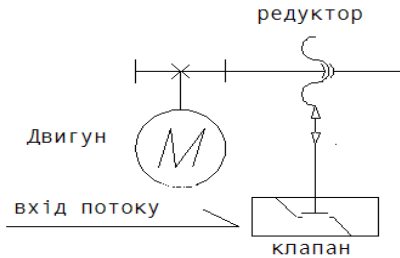


Рисунок 4.8 – Кінематична схема клапана

Контур клапана представлений структурною схемою (рис. 4.9).

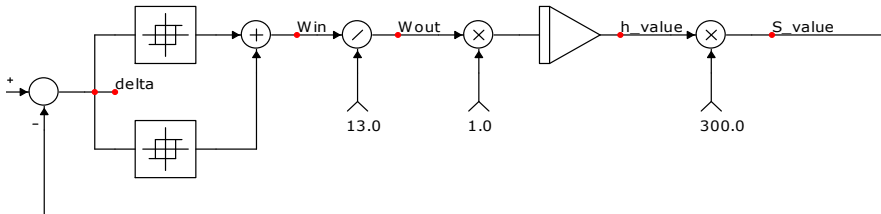


Рисунок 4.9 – Клапан. Структурна схема моделі

Модель клапана описується законами класичної механіки і синтезується виходячи з геометричних розмірів клапана.

Параметризується 3 блоками: редуктор; обертально-поступальна передача; затвор подачі рідини/ пара. Параметри моделі можна отримати з технічної документації, безпосереднім зняттям параметрів з реального устаткування, або їх аналітичним розрахунком. В ході моделінгу можуть бути виявлені нові закономірності, які необхідно врахувати, або такі, впливом яких можна знехтувати. Вхідний сигнал уставки перетину просвіту клапана в мм^2 моделюється програмованим джерелом (напруги).

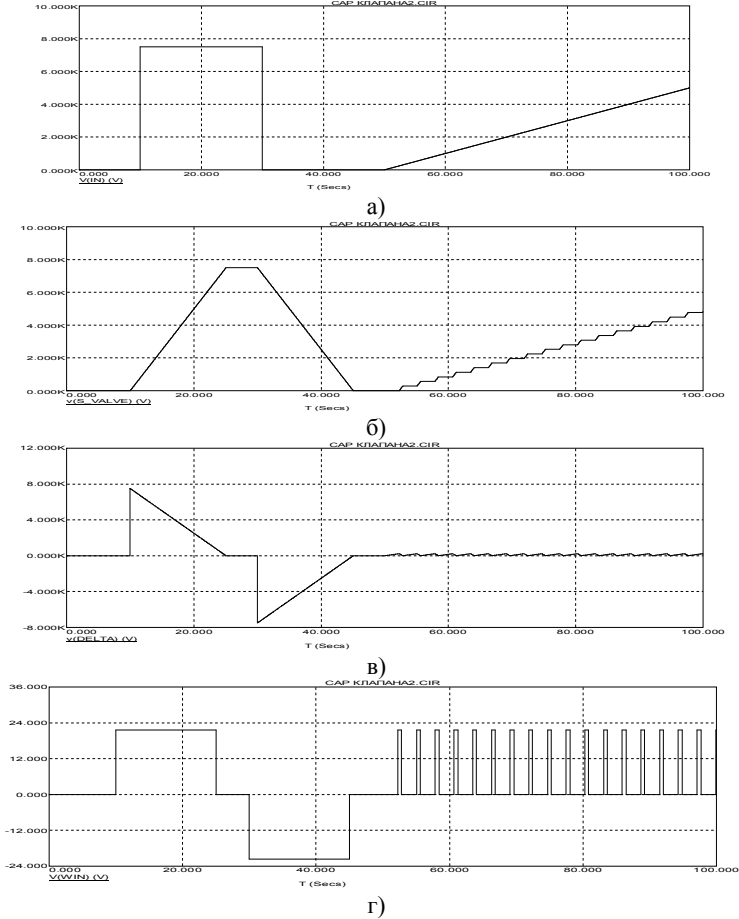
Контур складається з наступних елементарних ланок:

- ланка визначення величини помилки, реалізована на алгебраїчному суматорі із врахуванням знаку доданків (вихід: *delta* – відхилення поточного положення «вхід -» від уставки «вхід +»);
- ланка зони нечутливості (вихід: *Win* – частота обертання ведучої шестерні редуктора), реалізована на двох тригерах Шмітта (ланки гістерезису) із сполученими межами петель - для збереження ресурсу блоку керування електроприводом від «брязкоту» контактів;
- ланка «Редуктор» (блок ділення на коефіцієнт/передаточне число (вихід – *Wout* – частота обертання веденої шестерні редуктора);
- ланка інтегратора, яка моделює обертально-поступальну передачу із передаточним числом 1 мм/об, вихідна величина – поточне положення штока клапана *h_value* (мм);
- ланка множення для розрахунку поточного вікна просвіту клапана *S_value* (мм^2), відповідно до положення штока *h_value*.

Результати моделювання динаміки САР (перехідні процеси) представлені на рисунку 4.10.

На графіку 4.10, а[±] вхідний сигнал уставки перетину просвіту клапана (може мінятися як дискретно, так і безперервно), на рисунку 4.10, б відображена реакція клапана на вхідний вплив (видно роботу ланки зони нечутливості, яка переводить клапан в переривчастий режим роботи при швидкості зміни уставки, меншій за швидкість позиціонування штока клапана- у відсутності цієї ланки на малих швидкостях зміни уставки клапан переходив би у автоколивальний режим).

Рисунок 4.10, в відображає процес визначення помилки між уставкою та поточним значенням просвіту клапана *S_valve* (мм^2), відповідно до положення штока. На рисунку 4.10, г – частота і напрямок обертання електродвигуна переміщення клапана. Видно режим переривчастої роботи двигуна при повільній зміні уставки.



а – уставка; б – стан клапана; в – помилка уставка-поточне значення;
г – швидкість обертання двигуна

Рисунок 4.10 – Перехідні характеристики

4.4 Моделі ІППН в Micro-Cap

Імпульсний перетворювач постійної напруги (ІППН) можна віднести до системи автоматичного регулювання струму, або напруги. В англійській технічній літературі ці пристрої отримали назву імпульс-

них регуляторів (IP), імпульсних стабілізаторів та відносяться до класу *SMPS*, джерел вторинного електроживлення, тобто потужних пристроїв, що працюють в ключовому режимі, саме так і називається підрозділ бібліотеки програми Micro-Cap, де зберігаються макромоделі ІППН. Програми САD мають в своїх бібліотеках набір макромоделей для ІППН різних типів. Основний розподіл макромоделей – по типу перетворювача, роботу якого вони мають моделювати.

Крім макромоделей основних ІППН різного роду, наприклад, знижувального та підвищувального типів (Buck, Boost, Buck-Boost), до бібліотеки занесено макромоделі й інших перетворювачів (Додаток Д) [36, 37].

В Micro-Cap існує два способи моделювання ІППН: за допомогою *Switch* (ключових) та *Averaged* (усереднених) макромоделей, на належність до певного типу вказує префікс *A_* або *S_* у назвах файлів перетворювачів. Назви файлів мають інформативний характер та відображають властивості макромоделі згідно наведеної класифікації: наприклад, файл *A_BUCK_VM* вміщує усереднену макромодель перетворювача знижувального типу, в режимі контролю вихідної напруги.

Switch – макромоделі створені із урахуванням усіх нелінійностей, які властиві транзисторам, вентилям, трансформаторам, дроселям; їхнє використання обмежується режимами розрахунку перехідних характеристик. Однак, при редагуванні бібліотечних моделей та подальшому моделюванні перехідних характеристик за допомогою ключових макромоделей можливі проблеми незбіжності, що є слідством поганої обумовленості/жорсткості моделі.

В *Averaged*–макромоделях немає ключових елементів, функціонування пристрою в режимі малого сигналу описано квазістатичним нелінійним рівнянням із параметричними коефіцієнтами, які залежать від коефіцієнту заповнення. Ці економічні макромоделі набули найбільшого поширення, їх використовують для дослідження IP в частотному діапазоні. В робочих папках САD-програм вкладено готові для аналізу схеми: в них є навантаження, джерела живлення, контролер й перетворювач, представлений в *Averaged* моделях «чорним ящиком», схеми яких можна проглянути при переході в інформаційний режим. Існує також поділ за режимом контролю: за струмом (*current mode* – CM), або по напрузі (*voltage mode* – VM).

У моделях перетворювачів центральне місце займає інкапсульована макромодель системи керування – ядро, де керування здійсню-

ється шляхом зміни коефіцієнта заповнення, так звана «широтно-імпульсна» модуляція – PWM. У ключових макромоделях кількість виводів СК дорівнює кількості виводів у реальних мікросхемах, сама модель СК набагато складніша, ніж в усереднених макромоделях. Як правило, при проектуванні ІППН спочатку здійснюється аналіз за допомогою усереднених макромоделей, а потім – за допомогою ключових.

4.4.1 Моделі ШІМ-контролерів для дослідження САК в Spice-сумісних програмах ECAD

Контролер із модуляцією ширини імпульсу (ШІМ) вмикає та вимикає живлення, щоб контролювати рівень напруги або струму в навантаженні. Ця техніка модуляції виробляє серію імпульсів, робочий цикл яких залежить від опорного сигналу. ШІМ є ключовим принципом регулювання в багатьох пристроях: джерела безперебійного живлення (*UPS*), джерелах живлення змінного струму (*SMPS*), імпульсних перетворювачах постійної напруги (ІППН), регуляторах частоти (перетворювачах змінного струму), автоматизованих системах керування (АСК) напруги /струму або швидкості та інших САР та САК. В якості модулюючого сигналу зазвичай використовується пилоподібний сигнал.

Для дослідження САК та мехатронних систем на високому рівні абстракції можна використовувати програми ECAD [34]. Як правило, моделі для ШІМ-контролерів представлені в бібліотеках SPICE-сумісних програм призначені для спеціалізованих контролерів, або є невід’ємною частиною моделей імпульсних перетворювачів постійної напруги (ІППН).

Наприклад, для мікросхеми 100 В ШІМ-контролеру LM50201 для режиму контролю вихідного струму (*Current Mode PWM Control*), для мікросхеми інтегрованого мережевого інтерфейсу *Ethernet PD* та контролера ШІМ (LM5070), для регулятора поточного режиму вторинних джерел живлення, універсальних бездротових пристроїв малої потужності (NCP1200 / ON) тощо [31]. Ці моделі отримані як комбінація елементів принципової схеми інтегральної мікросхеми та поведінкових елементів, написані на спеціалізованій мові моделювання (*Spice*).

Моделі ШІМ розвивалися головним чином як компонент перетворювачів постійного струму та вони як правило є адекватними у режимі малого сигналу (*Small Signal Mode*). В Micro-Cap12 [12] є макромодель генератора ШІМ-сигналу, яка є складовою частиною ППІН перетворювача (файл DSP_SAMPLE circuit). Макромодель ШІМ-генератора має шість параметрів: *ModHigh*, *ModLow*, *ModFreq*, *ModType*, *PWMHigh* та *PWMLow*. *ModHigh* і *ModLow* визначають високий та низький рівні напруги модулюючого сигналу. Параметр *ModFreq* задає частоту модулюючого сигналу. *ModType* визначає форму модулюючого сигналу: пилкоподібний із заднім фронтом (*trailing edge sawtooth*), пилкоподібний із переднім фронтом (*leading edge sawtooth*), або трикутний сигнал (*triangle waveform*). Параметри *PWMHigh* та *PWMLow* задають значення високого та низького рівнів вихідного ШІМ-сигналу. Змінна перетворює параметр *ModFreq* в еквівалентний період (вимірюється в секундах). Внутрішній параметр *ModTrans* використовується для задання параметрів динамічного аналізу (*transition time*) для обраного типу модулюючого сигналу. Він задає transition time як *ModPeriod* поділений на 10000. Джерело напруги *ModRamp* генерує модулюючий сигнал (*sawtooth signal*). Поведінковий елемент (*Behavioral element*) – вихідне джерело напруги, генерує вихідний ШІМ сигнал V_{out} . Це функціональне джерело формує сигнал відповідно до рівняння:

$$V_{out} = V_o + V_a \cdot \tanh(1 \cdot 10^5 \cdot V_{IN,Ramp}). \quad (4.1)$$

Гіперболічний тангенс у виразі приймає значення від 1 до -1. Коефіцієнт $1 \cdot 10^5$ забезпечує високий коефіцієнт підсилення, який мінімізує час переходу між двома цими значеннями. Коли диференціальна напруга між вузлами *In* та *Ramp* $V_{IN,Ramp}$ є позитивною, функція *tanh* повертає 1. Коли диференціальна напруга є негативною, функція *tanh* повертає -1. Змінні V_o і V_a встановлюються за двома наступними командними операторами (синтаксис мови *Spice*):

```
.define Va (PWMHigh-PWMLow)/2;
.define Vo (PWMHigh+PWMLow)/2.
```

Ці дві змінні визначають рівні вихідної напруги імпульсного сигналу (високе та низьке значення *PWMHigh* та *PWMLow* відповідно).

Ця модель досить проста, заснована на елементі поведінки, але розрахунок гіперболічного тангенсу може викликати проблеми збіжності обчислювального алгоритму.

Ще одна економічна універсальна модель ШІМ контролера показана на рисунку 4.11 [38, 39]. Ця модель схожа на модель, описану вище, але заснована на більш простому алгоритмі для визначення вихідного сигналу і має втричі менший набір вхідних параметрів. Назва вхідного вузла для опорного сигналу – IN. У цій схемі генератор V1 імітує змінний вхідний / опорний сигнал. OUT – вихідний вузол для сигналу ШІМ. Сигнал модуляції генерується генератором пилкоподібного сигналу у межах макросу (передній фронт) багатопільовим джерелом напруги V2.

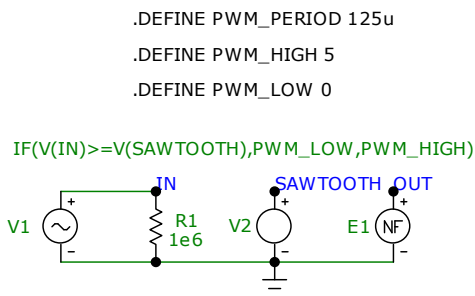


Рисунок 4.11 – Економічна модель ШІМ-контролера

Макрос має три внутрішні (локальні) параметри: *PWM_High* та *PWM_Low*, які визначають значення високої та низької напруги вихідного ШІМ сигналу та *PWM_Period*, що встановлює період Sawtooth сигналу. У цьому прикладі частота модулюючого сигналу становить 8 кГц. Резистор R1 (опір 1 МОм) забезпечує шлях постійного струму до землі від вхідного вузла.

Поведінка вихідного сигналу запрограмована функціональним джерелом E1. Воно емулює функцію компаратора та генерує вихідний імпульс. Відбувається порівняння вхідних і модулюючого сигналів: якщо значення вхідного / опорного сигналу більше за поточне значення пилкоподібного сигналу, то значення вихідного напруги прийма-

ється низьким; та навпаки, якщо вхідний сигнал менший за пилкоподібний сигнал, на виході приймається високе значення напруги.

Цей алгоритм прописаний в умовному операторі:

```
IF V(IN) ≥ V(SAWTOOTH)
THEN PWM_Low
ELSE PWM_High
```

Значення PWM_High та PWM_Low рівнів вихідної напруги наводяться у відповідних операторах «define». Ці командні оператори використовуються для створення та визначення значень символьних змінних. У мові Spice умовний оператор має конструкцію IF (b, z1, z2): якщо b є істиною, функція повертає z1, інакше вона повертає z2).

Результат моделювання відповідних сигналів модуляції наведено нижче на рис. 4.12. На верхньому графіку відображається пилкоподібний сигнал та змінний вхідний сигнал. Нижній графік показує згенерований ШІМ-сигнал.

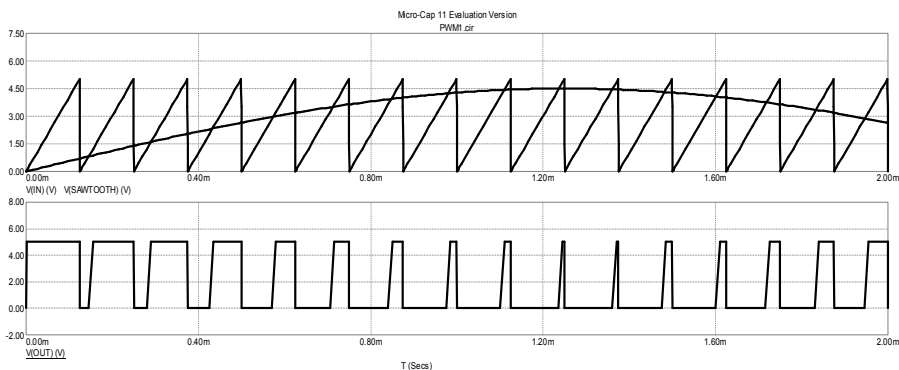


Рисунок 4.12 – Моделювання блока ШІМ в Micro-Cap12

Ще одна модель являє собою спрощену версію, в якій компенсуючий сигнал виробляється як постійний безперервний сигнал з варіацією його амплітуди (сигнал ступінчастої форми). Це припущення прийнятне за умови, якщо вихідний сигнал ефективно згладжує фільтр. Оскільки сигнал ШІМ регулює середнє значення напруги шляхом зміни коефіцієнта заповнення/шпаруватості (*duty cycle (D.C.)*), можна безпосередньо використовувати варіацію середнього / серед-

нього значення під час симуляції подібних систем із ШІМ-регульованим і згладженим вихідним сигналом.

Якщо $D.C.$ змінюється лінійно, ми можемо представити зв'язок параметрів сигналу із усередненим значенням напруги V_{AVG} :

$$V_{AVG} = V_m \cdot D.C. = V_m \cdot \frac{\tau}{T},$$

де V_{AVG} – амплітуда вихідної напруги;

T – період;

τ – тривалість імпульсу.

Ця модель являє собою пропорційний інтеграційний блок [35]. В обох моделях використовується каузальний підхід [40].

4.5 Проектувальник регуляторів та редактор лінійних систем в 20-sim

Редактор фільтрів і проектувальник регуляторів мають важливе значення при створенні САК та їх аналізі; вони мають зв'язок з редактором лінійних систем (*Linear System Editor*, LSE), який у свою чергу дає можливість синтезувати динамічні ланки чотирма способами: за допомогою простору станів (*State Space*), передавальної функції (*Transfer Function*), нулів-полосів 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.

Фактично, проектувальник регуляторів (CDE) має інтерфейс і всі можливості LSE, а додатково ще опції «*Sub System*» та «*Additional Ports*» (інтерфейс - на рисунку 4.11). LSE безпосередньо дозволяє аналізувати і досліджувати розроблену САК, використовуючи вбудовану можливість відтворення діаграм Бode, Найквіста, Нікольса та відображення карти нулів-полосів і h -функції (*Step Response*).

Редактор фільтрів дозволяє створювати різні типи фільтрів (Бесселя, 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.

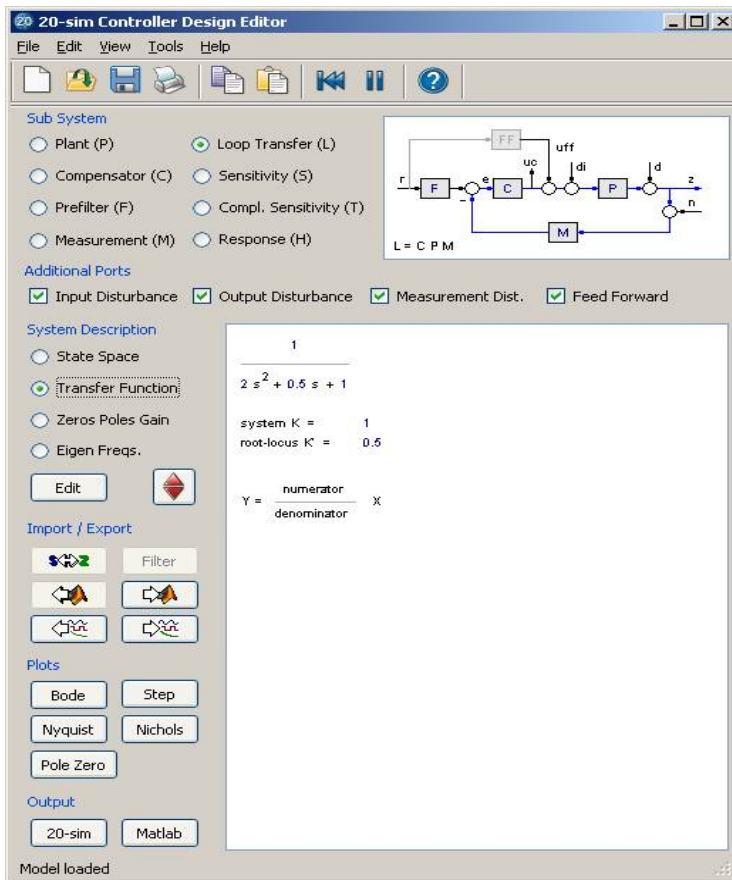


Рисунок 4.11 – Головне вікно CDE

Рисунки 4.12, 4.13, 4.14 показують опції редактора фільтрів для деяких типів регуляторів або фільтрів.

Наприклад, для опису універсального режекторного фільтру другого порядку, що складається з одного нуля та полюса (рис.4.12), застосовується наступні параметри:

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

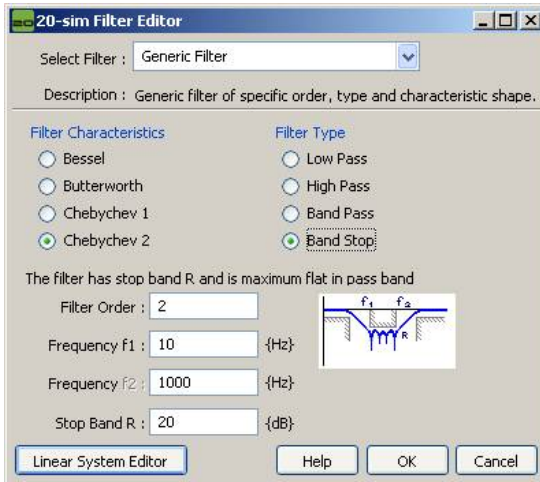


Рисунок 4.12 – Параметри фільтрів загального шаблону

Вікно для параметризації ПД наведено на рисунку 4.13; 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).

ПД-компенсатор (рис. 4.14) 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 ПД-компенсатора, знайденої в [4], можна сказати, що використовується

послідовне з'єднання внутрішніх елементів (П-, І-, Д- частин та фільтра).

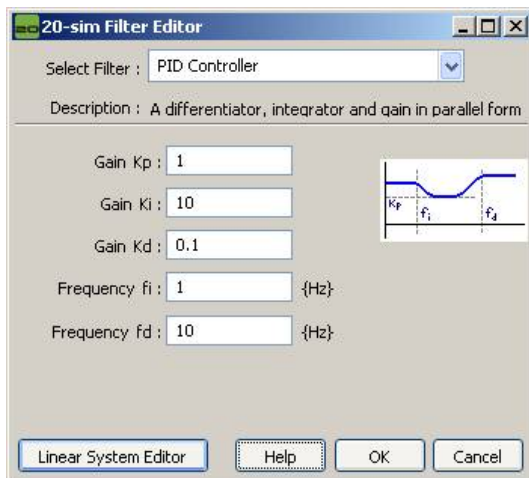


Рисунок 4.13 – Опис паралельного ПІД-регулятора

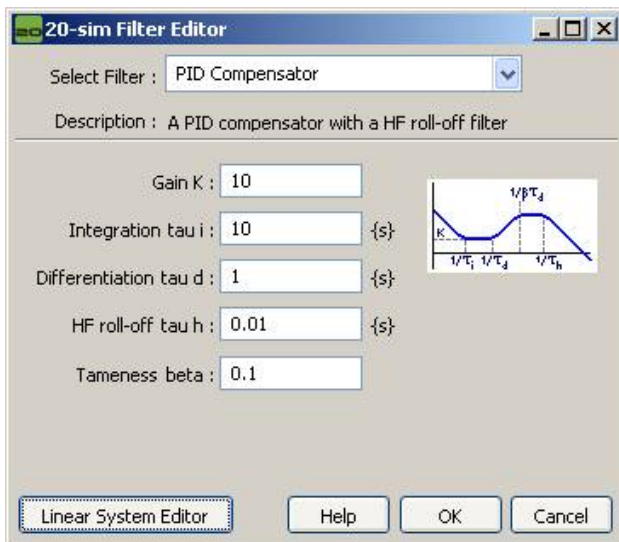


Рисунок 4.14 – Опис ПІД-компенсатора у редакторі фільтрів

Редактор фільтрів пропонує для синтезу декілька типів ПД-регуляторів з різним набором вхідних параметрів. Проєктувальник регуляторів (CDE) дозволяє синтезувати повну структурно завершену САК з контролем проміжних сигналів, використовуючи будь-яку форму опису, представлену в LSE.

Структурно повна система в CDE представлена на рисунку 4.15.

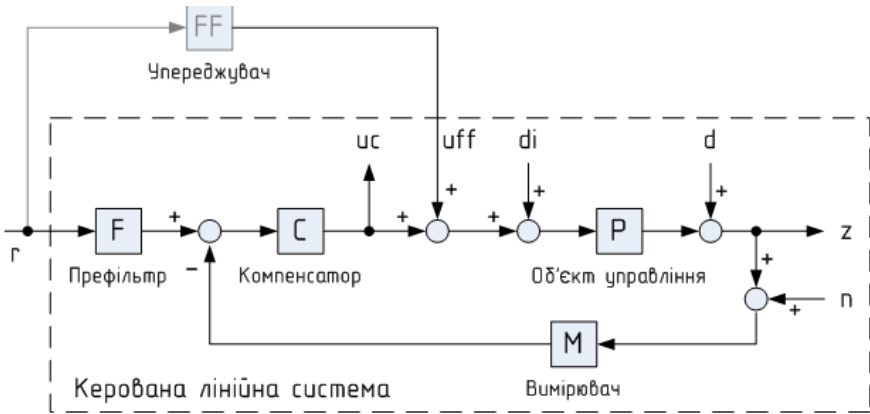


Рисунок 4.15 – Структура САК, що використовується в CDE

Така система складається з об'єкта керування P (Plant), регулятора або компенсатора C (Compensator), префільтра F (Prefilter), вимірювача M (Measurement), упереджувача FF (Feed Forward), та ряду вхідних/вихідних сигналів: r – вхідний вплив на систему або завдання, uc – чистий вихідний сигнал компенсатора, uff – вхідний сигнал від упереджувача, di – вхідне збурення об'єкта керування (input disturbance), d – вихідне збурення об'єкта керування (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 (Complex 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 керування 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 САЕ, розглянуті вище (окрім, *SamSim*), дозволяють працювати з ланкою чистого запізнення тільки в часовій області, але не можуть відпрацьовувати її у частотному домені. Приклад частотних характеристик моделі з вихідною затримкою, розробленою в редакторі лінійних систем 20-sim, приведено на рисунку 4.16.

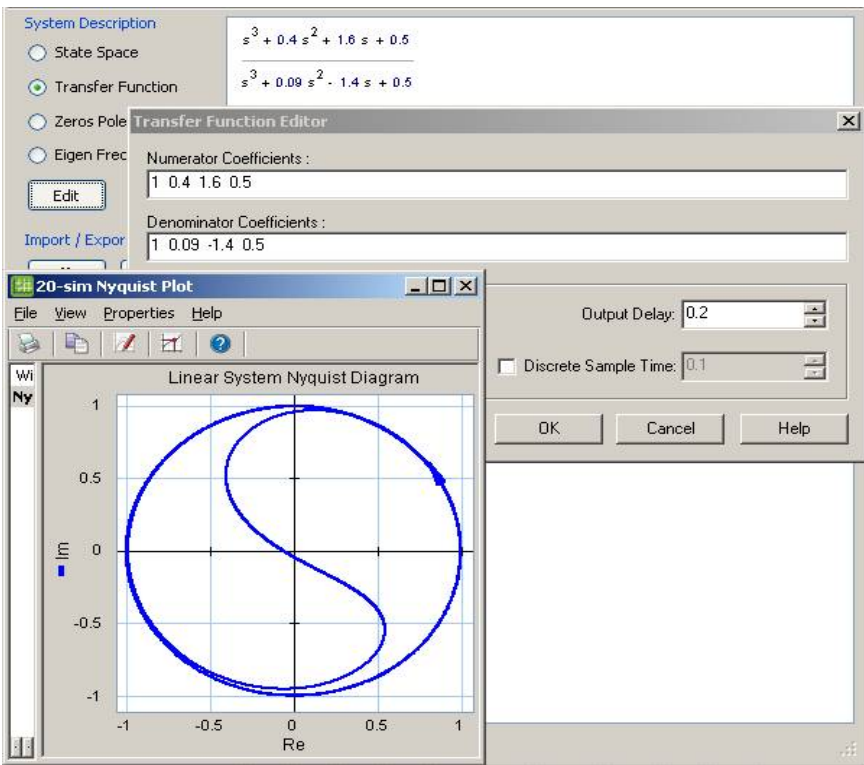


Рисунок 4.16 – ЧХ системи з вихідною затримкою (0,2с), зробленою в LSE

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).

4.6 Методика моделювання САК

Для дослідження САК 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 САК наведено в додатку Е. 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.

Треба відмітити, що задача вибору методу симуляції (алгоритму моделювання) дуже 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 САК (є необхідність у знаходженні оптимальних коефіцієнтів регуляторів при налагодженні САК; дослідження алгоритмів оптимізації розглядається в [49]).

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 [9], яка представлена у вигляді алгоритму в Додатку Ж.

Контрольні запитання

1. Класифікація та принципи дослідження САК.
2. Макроси з бібліотеки програми Micro-Cap, які використовуються для моделінгу САК.
3. Визначити критерії, за якими макроси, які використовуються для моделінгу САК можуть бути класифіковані.
4. Проектування регуляторів.
5. Методика аналізу САК.
6. Проектування фільтрів для САК.
7. Редактор лінійних систем в 20-sim.
8. Структура САК в редакторі CDE 20-sim.
9. Як буде виглядати структура САК замкненого типу?
10. Методика підвищення адекватності моделювання САК.

Завдання для практичного заняття/самостійної роботи

1. Класифікувати макромоделі блоків САК з бібліотеки Micro-Cap, згідно їхнього функціоналу.
2. Отримати частотні та динамічні характеристики обраного типу ІППН, визначити вихідні (вторинні) параметри.
3. Виписати модель ШІМ-контролера (PWM) на мові Spice, визначити структуру моделі, відновити її схему заміщення.
4. Дослідити моделі ІППН в програмі Micro-Cap (тип ІППН задає викладач).
5. Представити ІППН як САК, визначити його структурні елементи згідно рис.4.15.
6. Представити задану схему ІВС/САК в термінах структури САК (рис.4.15), що використовується в CDE, визначити всі змінні ІВС/САК для CDE.
7. Використовуючи програму 20-sim, синтезувати САК в CDE.
8. Використовуючи програму 20-sim, синтезувати САК в LSE.
9. Порівняти результати, зробити висновки про адекватність, точність.

ТЕМА №5

ОСНОВИ СИМУЛЯЦІЇ СИСТЕМ

Перелік питань:

- Графічний інтерфейс.
- Інструментальні засоби моделювання
- Основні поняття алгоритмів моделювання (*simulation*).

5.1 Перетворення ММ, заданої в графічному інтерфейсі

Графічний інтерфейс є в даний час стандартним і важливим компонентом сучасної програми моделювання. Він створює дружній інтерфейс між користувачем і програмою, дає можливість оперувати з графічними образами замість аналітичних виразів на етапі моделінгу що полегшує роботу і знижує ймовірність помилок при введенні інформації про систему. Графічний інтерфейс дозволяє вводити інформацію про досліджувану систему шляхом «малювання» на екрані монітора проектованої схеми у вигляді, зрозумілому широкому колу фахівців (див. п.1.1). Однак, цим роль графічного інтерфейсу не обмежується, завданнями графічного інтерфейсу, крім того, можуть бути:

- контроль за дотриманням деяких правил в процесі створення графічного зображення на екрані монітора (можуть накладатися обмеження на способи з'єднання компонентів і т.п.);

- перетворення інформації про схему в команди для симулятора;

- контроль процесу моделювання, візуалізація результатів.

Після побудови в головному редакторі модель підлягає компіляції з метою перетворення її графічного зображення (на високому рівні) та програмно-математичного опису (на низькому рівні) у спеціальний код симуляції. Компіляцію можна трактувати як перехід моделі від етапу «*modeling*» до етапу «*simulation*»; ця важлива процедура зумовлює ефективність згенерованого машинного коду, який буде використовуватися у симуляторі. Процедура компіляції складається з декількох етапів та кроків, які представлені в Додатку 3.

На першому етапі програма перевіряє модель на помилки: виконується пошук некоректного перетворення типів, що може призвести до втрати точності (наприклад, величина типу «*real*» підлягає перетворенню в величину типу «*integer*»), а також перевірка належного

використання величин, їх одиниць вимірювання. На другому етапі відбувається оптимізація структури математичної моделі, яка включає шість кроків. Спочатку відбувається аналіз каузальності, тобто визначається причинний порядок рівнянь і змінних; приклад на рисунку 5.1.

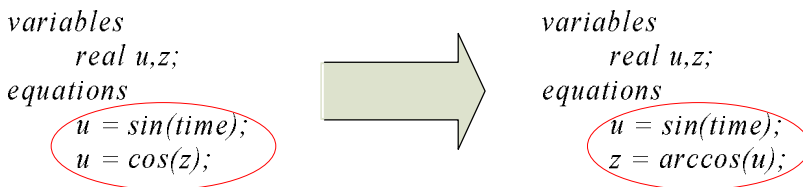


Рисунок 5.1 – Переформування рівнянь після аналізу каузальності

Потім рівняння з диференціальної форми трансформуються до інтегральної, наскільки це можливо. На третьому кроці диференціальні рівняння, що залишилися, розв'язуються безпосередньо (без перетворень), якщо це можливо. Четвертий крок характеризується процедурою вирішення проблеми одного з видів топологічних вироджень: виконується розв'язання алгебраїчних петель (*algebraic loops*) аналітично. На п'ятому кроці відбувається оптимізація структури рівнянь: деякі рівняння повинні бути обчислені лише один раз протягом кожного кроку симуляції, бо вони не впливають на динаміку моделі; в залежності від їх відношення стосовно динаміки моделі, ці рівняння можуть бути розраховані перед (вхідні рівняння) або після (вихідні рівняння) обчислення динаміки. На останньому кроці етапу оптимізації структури ММ проходить аналіз надлишковості. Багато модельних рівнянь просто здійснюють переприсвоєння різних змінних між собою, де значення деяких змінних відомо.

Етап компіляції коду складається з трьох кроків. По-перше, всі рівняння розбиваються до найпростішої форми і переписуються у правильний порядок виконання (*execution*). По-друге, з рівнянь створюється низькорівневий код інтерпретатора, що розуміється алгоритмами симуляції. По-третє, якщо включена опція вбудованого компілятора, то код інтерпретатора компілюється в специфічний машинний код, який з максимальною ефективністю використовує рідні (*native*) команди процесора.

5.2 Інструментальні засоби моделювання

Основні обчислювальні операції здійснюються блоками, що входять в інструментальні засоби (математичне ядро) САЕ. Спектр завдань, які вирішують програми математичного ядра, може бути досить широкий, від аналізу тільки лінійних моделей, до символічних обчислень та інтервальної арифметики [41]. Для пакетів, здатних повноцінно досліджувати ДС, основною частиною є колекції програм, що містять реалізації чисельних методів.

Сучасні універсальні системи моделювання містять як засоби вирішення статичних (аналітичних) завдань, так і потужні засоби розрахунку і аналізу динамічних процесів. Так велика частина завдань аналізу лінійних систем (розрахунок перехідних характеристик, отримання і перетворення типових ММ, побудова частотних характеристик, аналіз розміщення полюсів і нулів на комплексній площині і т.п.), не говорячи вже про задачі синтезу, вирішується програмами, що входять в блок аналітичних програм. Дана частина математичного ядра в значній мірі прив'язана до специфіки об'єкта проектування, його фізичним і інформаційним особливостям функціонування, а також до конкретних ієрархічних рівнів проектування.

Іншу частину математичного ядра складають програми, призначені для аналізу перехідних процесів у складних нелінійних системах, де застосування аналітичних методів неможливо. Ця частина є інваріантною до особливостей об'єкта моделювання, включає в себе методи і алгоритми, слабо пов'язані з особливостями ММ і використовуювані на багатьох ієрархічних рівнях. Найбільш важливими є програми числового інтегрування.

Числове інтегрування (те, що в іноземній літературі розуміється як «*simulation*») виконується спеціальною програмою – керуючою програмою моделювання або коротко модельатором (розв'язувачем, солвером). На модельатор зазвичай покладаються такі функції [6]:

- а) установка початкових умов і значень сигналів;
- б) керування модельним часом за принципом Δt або Δz , а при моделюванні, або використанні одночасно обох принципів;
- в) взаємодія з користувачем по ходу модельного експерименту, а також видача запитів і діагностичних повідомлень.

Для виведення, спостереження та обробки результатів моделювання використовується постпроцесор.

5.3 Основні поняття алгоритмів моделювання (simulation)

Всі реальні об'єкти функціонують у часі і сам хід часу необхідно моделювати так само, як і зміни всіх інших змінних. Процес моделювання являє собою взаємодію видів часу:

- а) реального часу, до моментів якого прив'язані події, що відбуваються в системі, що моделюється;
- б) модельного часу, який відраховується програмою моделювання і є моделлю реального часу;
- в) машинного часу, в якому функціонує апаратна частина системи моделювання.

Будь-який процес у ДС можна розглядати як зміну її стану, яка може відбуватися більш-менш рівномірно, або у формі різких змін, пов'язаних з появою подій. При комп'ютерному моделюванні модельний час може змінюватися дискретно, з деяким кроком h , особливість модельного часу в тому, що їм можна управляти. Необхідно узгодити процес вибору кроку і процес просування модельного часу з особливостями процесів в реальному об'єкті.

Основні принципи регламентації подій:

- принцип Δt ;
- принцип особливих станів Δz ;
- принцип послідовної проводки заявок;
- принцип паралельної роботи об'єктів (об'єктний принцип).

Найбільш часто застосовуються два способи просування модельного часу: «принцип Δt » та «принцип Δz ». Ілюстрація цих способів представлена на рисунку 5.2.

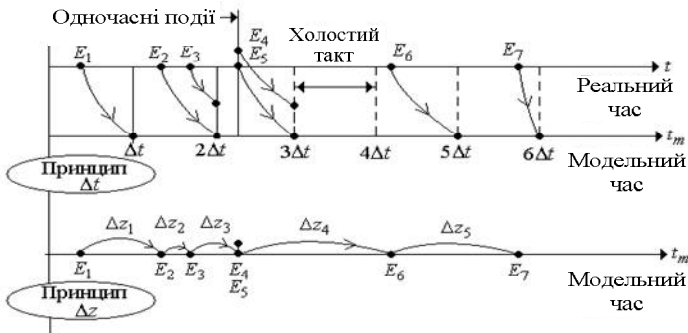


Рисунок 5.2 – Механізми просування модельного часу

Принцип побудови моделюючого алгоритму, що дозволяє визначати послідовні стани складної системи через деякі інтервали часу, називають «принципом Δt ». Він є універсальним (фактично, він використовується в EСAD при моделюванні аналогових електронних пристроїв, але сам крок автоматично варіюється в залежності від крутизни характеристики), але його використання призводить до збільшення часу моделювання при дослідженні аналогово-цифрових пристроїв.

За принципом Δt модельний час плине малими кроками Δt та може приймати тільки дискретні значення, кратні цьому часовому інтервалу. Величина кроку пов'язана із динамічними особливостями модельованого об'єкта. Вона може змінюватися в процесі моделювання, проте безпосередньо не прив'язана до подій, що відбуваються в модельованій системі, що призводить до того, що події можуть потрапити всередину часового кроку, а в результаті, можуть порушуватися причинно-наслідкові зв'язки між подіями. При малих значеннях часового кроку це не суттєво, в іншому разі – може привести до помилок моделювання.

Метод Δt доцільно використовувати якщо:

- моделюється безперервна система, процеси в якій являють собою безперервний ланцюг рівнозначних подій;
- в модельованій системі моменти появи подій обумовлені виконанням деяких умов, пов'язаних зі значеннями змінних системи, в результаті чого їх неможливо заздалегідь визначити.

Для МС реалізація принципу Δt є основним способом просування модельного часу, так як основу МС складають механічні пристрої, що володіють безперервною динамікою.

Принцип Δz (принцип особливих станів), припускає, що просування модельного часу обумовлено подіями, що відбуваються в модельованій системі. Як і в першому випадку, модельний час змінюється дискретно на величину Δz , однак ця величина прив'язана не до динамічних характеристик об'єкта, а являє собою часовий інтервал між послідовними подіями в системі. Величина Δz може мати довільну величину, в тому числі бути рівною нулю, якщо інтервал між подіями зневажливо малий. Моделювання за принципом Δz доцільно використовувати, якщо модельована МС є принципово дискретною, процеси в системі являють собою ланцюг подій, які розподілені в часі нерівномірно, або інтервали між ними великі.

Для МС досить характерним є режим, коли обробка моделі повинна бути пов'язана з роботою реального обладнання, тоді має місце моделювання в «режимі реального часу» (РРЧ). Режим реального часу – режим обробки даних, при якому забезпечується взаємодія обчислювальної системи з зовнішніми по відношенню до неї процесами у темпі, порівняному з швидкістю протікання цих процесів. Моделювання в РРЧ актуально при напівнатурному моделюванні і, особливо, при використанні моделей в контурі керування реальними технічними системами.

Реальні мехатронні об'єкти й МС описуються, як правило, системами нелінійних алгебраїчних і диференціальних рівнянь. Для більшості задач, що представляють практичний інтерес, розв'язок їх аналітичними методами неможливий. Результати можуть бути отримані шляхом побудови наближених розв'язків за допомогою чисельних методів інтегрування, зокрема кінцево-різницевих методів. Загальна ідея числового інтегрування ODE полягає в тому, що проводиться дискретизація незалежної змінної – часу t на інтервалі $[t_0, t_n]$ й заміна її рядом значень $[t_0, t_1, \dots, t_n]$ (принцип Δt). Відстань між двома сусідніми значеннями $h_i = t_i - t_{i-1}$ називається кроком інтегрування. В окремому випадку він може бути постійним на всьому заданому інтервалі зміни змінної t . У результаті, системі диференціальних рівнянь тим або іншим способом ставиться у відповідність система кінцево-різницевих рівнянь. Процедура інтегрування припускає розв'язок отриманої системи кінцево-різницевих рівнянь для фіксованих моментів часу $[t_0, t_1, \dots, t_n]$, починаючи з моменту t_0 , для якого визначений початковий стан досліджуваної системи x_0 . Відповідно, розв'язок виходить у вигляді сукупності значень $[x_0, x_1, \dots, x_n]$ для заданих моментів часу. У теорії чисельних методів розроблена велика кількість різних методів інтегрування, кожному з яких відповідає своя система кінцево-різницевих рівнянь. Ці методи можна розділити на основі наступної класифікації:

- а) методи явні і неявні;
- б) методи однокрокові і багатокрокові;
- в) методи першого, другого і т.д. порядків;
- г) методи з постійним / автоматичним вибором кроку.

Основними вимогами, що пред'являються до чисельних методів, є достатня точність і стійкість. Додатковими – універсальність, алгоритмічна надійність, помірні витрати машинного часу й оперативної пам'яті ЕОМ [42]. Слід враховувати, що практично всі ці характеристики мають сенс тільки стосовно до конкретного об'єкта дослідження, тому вибір відповідного методу інтегрування дуже важливий з точки зору ефективності дослідження.

Аналіз процесу функціонування технічного об'єкта чисельними методами завжди супроводжується помилками, які обумовлені неадекватністю моделі, наближеністю вихідних даних, властивостями використовованого методу інтегрування. Перші з цих факторів виникають на етапі отримання вихідної моделі. Останні – залежать від обраного методу числового інтегрування. Точність методу можна оцінити, проаналізувавши повну помилку на кожному кроці інтегрування, проте завдання це досить складне, так як припускає наявність точного рішення задачі. Повна помилка інтегрування на i -тому кроці включає наступні складові:

- помилка дискретизації, обумовлена заміною похідних кінцевими різницями;
- помилка обчислень, пов'язана з округленням чисел;
- помилка накопичення, що виникає внаслідок помилок на попередніх кроках інтегрування.

Помилка накопичення на j -тому кроці дорівнює повній помилці на попередньому кроці. Її оцінка пов'язана із дослідженням стійкості числового методу. Якщо метод стійкий, то помилка накопичення суттєво не зростає і загальну помилку інтегрування можна зв'язати в основному з помилкою дискретизації. Якщо навіть при невеликих погрешностях апроксимації при кожному кроці інтегрування h накопичена похибка зростає від кроку до кроку, то метод нестійкий і результати досліджень не вірні. Основними джерелами нестійкості процесу інтегрування є неузгодженість вибору методу інтегрування та методу керування кроком інтегрування з характером розв'язуваної задачі, з особливостями досліджуваної системи диференціальних рівнянь. Один і той же метод може бути досить ефективний при вирішенні однієї задачі, і неприйнятний для іншої. Стійкість визначається вибором формули інтегрування, кроком інтегрування і власними значеннями матриці Якобі системи диференціальних рівнянь.

Помилка дискретизації будь-якого стійкого методу прагне до нуля при $h \rightarrow 0$. Отже, за рахунок зменшення кроку можна зробити її якою завгодно малою. Однак, чим менше крок, тим більше буде потрібно кроків для вирішення конкретного завдання, тим більше позначаться на отриманому рішенні помилки округлення. На практиці, за рахунок обмеженої розрядності машинних слів, завжди існує величина кроку h_0 , менше якої внесок помилок округлення починає домінувати в сумарній помилці, що відображено на рисунку 5.3.

При малих значеннях кроку інтегрування мають великий вплив помилки округлення. У середній частині діапазону помилка зростає приблизно пропорційно кроку інтегрування (що відповідає методу першого порядку). Перевищення кроком значення $h_{кр}$ призводить до нестійкості процедури. Значення h_{max} та h_{min} можуть бути знайдені виходячи із заданої точності інтегрування.

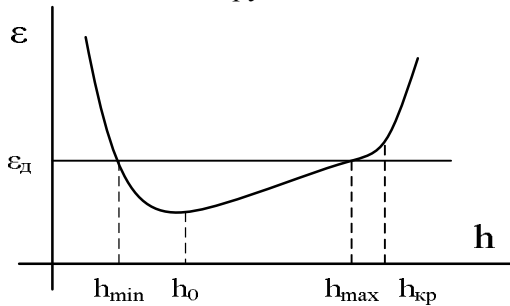


Рисунок 5.3 – Залежність помилки інтегрування від кроку інтегрування

Екстремальну величину кроку h_0 дуже важко встановити наперед, але в задачах, де не потрібна занадто висока точність, необхідний крок зазвичай буде значно більше, ніж цей мінімум, і основний внесок в повну помилку буде вносити помилка дискретизації.

5.4 Класифікація та аналіз існуючих методів інтегрування

Широке поширення отримали класичні методи інтегрування безперервних ДС, які можна розділити на групи за різними критеріями. При цьому кожен метод характеризується:

- величиною кроку (h) за часом: $h_n = t_{n+1} - t_n$ (та його адаптивністю);

- порядком точності ($O(h^q)$, q – порядок методу) при виконанні повного обсягу розрахунків на кожному кроці;
- стійкістю (пов'язана з величиною кроку; дуже важливий показник, що зумовлює застосовність алгоритму до тієї чи іншої задачі);
- кількістю врахованих результатів попередніх кроків на поточному кроці.

Класифікація основних методів інтегрування представлена в додатку I, порівняльні характеристики – в таблиці 5.1.

Таблиця 5.1 – Характеристика методів інтегрування

| Метод симуляції | Порядок методу | Явний (Я)/ Неявний (Н) | Одно- (О)/ Багато- (Б) кроковий | Для жорстких (Ж) чи м'яких (М) моделей | Для IDE | Для DAE | Змінний (З)/ Постійний (П) крок |
|-----------------|----------------|------------------------|---------------------------------|--|---------|---------|---------------------------------|
| FE | 1 | Я | О | М | - | - | П |
| BE | 1 | Н | О | М | + | - | П |
| AB2 | 2 | Я | Б (2) | М | - | - | П |
| RK2 | 2 | Я | О | М | - | - | П |
| RK4 | 4 | Я | О | М | - | - | П |
| RKF45 | 5 | Я | О | М | - | - | З |
| DOPRI853 | 8 | Я | О | Ж | - | - | З |
| VODE | BDF1-5 | Н | Б / змінний | Ж | - | - | З |
| | Adams1-12 | Н | Б / змінний | М | - | - | З |
| BDF | 1-5 | Н | Б / змінний | Ж | + | + | З |
| MeBDFi | 1-5 | Н | Б / змінний | Ж | + | + | З |

Тут використані наступні позначення:

AB2 – алгоритм *Adams-Bashford 2*

BDF – *Backward Differentiation Formula*

BE – алгоритм *Backward Euler* (неявний)

BFGS – алгоритм *Broyden-Fletcher-Goldfarb-Shanno*

BHHH – алгоритм *Berndt-Hall-Hall-Hausman*

DAE – *differential-algebraicequation*, диф.-алгебр. рівняння

DDE – *delaydifferenialequations*, диф. рівняння з затримками

DFP – алгоритм *Davidson-Fletcher-Powell*

DOPRI853 – алгоритм *Runge-Kutta-Dormand-Prince 8*

FE – прямий алгоритм Ейлера (явний)

ODAE – *over determined DAE*, перевизначені DAE

ODE – *ordinary differential equation*, звичайне диф. рівняння

RK – алгоритми Runge-Kutta 2, 4 та 5 порядку відповідно

RKF4 – алгоритм Runge-Kutta-Fehlberg 4/5

VODE, Adams – алгоритм *Vode Adams*

MeBDFi – *Modified Backward Differentiation Formula* (MeBDFi) –

модифікована формула диференціювання назад

На рисунку 5.4 наведені два способи (підходи) розв'язання алгебро-диференціальних рівнянь в САЕ.

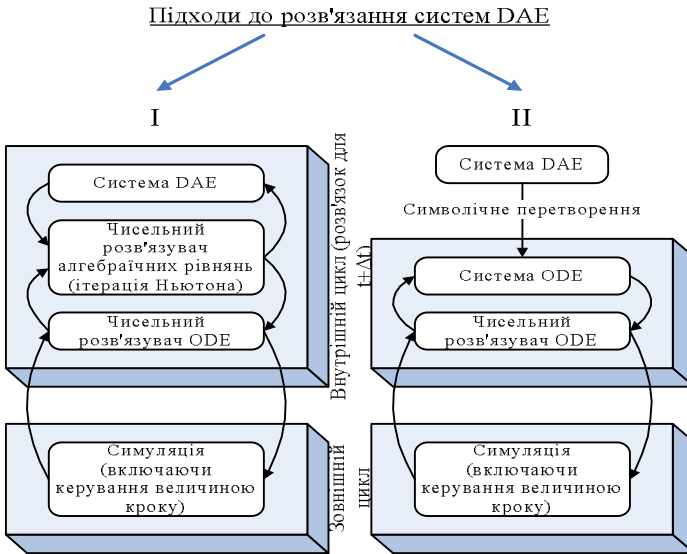


Рисунок 5.4 – Два підходи розв'язання DAE систем

Явні та неявні алгоритми відрізняються умовами старту процедури апроксимації функції на кроці інтегрування (процедури алгебраїзації): в явних (прямих) вся інформація для знаходження значення функції на новому часовому кроці відома з попереднього кроку, в неявних – значення функції на наступному кроці обчислюється одночасно із її похідною, тобто ці методи вимагають прогнозування та розв'язання отриманої СЛАР на кожному часовому кроці (тобто мають додаткові процедури лінеаризації, яка здійснюється, частіше за все, методом Ньютонa).

Неявні методи відрізняються підвищеною алгоритмічною надійністю – стійкістю, тобто мають здатність продукувати адекватний результат при варіації кроку інтегрування в досить широких межах, що є дуже цінною властивістю при моделюванні жорстких, погано обумовлених систем, що мають сингулярності [9].

До альтернативних алгоритмів належать спеціальні методи безпосереднього розв’язання ODE другого порядку (в той час як класичні методи працюють лише з ODE першого порядку, що були перетворені з вихідних ODE, які мають, у загальному випадку, високий порядок), геометричні інтегратори (симплектичні, симетричні, Мозера-Веселова), методи для систем з квантованим станом. Альтернативні алгоритми не отримали широкого поширення у сучасних програмах моделювання ДС.

5.5 Процедура каузалізації

Більшість ДС, як правило, описуються не одним рівнянням, а системою рівнянь, частіше за все ÷ диференціально-алгебраїчними (DAE). На рисунку 5.5 приведено приклад моделі ДС [1].

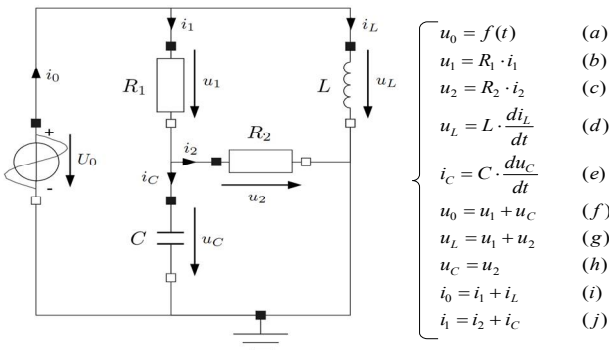


Рисунок 5.5 – Приклад ДС, що описується системою DAE

Більшість DAE у неявній формі, достатньо просто можуть бути перетворені до звичайних диференціальних рівнянь (ODE) у явній формі, використовуючи різні символічні алгоритми сортування. Але деякі системи DAE містять алгебраїчні петлі чи інші структурні сингулярності (топологічні виродження), що змушують ускладнювати методи перетворення за рахунок додавання допоміжних алгоритмів.

На рисунку 5.5 представлений набір рівнянь, що описують електричну схему, фактично це її ММ. Структура цієї системи рівнянь може бути зафіксована у так званій матриці інцидентій структури (*structure incidence matrix*), або матриці охоплення [43]. Спочатку, рівняння при побудові моделей в САЕ записуються в акаузальній формі, тобто знак рівності можна інтерпретувати як такий, що показує еквівалентність лівої та правої частин, замість того, щоб розуміти його як присвоєння правої частини рівняння лівій. Це можна зрозуміти, порівнявши рівняння (a) та (f) (рис. 5.5): обидва мають у лівій частині величину u_0 , але, очевидно, лише одне з них може бути використане для розв'язку відносно u_0 .

Використовуючи матрицю інцидентій можна проводити каузалізацію рівнянь, тобто відновлювати причинність процесів у даній ДС. Це не єдиний існуючий алгоритм каузалізації рівнянь; найбільш популярним є алгоритм Тар'яна (*Tarjan*) [43], оснований на теорії графів: він фіксує ту ж саму інформацію про рівняння, що і матриця інцидентій, але представляє її графічно у вигляді діграфа структури (*structure digraph*). Початковий діграф системи рівнянь представлений на рисунку 5.6 (позначення збережені).

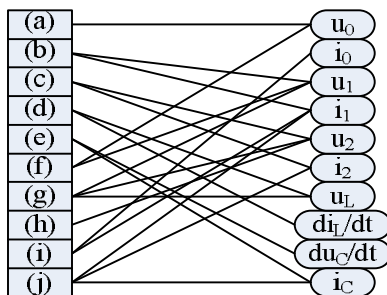


Рисунок 5.6 – Необроблений структурний діграф ДС

Лініями показані зв'язки між рівняннями (ліворуч) та змінними (праворуч): наприклад, змінна u_2 міститься у трьох рівняннях – (c), (g) та (h). Детальна методика побудови та оброблення діграфів структури з метою каузалізації рівнянь в [1, 43]. Таким чином, після каузалізації та при заданні початкових умов, буде отримана система рівнянь, що представлена на рисунку 5.7. Знаки рівності у цій системі рівнянь те-

пер позначають процедуру присвоєння. Рівняння вибудовані так, що жодна змінна не використовується до того, як буде обчислена.

$$\left\{ \begin{array}{ll} u_0 = f(t) & (a) \\ u_2 = u_C & (b) \\ i_2 = u_2 / R_2 & (c) \\ u_1 = u_0 - u_C & (d) \\ i_1 = u_1 / R_1 & (e) \\ i_C = i_1 - i_2 & (f) \\ u_L = u_1 + u_2 & (g) \\ \frac{du_C}{dt} = i_C / C & (h) \\ \frac{di_L}{dt} = u_L / L & (i) \\ i_0 = i_1 + i_L & (j) \end{array} \right.$$

Порядок виконання рівнянь
(executable sequence)

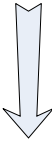


Рисунок 5.7 – Рівняння ММ, приведені до каузальної форми

Отже, рівняння відсортовані не тільки горизонтально (приведені до каузальної форми), але й вертикально – приведені до виконуваної послідовності (*executable sequence*).

Контрольні запитання

1. Графічний інтерфейс програм моделювання.
2. Інструментальні засоби моделювання.
3. Принципи просування модельного часу.
4. Помилка дискретизації.
5. Основні поняття алгоритмів моделювання.
6. Класифікація методів інтегрування.
7. Алгоритми інтегрування.
8. Повна помилка моделювання.
9. Процедура каузалізації.
10. Розв'язання проблем топологічних вироджень.

Завдання для практичного заняття/самостійної роботи

1. За заданою схемою отримати систему рівнянь (математичну модель), здійснити її каузалізацію.
2. Оберіть підхід та метод (алгоритм) для її розв'язання.

ТЕМА №6

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ. ОСОБЛИВОСТІ МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ ТА НЕЙРОННИХ МЕРЕЖ

Перелік питань:

- Імітаційне моделювання.
- Моделювання комп'ютерних мереж.
- Основні поняття та моделювання нейронних мереж.

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 [5]:

$$\begin{aligned} & \text{ext} F(X), \\ & X \in D_x \qquad (6.1) \\ & D_x = \{\psi(x), \varphi(x)\} \end{aligned}$$

де $F(X)$ – цільова функція;

X – вектор керованих (проектних) параметрів;

$\varphi(X)$ 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 однокритеріальної.

Існує 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.

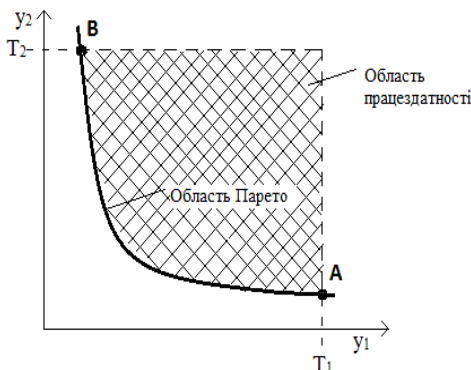


Рисунок 6.1 – Область працездатності

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 ТЗ. Дійсно в (6.2) не визначені норми вихідних параметрів.

Аналогічні недоліки властиві й *мультиплікативному критерію*, цільова функція якого має вигляд (її параметри описані вище):

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

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

$$S_j = (T_j - y_{ном}) / \delta_j,$$

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

Тоді цільова функція в максимінному критерії є:

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

Параметри цієї функції описані вище.

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

$$F(X) = \max \min_j Z_j(X), \quad (6.4)$$

$$X \in D_x, j \in [1, m],$$

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 керування 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: $\Delta\Phi_{\text{надійн}} = |\Phi^0 - \Phi_{\text{надійн}}^*|$, що вказує, наскільки знижується ефективність системи внаслідок можливих відмов її елементів в порівнянні з ефективністю ідеальної системи Φ^0 , 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 *перешкодозахисності 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^0 відноситься до так званих нормальних умов, коли перешкоди відсутні. Тоді в якості показника перешкодозахисності системи можна вибрати абсолютну величину різниці: $\Delta\Phi_{\text{перешкод}} = |\Phi^0 - \Phi_{\text{перешкод}}^*|$, що показує, наскільки змінюється ефективність системи під впливом перешкод із заданими характеристиками.

Деякими особливостями відрізняється оцінка *якості керування* в складній системі. Нехай в даній складній системі керування може бу-

ти організовано керування 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 керування. Тоді абсолютна величина різниці може служити *порівняльною оцінкою* варіантів керування A і B . Для того, щоб виконати *абсолютну оцінку* якості керування (наприклад, для варіанта A), необхідно знати *ідеальний варіант* керування, 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. Умови функціонування складної системи не залишаються постійними, вони схильні до збурень різної природи. Важливо знати, чи зберігають при наявності збурень показники якості системи свої значення, а якщо ні, то які є 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, уточнити суть «збереження необхідних властивостей».

Для системи, *стійкої* щодо функціоналу, що характеризує деяку властивість системи, 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 не можна підібрати обмежень на збурення, що забезпечують збереження відповідного функціоналу.

6.2 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+I) 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 [44].

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 класу:

– 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 на обслуговування;

– 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 проміжок часу (t_0, t_0+1) . Також величина R має сенс імовірності відмов для заявки.

Сутність методу *імітаційного моделювання* стосовно завдань масового обслуговування (СМО) 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.

6.3 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 Марківських випадкових процесів.

Загальні властивості динамічних систем:

- елемент складної системи функціонує в часі (в кожен момент часу t 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 каналами 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 каналами зв'язку і компоновка вхідних сигналів для елементів, які приймають сигнали;

- реакція елемента, який приймає сигнал, на вхідний сигнал.

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

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

де t – час;

α – параметр (сигнал) керування;

X – вектор вхідних сигналів;

k – оператор зв'язку.

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

З точки зору моделювання агрегат виступає як досить універсальний переробник інформації – він сприймає вхідні та управляючі сигнали та видає вихідні сигнали. (див. рисунок 6.2).

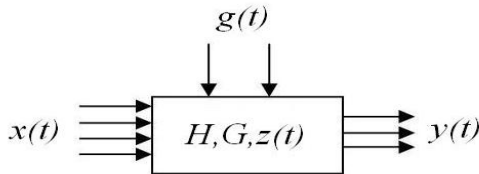


Рисунок 6.2 – Модель агрегативної системи

У кожний момент часу агрегат знаходиться в одному з можливих станів. Стан агрегату є елементом деякої множини Z . Якщо стан $z = (z_1, z_2, z_i)$ 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 вхідні контакти, до яких надходять в моменти часу τ_i керуючі 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 агрегату $z(t)$ з допомогою оператора G . За кінцевий інтервал часу в агрегат надходить кінцеве число вхідних 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 агрегативних систем 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 апарати опису статичної та динамічної структур системи, та відповідний набір для розв'язання рівнянь.

6.4 Скорочені теоретичні відомості по 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 [45], дозволяє створювати проекти обчислювальних мереж різної складності/топології та проводити їх аналіз, використовуючи технологію імітаційного моделювання. Цей пакет безкоштовний, 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 [46] і 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:

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

Задаються розміри блоків даних (транзакції) (*Transaction size*) і час між пакетами даних (транзакціями) (*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.

6.4.1 Приклад імітаційного моделювання мережі

Нижче наведено приклад моделювання мережі, що існувала на кафедрі Електронних систем на момент 1.06.04, та складалась з 20 робочих станцій, розташованих в двох кімнатах, відстань між якими 20 м. У одній з кімнат розташований файловий сервер, який виходить в інтернет 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 проєкт 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 розширення (*expand*). 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 work stations*. 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 Office Connect 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/вихідною буферизацією розміром в 100 кбайт (рис. 6.3).

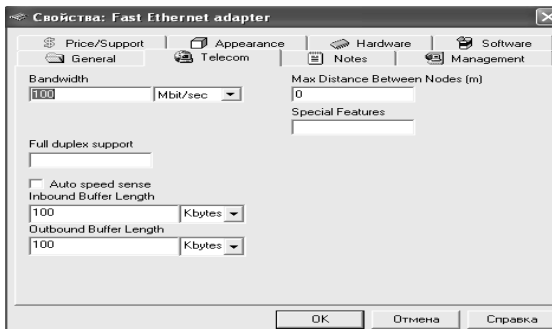


Рисунок 6.3 – Установка частоти передачі і розмірів буферів пристрою

За допомогою команди *link devices* (рисунок 6.4) 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 6.4 – Команда link devices

Вихід в інтернет здійснюється через маршрутизатор, сполучений з комутатором і супутниковою антеною. Натискаємо вкладку *router sand 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. Отримаємо наступну модель лабораторії (рисунок 6.5).

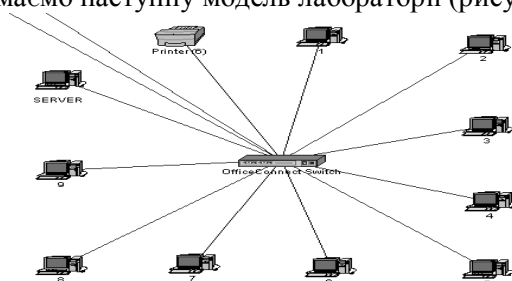


Рисунок 6.5 – Модель мережі лабораторії

Кожен пристрій має порти (рис. 6.6), через які з'єднується з іншими пристроями. Залежно від типу порту і типу кабелю зв'язку передача даних здійснюється з різною частотою (рис. 6.7). Необхідно виключити перевантаження каналів зв'язку і оптимізувати час очікування даних для кожного трафіку.

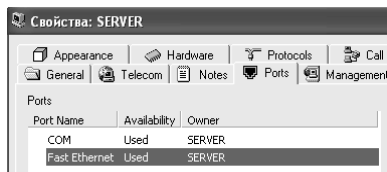


Рисунок 6.6 – Порти пристрою

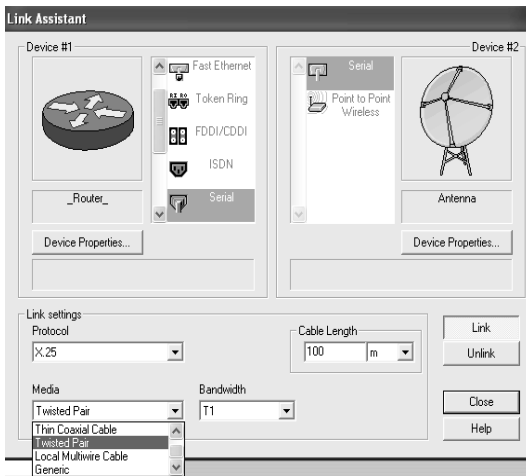


Рисунок 6.7 – Вибір типу зв'язку

Оберемо протокол маршрутизації Rip для протоколу TCP/IP (рис. 6.8), відкривши в меню *global* вкладку *modelsettings*:

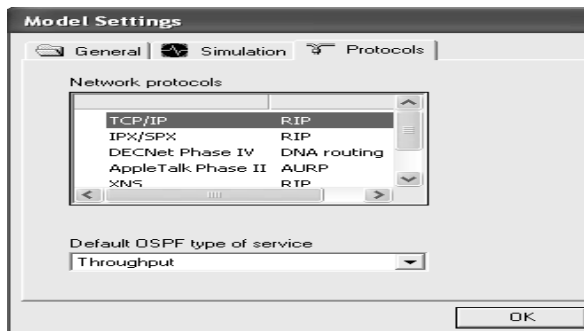


Рисунок 6.8 – Установка протоколів маршрутизації

Встановлюємо тип мережевого протоколу, що використовується кожною робочою станцією – TCP/IP (рис. 6.9), в контекстному меню обираємо вкладку *properties*:

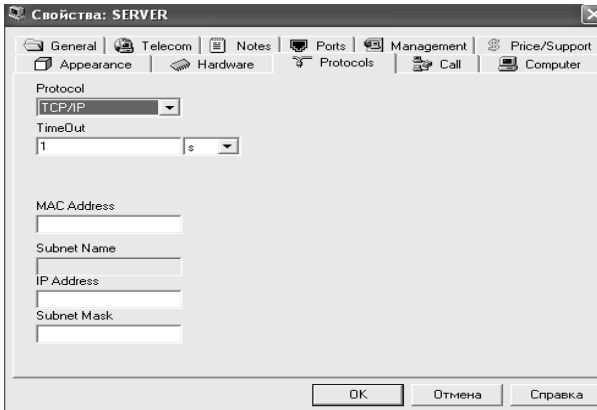
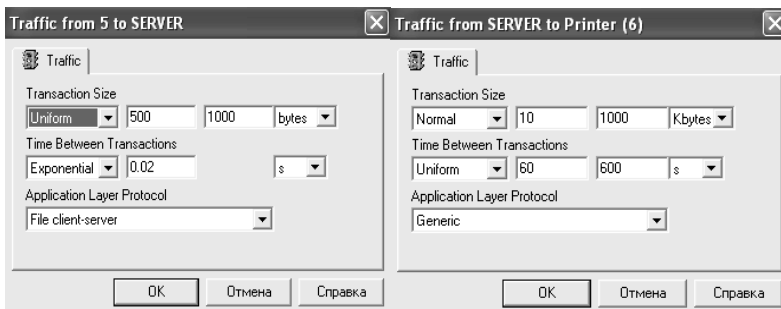


Рисунок 6.9 – Установка типу протокола сети

Обираємо команду *settraffic* (рис. 6.10). Задамо трафік між робочими станціями та сервером – *file server's client*, трафік між сервером і другою будівлею – *small office*.

Рисунок 6.10 – Команда *set traffic*

Задамо трафік для пар клієнт-сервер та принтер-сервер. Виберемо розподіл розміру транзакцій і паузи між ними (рис. 6.11):



а

б

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

Рисунок 6.11 – Установка параметров трафика

Змоделюємо передачу даних через супутник і міську мережу АТМ між двома будівлями, в одній з яких знаходиться наша мережа (рис. 6.12).

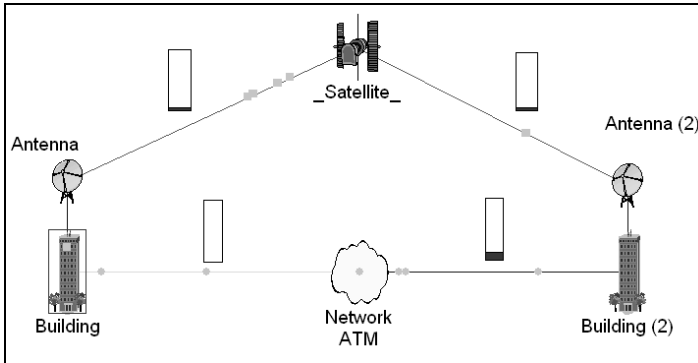


Рисунок 6.12 – Модель мережі: верхній рівень

Для цього встановимо на кожну будівлю супутникову антену вкладки *satellite communication equipment/ earth equipment/antennae/genericdevices*. 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. АТМ (*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.

На рисунку 6.13 спостерігається втрата даних у вигляді червоного спалаху. Причина – час очікування в каналі менший, ніж час проходження пакетів. Час очікування (*time out*) задається за допомогою мережевого протоколу, в нашому випадку – TCP/IP. Задамо більший час очікування (рис. 6.14).

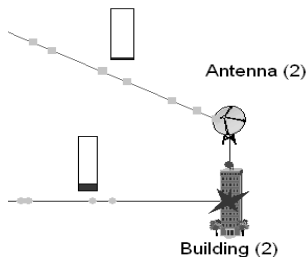


Рисунок 6.13 – Втрата даних

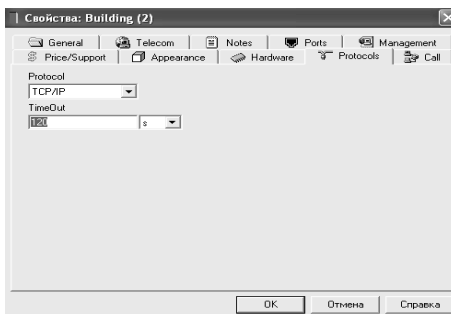
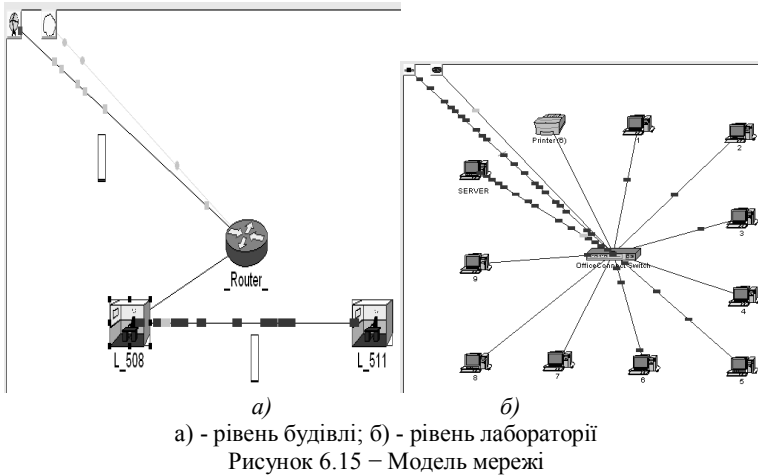


Рисунок 6.14 – Завдання часу очікування

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 %. Як бачимо, втрата даних припинилася (рис. 6.15). Модель мережі оптимізовано.



6.5 Проекування нейронних мереж

Редактор нейронних мереж в 20-sim підтримує два типи мереж:

а) адаптивні В-сплайн мережі (*Adaptive B-spline networks*) («В-сплайн» означає «базисний сплайн» [47]);

б) мережі на основі багат шарового перцептрона Румельхарта [47] (*Multi-layer Perceptron networks*), або MLP-мережі.

Основним елементом MLP-мережі, є штучний нейрон, який виконує просту математичну операцію (представлено на рисунку 6.16). Вхід нейрона складається зі змінних $x_1 \dots x_n$ і зсуваючого елемента, 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*).

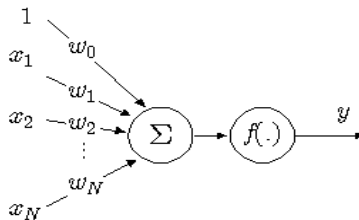


Рисунок 6.16 – Штучний нейрон

Таким чином, нейрон виконує розрахунки за формулою, параметри якої описано вище:

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

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

В таких типах нейронних мереж є 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. 6.16).

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 (*learningrate*); якщо цей показник великий, то перцептрон буде швидко підбирати коефіцієнти, але стане потенціально 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 [48] (в 20-sim *MLP*-мережі мають лише один прихований шар). Вікно редактора *MLP*-мережі показано на рисунку 6.17.

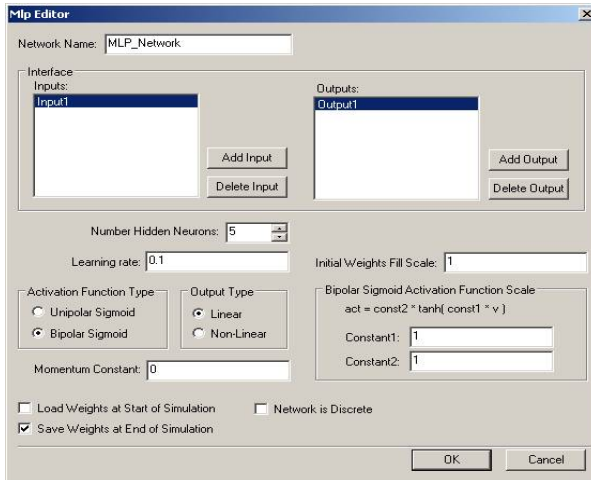


Рисунок 6.17 – Редактор *MLP*-мережі

Другий тип нейронної мережі, що підтримується в 20-sim – адаптивна *B*-сплайн мережа. Вікно та настройки редактора *B*-сплайн мережі представлені на рисунку 6.18.

B-сплайн мережа використовується для співставлення k входів до одного виходу на обмеженій області простору вхідних величин [48]. 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, ко-

жна одновимірна функція може бути апроксимована – сплайни ($F_1 \dots F_n$), поєднані з ваговими коефіцієнтами ($w_1 \dots w_n$), разом утворюють вихідну функцію.

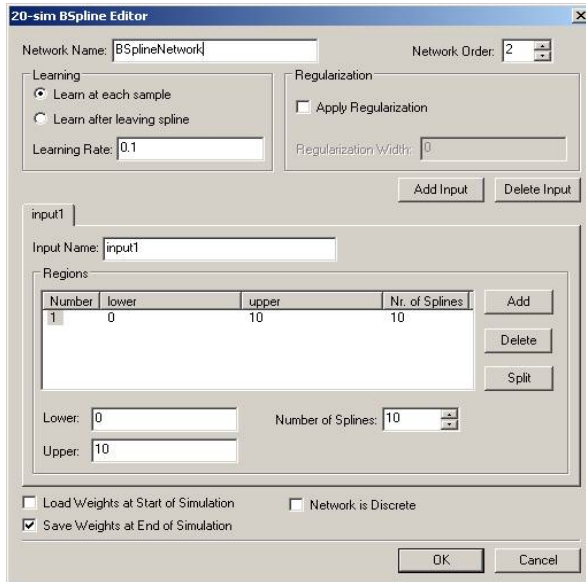


Рисунок 6.18 – Редактор B -сплайн мережі

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

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

Контрольні запитання

1. Критерії оптимальності систем.
2. Принципи імітаційного моделювання (виділити особливості, порівняно із моделюванням на схемотехнічному рівні).
3. Алгоритми та програми імітаційного моделювання. Метод Монте-Карло в програмі Мікро-Сар.
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.
8. Типи мереж.
9. Редактори мереж в 20-sim.
10. Поняття та область використання нейронної мережі.

Завдання для практичного заняття/самостійної роботи

1. В заданій схемі інформаційно-вимірювальної системи визначити інформаційну підсистему. Розкласти її на *SoftWare* та *Hardware*. Підібрати типи протоколів, способи зв'язку між підсистемами.
2. Сформувати концептуальну (словесну) модель об'єкта моделювання для системи Net Cracker.
3. Обрати математичний базис для опису моделі на етапі моделювання Net Cracker (види пакетів, типи розподілів).
4. Провести моделювання в Net Cracker.
5. Отримати модель штучного нейрона в програмі 20-sim.

ТЕМА №7

МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ЕСАД. ПАРАМЕТРИЧНИЙ СИНТЕЗ ТА ОПТИМІЗАЦІЯ

Перелік питань:

- Методика параметричного синтезу в ЕСАД.
- Синтез фільтрів в ЕСАД.
- Методики параметричної оптимізації в ЕСАД.
- Алгоритми оптимізації в САЕ.

7.1 Математичне забезпечення програм ЕСАД

Згадаємо стисло склад математичного забезпечення та алгоритми моделювання в ЕСАД [3]. Типове математичне забезпечення для моделювання електронних пристроїв на схемотехнічному та функціональному рівнях складається з компонентів:

- бібліотека моделей і параметрів дискретних приладів та інтегральних мікросхем;
- алгоритм формування математичної моделі схеми (ММС);
- чисельні методи розв'язування рівнянь ММС;
- алгоритми синтезу фільтрів;
- алгоритми параметричної оптимізації;
- засоби візуалізації та обробки результатів (постпроцесор).

Більшість складових математичного забезпечення розглянуто в попередніх дисциплінах (Комп'ютерне моделювання, САПР). Окреме місце займають алгоритми параметричного синтезу та оптимізації, які будуть розглянуті нижче. Структурний синтез зазвичай виконується евристичними методами [5].

ММС аналогової частини схеми отримується модифікованим методом вузлових потенціалів в формі системи нелінійних неоднорідних диференціальних рівнянь. Ці рівняння розв'язуються методами числового інтегрування змінного порядку зі змінним кроком; отримані нелінійні алгебраїчні рівняння лінеаризуються за допомогою модифікованого метода Ньютонна – Рафсона, на кожній ітерації якого фазові змінні знаходяться методом *LU*-факторизації. Отримані таблиці результатів (струми особливих гілок та напруги (потенціали) незалеж-

них вузлів) обробляються в постпроцесорі, зокрема, розраховуються нові залежності та будуються графіки.

В програмах EСAD аналогово-цифрові пристрої моделюються по окремих підпрограмах, але в єдиній оболонці, завдяки об'єднуючій ролі універсальних редакторів схем, в яких збираються схеми та компонуються моделі, та постпроцесорів, в яких виконується графічна обробка отриманих в результаті аналізу даних. Програма розглядає пристрій як аналогово-цифровий, якщо в описі схеми присутні макромоделі цифрових елементів і моделі та макромоделі аналогових приладів. Програмно розрізняються вузли аналогові, цифрові та А/Ц, Ц/А, у останні автоматично додаються джерела живлення із дискретними рівнями й прилади сполучення (інтерфейси А/Ц, Ц/А), які не відображуються на схемі, але забезпечують обмін даними між підпрограмами моделювання електричних процесів в аналоговій частині ланцюга і п'ятизначного моделювання цифрової частини. В цифровій частині схеми моделювання виконується подійне моделювання за правилами п'ятизначної логіки, де дискретні рівні сигналу приймають значення $1, 0, X, F, R$ (дозволені рівні напруги – алфавіт сигналу).

Мінімальне значення кроку інтегрування у цифровій частині визначається кроком дискретизації (частота дискретизації, як правило, 10ГГц), або параметром *Pin-to-Pin Delay*. Крок інтегрування у аналоговій частині визначається найменшою постійною часу у схемі. Загальна кількість кроків залежить від значення максимальної постійної часу в схемі, і в аналогово-цифровій схемі значно вища за кількість кроків у аналогових схемах тієї ж розмірності. Звичайно, аналогово-цифрові пристрої моделюються в режимі розрахунку перехідних процесів (*Transient*), однак інші режими – по постійному (*DC*) та змінному (*AC*) струму також доступні при певних спрощеннях схеми.

7.2 Автоматизований синтез фільтрів

Активні та пасивні фільтри мають регулярну структуру, будуються на основі обмеженої елементної бази (R, L, C та операційний підсилювач), тобто мають гомогенний базис, як слідство, на сьогоднішній день етапи структурного та параметричного синтезу для них повністю автоматизовані. Математичний апарат синтезу базується на теорії комплексних передатних функцій та методах поліноміальної апроксимації.

Розглянемо особливості автоматизованого синтезу фільтрів в ECAD. Синтез активних фільтрів в *MC* починається з вибору в меню *Design* вкладки *ActiveFilter*. Вікно завдання параметрів синтезу активних фільтрів представлено на рис. 7.1.

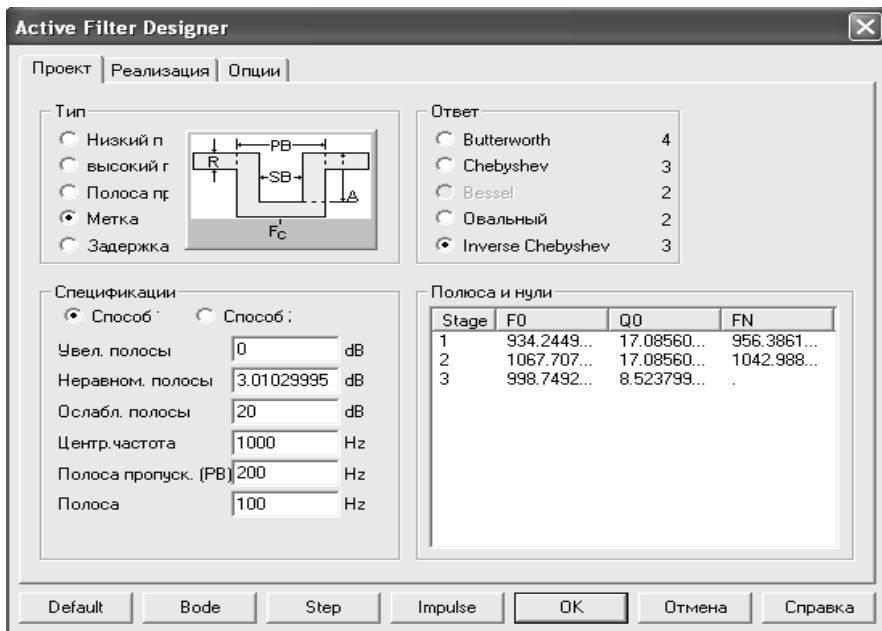


Рисунок 7.1 – Вікно синтезу активного фільтра

Розглянемо склад меню.

Перша вкладка *Design*. Пропонується 5 типів фільтрів:

Lowpass (низькочастотний)

Highpass (високочастотний)

Bandpass (смуговий)

Notch (загороджувальний)

Delay (фільтр затримки)

Перші 4 типи визначаються по АЧХ, фільтр затримки – по часовим характеристикам.

Пропонується 5 типів апроксимуючих поліномів: Баттерворта, Чебишева, Бесселя, Еліптичний, Інверсний Чебишева. Поліном Бесселя доступний тільки для синтезу фільтрів затримки. Для пасивних

фільтрів доступні тільки поліноми Баттерворта и Чебишева. Поліноміальні функції будуються в залежності від частоти для порівняння із реальними характеристиками та апроксимації методом найменших квадратів.

Mode 1, Mode 2 (режими 1 та 2).

Режим 1 дозволяє конкретизувати функціональні особливості фільтра, а саме, обрати значення частот пропускання, загасання, зрізу. Згідно заданим вимогам програма розраховує ступінь апроксимуючого поліному. В режимі 2, навпаки, можливо вказати ступінь поліному, та, відповідно, і кількість ступіней в синтезованій схемі фільтру.

PolesandZeros.

Секція *PolesandZeros* призначена для введення та редагування числових значень полюсів, нулів і добротності фільтру для оптимізації амплітуди та чутливості до шуму. Нулі фіксуються та не можуть змінюватися.

При натисканні кнопки *CodePlot* програма виводить графіки ідеальних АЧХ, ФЧХ та групової затримки для фільтра (реальний фільтр може мати дещо інші характеристики).

Impulse – програма виводить графік імпульсної характеристики, яка є реакцією схеми на одиничний імпульс.

У вкладці *Implementation* (Реалізація) програма пропонує обрати вид ступіней для реалізації схеми проектованого фільтра. Для цього необхідно в опції *StageValue* клацнути лівою кнопкою мишки по колонці *Circuit*, із запропонованого списку доступних типів обрати оптимальну:

| | | |
|------------------------|---|------------------------------|
| Sallen-Key | – | Саллен-Кея; |
| MFB (MultipleFeedback) | – | множинні зворотні зв'язки; |
| Tow-Thomas | – | Тау-Томаса; |
| Fleischer-Tow | – | Флейшера-Тау; |
| KHN | – | Кервіна-Хьюлсмана-Ньюкомба; |
| Acker-Mossberg | – | Аккерберга-Мосберга; |
| Tow-Thomas 2 | – | Тау-Томаса 2; |
| DABP | – | смуга пропускання подвійного |

підсилення (*DualAmplifierBandPass*).

Можна призначати одну ступінь для всіх ланцюгів, однак не всі ланцюги підходять для будь-якої передавальної функції. Ступінь Саллен-Кея є універсальною (рис. 7.2).

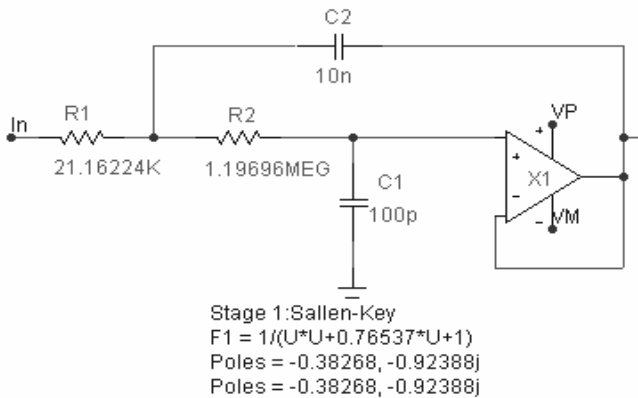


Рисунок 7.2 – Схема заміщення та параметри ступені Саллен-Кей

Секція *Circuit* дозволяє обрати, якого виду коло використовувати: стандартне пасивне LC коло чи подвійне. Секції *ResistorValue*, *CapacitorValue*, *InductorValue* дозволяють обрати значення резисторів, конденсаторів та котушок індуктивності точно або за існуючими стандартами. Секція *ImpedanceScaleFactor* – задається масштабний коефіцієнт, який множить значення R і L та ділить C, не змінюючи форму характеристики $F(p)$, але дозволяє наблизити значення R C L компонентів до стандартного ряду.

Вкладка *Options*.

Component Value Format – завдання формату значень компонентів.

PolynomialFormat – завдання значень поліноміальних коефіцієнтів.

Plot – програма буде ідеалізовані характеристики.

Number of DataPoints – кількість табличних точок (500).

Show Circuit – показати схему проєктованого фільтру.

Save To – дозволяє обрати, куди і в якому вигляді зберігати схему (Circuit як схема заміщення; *Macro* – «біла скриня»).

Опція *Opamp Model to Use* – обирання моделі ОП з трьох різних за рівнем складності.

OddLowPassStage – обирання останньої ступіні для НЧ фільтра (RC фільтр, RC з неінвертуючим підсилювачем, RC з інвертуючим підсилювачем).

OddHighPassStage – обирання останньої ступіні для ВЧ фільтра.

Зверніть увагу:

- існує протиріччя між крутістю спадання АЧХ і лінійністю ФЧХ, однак фільтри Баттерворта забезпечують компромісне рішення;

- пульсації в смузі пропускання відсутні у фільтрів Баттерворта, Бесселя та інверсного Чебишева;

- інверсний фільтр Чебишева має найбільшу крутість спадання АЧХ;

- фільтри Бесселя забезпечують найкращу лінійність ФЧХ в смузі пропускання та групової затримки;

- загасання корисного сигналу в смузі пропускання більше у фільтрів, які реалізовані на RC елементах, ніж на CL елементах;

- при однакових параметрах смуги пропускання та обраному типі апроксимації, реалізація через Саллен-Кей дає мінімальну схему;

- із ціллю мінімізації схеми фільтру необхідно обирати, якщо дозволяє ТЗ, невелику крутість фронтів АЧХ;

- при синтезі фільтрів використовується перехід у комплексну площину за допомогою оператора Лапласа(S).

Сутність параметричного синтезу фільтрів може бути представлена як мінімізація деякої функції помилок, або цільової функції. Відхилення АЧХ реального та ідеального НЧ-фільтра (різність ординат характеристик) може слугувати мірою похибок, яку необхідно зменшити шляхом цілеспрямованої зміни значень параметрів компонентів.

7.3 Синтез параметрів моделей приладів

Для параметризації моделей напівпровідникових приладів в МС [12] використовується Програма розрахунку параметрів математичних моделей аналогових компонентів *MODEL*, вона працює в інтерактивному режимі та виконує параметричний синтез (параметризацію) математичних моделей напівпровідникових приладів за їхніми паспортними даними, що вводяться в табличній, або графічній формі. При введенні графіків необхідно задати координати характерних точок: від двох до п'яти – чим більше даних, тем точніше оцінюються параметри

моделей. Якщо дані відсутні, то приймаються їхні значення по замовчанню. Приймається, що всі експериментальні дані виміряні при кімнатній температурі. Оптимізація здійснюється за допомогою прямого методу Пауелла.

Програма *MODEL* зберігає введені паспортні дані компонентів і розраховані параметри математичних моделей в бінарних файлах із розширенням *.MDL*. Ці файли можна завантажувати в програму *MODEL* для уточнення параметрів моделей, що зручно для відладки бібліотек моделей, отже *MDL*-файли має сенс зберігати для подальшого редагування.

Після завершення обробки бібліотеки, доцільно видалити з цього файлу вхідні дані і перетворити його в компакту форму - бінарні файли, що мають розширення *.LBR* та призначені для передачі в програму моделювання, де їх можна проглянути, редагувати та використати для побудови моделі системи.

Для обміну даними з програмою *PSpice* можна використати команду *File>Create SPICE file* для створення текстового файлу із розширенням *.LIB*. Бібліотечні файли з розширенням *.LBR* і *.LIB* підключаються до програми МС за допомогою файла *NOM.LIB* з каталогу *\DATA*, в якому перелічені директиви *.LIB* із вказанням імен файлів бібліотек. Після завантаження програми *MODEL* на екрані з'являється її меню, показане на рисунку 7.3.

Окрім стандартних кнопок керування вікнами, є наступні поля:

Меню команд – містить назви низхідних меню;

Текстові поля – містить чотири рядки *T1*, *T2*, *T3* і *T4*, зміст яких переноситься в бібліотеку моделей. Рядок *T1* містить ім'я компоненту (*Partname*), воно використовується для сортування в каталозі бібліотеки, інші – коментарі та додаткова інформація;

Таблиця даних – таблиця з двома, або трьома колонками для вводу значень координат характеристик компоненту, отриманих експериментально, або з довідників. Кількість колонок залежить від типу компоненту і характеру даних;

У вікні графіків характеристик моделі відмічені точки даних, якщо вони вводились користувачем. Близькість цих точок до синтезованих методами апроксимації графіків, свідчить про точність моделі. Числове значення середньоквадратичного відхилення у відсотках вказується зверху на рядку *Error*.

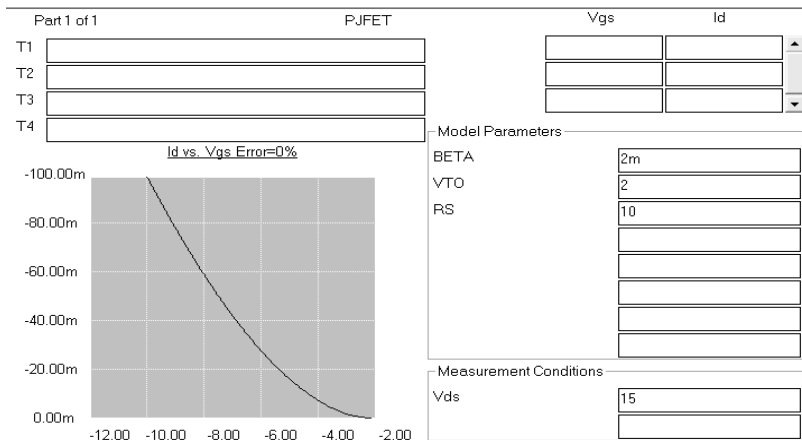


Рисунок 7.3 – Стартове вікно програми MODEL

Параметри моделі – таблиця параметрів моделі, отриманих в результаті обробки введених даних. Умови вимірювань – значення додаткових параметрів, при яких отримані експериментальні дані.

7.4 Методи оптимізації

При розробці моделей автоматизованих 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 коефіцієнтів регуляторів, 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, Кеслера (*Betragsoptimum*), Латцеля (*Betragsadaptation*), Пройса (*Betragsoptimum* за відсутності моделі об'єкта); метод з використанням прогнозуючого нейрокерування з нечіткими регуляторами, як альтернатива звичайного ПІД-регулятора тощо [49-51].

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 [52]):

- 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 (*Patternsearch* – 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 (*hillclimbing*) 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-Шміда, 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.

В практичних задачах оптимізації моделей, якщо передбачається, що цільова функція (ЦФ) є гладкою і кількість варійованих пара-

метрів (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 [53]):

- 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.

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. 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 не знайде розв'язок в області, де абсолютне значення градієнта функції невелике, але не нульове. Кінцево-різницеви квазі-Ньютонівські методи відрізняються від таких же методів, з використанням безпосередньо похідних, алгоритмами вибору довжини кроку, а також зміною рекомендованого критерію зупинки.

Всі техніки оптимізації можна класифікувати за різними категоріями; пропонується класифікація, що представлена таблицею 7.1.

Таблиця 7.1 – Класифікація методів оптимізації

| Класифікація | Клас методів | Примітка (підкласи та приклади) |
|---|--------------------------------|--|
| За наявністю обмежень на вектор варіюваних 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 (однопараметричні) | Варіюється один параметр (методи Фібоначчі, золотого перетину та інші) |
| | Багато-параметричні | Варіюється більше одного параметра (метод Нелдера-Міда та інші) |
| За характером шуканого рішення | Локальні | Метод Хука-Дживса, Пауела та ін. |
| | Глобальні | Методи відсікаючої площини, імітації відпалу, гілок і меж |
| За характером цільової функції (ЦФ) | Стохастичні | ЦФ має випадкові параметри |
| | Детерміновані | ЦФ апріорно детермінована |
| За способом досягнення результату | Алгоритмічні | Симплекс-метод та інші |
| | Ітеративні | Метод Ньютона, внутрішніх точок |
| | Евристичні | Еволюційні алгоритми |
| За кількістю попередніх врахованих кроків | Однокрокові | Враховується 1 попередній крок |
| | Багатокрокові | Враховується декілька попередніх кроків |
| За порядком похідних | Нульового порядку (прямі) | Пауела, золотого перетину, Хука-Дживса, Нелдера-Міда |
| | Першого порядку (градієнтні) | Градінтний спуск, BFGS-метод, Левенберга-Марквардта-1, методи DFP та SR1, спряжених градієнтів |
| | Другого порядку | З використанням гессіана ЦФ: метод Ньютона, Ньютона-Рафсона |

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 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 7.2.

Таблиця 7.2 – Порівняння 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 поч. 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 САК (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 К. Розширений алгоритм вибору методу оптимізації в *20-sim* представлений в Додатку Л [49].

7.5 Параметрична оптимізація в Micro-Cap

Параметрична оптимізація [12] 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 критеріях, вказаних в Додатку М. При цьому можлива оптимізація по максимальному (*Maximizes*), мінімальному (*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 екстремума 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.

Розглянемо приклад, коли в схемі диференційного підсилювача необхідно знайти значення опорів $R5$ та $R2$, при яких максимізується підсилення на частоті в 10 кГц. Для цих опорів задаються моделі $RMOD$ та у вікні оптимізатора задається параметр $RES\ RMOD$ та умови його варіації (приклад з бібліотеки MICRO CAP: папка DATA).

Досліджувана схема та результати моделювання (АЧХ у вихідному вузлі), отримані при оптимальному значенні опорів $R5$ та $R2$ наведені на рисунку 7.4.

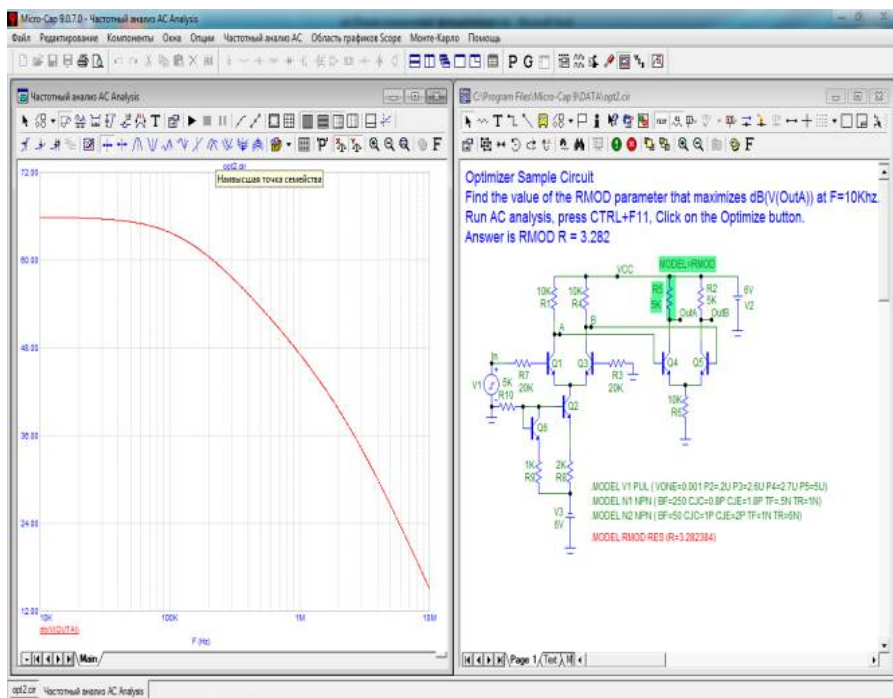


Рисунок 7.4 – Досліджувана схема та її АЧХ, отримана при оптимальному значенні параметрів

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;

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.

Приведемо 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+I*), 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.

На відміну від попередніх версій *Micro-Cap*, з восьмої версії програма сама проставляє діапазон і крок зміни параметра (можна редагувати).

d) 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.

e) 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 покровий метод Пауелла, який дозволяє «побачити» всі локальні екстремуми функції.

Контрольні запитання

1. Методика синтезу параметрів моделей.
2. Методика параметричної оптимізації в ECAD.
3. Алгоритми параметричної оптимізації в САЕ.
4. Методи оптимізації.
5. Рекомендації по вибору методу оптимізації.
6. Методика синтезу аналогових фільтрів в МС.
7. Алгоритми пошуку екстремумів.
8. Евристична оптимізація (самостійно).
9. Методи синтезу та проектування ПЛІС (самостійно).
10. Параметри, за якими оптимізуються пристрої в MICRO CAP.

Завдання для практичного заняття/самостійної роботи

1. Згідно завдання синтезувати схему фільтра в програмі Micro-Cap. Визначити ключові параметри частотних характеристик.
2. Згідно завдання зібрати схему фільтра типу, провести його аналіз та параметричну оптимізацію в програмі Micro-Cap.

ТЕМА №8

МЕТОДИКИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МОДЕЛІНГУ ТА СИМУЛЯЦІЇ

Перелік питань:

- Проблема точності моделінгу.
- Критерії якості моделювання. Усунення алгоритмічних збо-

їв.

8.1 Критерії якості моделювання

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 [42].

Універсальність, це застосованість математичного забезпечення до широкого класу об'єктів, що проектуються. 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 і не однозначною задачею, якщо вихідних функцій декілька (сигнал векторний). Зазвичай у таких випадках точність розраховується як зважена сума точності за складовими цього сигналу.

8.2 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 проектування, при деталізації проекту 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 реального 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:

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. 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 робочого навантаження і внутрішніх параметрів самої системи. Зазвичай таку оцінку проводять по кожному параметру окремо. Дані, отримані при оцінці чутливості моделі, можуть бути використані, зокрема, при плануванні експериментів: більша увага повинна приділятися тим параметрам, за якими модель є більш чутливою. Як бачимо, критерії якості протирічать один одному, отже задача проектування моделі є оптимізаційною.

8.3 Методика усунення алгоритмічних збоїв

В процесі моделювання в ECAD можливі наступні збої:

1. Аналіз не здійснюється, оскільки попередній аналіз МС виявив помилки, не сумісні з можливістю симуляції (це викликано топологічними виродженнями). Отже, на етапі верифікації ММС можливе отримання наступних повідомлень:

- «*The circuit missing a ground*». Оскільки ММС формується ММВП, то один з вузлів моделі схеми повинний бути заземлений. Якщо на схемі відсутнє заземлення, необхідно сформувати окрему схему з землею, щоб забезпечити незалежність вузлів;

- «*The matrix is singular*»: необхідно усунути в моделі схеми елементів, що приводять до появи топологічних вироджень, або додати такі елементи схеми, що усувають сингулярність, не допускати великих розкидів постійних часу (елементи можуть бути фіктивними, або такими, що емулюють паразитні ефекти).

2. Симуляція зупиняється, або результати не є адекватними.

Можливі наступні збої:

– відсутність коливань при моделюванні генераторних систем – для усунення цього необхідно задати в меню «*State Variable Editor*», або в меню компонентів початкові значення струмів, напруг, логічних рівнів;

– зупинка алгоритму та видача повідомлення «*Internal timestep too small*» – пов'язано з малим кроком інтегрування, який призводить до збільшення кількості ітерацій, у випадку нестійкості числового методу - до «зависання» системи. Для усунення цього необхідно провести корекцію схеми з метою зменшення її жорсткості, або змінити параметри *ABSTOL*, *RELTOL* в глобальних установах, обрати інший метод інтегрування.

– осциляції та «провали» характеристик - рекомендується використовувати більш стійкий метод Гіра (режим *Power Default* в глобальних установах); зменшити крок в меню *Analises Limits*;

– низька точність результату моделювання - рекомендується в якості ведучого елемента вибрати елемент із максимальною абсолютною величиною, пункт *PIVTOL* діалогового вікна *GLOBAL SETTING*.

– результати моделювання неадекватні - порядок величин *ABSTOL* і *RELTOL* діалогового вікна *GLOBAL SETTING* рекомендується пропорційно збільшувати зі зростанням порядку струмів і на-

пруг, які протікають в модельованій системі; збільшити кількість ітерацій параметра *ITL1* до 1000.

Основні параметри моделювання, які змінюються для жорстких математичних моделей наведені в таблиці 8.1.

Таблиця 8.1 – Параметри в глобальних настройках Micro Cap

| Ім'я | Сенс параметра |
|--------|--|
| ABSTOL | Абсолютна похибка розрахунків струмів у режимі <i>Transient</i> |
| CHGTOL | Припустима помилка розрахунку заряду в режимі <i>Transient</i> |
| GMIN | Мінімальна провідність (фіктивна) в матриці Якобі |
| ITL1 | Максимальна кількість ітерацій у режимі <i>DC</i> |
| ITL2 | Максимальна кількість ітерацій при розрахунку передатних функцій в режимі <i>DC</i> |
| ITL4 | Максимальна кількість ітерацій для поточного часу в <i>Transient</i> |
| PIVREL | Відносна величина елемента рядка матриці, необхідна для його виділення в якості ведучого елемента |
| PIVTOL | Абсолютна величина елемента рядка матриці, необхідна для його виділення в якості ведучого елемента |
| RELTOL | Припустима відносна помилка розрахунку напруг і струмів у режимі <i>Transient</i> |
| RMIN | Мінімальне значення опору навантаження напівпровідникових приладів |
| TRTOL | Коефіцієнт, що визначає припустиму помилку апроксимації в режимі <i>Transient</i> |
| VNTOL | Припустима помилка розрахунку напруг у режимі <i>Transient</i> |

Значення основних глобальних параметрів МС в стандартному режимі та в режимі моделювання жорстких та потужних схем наведено в таблиці 8.2 [12].

Таблиця 8.2 – Значення основних глобальних параметрів

| Параметр | Standard Default | Power Default |
|----------|------------------|---------------|
| ABSTOL | 1pA | 1uA |
| CHGTOL | 0.01pC | 1nC |
| GMIN | 1p | 1n |
| ITL1 | 100 | 200 |
| RELTOL | 1m | 10m |
| VNTOL | 1u | 1m |
| Метод | Trapezoidal | Gear |

Рекомендації для підвищення точності моделювання:

- на етапі моделінгу використовуйте найбільш повні бази даних по вхідних параметрах;
- спочатку перевірте використані вами моделі в статичному режимі, при необхідності – проведіть параметричну оптимізацію;
- спочатку використовуйте максимально прості моделі (наприклад, для активних елементів використовуйте моделі ключів);
- перевірте схему на «початкову схемну надійність», тобто, чи виконує система свою функцію в цілому (підсилює, випрямляє, фільтрує і т.д.);
- моделюйте складні схеми частинами;
- прогнозуйте хід та вид характеристик;
- при зображенні в постпроцесорі контролюйте крок по вісі X, для отримання плавних характеристик, різні фазові змінні зображуйте на різних графіках.

8.3.1 Рекомендації щодо уникання проблем з алгебраїчними петлями

Приклад алгебраїчної петлі представлений на рисунку 8.1.

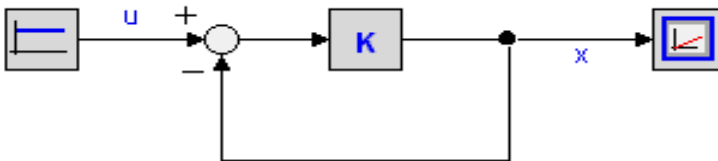


Рисунок 8.1 – Алгебраїчна петля у представленні блок-схемою (BD)

Зазвичай це контур зі зворотним зв'язком, у якому немає елементів «пам'яті», тому для обчислення змінних у петлі потрібні значення самих змінних. Зауважте, що на вході блока K одночасно заходять дані з «минулого» (функція « u ») і з «майбутнього» (функція « x »).

Хоча більшість САЕ-систем і має складні розвинені алгоритми пошуку аналітичного розв'язку, але не завжди можна уникнути появи непорушних алгебраїчних петель. Алгебраїчні петлі можуть призводити до зростання часу симуляції, а іноді й до зупинки симуляції при збої в ітераційному механізмі.

Тому можна рекомендовані деякі заходи для вирішення цих проблем:

а) зміна порядку обчислень. Алгебраїчні петлі зустрічаються, коли порядок обчислень є довільним. Якщо петлі з'являються у моделях на основі рівнянь (*equation model*), то можна змінити порядок обчислень, переписавши рівняння. Порядок розрахунків у моделях на основі зв'язаних графів може бути змінений шляхом введення каузальної інформації вручну (*hand-defined causality*);

б) введення паразитних елементів накопичення енергії (наприклад, маленька маса чи конденсатор) для розривання алгебраїчних петель. Однак, ці елементи вводять великі полюси в рівняння стану, що може призвести до значного збільшення часу симуляції;

в) видалення з алгебраїчних петель елементів, які не релевантні, або не мають суттєвого відношення (якими у другому наближенні можна знехтувати) до вихідної змінної при симуляції моделі (наприклад, невеликі амортизатори, дуже жорсткі пружини тощо). Проте, коректне видалення елементів не завжди можливе і потребує значних навичок моделювання та розуміння фізичних процесів;

г) поєднання (топологічне перетворення) однотипних елементів: іноді елементи одного типу можна об'єднати шляхом додавання значень їх параметрів, у багатьох випадках це зменшить кількість алгебраїчних петель.

8.3.2 Способи розв'язання високоіндексних топологічних вироджень ММ

Окрім алгебраїчних петель трапляються й інші структурні сингулярності при створенні моделей. Наприклад, може виникати проблема початкових умов для елементів накопичення енергії при певному їх з'єднанні, коли їх змінні не можуть вважатися незалежними (контур з ємностей наведено на рисунку 8.2).

Треба звернути увагу на рівняння моделі з позначкою (*f*) – це обмежуюче рівняння зв'язку (*constraint equation*), воно немає невідомих змінних, бо u_1 та u_2 обрані у якості змінних стану і вважаються відомими (у даній системі DAE). Існує кілька різних шляхів вирішення цієї проблеми.

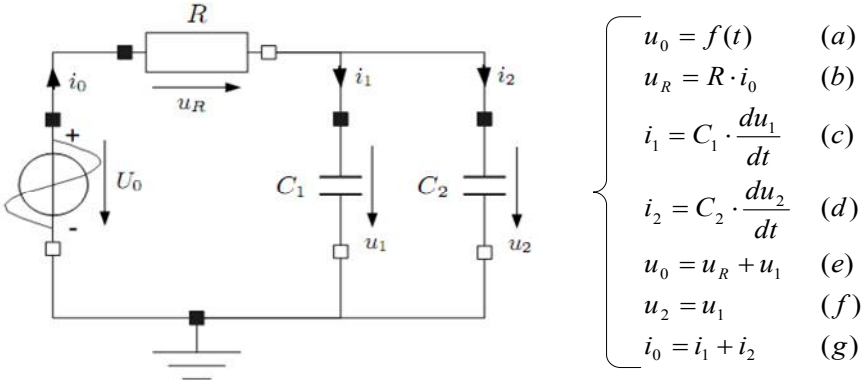


Рисунок 8.2 – Приклад структурної сингулярності у моделі електричного кола

Наприклад, можна змінити причинність рівняння (d), розв'язуючи його для змінної i_2 замість du_2/dt ; отже солвер повинен розв'язати рівняння відносно du_2/dt замість u_2 , таким чином використовуючи техніку числового інтегрування і диференціювання для розв'язку системи, блок-схема та каузалізовані рівняння якої приведені на рисунку 8.3.

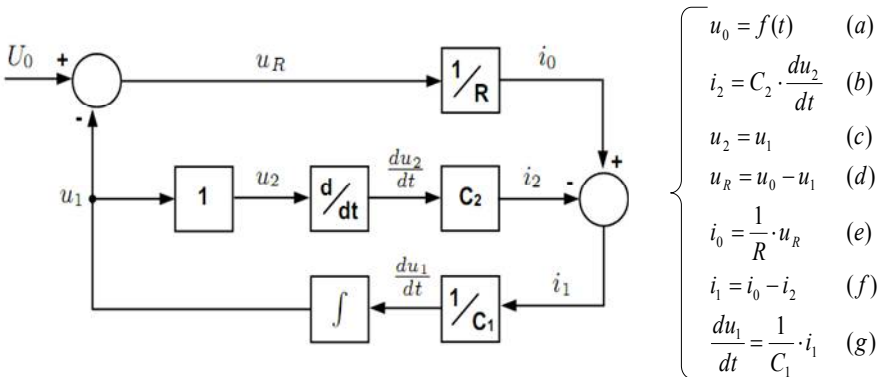


Рисунок 8.3 – Вирішення сингулярності зміною причинності у рівняннях

Але застосування числового диференціювання не найкращий спосіб для вирішення такої проблеми (принаймні, якщо використовується явна формула; краще у такому випадку використовувати неявну формулу, бо вона викликає ітерацію на кожному кроці) [43].

Краща реалізація вирішення проблеми представленої структурної сингулярності запропонована Пантелідесом [53]. Його алгоритм проводить видалення обмежуючого рівняння (для можливості подальшого використання символічного диференціювання) методом заміни, показаної на рисунку 8.4 (приклад для рівняння (f) з рисунку 8.3).

$$u_2 = u_1 \quad \longrightarrow \quad \frac{du_2}{dt} = \frac{du_1}{dt}$$

Рисунок 8.4 – Приклад заміни, що використовується у методі Пантелідеса

Після чого система DAE позбавляється розглянутої сингулярності і редукується до системи з алгебраїчною петлею (розв'язання яких вже розглядалося).

Взагалі, структурно сингулярні фізичні системи описуються у формі високоіндексних (високорангових) DAE [48]. Так званий індекс (ранг) збурення (*perturbation index*) є мірою обмежень, що з'являються серед рівнянь:

- DAE нульового рангу збурення (*index-0*) не містить жодної структурної сингулярності, немає алгебраїчних петель;
- DAE першого рангу (*index-1*) містить тільки алгебраїчні петлі;
- DAE, ранг яких більше одного (так звані високоіндексні DAE), містять алгебраїчні петлі та різні структурні сингулярності.

Алгоритм Пантелідеса – це метод символічного пониження індексу системи, застосовувати який можна кілька разів, щоразу понижуючи індекс збурення DAE на один, зрештою переходячи до задачі розв'язання алгебраїчних петель. Таким чином, наявність топологічних вироджень (структурних сингулярностей) ускладнює процедуру отримання математичної моделі системи в нормальній формі – потрібно або розв'язувати системи лінійних алгебраїчних рівнянь на кожному кроці числового інтегрування системи диференціальних рівнянь, або попередньо усувати топологічні виродження за допомогою змін моделі (вручну або автоматично).

8.3.3 Проблема жорсткості САК та способи її вирішення

Структурна складність САК, наявність нелінійних характеристик елементів і постійних часу динамічних ланок, що нерідко відрізняються на кілька порядків і залежать від параметрів реальних об'єктів ($\tau_1 = J \cdot \omega_0 / M_n$, $\tau_2 = L / R$, $\tau_3 = R \cdot C$), можуть призвести до появи жорстких (*stiff*) складових у різних частинах моделі системи («просторова жорсткість») і на різних інтервалах часу («часова жорсткість»). Встановити заздалегідь властивість жорсткості без цілеспрямованого вивчення поведінки системи не представляється можливим, тому для нелінійних моделей САК зі складною структурою необхідно використовувати спеціальні стійкі неявні багатокрокові різницеві методи числового моделювання [9].

Взагалі, поведінка жорсткої моделі характеризується ділянками з великими та малими змінами вихідної величини у часі, як показано на рисунку 8.5 [53].

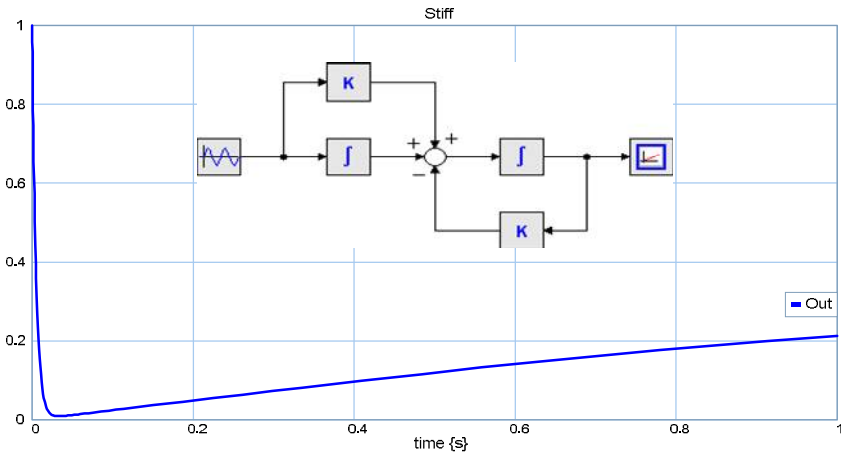


Рисунок 8.5 – Приклад жорсткої моделі: структура і осцилограма

Проблема, представлена на рисунку 8.5 може бути вирішена з використанням практично будь-якого алгоритму інтегрування (явного або неявного), але ефективність (та точність) симуляції буде різною: в 20-sim при моделюванні неявним методом BDF змінного порядку точ-

ності проводиться 263 обчислення моделі, а при інтегруванні явним методом Ейлера першого порядку точності з фіксованим кроком – 1001 обчислення моделі (величина кроку за часом дорівнює $0,001c$, що дозволяє отримати адекватні (але з малою точністю) результати моделювання). Таким чином, після дослідження жорсткої моделі можна зробити висновок, що доцільно застосовувати неявні методи для жорстких задач.

Жорсткість може залежати від:

- власних значень якобіану (узагальнення для нелінійних ODE);
- розмірності системи;
- вимог до точності симуляції;
- тривалості симуляції (діапазону часу модельного часу).

Можна сказати що, поведінка моделі не є жорсткою на інтервалі часу $0,2c < t < 1c$, але є жорсткою в діапазоні $0c < t < 1c$.

8.4 Використання ієрархічного підходу в моделюванні

Складні системи мають, як правило, ієрархічну структуру. Природні можливості людини дозволяють оперативно обробляти не більше 7 ± 2 одиниць інформації одночасно. У процесі проектування зручно зосереджуватися спочатку на поведінці окремих компонентів системи, а потім на їх взаємодії. При необхідності моделі компонентів можуть деталізуватися або, навпаки, укрупнюватися. Такий підхід, відомий як ієрархічне проектування, є типовим при розробці складних об'єктів і полягає в розбитті вихідної задачі на підзадачі.

В процесі проектування складної системи формуються певні уявлення про систему, що відображають її істотні властивості з тим або іншим ступенем детальності. У цих уявленнях можна виділити складові частини – рівні проектування. В один рівень, як правило, включаються уявлення, що мають загальну фізичну основу і допускають для свого опису використання одного і того ж математичного апарата.

Рівні проектування можна виділяти за ступенем детальності, з якою відображаються властивості проєктованого об'єкта. Тоді їх називають горизонтальними (ієрархічними) рівнями проектування. В результаті такого підходу об'єкт проектування піддається декомпозиції на фрагменти (підсхеми) і проектування кожного з них ведеться в певному сенсі самостійно. На кожному рівні ієрархії цей принцип

застосовується знову, що дозволяє замінити рішення однієї складної задачі багаторазовим розв'язанням задач меншої розмірності.

Використання принципу ієрархічного проектування дозволяє обмежити поточну складність проекту на прийнятному рівні через те, що в кожний момент часу проводяться операції тільки з двома суміжними рівнями декомпозиції об'єкта – структурним описом проєктованого в даний момент фрагмента і функціональним описом зовнішнього середовища. Інструментальною підтримкою ієрархічного проектування є багаторівневе моделювання.

При багаторівневому моделюванні різні фрагменти представляються на різних рівнях ієрархії, тобто з різним ступенем детальності. Наприклад, проєктована в даний момент часу частина об'єкта розкрита до рівня елементарних динамічних компонентів і імітується структурною моделлю, а інші фрагменти представлені на сусідньому більш високому рівні у вигляді функціональних моделей.

Завершивши проєктування одного фрагмента, можна згорнути його у функціональний блок і перейти до детальної моделі наступного фрагмента. Ця процедура повторюється багаторазово, на різних рівнях ієрархії проєктованого об'єкта. Перевагою такого підходу є те, що в полі зору перебуває в кожен момент часу мінімум необхідної інформації, не перевантаженої зайвими деталями. Описаний метод проєктування називається методом локальної деталізації об'єкту.

Програмною підтримкою багаторівневого моделювання, реалізованого в більшості мов графічного програмування, є процедура інкапсуляції, яка дозволяє згорнути будь-який смисловий фрагмент графічного представлення в одиничний блок. Крім того, що інкапсуляція служить основою отримання ієрархічно структурованих моделей, вона так само дозволяє розширити бібліотеку базових блоків блоками користувача, які можна багаторазово використовувати (наприклад, типові динамічні ланки). Такою можливістю володіє, зокрема, системи ECAD MC та *MultiSim*, пакет *LabVIEW*, пакет *Simulink* і засновані на ньому пакети *SimMechanics*, *SimPowerSystems*, *20-sim* тощо.

Можливе також використання діакоптичного підходу, при якому попередні та наступні каскади емулюються генераторами та активно-реактивним навантаженням відповідно (досить широко використовується при оцінці початкової схемної надійності).

Контрольні запитання

1. Критерії якості моделей.
2. Критерії ефективності моделювання.
3. Причини в алгоритмічних збоїв ECAD.
4. Методика усунення причин алгоритмічних збоїв.
5. Рекомендації для підвищення точності моделювання.
6. Структурні сингулярності та шляхи розв'язання цієї проблеми в програмах ECAD та CAE.
7. Переваги неявних методів числового інтегрування.
8. Глобальні параметри для симуляції в ECAD.
9. Інкапсуляція, діакоптика.
10. Евристичні методи прийняття рішення (самостійно).

Завдання для практичного заняття/самостійної роботи

1. Зібрати схему із різними постійними часу, промоделювати при різних значеннях Settings, обираючи різні методи інтегрування.
2. Спробуйте різні методи спрощення моделі та навпаки, задавайте надмірну точність (паразитні параметри), порівняйте час моделювання, визначте проблеми симуляції.

ПИТАННЯ ДО ЗАЛІКУ

1. Мета та завдання моделювання.
2. Рівні проєктування і моделювання електронних пристроїв.
3. Особливості математичного забезпечення рівнів проєктування і моделювання ІВТ, програмне забезпечення.
4. Розвиток методів моделювання систем, їх місце у автоматизованому проєктуванні.
5. Математичний базис на різних рівнях проєктування .
6. Критерії якості моделей.
7. Сучасні програми математичного моделювання. Ланка CAE-CAD-CAM.
8. Програми функціонального та системного проєктування .
9. Каузальне моделювання. Програми, підходи.
10. Акаузальне моделювання. Програми, підходи.
11. Процедура каузалізації.
12. Можливості CAE.
13. Порівняння програм класу CAE.
14. Поняття динамічної, мехатронної системи.
15. Моделювання мехатронних систем.
16. Розвиток методів та алгоритмів моделювання.
17. Макромоделі регуляторів
18. Класифікація пристроїв автоматики, їхні макромоделі.
19. Принципи моделювання систем на структурному рівні засобами CAD.
20. Поведінкові макромоделі систем.
21. Види та методика аналізу в програмах CAD.
22. Структура типової програми класу ECAD.
23. Математичне забезпечення ECAD.
24. Наскрізне проєктування .
25. Алгоритми синтезу.
26. Алгоритми аналізу.
27. Алгоритми оптимізації.
28. Підготовка конструкторської документації.
29. Поняття життєвого циклу виробу та віртуального виробництва.
30. Види та методика аналізу в програмах CAE.

31. Структура типової програми класу САЕ.
32. Підпрограми синтезу регуляторів засобами САЕ.
33. Частотні характеристики ПД-регулятора.
34. Роль САЕ програм в системах віртуального виробництва.
35. Інструментальні засоби програм віртуального виробництва.
36. Моделювання систем інформаційної електроніки.
37. Моделювання комп'ютерних мереж засобами NetCracker.
38. Моделювання нейронних мереж.
39. Поведінкові макромоделі, поведінкові алгоритми.
40. Способи «просування часу» в алгоритмах аналізу.
41. Порівняння підходів моделювання аналогових та цифрових систем.
42. Порівняння підходів моделювання: каузального та акаузального.
43. Порівняння підходів моделювання на макрорівні: системний та структурний підходи.
44. Моделювання сенсорів.
45. Проблеми моделювання систем керування із мікроконтролерами / мікропроцесорами.
46. Принципи імітаційного моделювання систем масового обслуговування.
47. Проблеми точності моделювання.
48. Проблема економічності моделювання.
49. Напрями підвищення ефективності моделювання.
50. Рекомендації по вибору оптимального програмного та математичного забезпечення (згідно індивідуальному завданню).

ЗМІСТ ТА ОБСЯГ КОНТРОЛЬНИХ РОБІТ

1. У **вступі** актуалізується задача моделювання (розробки моделі та симуляції) обраного об'єкту дослідження, вказуються шляхи та етапи (у послідовності) досягнення цілей дослідження.

2. **Теоретична частина** контрольної роботи містить основні відомості та методики моделювання інформаційно-вимірювальних систем, САК та САР на функціональному та системному рівнях, особливості моделей та макромоделей на даних рівнях. В даному розділі роботи необхідно проаналізувати об'єкт, визначити мету дослідження та розробити концептуальну модель, прийняти рішення стосовно способу моделювання (каузальне, або акаузальне) та обрати згідно певних критеріїв програмне забезпечення для віртуального експерименту.

3. В **розрахунковій частині** подаються схеми, рівняння або інші способи представлення моделей, які обираються для моделювання інформаційно-вимірювальних систем. Приймається рішення стосовно інструментального набору для адаптації розробленої моделі до умов обраного програмного середовища. Розділ має містити результати модельного експерименту – характеристики, таблиці, або інші форми представлення інформації, на основі яких приймається рішення про ефективність моделі за обраними критеріями якості (точність, універсальність, економічність, ергономічність) та про необхідність параметричної та структурної оптимізації моделі. Моделлю може бути схема заміщення, система рівнянь, граф, алгоритм, таблиця істинності/переходів, концептуальний опис.

4. **Висновки** складаються за результатами експериментів, причому необхідно проаналізувати кількісно точність, якісно - економічність обраних моделей та їхню універсальність, показати межі адекватності моделі, напрямки її вдосконалення та перспективи використання.

5. **Список посилань** (мінімум 10 джерел) складається в порядку появи посилань за текстом згідно правил.

Всі прийняті рішення в контрольній роботі повинні бути обгрунтованими. Контрольна робота має бути захищена впродовж семестру. Робота повинна бути оформлена згідно норм і правил ДСТУ3008:2015. Обсяг роботи – 20 аркушів А4 (мінімум).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Laboratories, Sandia & James, Dr & Campbell, E & Anderson, Dennis & Lawton, Craig & Shirah, Donald & Dennis, Dr & Longsine, E. (2005). System of systems modeling and simulation. URL: https://www.researchgate.net/publication/228857911_System_of_systems_modeling_and_simulation
2. Обод І. І., Заволодько Г. Е., Свид І. В. Математичне моделювання систем: навчальний посібник. – Харків: Друкарня МАДРИД, 2019. 268 с.
3. Василенко, О.В. Навчальний посібник з дисципліни «Комп'ютерне моделювання» для студентів спеціальності 152. НУ «Запорізька політехніка, 2020. 175 с.
4. 20-sim 4.3 ReferenceManual / Ir.C. Kleijn, Ir.M.A. Groothuis, Ir.H.G. Differ // ControllabProducts B.V.– 1099 p. URL: <https://books.google.al/books?id=zoLnXeaS5iMC&printsec=frontcover#v=onepage&q&f=false>
5. Системи автоматизованого проектування: конспект лекцій [Електронний ресурс]: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», спеціалізації «Комп'ютерноінтегровані системи та технології в приладобудуванні» / КПІ ім. Ігоря Сікорського; автори: К.С. Барандич, О.О. Подолян, М.М. Гладський. – Електронні текстові дані (1 файл 3,05 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. 97 с.
6. Dynamic System Model. URL: <https://www.sciencedirect.com/topics/computer-science/dynamic-system-model>
7. What Is the Difference Between CAD, CAE and CAM? URL: <https://online.egr.msu.edu/articles/cad-vs-cae-vs-cam-what-is-the-difference/>
8. What is System Dynamics? URL: https://systemdynamics.org/what-is-system-dynamics/?gad_source=1&gad_campaignid=22532377806&gbraid=0AAA AACVfjiQhzzLvG4SurIOTiXi8l395E&gclid=CjwKCAjwqubPBhBOEiwAzgZX2kZLMxw1JI1UcjLCbr0LCFCAL-z4kjpgvXYSgouafuUD2DoQHnK0mLhoCcJ8QAvD_BwE
9. Shmatko, O., Herasymov, S., Milevskiy, S., Balitskiy, N., Pohasii, S., Aleksieiev, M., Vlasov, I., Melenti, Y., Kovalenko, Y., &

- Peleshok, Y. (2025). Development of a method for assessing the efficiency of technical systems' computer dynamic simulators. *Eastern-European Journal of Enterprise Technologies*, 2 (134), 50–61.
<https://doi.org/10.15587/1729-4061.2025.327558> URL:
<https://journals.uran.ua/eejet/article/view/327558>
10. Simulink. URL:
<https://uk.mathworks.com/products/simulink.html>
11. Василенко О.В. Методичні вказівки до лабораторних занять з дисципліни «Комп'ютерне моделювання». Запоріжжя: НУ «Запорізька політехніка, 2020. 72 с.
12. Micro-Cap 12, Electronic Circuit Analysis Program. Reference Manual, Spectrum Software URL:
<https://gotroot.ca/spectrum/www.spectrum-soft.com/index-2.html>
13. Simulation Tools. URL:
http://www.idsia.ch/~andrea/Andrea_Rizzoli_Home_Page/Sim_Tools.html
14. The HDL Handbook. URL:
<https://www.sciencedirect.com/book/edited-volume/9780123821713/the-hdl-handbook>
15. VHDL як алгоритмічна мова для опису паралельних алгоритмів. URL: <https://kanyevsky.kpi.ua>
16. SimApp, Ordering Information. URL:
<http://www.simapp.com/order-simapp.php>
17. Differential Algebraic System Solver. URL:
<http://acronyms.thefreedictionary.com/DASSL>
18. CVODES. URL:
<https://computing.llnl.gov/projects/sundials/cvodes>
19. MapleSim by Maplesoft. Advanced System Level Modeling and Simulation URL: <https://altair.com/maplesim>
20. Simscape. Model and simulate multidomain physical systems. URL:
<http://www.mathworks.com/products/simscape/index.html>
21. MathWorksStore. URL:
<http://www.mathworks.com/store/product>
22. MathWorks. URL:
<https://uk.mathworks.com/>
23. MathWorks Documentation Center, SolverPane. URL:
<http://www.mathworks.com/help/simulink/gui/solver-pane.html>
24. VisSim/Altair Embed. URL: <https://altair.com/embed/>

25. Floudas, C. A., Pardalos, P. M., (Eds.) Encyclopedia of Optimization, (2nd ed. 2009); Springer-Verlag NewYork, LLC, 2009. 4626p.: 613 illus.

26. VisSim/Altair Embed trial. URL: <https://altairone.com/Marketplace?app=Embed>

27. The Resource Library is your hub for Altair content. URL: <https://altair.com/resourcelibrary/?category=Webinars&product=Altair%20Embed&lang=English>

28. Altair Embed® Applications. URL: <https://altair.com/embed-applications>

29. Dymola Dynamic Modeling Laboratory. User's Manual. URL: <https://people.inf.ethz.ch/cellier/Lect/MMPS/Refs/Dymola5Manual.pdf>

30. Versions of 20-sim. URL: <http://www.20sim.com/home/versions.html>

31. Василенко О.В. Аналіз програм для моделювання мехатронних систем. *Радіоелектроніка, інформатика, управління*, 2015. №2. С.16-31.

32. Трет'як А.В. Основи робототехніки: навчальний посібник для студентів спеціальностей 133 «Галузеве машинобудування», 141 «Електроенергетика, електротехніка та електромеханіка», 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» / А.В. Трет'як, А.М. Кльон. – Полтава, видавництво національного університету «Полтавська політехніка імені Юрія Кондратюка», 2024. 135 с.

33. Build a Robot Step by Step. URL: <https://uk.mathworks.com/help/robotics/ug/build-a-robot-step-by-step.html>

34. Василенко О.В. Modeling of multidomain automatic control Systems in ECAD. *Вісник Академії митної служби України. Серія: Технічні науки*, 2015. №1. С.13-19.

35. Приймак, Б. І. Теорія автоматичного керування. Лінійні системи. Навчальний посібник. К. : КПІ, 2023. – 310 с.

36. PowerEsim Switch Mode Power Supply Design Tool. URL: <https://www.infineon.com/design-resources/simulation-modeling/poweresim-switch-mode-power-supply-simulation-tool>

37. Загальна електротехніка і основи електроніки: навчальний посібник / Співак. В.М., Гуржий А.М., Нельга А.Т., Іт'якін О.С.– Київ: КПІ, 2020. 266 с.

38. Vasylenko O.V. PWM controller's models for investigation ACS in SPICE-family ECAD programs [Text] / O.V. Vasylenko, G.V. Snizhnoi // *Електротехніка та електроенергетика*, 2018. № 1. С.64-71.

39. Vasylenko O. Simulation of ACS for Magnetic Susceptibility Measurements in ECAD Based on Time Domain Functions / O. Vasylenko, V. Reva, G. Snizhnoi [Electronic resource] // *Proceeding soft the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019)*, Zaporizhzhia, Ukraine, April 15–19, 2019 (CEUR Workshop Proceedings, Vol. 2353, P. 689–701). URL: <http://ceur-ws.org/Vol-2353/paper55.pdf>

40. Simulink for System Modeling and Simulation. URL: <https://uk.mathworks.com/solutions/system-modeling-simulation.html>

41. Інтервальний аналіз. URL: https://nmetau.edu.ua/file/07_7.6_lbr_gr_rbr_.pdf

42. Переверзев А.В., Василенко О.В. Моделювання в електроніці. Учбовий посібник для студентів напряму «Електроніка». – Запоріжжя: ЗДІА, 2003. – 160 с.

43. Cellier F.E., Kofman E. Continuous system simulation, Springer Verlag, New York, 2006. 643 p.

44. Литвинов А. Л. Теорія систем масового обслуговування: навч. посібник / А. Л. Литвинов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. Харків: ХНУМГ ім. О. М. Бекетова, 2018. 141 с.

45. Netcracker. URL: www.netcracker.com

46. Гайтан О.М. Порівняльна характеристика програмних середовищ моделювання систем масового обслуговування. URL: <https://journals.nupp.edu.ua/mist/uk/article/view/483>

47. Штучні нейронні мережі. URL: <https://itmaster.biz.ua/programming/vision/neural-networks.html>

48. І.В.Шелевицький, М.О.Шутко, В.М.Шутко, О.О. Колганова. Сплайни в цифровій обробці даних і сигналів, 2007. 232с., іл. ISBN 078-966-2915-86-0

49. Василенко, О.В. Вибір методу оптимізації систем автоматичного керування в системах автоматизованого інжинірингу / О.В. Василенко, Я.І. Петренко // *Енергетика і автоматика*, 2017. №1. С.75-89.

50. What is PID? Learn how PID controls work in real industrial control systems. URL:

<https://www.valmet.com/automation/services/process-performance/pid-tuning-process-control/pid-tuning-tutorial/>

51. Ziegler J.G., Nichols N.B. Optimum settings for automatic controllers. URL:

https://skoge.folk.ntnu.no/publications_others/1942_ziegler-nichols.pdf

52. GLOPT – A Program for Constrained Global Optimization. URL: https://www.mat.univie.ac.at/~herman/papers/glopt_k.pdf

53. Pantelides C. The Consistent Initialization of Differential-Algebraic Systems. SIAM Journal of Scientific and Statistical Computing. URL: <https://epubs.siam.org/doi/10.1137/0909014>

54. The Index of General Nonlinear DAEs. URL: <https://link.springer.com/article/10.1007/s002110050165> 72:173-196, 1995.

55. Petrenko Y.I. Research of algorithms and development of modeling and simulation technique of automatic control systems in 20-sim/ Матеріали XVII НТК студентів, магістрантів, аспірантів і викладачів ЗДІА, 2013. С. 48-49.

56. Василенко, О.В. Модель крокового двигуна для дослідження систем автоматичного позиціонування в ECAD / О.В. Василенко, Є.Л. Жавжаров // *Електротехніка та електроенергетика*, 2017. №1. С.31-38.

57. Bond Graphs in Dynamic System Modeling. URL: <https://www.scribd.com/document/505453781/Bond-Graph-Modelling-introduction-full-version-2014>

58. Методичні вказівки до лабораторних занять з дисципліни «Системи автоматичного керування» для студентів спеціальностей: 175 «Інформаційно-вимірювальні технології», освітня програма: «Інформаційні системи моніторингу і контролю»; 176 «Мікро- та наносистемна техніка», освітня програма: «Мікро- та наноелектронні прилади і пристрої» першого (бакалаврського) рівня вищої освіти денної й заочної форм навчання / Укл.: Ольга Василенко. – Запоріжжя: НУ «Запорізька політехніка», 2025. 49 с.

ДОДАТОК А

Вхідний контроль, дисципліна «Комп'ютерне моделювання»

Варіант 1

- | | | |
|----|------------------------------------|----------|
| 1. | Що таке система? | 20 балів |
| 2. | Які програми для моди знаєте? | 20 балів |
| 3. | Основні логічні функції. | 20 балів |
| 4. | Що таке рівняння динаміки системи? | 20 балів |
| 5. | Що таке аналіз? | 20 балів |

Варіант 2

- | | | |
|----|-------------------------------|----------|
| 1. | Що таке синтез? | 20 балів |
| 2. | Що таке підсистема? | 20 балів |
| 3. | Що вивчає алгебра логіки? | 20 балів |
| 4. | Що таке система? | 20 балів |
| 5. | Що таке функціональний вузол? | 20 балів |

Варіант 3

- | | | |
|----|---|----------|
| 1. | Чим відрізняються поняття регулювання та керування? | 20 балів |
| 2. | Що таке прямий зв'язок? | 20 балів |
| 3. | Що таке інформаційно-вимірювальна система? | 20 балів |
| 4. | Що таке елемент системи? | 20 балів |
| 5. | Що таке неперервні САК? | 20 балів |

Варіант 4

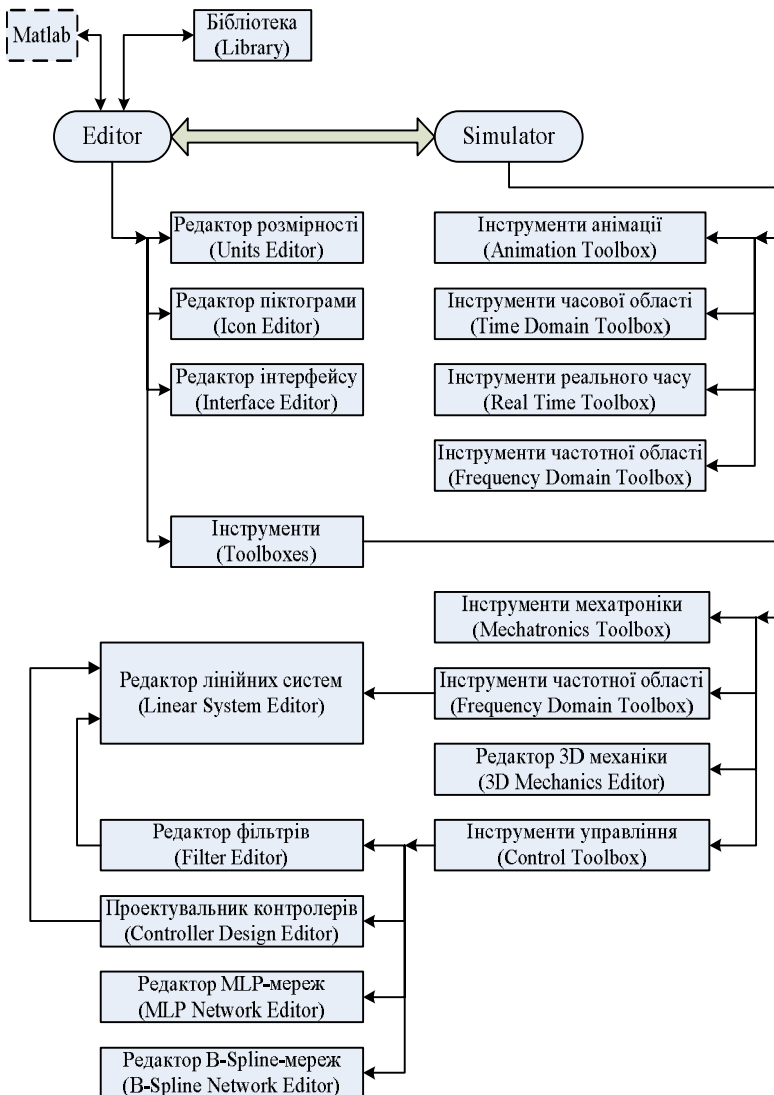
- | | | |
|----|------------------------------------|----------|
| 1. | Що таке фаза проєктування ? | 20 балів |
| 2. | Що таке динамічна система? | 20 балів |
| 3. | Що таке перехідна функція системи? | 20 балів |
| 4. | Що таке зворотний зв'язок? | 20 балів |
| 5. | Що таке дискретні САК? | 20 балів |

Варіант 5

- | | | |
|----|--|----------|
| 1. | Статичні характеристики основних блоків ІВТ. | 20 балів |
| 2. | Що таке передавальна функція системи? | 20 балів |
| 3. | Як змінюється сигнал в аналогових системах? | 20 балів |
| 4. | Що таке системний аналіз? | 20 балів |
| 5. | Що таке автоматична система? | 20 балів |

ДОДАТОК Б

Структура САЕ



ДОДАТОК В

Таблиця В.1 - Порівняння можливостей САЕ

| Особливості | Інструменти | | | | | | | | | | |
|-----------------------------|-------------|----------|--------|----------|--------|--------|--------|----------|-------------|-------|--------------|
| | ACSLx | O-Matrix | MATLAB | Simscape | 20-sim | AMESim | Dymola | MapleSim | SimulationX | EASY5 | MathModelica |
| Моделювання | | | | | | | | | | | |
| процедурне | + | + | + | + | + | + | + | + | + | + | + |
| на основі рівнянь | + | - | + | + | + | + | + | + | + | - | + |
| каузальне | + | - | + | + | + | + | + | + | + | + | + |
| акаузальне | - | - | + | + | + | + | + | + | + | - | + |
| Мова програмування | | | | | | | | | | | |
| власна | + | + | + | + | + | + | - | + | + | + | - |
| Modelica | - | - | - | - | - | + | + | + | + | - | + |
| Бібліотеки | | | | | | | | | | | |
| блок-схеми (Control/Signal) | + | - | + | + | + | + | + | + | + | + | + |
| електричні | - | - | + | + | + | + | + | + | + | + | + |
| термічні | - | - | + | + | + | + | + | + | + | + | + |
| гідравлічні | - | - | + | + | + | + | + | + | + | + | + |
| механічні (1D) | - | - | + | + | + | + | + | + | + | - | + |
| механічні (3D), Multi-body | - | - | + | + | + | + | + | + | + | + | + |
| автомобільних систем | + | - | + | + | - | + | + | + | + | - | + |

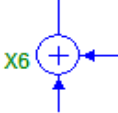

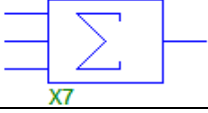
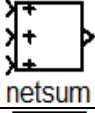
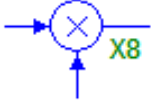
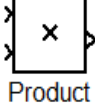


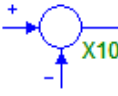
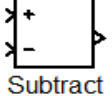
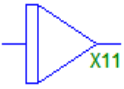

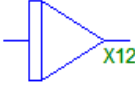

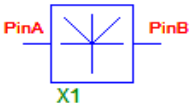
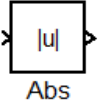
Продовження таблиці В.1

| Особливості | Інструменти | | | | | | | | | | |
|------------------------------------|-------------|----------|--------|----------|--------|--------|--------|----------|-------------|-------|--------------|
| | ACSLx | O-Matrix | MATLAB | Simscape | 20-sim | AMESim | Dymola | MapleSim | SimulationX | EASY5 | MathModelica |
| Взаємодія | | | | | | | | | | | |
| експортування коду | + | - | + | + | + | + | + | + | + | + | + |
| імпорт з Simulink | - | - | | | - | + | - | + | + | + | - |
| експорт до Simulink | + | - | | | + | + | + | + | + | + | - |
| Simulink co-simulation | - | - | | | + | + | - | - | + | + | - |
| експорт до ADAMS | - | - | - | + | + | + | - | - | - | + | - |
| Microsoft.NET | - | - | + | + | + | + | - | + | + | + | + |
| Можливості програмування | | | | | | | | | | | |
| обмежені | | | | | | | • | | | | |
| середні | | | | | • | • | | | • | • | |
| широкі | • | • | • | • | | • | | • | | | • |
| попередня обробка | + | + | + | + | + | + | - | + | + | - | + |
| підтримка після обробки | + | + | + | + | + | + | + | + | + | - | + |
| Інтерактивні веб-публікації | | | | | | | | | | | |
| | - | - | + | + | - | - | - | + | - | - | - |

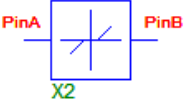

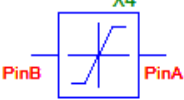

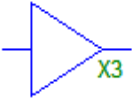




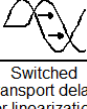

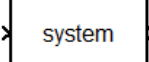
ДОДАТОК Г

Умовні зображення блоків для моделювання САК в MICRO-CAP та MatLab

Таблиця Г.1 - Умовні зображення блоків для моделювання

| N | Назви в програмах | MC12 | SIMULINK |
|---|---|---|---|
| 1 | Суматор <i>Sum-Sum</i> |  |  |
| 2 | 3-ходовий суматор <i>Sum3-Sumofelement</i> |  |  |
| 3 | Мультиплікатор <i>Mul-Product</i> |  |  |
| 4 | Дільник <i>Div-Divide</i> |  |  |
| 5 | Субтрактор <i>Sub-Subtract</i> |  |  |
| 6 | Інтегратор <i>Int-Integrator</i> |  |  |
| 7 | Диференціатор <i>Dif-Derivative</i> |  |  |
| 8 | Абсолютне значення <i>ABS- ABS</i> |  |  |

Продовження таблиці Г.1

| N | Назви в програмах | MC12 | SIMULINK |
|----|--|---|---|
| 9 | Зона нечутливості <i>Slip-Dead zone</i> |  |  |
| 10 | Амплітудний обмежувач <i>Clip-Saturation</i> |  |  |
| 11 | Лінійний Підсилювач <i>Amp-Gain</i> |  |  |
| 12 | Генератор шуму <i>Noise-UniformRandomNumber</i> |  |  |
| 13 | Затримка сигналу <i>Delay-Transportdelay</i> |  |  |
| 14 | Передаточна функція <i>F-S Function</i> |  |  |

ДОДАТОК Д

Класифікація моделей імпульсних регуляторів в Micro-Cap

Таблиця Д.1 - Класифікація моделей імпульсних регуляторів в MC12

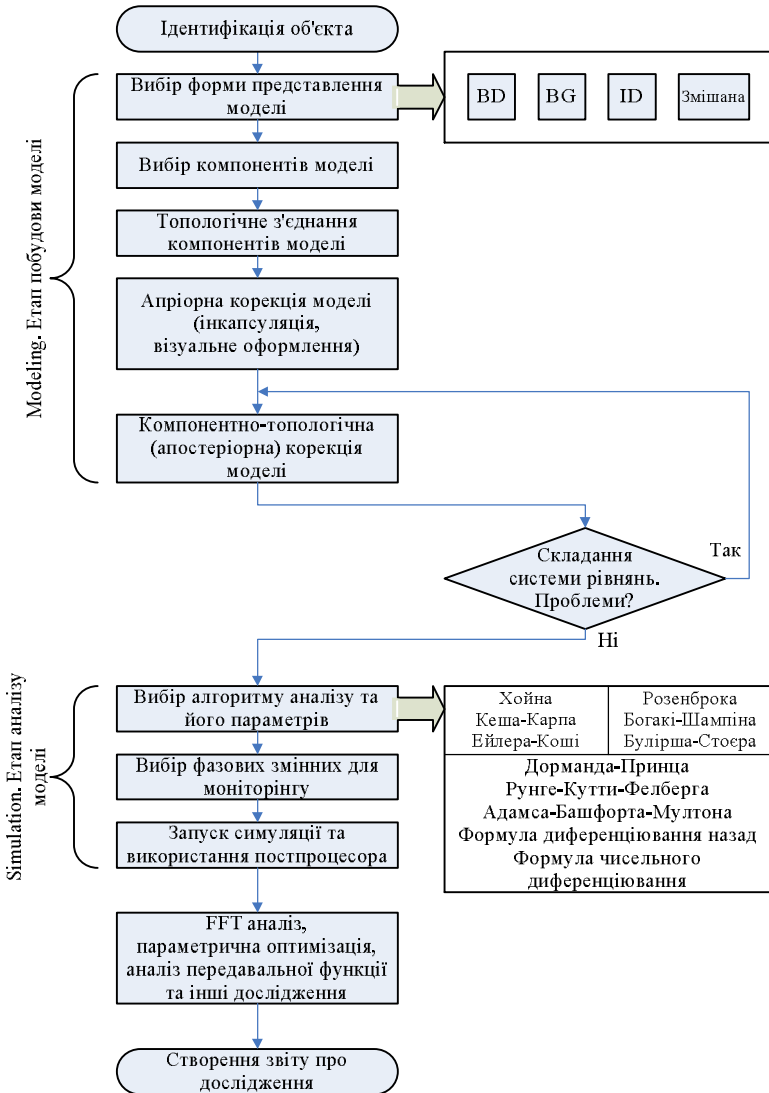
| Ім'я файлу схеми | Тип перетворювача | Модель ядра |
|------------------|--|-------------|
| S_2FLY_CM | Зворотноходовий двохключевий | 2SWITCHCM |
| S_2FOR_CM | Прямоходовий двохключевий | 2SWITCHCM |
| S_BOOST_CM | Підвищувальний (контроль за струмом) | PWMCM |
| S_BOOST_VM | Підвищувальний (контроль за напругою) | PWMVM |
| S_BUCK_CM | Знижувальний із ключем та діодом | PWMCM |
| S_BUCK_SYN | Знижувальний синхронний із <i>p</i> - і <i>n</i> -канальними MOSFET | PWMVM |
| S_BUCK_SYN2 | Знижувальний синхронний із <i>p</i> - і <i>n</i> -канальними MOSFET | PWMCM |
| S_BUCK_VM | Знижувальний із <i>n</i> -канальним MOSFET і діодом (контроль за напругою) | PWMVM |
| S_BUCKBOOST_CM | Знижувальний інвертуючий ключом та діодом (контроль за струмом) | PWMCM |
| S_BUCKBOOST_VM | Знижувальний інвертуючий із ключом та діодом (контроль за напругою) | PWMVM |
| S_FLYBACK_CM | Зворотноходовий одноключевий (контроль за струмом) | PWMVM |

Продовження таблиці Д.1

| Ім'я файлу схеми | Тип перетворювача | Модель ядра |
|------------------|--|-------------|
| S_FLYBACK_VM | Зворотноходовий одноключевий (контроль за напругою) | PWMVM |
| S_FORWARD_CM | Прямоходовий одноключевий (контроль за струмом) | PWMCM |
| S_FORWARD_VM | Прямоходовий одноключевий (контроль за напругою) | PWMVM |
| S_FULL_CM | Мостова із безпосереднім керуванням ключами (контроль за струмом) | FULL_CM |
| S_FULL_VM | Мостова із безпосереднім керуванням ключами (контроль за напругою) | FULL_VM |
| S_FULL_XFMR | Мостова, верхні ключі управляються ч/з трансформатор | PUSH_CM |
| S_HALF_CM | Напівмостова із безпосереднім керуванням ключами (ко нтроль за струмом) | HALF_CM |
| S_HALF_VM | Напівмостова із безпосереднім керуванням ключами (ко нтроль за напругою) | HALF_VM |
| S_HALF_XFMR | Напівмостова, верхні ключі керується ч/з трансформатор | PUSH_CM |
| S_PUSH_CM | Двохтактна прямоходова (контроль за струмом) | PUSH_CM |
| S_PUSH_VM | Двохтактна прямоходова (контроль за напругою) | PUSH_VM |

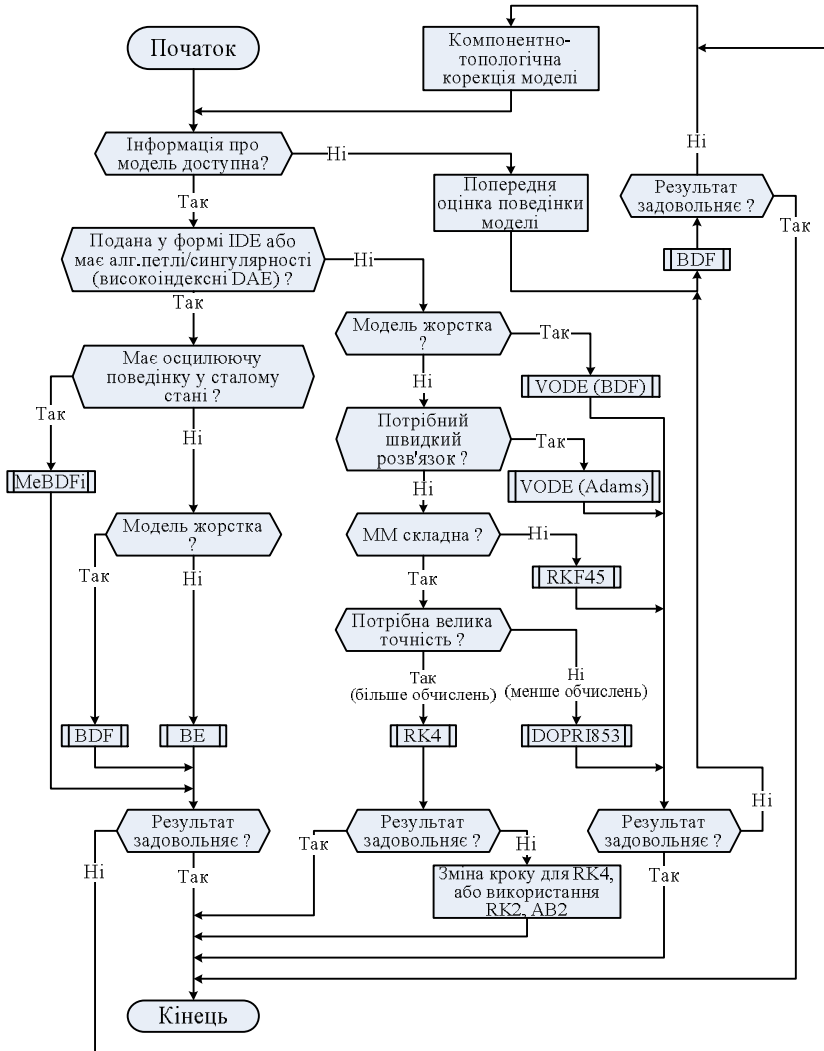
ДОДАТОК Е

Згорнутий алгоритм моделювання САК



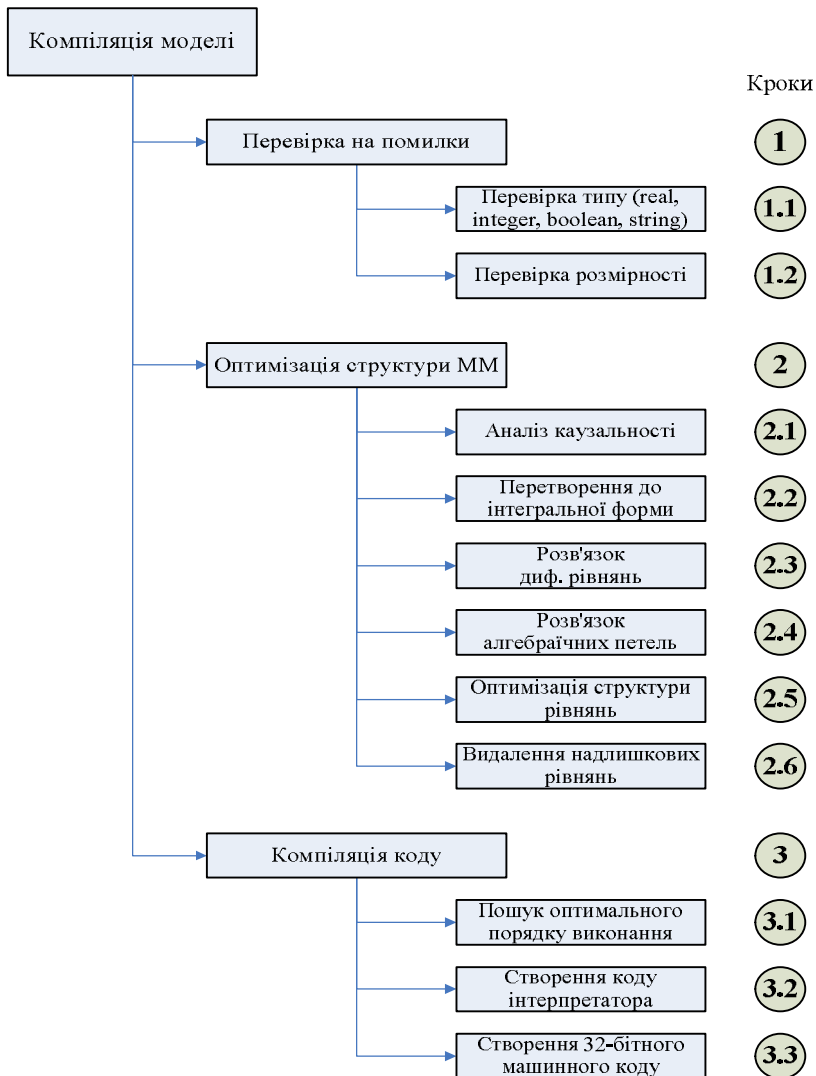
ДОДАТОК Ж

Алгоритм вибору методу інтегрування в 20-sim



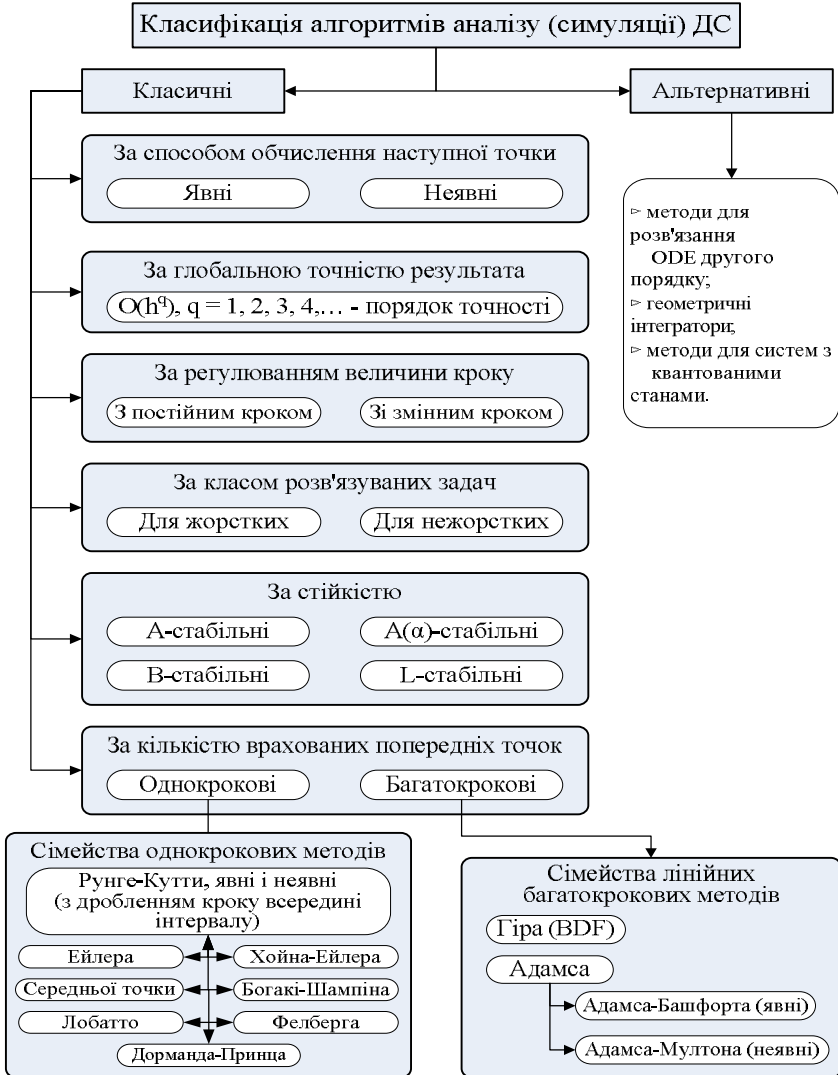
ДОДАТОК 3

Процедури етапу компіляції моделі в САЕ



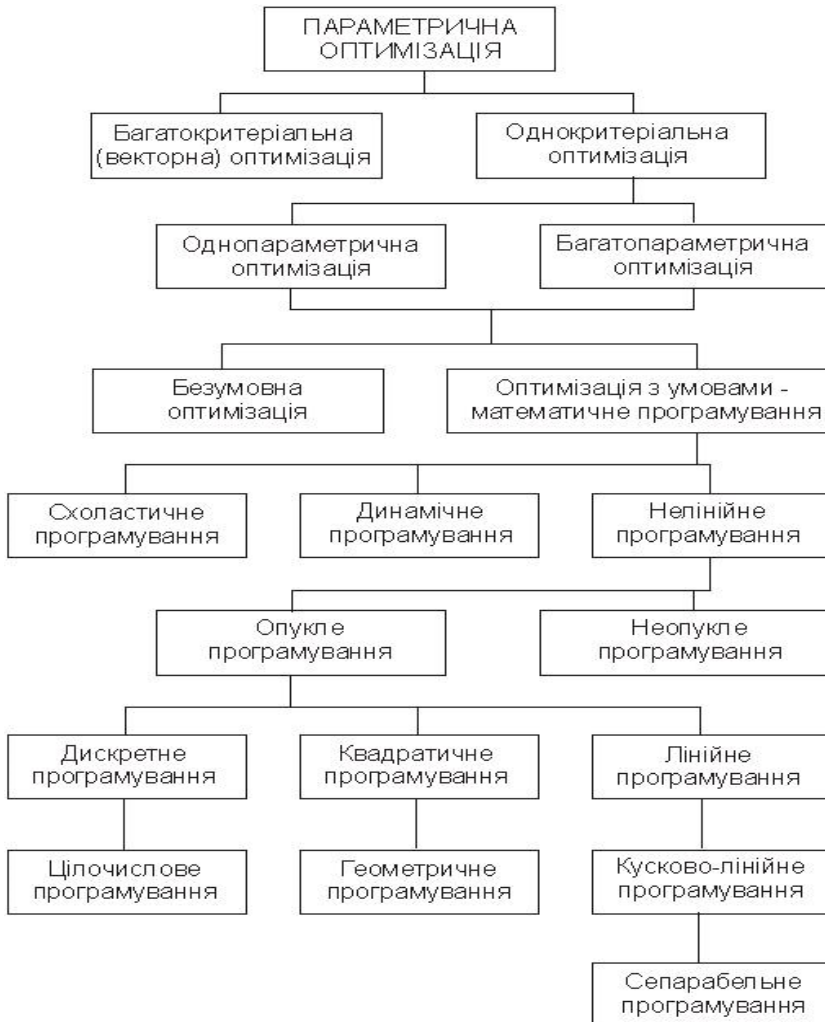
ДОДАТОК І

Класифікація методів симуляції



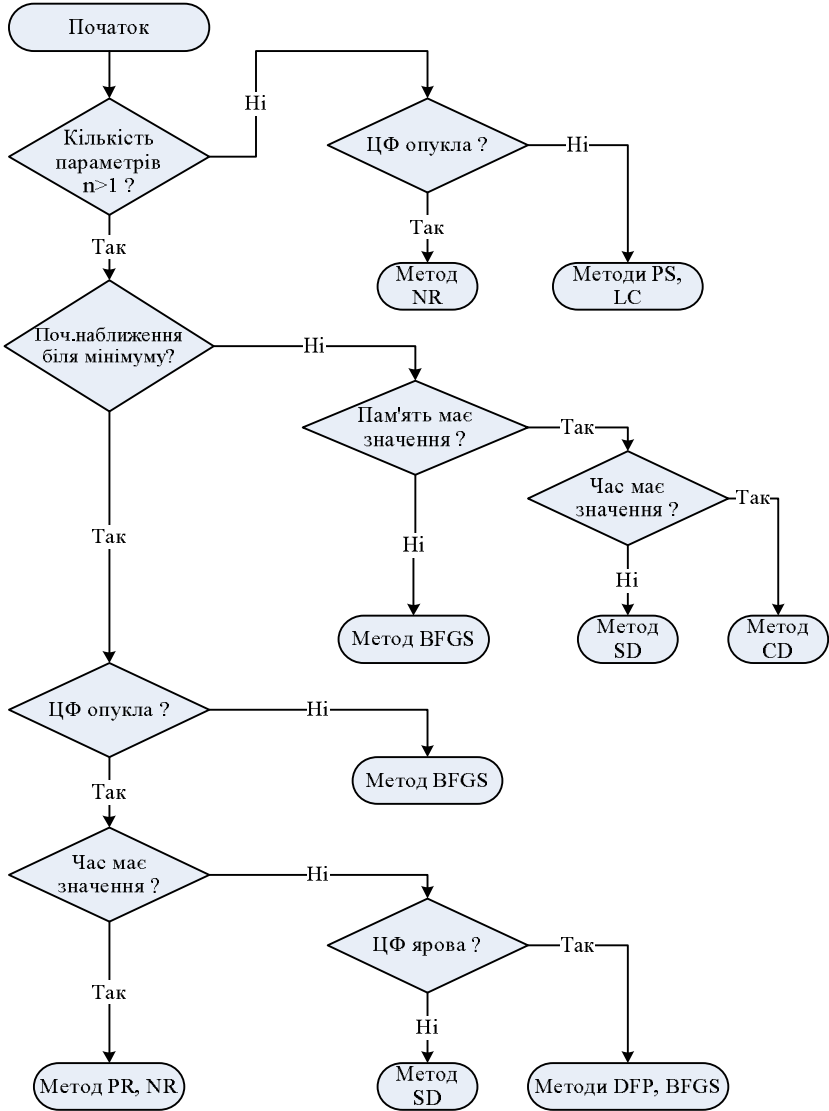
ДОДАТОК К

Класифікація задач оптимізації



ДОДАТОК Л

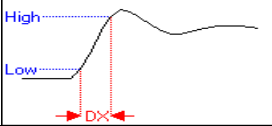
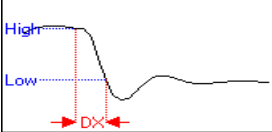
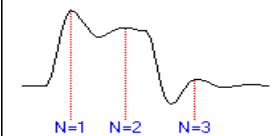
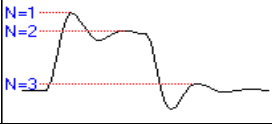
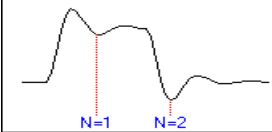
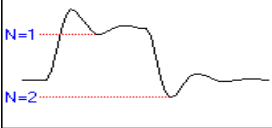
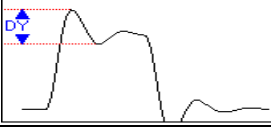
Алгоритм вибору методу оптимізації в 20-sim



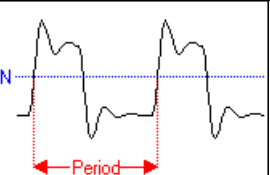
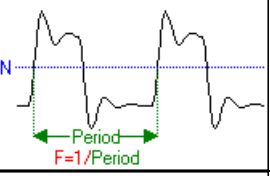
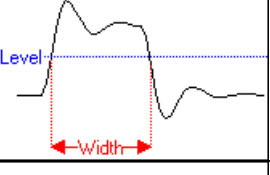
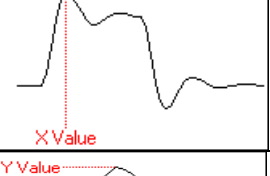
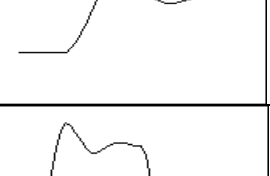
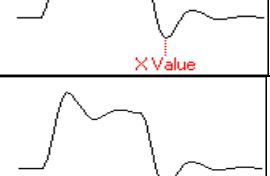
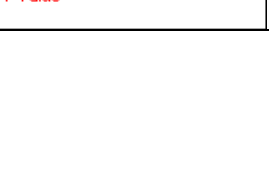
ДОДАТОК М

Параметри, що оптимізуються в MICRO CAP

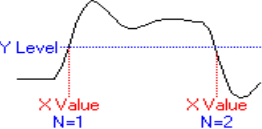
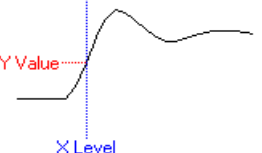

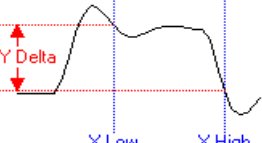

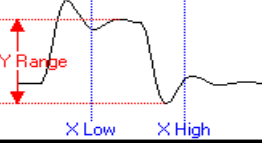
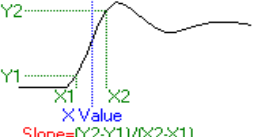
Таблиця М.1- Параметри, що оптимізуються

| Ілюстрація | Функція | Опис |
|---|--------------------|---|
|  | <i>Rise_Time</i> | Знайти зміни у виразі X , коли вираз Y збільшується |
|  | <i>Fall_Time</i> | Знайти зміни у виразі X коли вираз Y зменшується |
|  | <i>Peak_X</i> | Знайти n -ий локальний максимум у виразі X |
|  | <i>Peak_Y</i> | Знайти n -ий локальний максимум у виразі Y |
|  | <i>Valley_X</i> | Знайти n -ий локальний мінімум у виразі X |
|  | <i>Valley_Y</i> | Знайти n -ий локальний мінімум у виразі Y |
|  | <i>Peak_Valley</i> | Знайти n -ий локальний пік западини у виразі Y |

Продовження таблиці М.1

| Ілюстрація | | Функція | Опис |
|---|--|------------------|--|
|  | | <i>Period</i> | Знайти n -ий період у виразі Y . |
|  | | <i>Frequency</i> | Знайти n -у частоту у виразі Y . |
|  | | <i>Width</i> | Знайти $n-i$ X відмінності в послідовності крапок, що проходять через рівень Y |
|  | | <i>High_X</i> | Знайти найбільше значення у виразі X у виділеній частині |
|  | | <i>High_Y</i> | Знайти найбільше значення у виразі Y у виділеній частині |
|  | | <i>Low_X</i> | Знайти найменше значення у виразі X у виділеній частині |
|  | | <i>Low_Y</i> | Знайти найменше значення у виразі Y у виділеній частині |

Кінець таблиці М.1

| Ілюстрація | Функція | Опис |
|---|------------|--|
|  | X_Level | Знайти значення X в n -ому випадку, де вираз $Y =$ вказаному рівню Y |
|  | Y_Level | Знайти значення Y в n -ому випадку, де вираз $X =$ вказаному рівню X |
|  | X_Delta | Знайти різницю X в n -ому випадку діапазону виразу Y |
|  | Y_Delta | Знайти різницю Y в n -ому випадку діапазону виразу X |
|  | X_Range | Знайти межі $X_{\min-\max}$ в межах n -го випадку вказаного діапазону |
|  | Y_Range | Знайти межі $Y_{\min-\max}$ в межах n -го випадку вказаного діапазону X |
|  | $Slope$ | Знайти крутизну нахилу (дельта Y /дельта X) у n -ому випадку вказаного значення X |