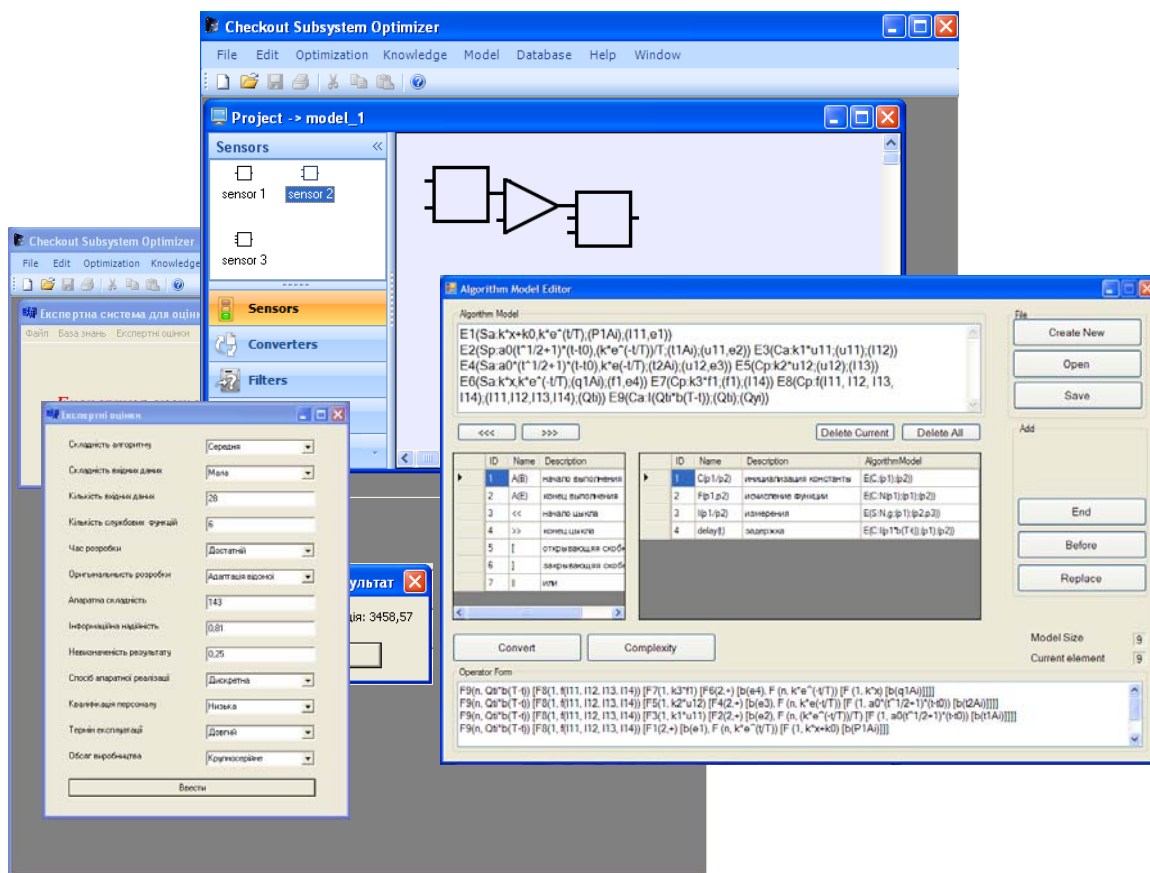


В. М. Дубовой, О. Д. Никитенко

ОПТИМІЗАЦІЯ ПІДСИСТЕМ ЗБОРУ ДАНИХ АСУТП В УМОВАХ КОМБІНОВАНОЇ НЕВИЗНАЧЕНОСТІ



Міністерство освіти і науки, молоді та спорту України
Вінницький національний технічний університет

В. М. Дубовой, О. Д. Никитенко

**ОПТИМІЗАЦІЯ ПІДСИСТЕМ ЗБОРУ ДАНИХ АСУТП
В УМОВАХ КОМБІНОВАНОЇ НЕВИЗНАЧЕНОСТІ**

Монографія

УНІВЕРСУМ – Вінниця
2011

УДК 004.942:004.051:004.021

Д 79

Рецензенти:

А. А. Тунік, доктор технічних наук, професор,

Р. Н. Квєтний, доктор технічних наук, професор

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки, молоді та спорту України (протокол № __ від __.__.2011 р.)

Дубовой В. М., Никитенко О. Д.

Д 79 Оптимізація підсистем збору даних АСУТП в умовах комбінованої невизначеності. Монографія. – Вінниця: УНІВЕРСУМ-Вінниця, 2011. – 169 с.

ISBN

В монографії розвинуто інформаційну технологію оптимізації структури підсистем збору даних АСУТП з метою підвищення їх ефективності в умовах комбінованої невизначеності. Моделювання і проектування структури підсистем збору даних здійснюється на основі алгоритмічної моделі. Розроблений метод моделювання підсистем збору даних в умовах невизначеності базується на формальній системі алгоритмічних моделей і використовує операторне подання перетворень невизначених даних.

Робота розрахована на науковців, інженерно-технічних працівників, аспірантів та студентів, які займаються питаннями проектування підсистем збору даних АСУТП.

УДК 004.942:004.051:004.021

ISBN

© В. Дубовой, О. Никитенко, 2011

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	5
ВСТУП.....	6
1. АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА ТЕХНОЛОГІЙ СТРУКТУРНОГО ПРОЕКТУВАННЯ ПІДСИСТЕМ ЗБОРУ ДАНИХ.....	8
1.1. Проблеми моделювання і проектування структур підсистем збору даних.....	8
1.1.1.Роль підсистем збору даних у підвищенні ефективності АСУ ТП	9
1.1.2. Види підсистем збору даних.....	11
1.1.3.Системи структурного проектування підсистем збору даних	15
1.2. Моделі підсистем збору даних.....	18
1.3. Застосування алгоритмічного підходу до дослідження систем	22
1.3.1. Змістовні логічні схеми алгоритмів як апарат формалізованого опису алгоритмів роботи підсистем збору даних.....	23
1.3.2. Алгоритми обчислювальних процесів та алгоритмічна алгебра.	25
1.3.3. Методи оцінювання характеристик підсистем збору даних.....	31
1.4. Методи структурного моделювання систем в умовах невизначеності.....	37
2. РОЗРОБКА МЕТОДУ ПОБУДОВИ АЛГОРИТМІЧНИХ МОДЕЛЕЙ ПІДСИСТЕМ ЗБОРУ ДАНИХ.....	42
2.1. Комплекс моделей інформаційної технології структурної оптимізації підсистем збору даних АСУТП.....	42
2.2. Розробка методу моделювання програмно-апаратного комплексу підсистем збору даних	44
2.2.1. Формальна система алгоритмічних моделей.....	44
2.2.2.Алгебра алгоритмічних моделей.....	51
2.3. Перетворення алгоритмічних моделей в умовах невизначеност	52
2.3.1. Отримання операторної форми алгоритмічної моделі.....	53
2.3.2. Еквівалентність алгоритмічних моделей в умовах невизначеності	60
2.4. Оцінювання характеристик підсистем збору даних.....	65
2.4.1. Оцінювання складності.....	66

2.4.2.Оцінювання невизначеності результатів збору даних.....	76
2.4.3.Оцінювання швидкодії.....	81
2.4.4.Оцінювання надійності.....	83
2.4.5. Комплексна оцінка ефективності підсистем збору даних	88
2.4.6. Метрики у просторі структур підсистем збору даних	94
3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПІДСИСТЕМ ЗБОРУ ДАНИХ В УМОВАХ НЕВИЗНАЧЕНОСТІ НА ОСНОВІ АЛГОРИТМІЧНИХ МОДЕЛЕЙ.....	97
3.1. Концептуальна структура інформаційної технології.....	97
3.2. Оптимізація підсистем збору даних.....	102
3.2.1.Оптимізація в умовах повної визначеності.....	104
3.2.2.Оптимізація в умовах невизначеності.....	114
3.3. Модель розвитку підсистеми збору даних	120
3.4. Алгоритм оптимізації підсистеми збору даних.....	121
4. ПРАКТИЧНА РЕАЛІЗАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ.....	126
4.1. Розробка програмного забезпечення інформаційної технології оптимізації підсистем збору даних АСУТП	127
4.2. Експериментальні дослідження ефективності інформаційної технології	139
ПІСЛЯМОВА.....	150
ЛІТЕРАТУРА.....	151

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AM	Алгоритмічна модель
АСУТП	Автоматизована система управління технологічним процесом
БД	База даних
БЗ	База знань
ІТ	Інформаційна технологія
ЛСА	Логічні схеми алгоритмів
ПЗД	Підсистема збору даних
САА	Система алгоритмічних алгебр
СК	Системи керування
СК ТП	Системи керування технологічним процесом
УФН	Узагальнююча функція невизначеності
SCADA	Supervisory Control And Data Acquisition
ССП	Система структурного проектування
UML	Unified Modeling Language
$G\{V, P\}$	Граф
$\beta(x)$	Узагальнююча функція невизначеності
$X[N]$	Вектор параметрів технологічного процесу
$X''[n]$	Вектор контрольованих параметрів
$X'[k]$	Вектор параметрів, які оцінюються
Q	Інформаційно-вартісний показник ефективності функціонування ПЗД
θ_S	Міра складності системи S
σ	Розкид (СКВ) результату
P_H	Надійність ПЗД
T_u	Тривалість циклу збору даних (швидкодія підсистеми)
I_u	Кількість інформації про технологічний процес за один цикл
AS	Алгебраїчна система
OP	Множина операцій над алгоритмічними моделями
op^{-1}	Обернена операція
M	Алгоритмічна модель
B	Вірогідність співпадання результатів в умовах невизначеності
τ_0	Інтервал кореляції стану технологічного процесу
C	Вартісна функція

ВСТУП

Останніми роками зріс інтерес до проблем побудови високоефективних та високонадійних підсистем збору даних (ПЗД) у різних галузях людської діяльності, зокрема в АСУ ТП, що в свою чергу, приводить до впровадження інформаційних технологій проектування підсистем збору даних. З одного боку це пов'язано із значним прогресом в області обчислювальної техніки, розвитком інформаційних технологій, підвищенням ступеня автоматизації, а з іншого необхідністю ефективного проектування таких підсистем, оскільки забезпечення якості збору, передачі та обробки інформації і задоволення постійно зростаючих вимог до надійного і безпечного функціонування АСУ ТП впливає на результат функціонування системи в цілому.

При проектуванні ПЗД постає необхідність оптимізації її структури, яка забезпечить необхідну якість функціонування АСУ ТП, причому важливим питанням є вибір критерію оптимізації, знаходження оптимального компромісу між швидкодією і вартістю системи, складністю і надійністю, оптимального розподілу функцій між програмними і апаратними засобами ПЗД з урахуванням особливостей контрольованого технологічного процесу.

Для формалізації та оптимізації підсистем збору даних використовують різноманітні моделі. Оскільки ПЗД містить апаратні та програмні засоби, в основному орієнтовані на комп'ютерну техніку, то її модель має бути універсальною. Такими універсальними моделями є алгоритмічні моделі (АМ), які широко використовуються для дослідження процесів і проектування систем.

У більшості теоретичних задач структурної оптимізації мова йде про постановки і методи розв'язання задач, що не містять невизначеностей. Проте, в переважній більшості практичних задач функціонування ПЗД відбувається в умовах невизначеності вхідних даних, зовнішніх впливів і параметрів системи, причому ступінь цієї невизначеності, а також її природа можуть суттєво відрізнятись. В цих умовах задача оптимізації структури ПЗД має певні особливості, і наразі немає загальновизнаних підходів до її розв'язання. Існують роботи, в яких розглядаються теоретичні питання, пов'язані з алгоритмічними моделями ПЗД, але мало досліджено питання врахування невизначеності при моделюванні і оптимізації структури ПЗД на основі алгоритмічних моделей.

Отже, однією з головних проблем проектування ПЗД є оптимізація структури в умовах невизначеності частини параметрів, зокрема актуальною теоретичною і прикладною задачею, яка розглядається в монографії, є розвиток інформаційної технології та вдосконалення методів оптимізації структури ПЗД на основі алгоритмічних моделей з метою врахування в них умов невизначеності.

1. АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА ТЕХНОЛОГІЙ СТРУКТУРНОГО ПРОЕКТУВАННЯ ПІДСИСТЕМ ЗБОРУ ДАНИХ

Збір і аналіз даних - типові задачі автоматизації. Для їх реалізації вітчизняні й закордонні виробники випускають системи збору даних різної конфігурації. Структурне проектування підсистем збору даних [69] є важливим питанням, оскільки ПЗД відрізняються великим різномаяттям методів отримання даних, рівнем достовірності, зручністю у використанні, модульністю, наявністю власного програмного забезпечення.

1.1. Проблеми моделювання і проектування структури підсистем збору даних

Автоматизована система управління технологічними процесами складається з виділених за функціональною чи структурною ознакою підсистем, кожна з яких відповідає конкретній меті та задачам. Підсистема збору даних є комплексом засобів збору й обробки даних, необхідних для організації керування об'єктами [44].

При створенні підсистем збору даних ставиться завдання відібрати й автоматизувати трудомісткі, повторювані та рутинні операції над великими масивами даних [73, 95, 98, 99, 160, 172, 201], що є складною задачею, яка потребує ретельного моделювання і проектування структури ПЗД.

В іноземній літературі термін ПЗД досить часто зустрічається під аббревіатурою SCADA (Supervisory Control And Data Acquisition). Але здебільшого під ПЗД розуміють апаратно-програмні комплекси збору даних, системи диспетчерського управління і збору даних, в більшості випадків для систем керування технологічними процесами. Рідше під SCADA розуміють засіб для створення програмного забезпечення та проектування підсистем збору даних.

Розвиток підсистем збору даних зумовлений підвищенням складності об'єктів керування, розпаралелюванням процесів в системах керування, масовим використанням потужних та недорогих мікроконтролерів, пошуком нових шляхів підвищення якості й скорочення виробничих витрат та широким введенням в експлуатацію багатофункціональних об'єктів. Це призвело до набуття такими системами розподіленого характеру та суттєвого ускладнення їх структури.

1.1.1. Роль підсистем збору даних у підвищенні ефективності АСУ ТП.

Сучасні технології збору даних отримали широке застосування і використовуються як в науково-дослідницьких цілях, так і в різних задачах автоматизації промисловості. Для збору даних здебільшого використовуються комп'ютери (контролери) з послідовними і паралельними портами на основі різних стандартних шин (PCI, PXI, CompactPCI, PCMCIA, FireWire). В деяких задачах використовуються віддалені пристрої збору даних та промислові мережі, а іноді ці пристрої вбудовуються безпосередньо в комп'ютер і передають дані напряму в його пам'ять. Усі ці способи побудови ПЗД характеризуються різною швидкістю, надійністю, просторовим розташуванням тощо. Тому вибір правильної структури ПЗД має привести до підвищення ефективності функціонування об'єктів, для яких вона розробляється.

Розмаїття вимог, зумовлених конкретними задачами, не може бути забезпечено якою-небудь одною структурою підсистеми збору даних. Залежність організації збору даних від організації технологічного процесу і необхідність забезпечення їх сумісності зумовили появу цілого ряду структур підсистем збору даних і обробки інформації. Схема основних операцій, які реалізуються в подібних системах, наведена на рис. 1.1. Зв'язки між окремими блоками (операціями) можуть бути різними і, таким чином, відображати розподіл потоків даних в ПЗД у часі і просторі [73].

Підсистеми збору даних складаються з трьох головних компонент (рис. 1.2).

Віддалений термінал (Remote Terminal Unit) - призначений безпосередньо для отримання даних з технологічного процесу в режимі реального часу за допомогою сенсорів, контролерів, спеціалізованих багатопроцесорних комплексів або бортових комп'ютерів в залежності від призначення системи збору даних. Останніми роками існує тенденція до інтелектуалізації цього компонента.

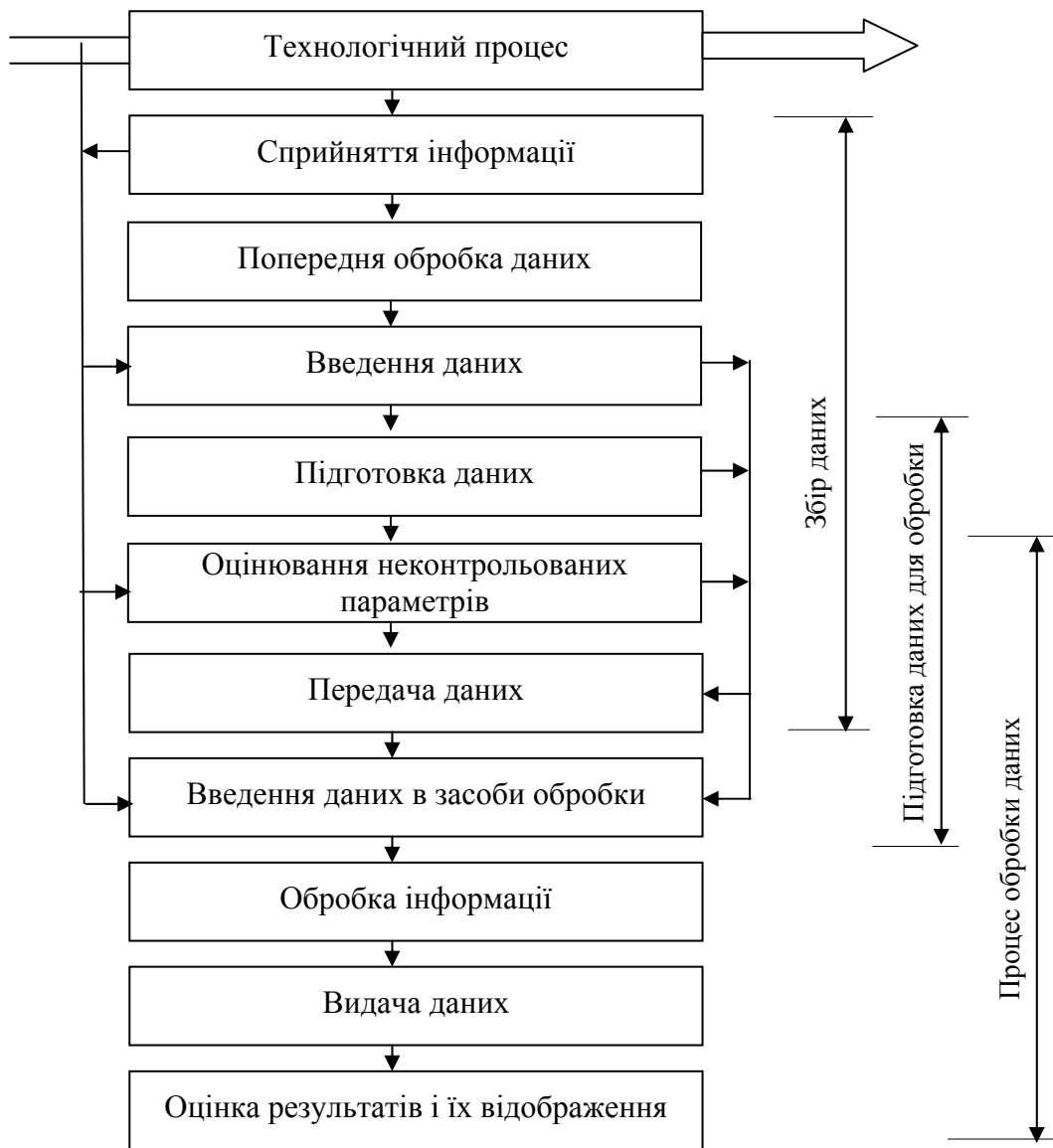


Рис. 1.1. Схема операцій, які реалізуються в процесі збору та обробки інформації

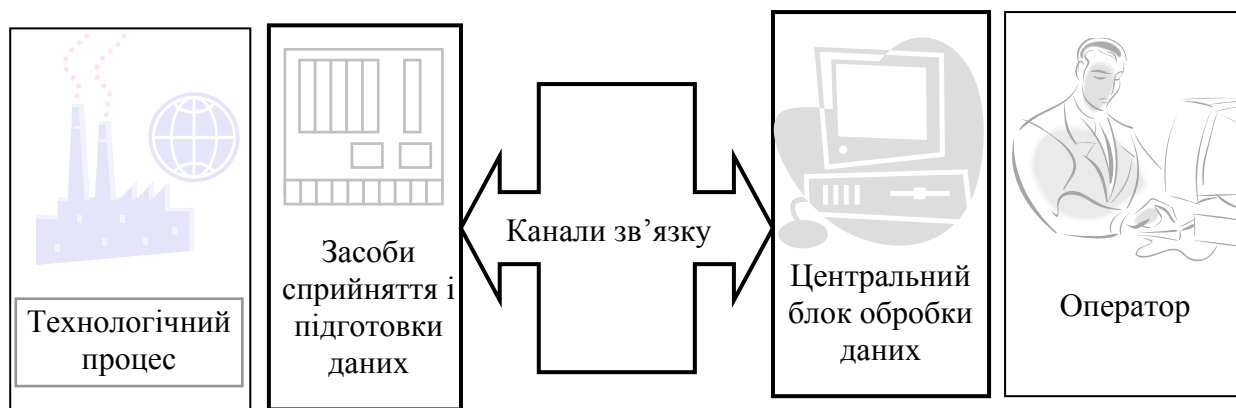


Рис. 1.2. Структурні компоненти ІЗД

Головний термінал (Master Terminal Unit) – здійснює обробку інформації вищого рівня і, головним чином, забезпечує зручний людино-машинний інтерфейс. В залежності від складності ПЗД головний термінал може бути реалізований у вигляді великого обчислювального комплексу або як окремий комп'ютер, під'єднаний до каналів зв'язку. Найчастіше сучасні ПЗД реалізуються на основі технології клієнт-сервер, що складається з 4 компонент: інтерфейс користувача; керування даними (здебільшого на основі реляційних баз даних); компонент мережі й служби (використовує стандартні мережеві технології і протоколи); служби реального часу [137].

Канали зв'язку, або Communication System, дозволяють передавати інформацію з віддалених об'єктів на інтерфейс оператора. Причому архітектура системи зв'язку залежить від вимог до надійності зв'язку, наявної матеріальної бази, кількості віддалених об'єктів, відстані між головним та віддаленим терміналами тощо. На рис. 1.3 показано зв'язок ПЗД з технологічним процесом і центральним блоком управління (ЦБУ).

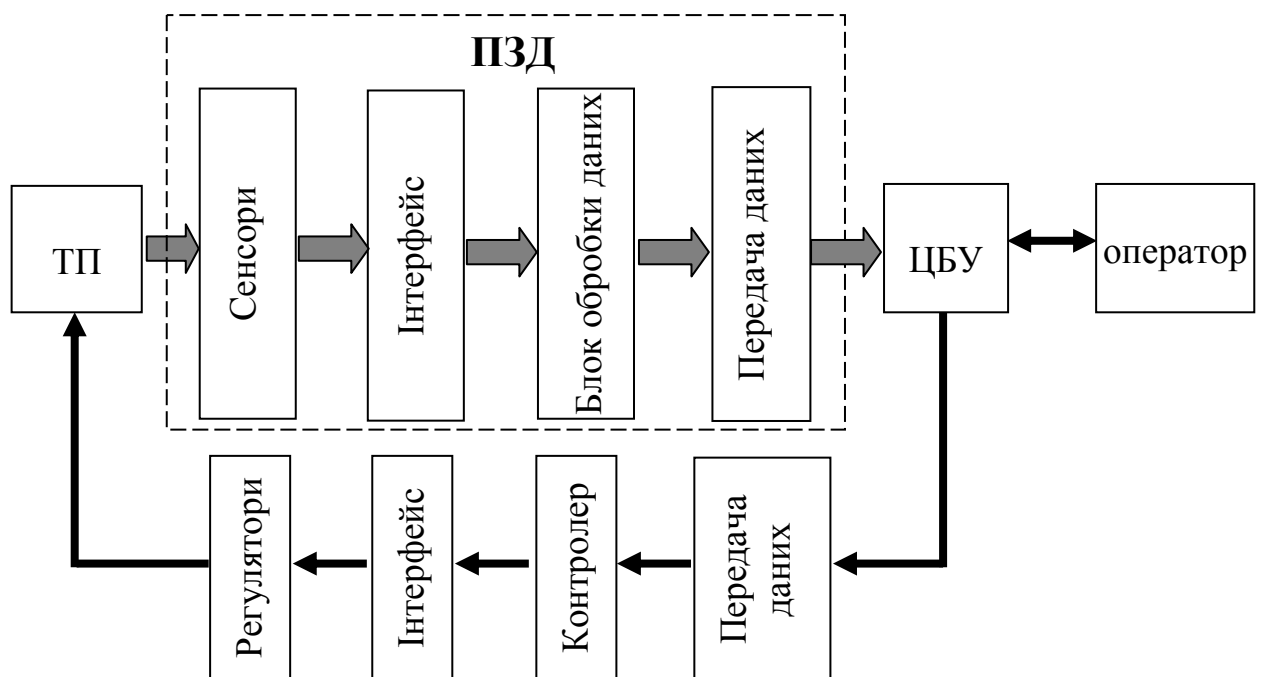


Рис.1.3 Місце ПЗД в АСУТП

1.1.2. Види підсистем збору даних. Підсистеми збору даних є складним комплексом програмно-апаратних засобів, які можуть поєднуватися і виконуватися у різний спосіб. Використання інтелектуальних сенсорів наділяє ПЗД властивостями розподіленої обробки даних. Керування такими системами здійснюється за допомогою мереж комп'ютерів та контролерів, що

суттєво збільшує розмаїття ПЗД. Класифікація ПЗД наведена на рис. 1.4.

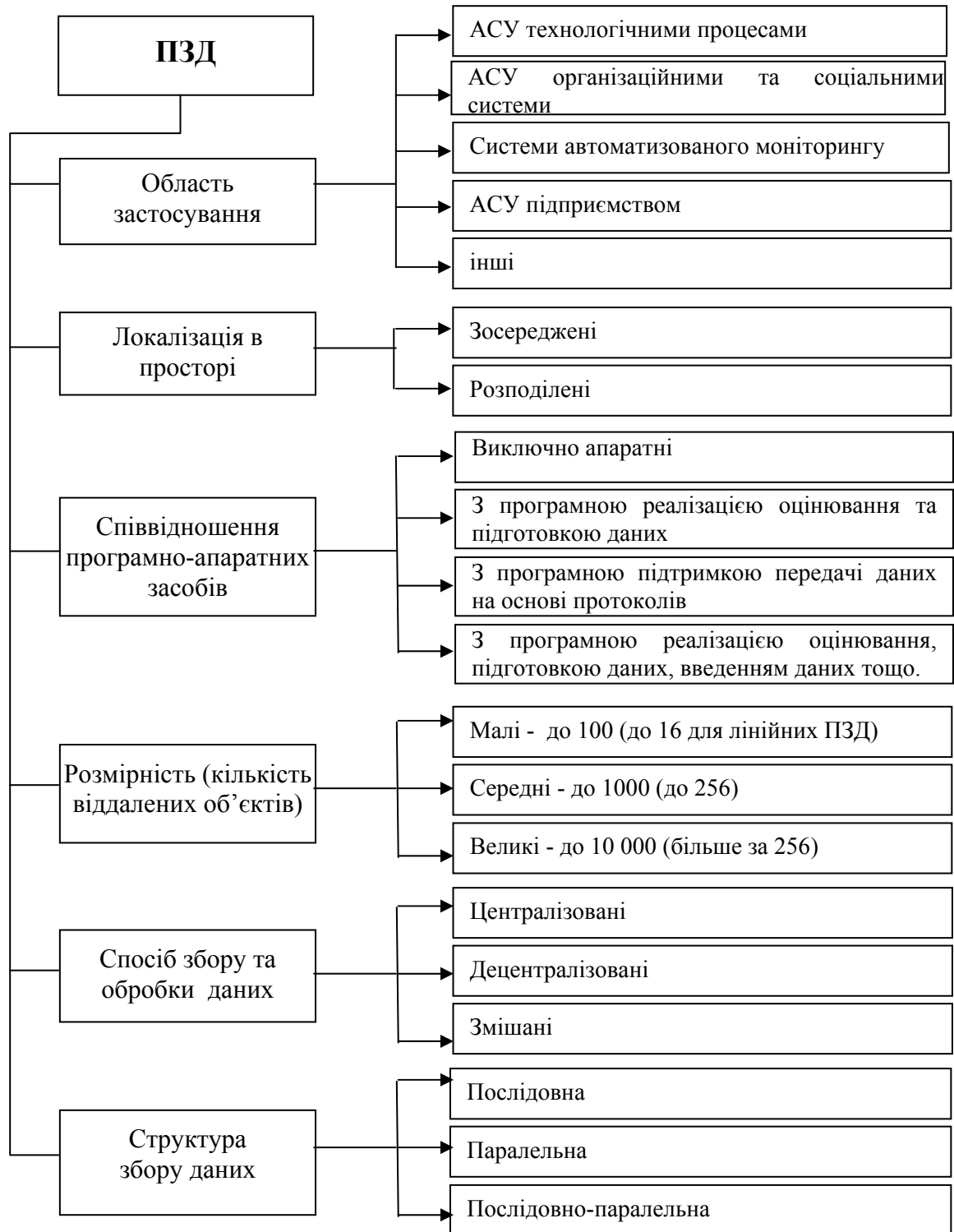


Рис.1.4. Класифікація підсистем збору даних

Наведена класифікація орієнтована на ознаки, які є істотними при проектуванні ПЗД. Важливо визначити розподіл апаратно-програмних операцій, множини елементів ПЗД, зв'язків між елементами, функцій цих елементів, множину входів та множину виходів. Характеристика підсистем збору даних повинна враховувати можливість втручання оператора в процес

збору, просторовий розподіл джерел інформації, необхідність передачі даних каналами зв'язку, можливості технічних засобів і умови їх експлуатації та обслуговування.

Розподіл елементів ПЗД в просторі і їх віддаленість від центрального блока обробки даних визначають три форми організації таких підсистем – в безпосередній близькості від протікання технологічного процесу, в ряді територіально розподілених пунктів (децентралізований збір) і в одному дистанційно віддаленому від процесу центральному пункті (централізований збір даних) [73]. Збір даних в місці протікання процесу може здійснюватись автоматично або оператором, який спостерігає за приладами (рис. 1.5, а).

На рис. 1.5, б зображена загальна схема автоматичного децентралізованого збору даних з використанням в окремо розташованих пунктах автоматичних реєстраторів сигналів, а в деяких випадках – контролерів. Пристрої збору і обробки працюють за автономними програмами.

Організація зв'язку між центральним блоком обробки даних і первинними джерелами може бути організована за типовими структурами: паралельна (рис. 1.6, а) послідовна (рис. 1.6, б), та паралельно-послідовна (рис. 1.6, в) структури.

Кожна з цих структур має свої переваги і недоліки, зокрема, паралельна характеризується швидкістю обробки, але високою ціною реалізації такої структури. Перевагами таких ПЗД є паралелізм в роботі і, як наслідок, більша швидкодія і підвищена надійність. Недолік – більша кількість ліній зв'язку, прокладених від первинних джерел до центрального блока обробки даних (у відповідності з числом виміряних і контрольованих параметрів та характеристик процесу), що, в свою чергу, призводить до підвищення вартості такої структури. Цей недолік відсутній в централізованому зборі даних (рис. 1.5, в). Але при цьому система є менш швидкодіюною. Крім того, в подібних системах знижується надійність через використання єдиного центру збору і єдиного каналу передачі даних, відмова яких призводить до повної відмови всієї підсистеми збору.

Структуру ПЗД також може визначати технологічний процес, який відбувається в різних умовах [3, 7]. Таким чином, комбінуючи ознаки і характеристики структур ПЗД, маємо значну кількість варіантів структур, серед яких необхідно обрати оптимальну для конкретних умов функціонування.

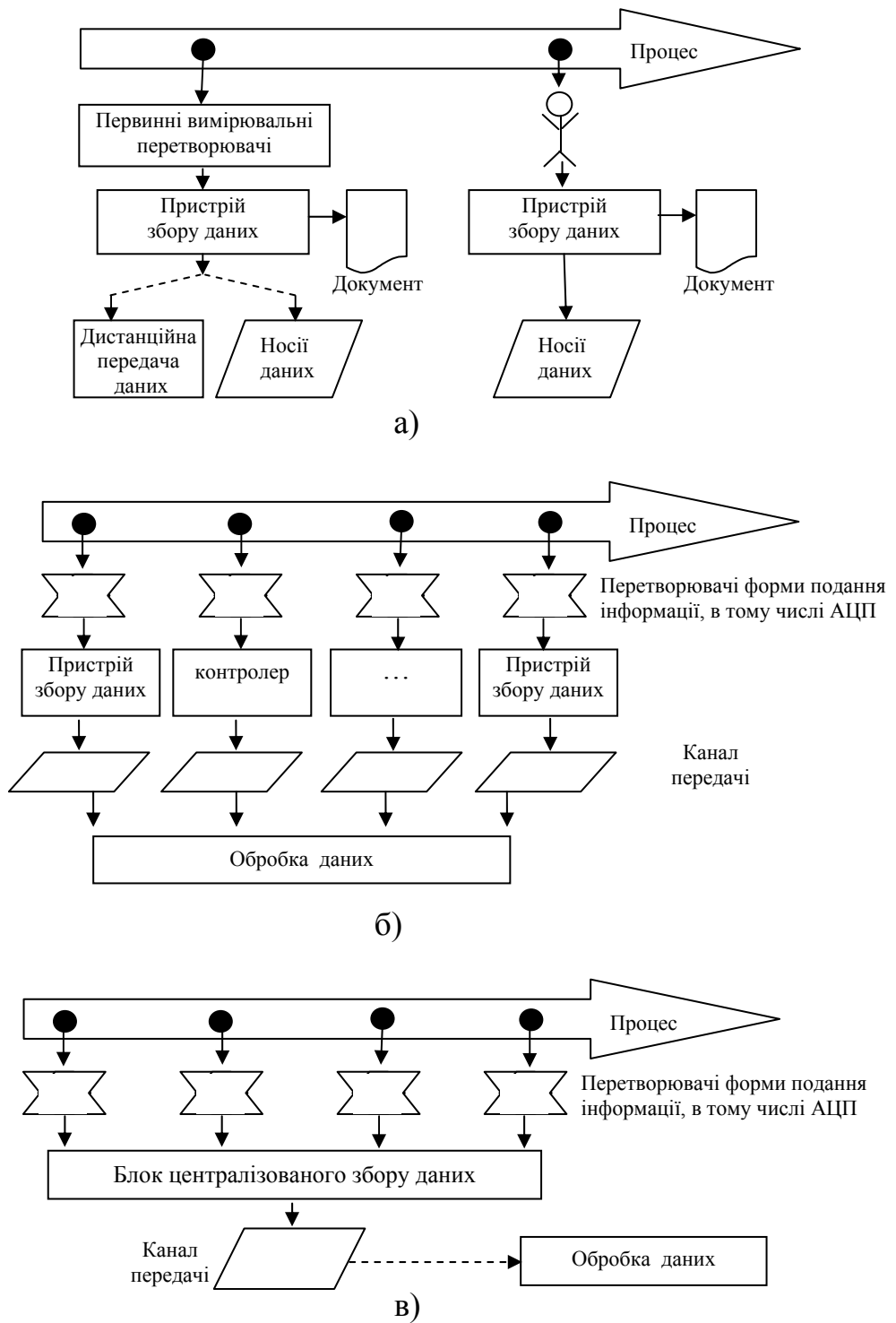


Рис. 1.5. Форми організації ПЗД

а) схема збору даних у місці протікання технологічного процесу

б) схема децентралізованого збору даних

в) схема централізованого збору даних

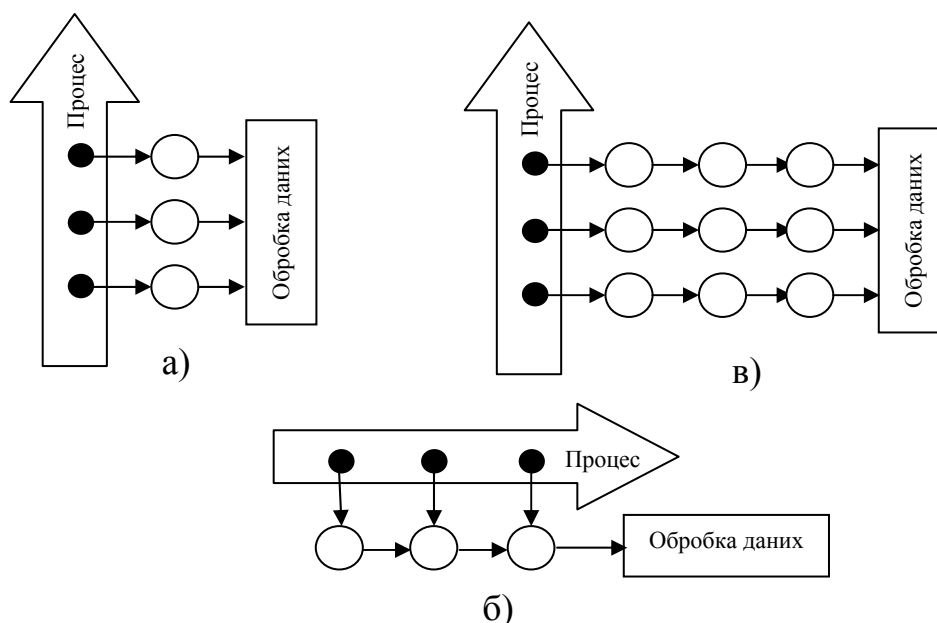


Рис. 1.6. Структура зв'язку з первинними джерелами
 а) паралельна структура; б) послідовна структура;
 в) паралельно-послідовна структура

1.1.3. Системи структурного проектування підсистем збору даних.

Розробка підсистеми збору та обробки даних вимагає здебільшого багато часу та великих матеріальних витрат [21, 98]. Застосування систем структурного проектування (ССП) дозволяє скоротити початкові витрати та витрати на впровадження системи в декілька разів. Крім того, система, створена «вручну», далека від прийнятих стандартів і є закритою, тобто не дозволяє взаємодіяти з іншими системами [89]. Питання правильного вибору ССП є актуальним, оскільки розробка і вдосконалення ПЗД є важливим та матеріально витратним етапом в проектуванні системи в цілому. Вирішити його можна, розглянувши функціональні можливості ССП (рис. 1.7).

Визначимо декілька популярних систем проектування збору даних (табл. 1.1.) і на їх прикладі розглянемо основні можливості ССП. З аналізу таблиці видно, що більшість систем відрізняється лише технічною реалізацією і деякими специфічними можливостями. Такі можливості, як збір первинної інформації від первинних джерел, керування й реєстрації сигналів про аварійні ситуації, зберігання інформації з можливістю її обробки, візуалізація подання інформації у вигляді графіків та інші важливі функціональні характеристики ПЗД реалізовані в кожній з наведених ССП.

Але деякі з цих систем мають певні особливості, зокрема, до складу Trace-Mode 6 входить найбільша кількість бібліотек ресурсів [202], які готові до використання в прикладних проектах, з них драйвери до 2305 контролерів, 1116 графічних зображень технологічних об'єктів, 596 анімованих об'єктів; більше 150 алгоритмів обробки даних, комплексні технологічні об'єкти, що робить цю систему безумовним лідером за цим показником.



Рис. 1.7. Характерні риси ССП

Важливим є питання програмно-апаратної платформи, на якій реалізована підсистема збору даних, оскільки від цього залежить її експлуатаційна вартість і поширення на існуючі обчислювальні засоби. Більшість з систем реалізована на платформах MS Windows, оскільки там існує можливість застосування зручного людино-машинного інтерфейсу.

Мережний сервіс – це також важливе питання, оскільки він впливає на ефективність функціонування ПЗД в різноманітному середовищі (об'єкти керування, апаратура, що реєструє інформацію, робочі місця операторів,

сервери баз даних тощо). Від таких систем вимагають підтримку стандартних мережних протоколів (NETBIOS, TCP/IP і ін.) і середовищ (ETHERNET та інші), а також мережних промислових інтерфейсів (PROFIBUS, CANBUS, LON, MODBUS і т.д.). Відрізняють ці системи лише набором та кількістю підтримуваних мережних інтерфейсів.

Таблиця 1.1

Перелік поширених ССП ПЗД та їх властивості

Системи структурного проектування ПЗД	Функціональні характеристики	Наявність засобів оптимізації структури ПЗД	Технічні характеристики	Експлуатаційні характеристики	Відкритість системи	Багаторівневість систем автоматизації	Велика кількість бібліотек ресурсів	Можливість підтримки стандартних мережних протоколів і середовищ	Якість технічної підтримки та додаткових послуг
Factory-Link (U.S.DATA-Co., США)									
InTouch (Wonderware, США)									
Genesis (Iconics, США)									
RealFlex(BJ.Software.Systems, США)									
Sitex (Jade.Software, Англія)									
FIX (Intellution, США)									
Trace-Mode (AdAstra, Росія)									
IGSS (Seven.Technologie, Данія)									
Image (Технолинк, Росія)									

Експлуатаційні характеристики впливають на вартість реалізації і важливі для простоти та швидкості розроблених прикладних систем.

Важливою характеристикою ССП є відкритість, оскільки це дозволяє підключати до таких систем незалежно розроблені компоненти та модулі, що реалізують ті можливості, яких немає в існуючих ССП. Тому більшість з них має можливість доступу до системних засобів (доступ до графічних функцій, функцій роботи з базами даних і т.д.). Крім того, майже всі ССП ПЗД дозволяють розробляти власні програмні модулі й драйвери нових пристроїв нижнього рівня.

Оскільки основні можливості ССП ПЗД близькі та реалізовані приблизно в однаковій мірі, то велика увага приділяється якості технічної підтримки та додаткових послуг з освоєння й впровадження кінцевої системи збору даних, звичайно, важливе значення має кількість та якість бібліотек ресурсів, які готові до використання в прикладних проектах. Тобто,

скорочення витрат системних інтеграторів і кінцевих користувачів, зменшення вартості супроводу кінцевої системи та технічні можливості цих систем впливають на пріоритетність ССП ПЗД при їх виборі. Наприклад, на відміну від інших аналогічних SCADA-систем, окрім стандартного набору можливостей, ССП TRACE MODE використовує унікальну технологію playback (графічний огляд архівів на робочих місцях керівників) та має спеціальні методи підвищення надійності АСУТП (гаряче резервування серверів реального часу, промислових контролерів, мережевих адаптерів тощо).

Отже, за функціональними можливостями всі розглянуті системи в цілому порівнянні. Побудова ПЗД на основі розглянутих ССП значно скорочує набір необхідних знань в області класичного програмування, дозволяючи сконцентруватись на поставленій задачі.

Але виходячи з аналізу табл. 1.1, можна зробити висновок, що у поширених ССП не розв'язана задача структурної оптимізації проєктованих систем (окрім оптимального резервування і забезпечення стійкості в системах RealFlex, Trace-Mode та ін.). Для подальшого розвитку інформаційної технології оберемо базову технологію ССП TRACE MODE як таку, що має, наразі, найвищі показники.

1.2. Моделі підсистем збору даних

Модель підсистеми збору даних є основою для розв'язання задачі оптимізації структури ПЗД. Існують різні моделі, за допомогою яких можна описати структуру та роботу ПЗД, зокрема, структурні, динамічні, статичні та інші (рис. 1.8). Розглянемо деякі переваги і недоліки використання цих моделей для проєктування та моделювання ПЗД.

Моделі структури подають об'єкт моделювання з точки зору його складу та взаємозв'язку частин між собою й зовнішнім середовищем.

Структурні моделі можуть зображуватись у різних просторах: у просторі фізичних зв'язків, у просторі логічних зв'язків, у вигляді розташування у геометричному просторі.

Найпоширенішою та найуніверсальнішою формою подання структурних моделей та основним математичним апаратом є граф. Для опису графів

використовуються матриці та списки (матриця суміжності, матриця

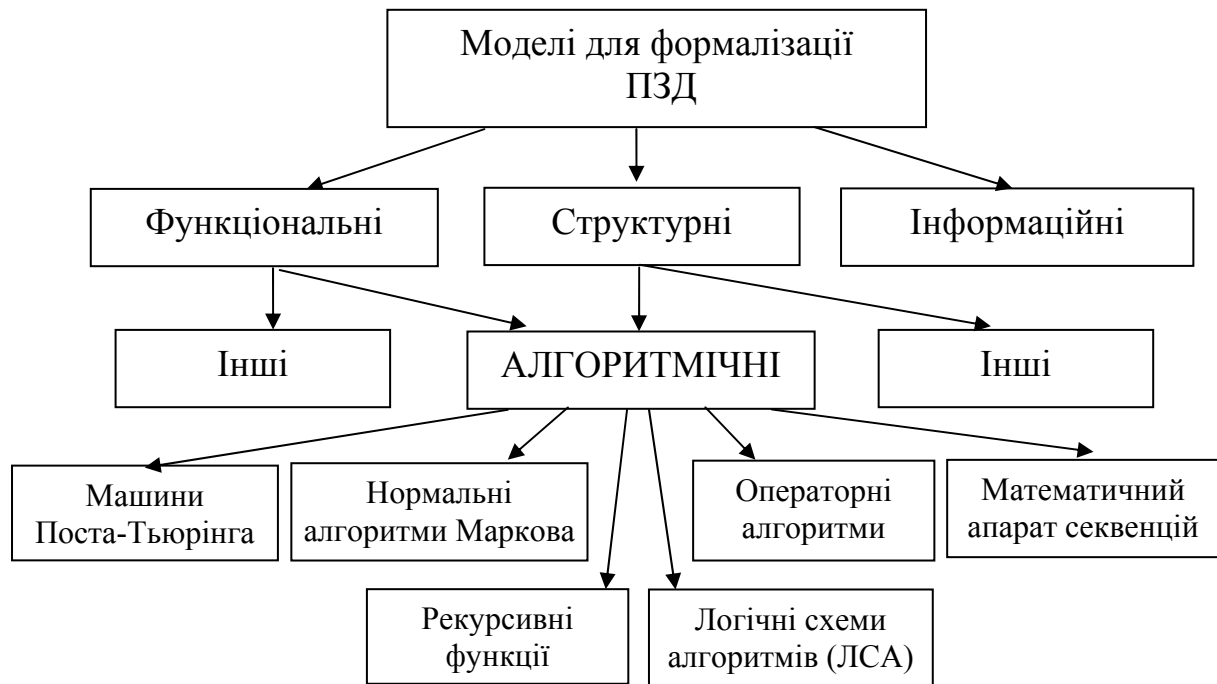


Рис. 1.8. Види моделей ПЗД

інциденції, списки пар вершин) [63, 76].

Функціональні моделі описують процеси, які відбуваються в об'єкті моделювання, та поведінку об'єкта у часі. Вони використовуються у комплексі із структурними моделями.

Модель динаміки системи може подаватися у різних ізоморфних та гомеоморфних формах [43, 78]: диференціальні рівняння, операторні рівняння, інтегральні рівняння, спектральні рівняння, передаточні функції, комплексні частотні передаточні функції, амплітудно-фазові частотні характеристики, перехідні функції і характеристики.

Найпоширенішою моделлю динаміки є *диференціальне рівняння* [43], а саме лінійне диференціальне рівняння, яке описує динаміку лінійних об'єктів з постійними параметрами

$$a_b \frac{d^n y(t)}{dt^n} + a_{b-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_1 y(t) = b_m \frac{d^m x(t)}{dt^m} + b_{m-1} \frac{d^{m-1} x(t)}{dt^{m-1}} + \dots + b_1 x(t) + b_0. \quad (1.1)$$

Якщо розглядати лінійну систему, то її модель у формі зображення за Лапласом подається передаточною функцією у вигляді дрібно-раціонального виразу [43]

$$W(p) = \frac{Y(p)}{X(p)} = \frac{a_n p^n + \dots + a_1 p + a_0}{b_m p^m + \dots + b_1 p + b_0}, \quad (1.2)$$

де $Y(p)$, $X(p)$ – зображення за Лапласом вихідної та вхідної величин лінійної системи.

Програмна реалізація лінійної динамічної моделі здійснюється на основі її дискретного подання.

З (1.2) знаходимо

$$b_m p^m y + \dots + b_1 p y + b_0 y = a_n p^n x + \dots + a_1 p x + a_0 x. \quad (1.3)$$

У дискретній системі аналогом похідних є відповідні різниці. В загальному випадку:

$$z^{(k)}(t_0) = \frac{1}{\Delta t^k} \sum_{i=0}^k (-1)^i C_k^i z_{-i}, \quad (1.4)$$

де t_0 – момент надходження останнього даного, поточний момент часу;

Δt = інтервал дискретизації;

$z_{-i} = z(t_0 - \Delta t \cdot i)$.

Підставляючи (1.4) в (1.3), отримаємо дискретний вираз рівняння перетворення системи

$$\sum_{j=0}^m \left[\frac{b_j}{\Delta t^j} \sum_{i=0}^j (-1)^i C_j^i y_{-i} \right] = \sum_{j=0}^n \left[\frac{a_j}{\Delta t^j} \sum_{i=0}^j (-1)^i C_j^i x_{-i} \right].$$

Виділивши з лівої частини рівняння значення вихідної величини у поточний момент часу та зробивши деякі елементарні перетворення, отримують [43] дискретну модель лінійної динамічної системи

$$y_0 = \sum_{i=0}^n K_{x_i} x_{-i} + \sum_{i=1}^m K_{y_i} y_{-i}, \quad (1.5)$$

$$\text{де } K_{x_i} = \frac{(-1)^i \sum_{j=i}^n C_j^i \frac{a_j}{\Delta t^j}}{\sum_{j=0}^m \frac{b_j}{\Delta t^j}}; \quad K_{y_i} = \frac{(-1)^{i+1} \sum_{j=i}^m C_j^i \frac{b_j}{\Delta t^j}}{\sum_{j=0}^m \frac{b_j}{\Delta t^j}}.$$

Якщо при синтезі структури системи існує необхідність урахування ряду логічних виразів, то використовують математичний апарат секвенцій

[104], який є таким же універсальним, як і класична мова кінцевих автоматів - мова автоматних таблиць, проте описи мовою секвенцій є більш компактними. Крім того, цей математичний апарат зручніший для синтезу систем з розподіленою пам'яттю. Але для опису структури ПЗД, де необхідно враховувати динаміку процесів отримання даних і невизначеність умов, цей апарат є недостатньо ефективним.

Поширеність моделей для проектування того чи іншого процесу зображена інтенсивністю сірого кольору (табл. 1.2).

Не дивлячись на великий вибір моделей опису процесів функціонування систем, серед яких: марковські та напівмарковські процеси, моделі теорії масового обслуговування, формальні граматики, мережі Петрі, логічні автомати і схеми алгоритмів Ляпунова, логіко-лінгвістичні моделі, мережі GERT, PERT, функціональні і функціонально-семантичні мережі, ймовірнісні та нечіткі моделі, жодна з них не позбавлена недоліків і не може розглядатися як універсальний метод опису функціонування будь-яких ПЗД.

Таблиця 1.2

Область застосування моделей з погляду проектування ПЗД

Види моделей	Проектування алгоритмів	Проектування логічних схем	Проектування вимірювальних систем	Проектування процесу передачі даних	Проектування інтерфейсів	Проектування структури систем	Проектування динаміки системи
Логіки							
Динаміки							
Статики							
Обслуговування							
Графові							
Алгоритмічні							
Інформаційні							

З великої кількості моделей для структурного проектування ПЗД найбільш зручними є алгоритмічні моделі. Вони є основою програмування, використовуються для дослідження та опису процесів при проектуванні ПЗД. Алгоритмічні моделі подають об'єкт у вигляді послідовності дій, які

дозволяють просто і швидко отримати деякі його характеристики [55, 75, 76]. Більшість ПЗД містить в своєму складі комп'ютер, тому з тотальним переходом на комп'ютеризоване керування актуальність розвитку і застосування алгоритмічних моделей при оптимізації та моделюванні ПЗД постійно зростає.

1.3. Застосування алгоритмічного підходу до дослідження систем

Алгоритмічні моделі (алгоритми) є основою програмування [23, 79, 97, 168, 169], використовуються для опису процесів в інформаційних системах [31, 60] і системах контролю та керування технологічними процесами [108]. Створення алгоритмічної моделі у багатьох випадках є кінцевим етапом проектування системи. Важливим практичним застосуванням алгоритмічних моделей є їх використання при розробці АСУТП.

Вже в перших серйозних роботах про великі системи [79, 97, 168, 169] почав формуватися алгоритмічний підхід до дослідження систем, який відрізняється тим, що опис досліджуваної системи створюється у вигляді алгоритмічної моделі, призначеної для реалізації на ЕОМ. Широкі можливості алгоритмічного підходу дозволили не тільки істотно просунути вперед дослідження систем керування, але й привели до розробки нових алгоритмічних мов, призначених спеціально для алгоритмічного моделювання (SIMULA, DINAMO тощо), а також стали основою для появи і швидкого розвитку ідей штучного інтелекту [170].

Основою для побудови алгоритмічної моделі може служити концептуальний опис системи і її окремих властивостей, експериментальні дані системи, фізична, іконографічна або математична модель системи, тобто, в кінцевому рахунку будь-яка інформація про систему, яка дозволяє побудувати потрібну алгоритмічну модель.

Алгоритмічна модель сприяє ефективному використанню ЕОМ при розв'язанні таких важливих задач, як аналіз і синтез структури системи керування, вибір оптимальних параметрів, оцінювання ефективності системи, побудова оптимальних алгоритмів обробки даних тощо [4, 13, 127, 128].

Ефективним засобом комплексного дослідження систем є апарат змістовних логічних схем алгоритмів та алгоритмічна алгебра Глушкова.

1.3.1. Змістовні логічні схеми алгоритмів як апарат формалізованого опису алгоритмів роботи підсистем збору даних. Для формалізованого опису алгоритмів роботи інформаційно-вимірювальних систем (ІВС) запропонований апарат змістовних логічних схем алгоритмів [170].

Опис принципу і послідовності дії ІВС в більшості випадків проводиться словесно, природною мовою. Він стає громіздким для складних систем збору даних, це ускладнює аналіз роботи, порівняння варіантів реалізації таких засобів та підготовку їх програмного забезпечення.

Логічні схеми алгоритмів, які використовуються для розв'язання цієї задачі, вирізняються серед способів формального опису роботи підсистем збору даних компактністю запису, можливістю подання алгоритму з наперед встановленим ступенем деталізації, відомим зв'язком з логічними схемами програм, можливістю мінімізації обсягу деяких видів алгоритмів [171].

У ЛСА застосовуються множини функціональних $\{A_{ij}\}$, логічних $\{w_{ij}\}$ операторів, а також правила послідовності виконання алгоритму в залежності від умов w_i . У ЛСА об'єднуються оператори, які визначають обмін інформаційними і службовими сигналами між функціональними блоками системи, а також перетворення цих сигналів. У складних моделях виділяються групи пов'язаних між собою операторів. Окрім літерних позначень у ЛСА використовують символи, якими позначається порядок виконання операторів або особливості апаратної частини системи. Для прикладу наведені основні елементи ЛСА в табл. 1.3 [170].

Таблиця 1.3

Умовні позначення логічних схем алгоритмів

Зміст	Позначення	Зміст	Позначення
Символи		Оператори	
Паралельне перетворення		Інформаційні оператори	$I(I^*)$
Перенесення ЛСА	\times	Логічні оператори	$\omega(\omega^*)$
Продовження ЛСА	...	Командні оператори	$\Phi(\Phi^*)$
Об'єднання перетворень	{ }, []	Почати	$\Phi(B)$
Скорочене позначення	$\langle \cdot \rangle$	Записати	$\Phi(R)$
Розшифрування	$:=$	Завершити	$\Phi(E)$
Виконання алгоритму програмним шляхом	\sqcap	Адреса	$Ad(Ad^*)$

При розробці ЛСА передбачається, що вони повинні:

- описувати функціонування як апаратної, так і програмно-керованої частин ІВС;
- по можливості наочно відображати зміст перетворень, що описуються;
- описувати не тільки інформаційні перетворення, а й службові операції;
- описувати функціонування ІВС з різною деталізацією.

На основі ЛСА підсистеми збору даних можуть бути подані у вигляді сукупності пов'язаних між собою функціональних блоків, під ними розуміють частини системи, яким необхідна організація спільної та узгодженої роботи [171]. Функціональні блоки поділяються на інформаційні і керуючі, які виконують функції організації обміну інформаційними і керуючими сигналами в системі (табл. 1.4).

Таблиця 1.4

Позначення функціональних блоків і перетворень

Назва блока	Умове позначення	Назва блока	Умове позначення
Датчик	$I(x_0/x_1)$	Перетворювач коду	$I(z_1/z_2)$
Аналоговий перетворювач	$I(x_i/x_{i+1})$	Цифровий обчислювальний пристрій	$I(CP/Z)$
Аналоговий комутатор	$I(X/x_i)$	Цифровий комутатор	$I(Z/z_i)$
Аналоговий ЗП	$I(S:X)$	Цифровий ЗП	$I(S:Z)$
Аналоговий пристрій порівняння	$I(CR:x_i;x_j)$	Цифровий пристрій порівняння	$I(CR:z_i;z_j)$
Аналоговий канал зв'язку	$I(x)$	Цифровий канал зв'язку	$I(Z)$
Пристрій реєстрування	$I(W:x)$	Дисплей	$I(R:X;Z)$ $I(W:X;Z)$
Показуючий пристрій	$I(R:x)$	Цифровий індикатор	$I(R:z)$
Аналого-цифровий перетворювач	$I(x/z)$	Таймер	$I(T)$
Цифро-аналоговий перетворювач	$I(z/x)$	Обчислювальні операції	$\sum \Delta \int \prod r d$

Розглянемо для прикладу ЛСА аналого-цифрову частину інформаційно-вимірювальної системи, яка складається з аналогових вимірювальних каналів і системних аналого-цифрових пристроїв [170]. Функціонування одноступінчатої паралельно-последовної структури при циклічній роботі комутатора може бути описано ЛСА

$$\Phi(B)[I(x_{01}/x_{11}) \parallel \dots \parallel I(x_{0i}/x_{1i}) \parallel \dots \parallel I(x_{0n}/x_{1n})] \times \\ \times \left[\Phi(i := i + 1; i_0 = 0) I(X_1/x_{1i}) I(x_{1i}/z_i) I(z_i) \omega(i = n) \right] \Phi(E)$$

Кількість ступенів структури визначається числом послідовно розташованих вузлів, в яких змінюється об'єм інформаційних каналів.

До основних недоліків ЛСА можна віднести необхідність складання і постійного використання списків операторів $\{A_{ij}\}$ і $\{w_{ij}\}$ з розшифруванням їх змісту, відсутність ефективних методів мінімізації записів. Ці недоліки слабо проявляються при відносно простих і загострюються при складних ЛСА.

Апарат ЛСА дозволяє розробляти алгоритми роботи ПЗД, може суттєво зменшити об'єм текстового матеріалу і покласти початок розробці структури ПЗД, зокрема формалізації проєктованого програмного забезпечення та апаратних засобів.

1.3.2. Алгоритми обчислювальних процесів та алгоритмічна алгебра.

Первісно під алгоритмічними моделями розуміли виключно алгоритми обчислень. Останнім часом поняття алгоритму узагальнене до алгоритмічної моделі. Вченими було запропоновано безліч еквівалентних означень алгоритму [81, 133, 142]. Уточнення поняття алгоритму пов'язано з уточненням алфавіту даних і форми їх подання, структури пам'яті і розміщення в ній даних, елементарних кроків алгоритму і механізму реалізації алгоритму. Основні типи алгоритмічних моделей [142], розрізняються вихідними трактуваннями, що таке алгоритм (див. рис. 1.8):

1. Перший тип трактує алгоритм як деякий детермінований пристрій, здатний виконувати в кожен момент лише строго фіксовану множину операцій. Основною теоретичною моделлю такого типу є машина Тьюрінга.

2. Другий тип пов'язує поняття алгоритму з традиційним поданням – процедурами обчислення значень числових функцій. Основною теоретичною

моделлю цього типу є рекурсивні функції – історично перша формалізація поняття алгоритму.

3. Третій тип алгоритмічних моделей – це перетворення слів у довільних алфавітах, у яких операціями є заміни частини слів іншим словом. Основною теоретичною моделлю цього типу є нормальні алгоритми Маркова.

Згадані вище три головні алгоритмічні моделі математично еквівалентні, але породили різні напрямки в програмуванні. Так, мікропрограмування будується на ідеях машин Тьюрінга, структурне програмування запозичало свої конструкції з теорії рекурсивних функцій, мови символічної обробки інформації (РЕФАЛ, ПРОЛОГ) беруть початок від нормальних алгоритмів Маркова і систем Поста [97].

Алгоритми в залежності від мети, початкових умов задачі, шляхів її розв’язання, визначення дій виконавця підрозділяються на типи, показані на рис. 1.9.

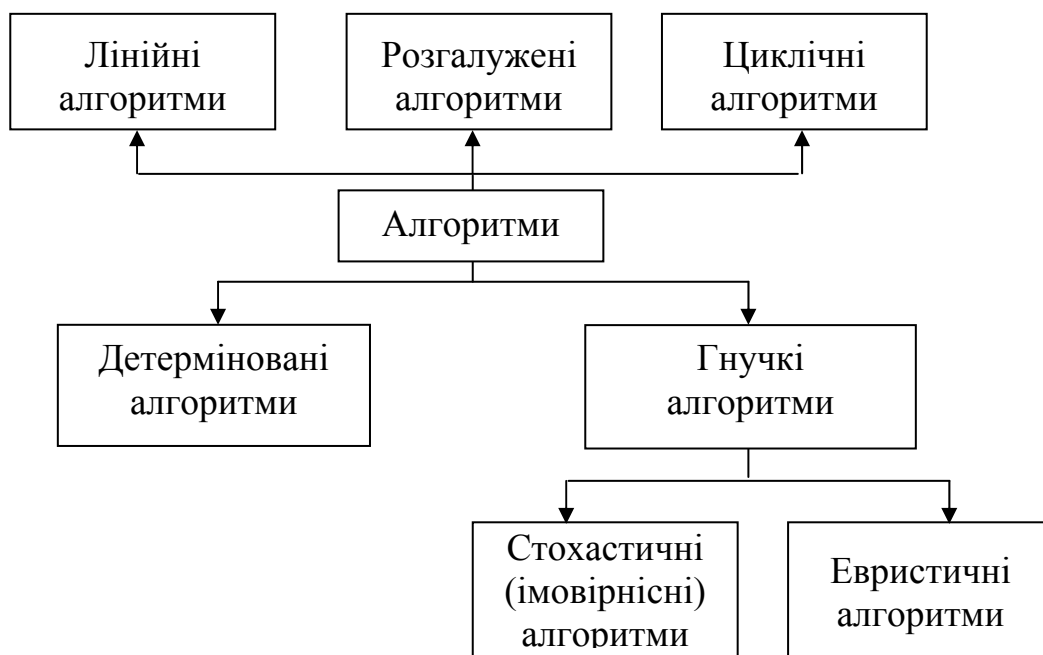


Рис. 1.9. Загальна класифікація алгоритмів в залежності від шляху розв’язання задачі та визначення дій виконавця

Розрізняються також такі форми подання алгоритмів: словесна (запис природною мовою), графічна (зображення з графічних символів, наприклад, структурна схема алгоритму, псевдокоди (напівформалізовані описи

алгоритмів умовною алгоритмічною мовою, що містять як елементи мови програмування, так і фрази природної мови, загальноприйняті математичні позначення й ін.), програмна (тексти мовами програмування).

Зазвичай формулюють кілька загальних властивостей алгоритмів, що дозволяють відрізнити алгоритми від інших інструкцій (табл. 1.5) [169].

Таблица 1.5

Властивості алгоритму

Властивість	Опис властивості алгоритму
Дискретність (роздільність)	Алгоритм повинен зображувати процес розв'язання задачі як послідовне виконання простих кроків. Кожна дія, яка передбачена алгоритмом, виконується тільки після того, як закінчилось виконання попередньої дії.
Визначеність	Кожне правило алгоритму повинно бути чітким та однозначним. Завдяки цій властивості виконання алгоритму носить детермінований характер.
Результативність (кінцевість)	Алгоритм повинен приводити до розв'язку задачі за кінцеву кількість кроків. Алгоритм є скінченим об'єктом, що є необхідною умовою його технічної реалізації;
Масовість	Алгоритм розв'язання задачі розробляється в загальному вигляді, тобто може застосовуватись для класу задач, які відрізняються тільки вихідними даними.
Правильність	Застосування алгоритму до правильних вхідних даних повинно приводити до отримання необхідних результатів. Найчастіше правильність перевіряється на наборі тестів.
Ефективність	Алгоритм повинен забезпечити розв'язання задачі за мінімальний проміжок часу з мінімальними витратами пам'яті. Для оцінювання алгоритмів існує багато критеріїв. Найчастіше оцінка алгоритму складається з оцінки часових витрат на розв'язання задачі в залежності від об'єму вхідних даних. Використовується також термін „трудомісткість алгоритму”. Фактично ця оцінка зводиться до оцінки кількості основних операцій, що їх виконує алгоритм, оскільки кожна конкретна операція виконується за кінцевий час.

Важливою властивістю алгоритмічних моделей є їх еквівалентність, оскільки одну і ту ж систему можна описати декількома різними, але

еквівалентними АМ, що будуть відрізнятися ступенем деталізації, паралельним або послідовним виконанням деяких операцій тощо.

Одним з найважливіших питань застосування алгоритмічних моделей для проектування систем є еквівалентні перетворення таких моделей. Значні результати у дослідженні еквівалентності алгоритмів належать А. А. Ляпунову, який ввів поняття схем програм [91], та іншим дослідникам [32, 55, 70, 75, 178]. На базі стандартних схем алгоритмів введені основні поняття й властивості, пов'язані з алгоритмічними моделями, головне з яких - відношення функціональної еквівалентності алгоритмічних моделей.

Ідеї Ляпунова були розвинені наприкінці 50-х і в 60-і роки А. П. Єршовим, Н. А. Криницьким, Л. А. Калужніним, Р. І. Подловченко і Ю. І. Яновим, який в [178] формалізував поняття схеми програми, визначив відношення еквівалентності схем і дослідив проблему еквівалентності для класу схем, що одержали згодом назву схем Янова. Н. А. Криницький [75] досліджував проблему еквівалентності й еквівалентних перетворень стандартних схем, причому для підкласу схем без циклів (тобто схем, граф яких не містить контурів) знайдений алгоритм розпізнавання еквівалентності й побудована повна система перетворень, що дозволяє будь-яку пару еквівалентних схем автоматично перетворити одна в одну. Графова форма схем була запропонована Л. А. Калужніним [55].

Клас стандартних схем характеризується базисом класу B і структурою схеми. Базис класу фіксує символи, з яких будуються схеми, вказує їх роль (змінні, функціональні символи й т.п.), задає вигляд виразів і операторів схеми [70]. Інтерпретація перетворює стандартну схему у конкретну алгоритмічну модель. Інтерпретацією базису B у області інтерпретації D називається функція I , що зіставляє кожному елементу (змінним, константам, функціональним символам, предикатам) з базису B деякі всюди визначені функції й елементи з області інтерпретації D . Пара (S, I) , де S - схема в базисі B , а I - інтерпретація цього базису, називається інтерпретованою стандартною схемою алгоритму або алгоритмічною моделлю.

В [70] введене відношення еквівалентності для стандартних схем алгоритмів в одному базисі. Якщо схеми S_1 і S_2 побудовані у двох різних базисах B_1 і B_2 , то можна їх "привести" до одного базису, за який взяти об'єднання базисів B_1 і B_2 . Стандартні схеми S_1 і S_2 у базисі B функціонально еквівалентні $(S_1 \sim S_2)$, якщо для будь-якої інтерпретації I базису B програми (S_1, I) і (S_2, I) або обидві зациклюються, або обидві зупиняються з однаковим

результатом, тобто $val(S_1, I) \approx val(S_s, I)$. Еквівалентними перетвореннями алгоритмічної моделі будемо називати таку послідовність операцій над моделлю, що не змінює зміст результатів роботи системи. З аналізу зрозуміло, що еквівалентні перетворення досліджувались багатьма науковцями, але в їх роботах не врахована невизначеність, яка присутня при функціонуванні ПЗД.

Проблема еквівалентних перетворень алгоритмів уже багато років привертає увагу спеціалістів [102, 115, 116, 159]. Вона вирішувалась в різних постановках для різних форм завдання алгоритмів: машин Тьюринга, операторних схем алгоритмів, рекурсивних функцій і т.д. Основна задача складалася в знаходженні системи породження алгоритмів, які еквівалентні заданому. В рамках оптимізації структури системи на основі алгоритмічної моделі ця проблема в порівнянні з роботами [30, 102, 147] набуває зовсім нового значення і потребує розробки нового теоретичного апарату.

Існує багато апаратів для формалізації перетворень алгоритмічних моделей, зокрема, алгоритмічна алгебра Глушкова [29], основні елементи якої наведені у табл. 1.6.

Таблиця 1.6

Операції системи алгоритмічних алгебр

Операція	Позначення	Пояснення
Множення (композиція)	$A \times B$ чи AB	Операція асоціативна $(AB)C = A(BC)$ Мають місце і такі співвідношення: $AE = EA = A$, $AN = NA = N$ Тут оператори (E – тотожний, N – невизначений) відіграють роль констант.
α -диз'юнкція ($\alpha \in \hat{g}$)	$P_\alpha \vee Q$	Для довільного стану $m \in M$: 1) $R(m) = P(m)$, якщо умова $\alpha(m) = 1$ 2) $R(m) = Q(m)$, якщо умова $\alpha(m) = 0$ 3) $(P_\alpha \vee Q) = N$, якщо $\alpha = \mu$
α -ітерація ($\alpha \in \hat{g}$)	$\{P_\alpha\}$	Являє собою новий оператор S . Виконання оператора $\{P_\alpha\}$ полягає в перевірці умови α і застосуванні оператора P доти, поки ця умова залишається помилковою.

Апарат алгебри алгоритмів дозволяє зобразити алгоритми і робити над

ними еквівалентні перетворення, як вони виконуються в звичайній алгебрі чи аналізі [128]. Для реалізації апарату алгебри алгоритмів кожному операційному автоматові поставлено у відповідність дві алгебри: алгебру операторів \hat{y} і алгебру умов \hat{g} . Операторами з \hat{y} є будь-які однозначні перетворення інформаційної множини M (станів автомата). До операторів відносять, зокрема, усі мікрооперації, визначені на операційному автоматі (табл. 1.7)..

Таблиця 1.7

Порівняння елементів алгоритмічних моделей

Елементи ЛСА		Операції системи алгоритмічних алгебр		Типові елементи схем програм	
Зміст	Позначення	Зміст	Позначення	Зміст	Позначення
Функціональний оператор	A_i	оператор	\hat{y}	функція	$F(x)$
Логічний оператор	w_i	умова	\hat{g}	логічний вираз (функція)	L
Перетворення сигналів	x/y	присвоювання		процес визначення значення ↓ ↓	$y := F(x)$
Виконання одного з декількох перетворення (або)	$A_1 A_2 A_3$		$((A1_{\alpha_1} \vee A2_{\alpha_2}) \vee A3) = N$	вибір 	case x of 1: A_1 2: A_2 3: A_3
Об'єднання перетворень	$[\cdot], \{ \cdot \}$	множення (композиція)	$A \times B$ чи AB	складний оператор	Begin End
Виконання окремого алгоритму				окремо визначений алгоритм 	Procedure A
Перенесення виконання алгоритму вперед		α -диз'юнкція ($\alpha \in \hat{g}$)	$(P_{\alpha} \vee Q) = N$	умовний перехід 	if α then P else Q
Перенесення виконання алгоритму назад		α -ітерація ($\alpha \in \hat{g}$)	$\{P_{\alpha}\}$		repeat P until α

Умови на \hat{g} подані відображенням множини M у триелементну множину $\{1,0,\mu\}$ (істина, хибність, невизначеність). До умов з \hat{g} відносять всі елементарні логічні умови, значення яких є вихідними сигналами операційного автомата. Для умов з \hat{g} вводяться узагальнені булеві операції диз'юнкції, кон'юнкції і заперечення. Крім цих операцій визначається операція лівого множення умов на оператори. Якщо α - умова, а P - оператор, то добуток $P\alpha$ являє собою нову умову β , перевірка якої еквівалентна перевірці умови α після виконання оператора P .

Система $(\tilde{y}; \hat{g})$ називається системою алгоритмічних алгебр (САА). До складу алгебри \tilde{y} включаються всі операції автомата, а до складу алгебри \hat{g} - усі визначені на множині M елементарні логічні умови. У систему алгоритмічних алгебр введені операції, що показані в табл. 1.6 [127].

До побудованих у такий спосіб множин $\tilde{y}^{(0)}$ і $\hat{g}^{(0)}$ операторів і умов застосовуються наведені у таблиці операції. При цьому виникають нові оператори й умови, що разом із уже наявними складають множини $\tilde{y}(1)$ й $\hat{g}(1)$. До цих множин знову застосовується зазначений спосіб утворення нових операторів і умов. Об'єднавши всі побудовані множини $\tilde{y}(i)$ та $\hat{g}(i)$, одержимо шукані алгебри \tilde{y} і \hat{g} .

Порівняння апарату ЛСА, системи алгоритмічних алгебр Глушкова, засобів зображення граф-схем програм свідчить про їх тісний зв'язок, показаний у табл. 1.7.

З аналізу ЛСА видно, що алгоритмічні моделі як апаратної, так і програмної частин підсистем збору даних, можуть бути подані у алгебраїчній формі.

Алгебраїчна форма алгоритмічної моделі ПЗД дозволяє здійснити перехід до узагальненої операторної моделі [19], яка враховує невизначеність початкових даних, вхідних впливів та структурних змін системи [69]. Для відображення в алгоритмічній моделі операцій над функціями невизначеності введені означення операторів перетворення функцій [28]. Алгебраїчна форма дозволяє здійснювати формальні перетворення і оцінювати характеристики алгоритмічних моделей.

1.3.3. Методи оцінювання характеристик підсистем збору даних.
Одним з важливих кроків аналізу підсистем збору даних є оцінювання

характеристик системи (швидкодії, програмно-апаратних витрат, надійності та ін.).

Зокрема, важливою задачею для оптимізації структури ПЗД є визначення вимог до складності підсистем збору даних. Складність визначається як для програмної частини, так і для апаратної.

Складність системи – це якісна характеристика, тому не існує однозначної методики її оцінювання. Але можна розглядати складність структури та поведінки системи, складність її вивчення, створення та використання. Крім таких підходів до оцінювання складності існує поділ на види складності, які залежать від задачі, яка розв'язується, наприклад, складність аналізу існуючої системи, складність синтезу нової системи, складність тиражування створеної системи, складність репродукції існуючої системи [162].

Алгоритмічні моделі стали об'єктом дослідження при визначенні характеристик об'єктів, для моделювання яких вони призначені [133]. У цій області виділяються задачі одержання верхніх і нижніх оцінок складності алгоритмів.

Для одержання верхніх оцінок достатньо інтуїтивного поняття алгоритму [133]. Для цього будується неформальний алгоритм розв'язання конкретної задачі, і потім він формалізується для реалізації на певній мові. Якщо складність (час чи пам'ять) обчислення для цього алгоритму не перевищує значення деякої функції при всіх значеннях аргументу, то ця функція береться за верхню оцінку складності. Складність алгоритмів обчислень може визначатися різними способами [58, 115, 165, 167].

Велика розмірність системи (простору станів чи числа компонентів) не обов'язково означає велику складність системи і навпаки. Складність послідовних алгоритмів звичайно виміряється числом арифметичних дій. Складність паралельних алгоритмів виміряється часом, протягом якого алгоритм може бути реалізований на k -процесорній машині [58].

Проблему обчислювальної складності можна досліджувати з різних точок зору

Важливе місце у вирішенні проблеми складності займає аксіоматичний підхід до цієї проблеми. В [58] введені аксіоми (табл. 1.8) часової складності.

В табл. 1.8 аксіоми 2-4 визначають складність системи, яка складається з послідовно-паралельно чи ієрархічно з'єднаних підсистем (зі зворотним зв'язком). Ієрархічні аспекти відбиті в аксіомі 1, динамічні – в аксіомі 5.

Ці аксіоми зручні для різних алгебраїчних підходів до аналізу й обчислення складності, але цих аксіом виявляється достатньо для визначення складності систем лише у деяких випадках, зокрема, якщо розглядати час як міру складності системи. Аксіома 2 з цієї точки зору є вірною. Але якщо враховувати не тільки час, але й зміст перетворень, які відбуваються в підсистемі, та кількість невизначених даних, то дана аксіома не є правильною, і такий підхід до визначення складності є нехарактерним для ПЗД.

Таблиця 1.8

Аксіоми системної складності

Назва аксіоми	Зміст аксіоми
1. Ієрархія	Якщо Σ_0 підсистема Σ , то $\theta(\Sigma_0) \leq \theta(\Sigma),$ тобто, підсистема не може бути більш складною, ніж система в цілому.
2. Паралельне з'єднання	Якщо $\Sigma = \Sigma_1 \oplus \Sigma_2 \oplus \dots \oplus \Sigma_k$, тобто, Σ є паралельним з'єднанням систем $\{\Sigma_i\}$, то $\theta(\Sigma) = \max_{1 \leq i \leq k} \theta(\Sigma_i).$
3. Послідовне з'єднання	Якщо $\Sigma = \Sigma_1 \otimes \Sigma_2 \otimes \dots \otimes \Sigma_k$, тобто, Σ є послідовним з'єднанням систем $\{\Sigma_i\}$, то $\theta(\Sigma) \leq \theta(\Sigma_1) + \theta(\Sigma_2) + \dots + \theta(\Sigma_k).$
4. З'єднання зі зворотним зв'язком	Якщо присутня операція зворотного зв'язку (\rightarrow) з системи Σ_2 в систему Σ_1 , то $\theta(\Sigma_1 \oplus \Sigma_2) \leq \theta(\Sigma_1) + \theta(\Sigma_2) + \theta(\Sigma_2 \rightarrow \Sigma_1).$
5. Нормалізація	В класі систем, які задовольняються цим аксіомам, виділена підмножина систем F , для яких $\theta(\Sigma) = 0 \text{ для всіх } \Sigma \in F.$

Із системно-теоретичної точки зору найбільш розвинутою є теорія складності процесів, що моделюються кінцевими автоматами [58]. Складність залежить від структури компонентів і довжини обчислень. Таким чином, складність враховує не тільки загальне число обчислень, але й внутрішню складність підавтоматів.

Складність керування виходить з кількості обчислень, необхідних для повної керованості системи. У більшості практичних ситуацій внутрішня складність керованої системи буде перевищувати складність керування. Чим менше ця різниця, тим більш стійко (збалансовано) поводить себе система [166].

Існує декілька способів оцінювання складності апаратної частини систем, зокрема електронних схем. Часто використовують оцінку за Квайном, яка визначається як сумарне число входів усіх логічних елементів [156]. Іншими підходами є визначення складності в залежності від кількості елементів і зв'язків системи; кількості станів системи; об'єму обчислень, необхідних для вивчення системи; кількості двійкових розрядів, необхідних для ідентифікації системи. Але ці підходи не є універсальними і не дозволяють повною мірою оцінити складність системи.

Досить часто при аналізі складності апаратних засобів використовуються функції Шеннона при реалізації функцій алгебри логіки схемами функціональних елементів в довільному базисі. Складність $L(S)$ схеми S , в такому випадку, дорівнює сумі ваг всіх елементів схеми S . Складність $L(f)$ функції f є нижньою межею складностей схем, які реалізують f . Функція $L(n)$, яка дорівнює максимуму складностей функцій від n аргументів, називається функцією Шеннона [56].

В [162] складність C оцінюють на основі витрат E , часу T , складності методів M і ступеня адекватності A . Складовими показниками складності є її структура (кількість елементів і зв'язків), ступінь відповідності моделі системи (опису) реальній системі, час і витрати на вивчення (створення) системи, які залежать від складності обраних методів

$$C = \langle T, E, M, A \rangle,$$

де $T \in [0; \infty]$, $E \in [0; \infty]$, $M \in [0; \infty]$, $A \in [0; 1]$.

Складність системи тісно пов'язана з іншими характеристиками, які визначають ефективність ПЗД. Такими характеристиками ПЗД є невизначеність результатів збору даних, надійність та швидкодія системи,

оскільки ці характеристики значно впливають на ефективність функціонування ПЗД.

В теорії надійності систем [34] виділяють структурний та функціональний підходи [174].

Структурний підхід близький до класичної теорії надійності систем, його моделі надійності побудовані на основі структури системи та характеристик надійності її елементів, основними показниками якої є ймовірність безвідмовної роботи та коефіцієнт готовності. Як початкові дані для моделювання надійності використовуються ймовірно-часові характеристики відмов – випадкових подій втрати працездатності [68, 141].

Функціональний підхід ґрунтується на теорії надійності алгоритмів [17, 50, 139, 151], теорії надійності людино–машинних систем [10, 27, 35, 53, 143], теорії надійності трудових та технологічних процесів [22, 37, 150]. Моделі надійності такого підходу побудовані на основі алгоритму функціонування системи, а узагальненим показником надійності виступає ймовірність досягнення мети, яка для прикладних задач інтерпретується показниками безпомилковості, достовірності, своєчасності тощо. На відміну від структурного підходу теорія та практика функціонального підходу менше досліджена. Іноді ці два підходи комбінують для опису надійності системи [175].

Надійність ПЗД включає в себе надійність конструктивних елементів (надійність апаратних засобів) і обчислювальної системи (надійність програмних засобів). Під надійністю програм розуміють властивість правильно та своєчасно виконувати передбачені специфікацією функції в процесі взаємодії з операційним середовищем (сукупність технічних засобів обчислювальної системи, інформації, що обробляється, та людини), тобто функціональну надійність.

Існує декілька десятків емпіричних, ймовірнісних та статистичних моделей для оцінювання числових показників надійності програмних засобів [57]. Найбільш простими є емпіричні моделі, які основані на аналізі існуючої інформації про функціонування раніше розроблених програм. Наприклад, модель Холстеда оцінює кількість помилок в програмі після закінчення її розробки

$$N_{oui} = K_{HO} V_{on} \log_2(\eta_1 + \eta_2), \quad (1.6)$$

де K_{HO} - коефіцієнт пропорційності,

V_{on} - кількість операторів в програмі,

η_1, η_2 - число операторів та операндів в програмі.

Оцінку надійності конструктивних елементів, приладів чи систем знаходять за допомогою показників структурної надійності, наприклад, ймовірності безвідмовної роботи.

Сумарна інтенсивність відмов Λ та ймовірність безвідмовної роботи P_T визначаються за формулами:

$$\Lambda = \sum_i \lambda_i k_{1i} k_{2i}, \quad (1.7)$$

$$P_T = e^{-\Lambda T}, \quad (1.8)$$

де k_{2i} - коефіцієнт навантаження i -го елемента;

k_{1i} - коефіцієнт, який визначає умови експлуатації i -го елемента;

λ_i - інтенсивність відмови i -го елемента;

T – інтервал часу, протягом якого i -й елемент безвідмовно працює.

Швидкодія підсистем збору даних – це комплексне поняття, оскільки такі системи складаються з різних частин, оцінка швидкодії яких здійснюється по-різному.

Швидкодія підсистеми передачі даних залежить від протоколу передачі [163] та умов передачі. Оцінювання швидкодії здійснюється згідно з теоремою Шеннона [90]. Ця теорема регламентує передачу певної кількості інформації I за певний час T при смузі пропускання F і відношенні "сигнал/шум" в каналі S/N

$$I = FT \log (1 + S/N). \quad (1.9)$$

Швидкодію підсистеми регулювання можна оцінювати за часом закінчення перехідного процесу при заданій допустимій похибці [61].

Швидкодія засобів вимірювання визначається часом з моменту зміни вимірюваної величини до моменту встановлення показу приладу [24].

Таким чином, методи оцінювання характеристик окремих складових ПЗД ґрунтуються на різних моделях і підходах, які необхідно об'єднати для отримання комплексної оцінки системи в цілому. Тому розробка і удосконалення методів оцінювання складності, надійності та швидкодії підсистем збору даних мають важливе значення.

1.4. Методи структурного моделювання систем в умовах невизначеності

В переважній більшості практичних задач функціонування систем відбувається в умовах невизначеності вхідних даних, причому ступінь цієї невизначеності може суттєво різнитися [149, 173, 177]. Джерелом невизначеності можуть бути: недостатнє знання предметної області, нестача точної інформації про значення даних, невизначеність цілей тощо [14, 16, 48, 49, 64-67]. Функціонування систем в умовах окремих видів невизначеності інтенсивно вивчається багатьма дослідниками [16, 28, 48, 49, 82, 83, 106, 107, 154, 165, 192, 193, 195].

Для вирішення задачі синтезу систем в умовах невизначеності (УН) розроблено багато методів. Історично першим склався адаптивний підхід, орієнтований, в основному, на вирішення задач керування лінійними об'єктами в умовах апріорної параметричної невизначеності. Його ефективність обмежується необхідністю знання структури моделі об'єкта, що у більшості відомих на сьогодні алгоритмів адаптації повинна бути лінійною за параметрами [138, 148]. Структура моделі у цих задачах характеризується невизначеністю порядку передаточної функції.

Для оцінювання невідомих величин та процесів в умовах невизначеності існують статистичні та гарантовані *методи оцінювання* [81], зокрема методи фільтрації Калмана, байєсівський підхід, метод еліпсоїдального оцінювання тощо [12]. Недоліком статистичних методів оцінювання є необхідність знання статистичних властивостей об'єкта. Альтернативним до стохастичного є теоретико-множинний або гарантований підхід до трактування невизначеності.

Методи оцінювання та ідентифікації нелінійних об'єктів керування в реальному часі, як правило, зводяться до лінеаризації моделі та оцінювання невідомих величин стохастичними або гарантованими методами, або до використання непараметричних моделей та так званих інтелектуальних методів.

Необхідність керування в реальному часі об'єктами, що характеризуються високим рівнем нелінійності і "зашумленості", в умовах

структурної і параметричної апріорної та поточної невизначеностей привела до появи нового покоління моделей - нечітких [101, 111], нейромережових [181] і гібридних нечітких нейромереж [8, 18, 98].

Зараз ці напрямки знаходяться в стадії бурхливого розвитку, про що свідчить потік публікацій в IEEE Transactions on Neural Networks, Fuzzy Systems, Automatic Control [184, 198].

Окрім вищезазначених, існує багато інших методів, що можуть бути використані при структурному моделюванні ПЗД в умовах невизначеності. Останнім часом, наприклад, набули поширення методи, що базуються на так званому μ -аналізі [5, 81], апараті теорії невизначених графів, невизначених алгоритмів, методі узагальнюючих функцій невизначеності тощо.

Апарат теорії графів вже давно використовується для моделювання систем збору даних [20, 45, 76]. Останнім часом зріс інтерес дослідників до систем з досить великою кількістю вузлів, структура яких нерегулярна, складна і динамічно змінюється з часом [19, 197]. Прикладами таких ПЗД можуть бути соціальні, інформаційні, технологічні і біологічні системи. Для дослідження таких систем використовують невизначені графи (нечіткі і стохастичні). Зокрема, для моделювання комплексних мереж здебільшого використовують такі підходи [51]: випадкові пуассонівські графи і узагальнені випадкові графи [188, 201]; марковські випадкові графи і модель блукання по «графу графів» з ймовірностями, пропорційними бажаним властивостям [189, 190, 191]; еволюційна модель росту мережі [149], модель Прайса [199]. Здебільшого ці підходи засновані на генеруванні випадкового графу з вже відомою кількістю вершин і заданими ймовірнісними властивостями.

Ймовірнісний граф [187, 196] (рис. 1.10), який зображує схему алгоритму, дозволяє оцінити часові характеристики алгоритму, витрати процесорного часу, трудомісткість тощо [63].

Аналогічний вигляд мають нечіткі графи, які застосовуються у випадках, коли введені користувачем дані є результатом експертного оцінювання [8, 40, 47, 103, 140, 194, 204].

Нечіткі алгоритми визначаються впорядкованою множиною нечітких інструкцій (нечітких висловлювань), що містять поняття, які формалізуються нечіткими множинами [179]. Під нечіткими інструкціями розуміють

інструкції, що містять нечіткі поняття, які, здебільшого, містять словосполучення «приблизно», «трохи більше», «менше ніж» та ін.

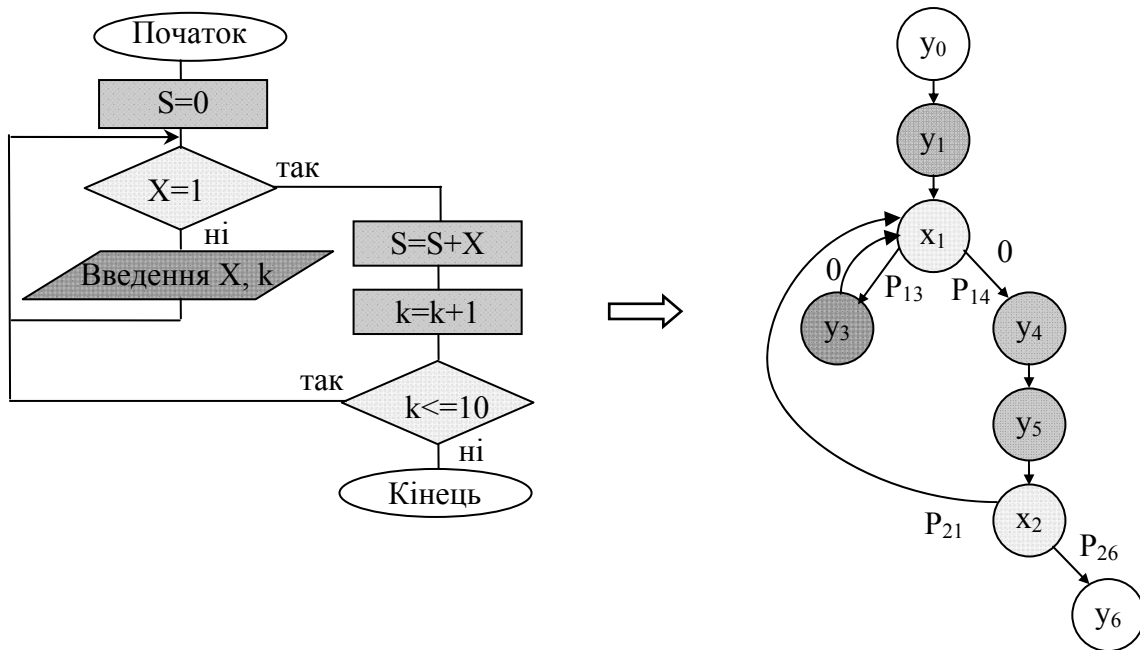


Рис. 1.14. Приклад імовірнісного графу

Метод узагальнюючих функцій невизначеності.

Існуючі методи моделювання розглядають структурні моделі систем в умовах невизначеності стохастичного [146] чи нечіткого [111] типу. В наявних роботах розглядається обмежений клас задач і видів невизначеності при побудові алгоритмічних моделей. Не існує загального підходу і мало досліджені алгоритмічні моделі, в яких для їх побудови одночасно використовується інформація, отримана з різних джерел: за допомогою експертної оцінки, за допомогою вимірювання і статистичної обробки і т.п. У той же час наразі розвивається *метод узагальнюючих функцій невизначеності* для моделювання та проектування систем [28, 38, 186].

Система узагальнюючих функцій [28] складається з правил утворення формул, правил переходу від формальних систем чітких чисел, випадкових величин, нечітких чисел до системи узагальнюючих функцій і назад. Для узагальнюючої функції визначені також правила узагальнення математичних операцій.

Під узагальнюючою функцією [38] розуміють додатно визначену

функцію на проміжку можливих значень аргументу, що позначається $\beta(x)$ і характеризує можливість π або ймовірність p прийняття аргументом значення з певного інтервалу $[x_1, x_2]$, $x_1 \in B$, $x_2 \in B$ за правилами:

$$p = \frac{\int_{x_1}^{x_2} d[\beta(x)]}{\int_B d[\beta(x)]}, \quad \pi = \frac{\int_{x_1}^{x_2} d[\beta(x)]}{\max_B \int_{[x_{i-1}, x_i]} d[\beta(x)]}, \quad (1.10)$$

де $x_{i-1}, x_i \in B$, $i = \overline{1, n}$,

n – кількість інтервалів розбиття B .

Для узагальнюючої функції визначені також правила узагальнення математичних операцій [28]. Всі операції поділені на три групи: нелінійні унарні, нелінійні бінарні, інтегро-диференціальні. За основу визначення цих операцій прийнятий операторний метод перетворення [19], що використовує інтегральні оператори виду:

$$\beta_Y(y) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \beta_X(\bar{x}) \varphi(\bar{x}, y, J, L) d\bar{x}, \quad (1.11)$$

де φ - ядро оператора, яке задовольняє умову

$$\int_{-\infty}^{+\infty} \varphi(\bar{x}, y, J, L) d\bar{x} = 1, \quad (1.12)$$

де J і L – характеристики операції, яка виконується;

n – кратність інтегрування, яка залежить від розмірності вектора \bar{x} і характеристик операцій J і L .

В [28] визначені також відносини порівняння невизначених даних в системі G :

Означення 1. Невизначені дані x , y вважаються рівними $X \equiv Y$, якщо $\beta_X = \beta_Y$.

Означення 2. Для невизначених даних $X \geq Y$, якщо $Z = X - Y$ і

$$\int_0^{+\infty} \beta_Z dz > \int_{-\infty}^0 \beta_Z dz.$$

Для оцінювання ступеня еквівалентності алгоритмів в умовах

невизначеності, розглянутого в п. 1.3.2, введемо поняття ступеня рівності.

Означення 3. Ступенем рівності невизначених даних x і y , які характеризуються функціями невизначеності β_x і β_y , відповідно, будемо називати величину

$$d = \int_{-\infty}^{+\infty} \beta_{xy}(x = \xi, y = \xi) d\xi = \int_{-\infty}^{+\infty} \beta_x(\xi) \cdot \beta_y(\xi) d\xi. \quad (1.13)$$

Очевидно, $d=1$ при $x = y$ за означенням 1.

Головною перевагою методу є його багаторівневий характер, що дозволяє поєднувати моделювання процесів, які проходять в окремих перетворювачах сигналів, і в розподілених системах в цілому. Зазначений метод, на основі моделювання динаміки інформаційних потоків, дозволяє врахувати в одній моделі алгоритмічний, структурний і статистичний аспекти функціонування розподілених ПЗД, оптимізувати інформаційні процеси як на мікро-, так і на макрорівні [38].

2. РОЗРОБКА МЕТОДУ ПОБУДОВИ АЛГОРИТМІЧНИХ МОДЕЛЕЙ ПІДСИСТЕМ ЗБОРУ ДАНИХ

Інформаційна технологія оптимізації структури підсистем збору даних (ПЗД) складається з методів і алгоритмів, які ґрунтуються на комплексі взаємопов'язаних моделей.

2.1. Комплекс моделей інформаційної технології структурної оптимізації підсистем збору даних АСУТП

Схема зв'язку моделей, які використовуються при оптимізації структури ПЗД, зображена на рис. 2.1.

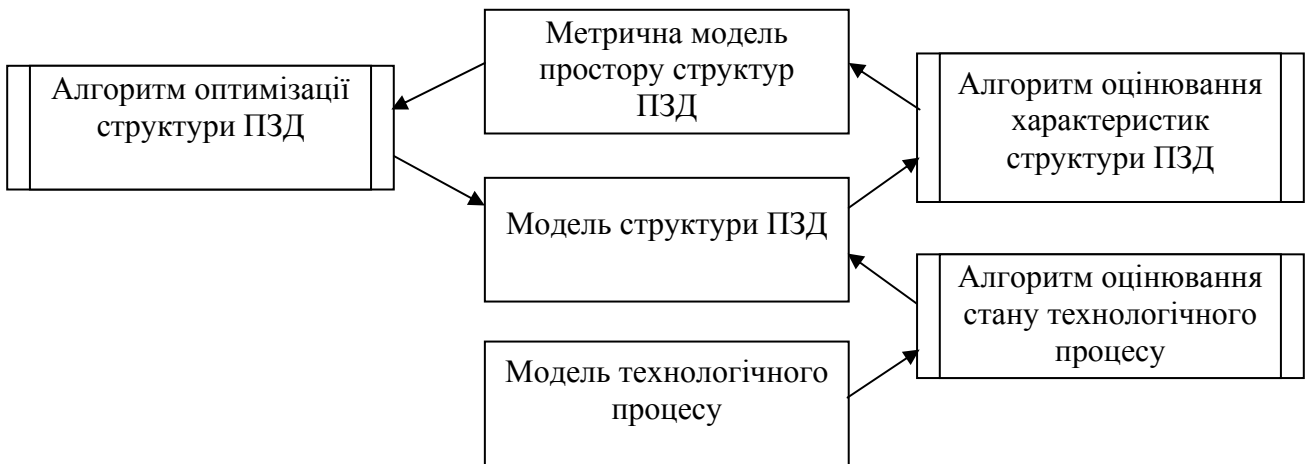


Рис. 2.1. Зв'язок моделей інформаційної технології оптимізації структури ПЗД

Основою для всього комплексу є модель технологічного процесу як об'єкта контролю. Нехай модель технологічного процесу подана системою рівнянь

$$F(X) = 0, \quad (2.1)$$

де X – вектор параметрів стану технологічного процесу.

Нехай також підсистема збору даних АСУТП безпосередньо сприймає інформацію про вектор параметрів $X'[n] \subseteq X[N]$, $n \leq N$.

Якщо ефективність функціонування технологічного процесу визначається показником $Q\{X'\}$, де $X'[k] \subseteq X[N]$, $k \leq N$, то для вирішення задач контролю та керування технологічним процесом необхідно отримати розв'язок системи

рівнянь (2.1):

$$X'[k] = F^{-1}(X''[n]), \quad (2.2)$$

де F – оператор моделі об'єкта;

$X'[k]$ – вектор параметрів технологічного процесу;

k – кількість параметрів технологічного процесу, які розраховуються;

$X''[n]$ – вектор контрольованих параметрів;

n – розмірність вектора контрольованих параметрів.

Якщо $k = n$, то система рівнянь за умови невиродженості і сумісності може бути розв'язана з точністю, яка визначається обчислювальною похибкою обраного алгоритму розв'язання і похибками початкових даних (результатів вимірювання X''). Якщо $k \neq n$, то розв'язок може бути отриманий за допомогою алгоритму оцінювання [39], який мінімізує середню квадратичну похибку результатів. В обох випадках похибка результату залежить від вибору множини контрольованих параметрів технологічного процесу, що, в свою чергу, впливає на структуру ПЗД.

Модель структури ПЗД, яка залежить від множини контрольованих параметрів, їх типів, розташування точок контролю у просторі, а також вимог до метрологічних характеристик кожного вимірювального каналу, є об'єктом оптимізації запропонованої інформаційної технології. Для здійснення оптимізації необхідно у просторі варіантів структур ПЗД визначити метрику, яка дозволить застосувати чисельні методи оптимізації.

Метрику ρ у просторі структур ПЗД визначимо як функціонал від комплексу характеристик ПЗД

$$\rho = F[c, \theta_S, P_H, I, T_u, \sigma], \quad (2.3)$$

де c – вартість ПЗД;

θ_S – складність;

P_H – надійність;

I – інформативність (кількість інформації за один цикл роботи підсистеми);

σ – розкид (СКВ) результату (невизначеність результатів збору даних про стан об'єкта);

T_u – тривалість циклу збору даних (швидкодія підсистеми).

Таким чином, постає задача розробки такої моделі програмно-апаратного комплексу ПЗД, яка дозволить формалізувати оцінювання зазначених

характеристик і побудувати метричний простір структур, на якому здійснюватимемо оптимізацію ПЗД.

2.2. Розробка методу моделювання програмно-апаратного комплексу підсистем збору даних

Розв'язання задачі оптимізації структури ПЗД потребує розробки моделі підсистеми, яка дозволяє застосування формальних процедур оптимізації. Такою є алгоритмічна модель.

2.2.1. Формальна система алгоритмічних моделей. Всі операції, які виконуються у ПЗД, можна розділити на такі, які вирішуються за допомогою програмних засобів, і такі, які вирішуються за допомогою апаратних засобів.

Алгоритмічні моделі як апаратної, так і програмної частин ПЗД, можуть бути подані у алгебраїчній формі. У формалізованому вигляді запропонуємо означення алгоритмічної моделі за допомогою нотації Бекуса – Наура (наведені лише основні означення):

$\langle \text{алгоритмічна модель} \rangle$	$::= \langle \text{ідентифікатор} \rangle ! A(B) \langle \text{блок} \rangle A(E)$
$\langle \text{ідентифікатор} \rangle$	$::= M \langle \text{номер} \rangle$
$\langle \text{блок} \rangle$	$::= \langle \text{ідентифікатор} \rangle ! \langle \text{пусто} \rangle ! \langle \text{елемент} \rangle ! [\langle \text{блок} \rangle \langle \text{блок} \rangle]$
$\langle \text{елемент} \rangle$	$::= \langle \text{перетворення} \rangle ! \langle \text{умова} \rangle ! \langle \text{знак} \rangle$
$\langle \text{перетворення} \rangle$	$::= E \langle \text{номер} \rangle \langle \text{операція} \rangle ; \langle \text{вхідні дані} \rangle ; \langle \text{результати} \rangle$
$\langle \text{умова} \rangle$	$::= w \langle \text{номер} \rangle \langle \text{логічний вираз} \rangle ; \langle \text{вхідні дані} \rangle \langle \text{знак переходу} \rangle$
$\langle \text{знак} \rangle$	$::= \langle \text{знак переходу} \rangle ! $
$\langle \text{знак переходу} \rangle$	$::= _ \langle \text{номер} \rangle ! _ \langle \text{номер} \rangle$
$\langle \text{операція} \rangle$	$::= \langle \text{пусто} \rangle ! \langle \text{тип операції} \rangle : [\langle \text{зміст операції} \rangle]$
$\langle \text{тип операції} \rangle$	$::= \langle I \rangle ! \langle S \rangle ! \langle C \rangle ! \langle T \rangle ! \dots$
$\langle \text{вхідні дані} \rangle$	$::= \langle \text{пусто} \rangle ! (\langle \text{ім'я} \rangle : [\langle \text{тип} \rangle] ! \langle \text{значення вхідного даного} \rangle)$
$\langle \text{результати} \rangle$	$::= \langle \text{пусто} \rangle ! (\langle \text{ім'я} \rangle : [\langle \text{тип} \rangle])$

Примітки:

1. Службові символи нотації: $\langle \rangle$ – межі синтаксичної конструкції; $::=$ – “це”; $!$ – “або”; \dots - “інші”; $[\dots]$ – необов’язкові елементи.

2. $_ \langle \text{номер} \rangle$ - перехід вперед; $_ \langle \text{номер} \rangle$ - перехід назад.

3. $\langle I \rangle$ - інформаційні операції; $\langle S \rangle$ - вимірювальні операції; $\langle C \rangle$ - обчислювальні операції; $\langle T \rangle$ - операція передачі.

4. Невизначені змінні позначаються іменами довжиною не більше 4 символів.

5. $C: I(x * g)$ - позначення інтегро-диференціального (динамічного)

перетворення у формі інтеграла Дюамеля, де x - вхідна функція, g - ядро перетворення (імпульсна перехідна функція динамічного перетворення),

6. { (,) , пропуск } - роздільники.

7. Апаратна або програмна реалізація при необхідності явного опису позначається індексами h і s відповідно біля знака операції: $E(on_h;();())$.

8. Пустий оператор позначається $E()$.

Приклади типових записів наведені у табл. 2.1.

Таблиця 2.1

Приклади типових записів алгоритмічної моделі

Запис алгоритмічною мовою	Алгебраїчна форма (R- форма)	Коментар
Перевірка умови і розгалуження $if(a) then n1 else n2$	$w_1(a) \lfloor_{n_1} M_{n_2} \lfloor_{n_3} M_{n_1} M_{n_3}$	a - логічна змінна, $n_1 \in N$ - номер оператора алгоритму, до якого здійснюється перехід при істинному значенні a , $n_2 \in N$ - при помилковому, N - нумерована множина операторів алгоритму
Обчислення функції $p_2 = N(p_1)$	$E(C : N(p_1); (p_1); (p_2))$	p_1 - вихідні дані; p_2 - результат обчислення; N - формула обчислення.
Ініціалізація констант $\# define p_2, p_1$	$E(C; (p_1); (p_2))$	p_1 - значення константи; p_2 - ім'я константи.
Вимірювання $import (&p_1)$ $p_2 = N(p_1)$ $p_3 = \varepsilon$	$E(S : N, g; (p_1); (p_2, p_3))$	p_1 - вимірювана величина; p_2 - результат вимірювання; p_3 - похибка вимірювання; N - рівняння перетворення; g - динамічна характеристика перетворення.
Введення експертних даних $scan (&p_1, &p_2);$ $p_3 = (p_1 + p_2) / 2;$ $p_4 = (p_2 - p_1) / \sqrt{12}$	$E(I : Ex; (p_1, p_2); (p_3, p_4))$	p_1, p_2 - ліва та права границі оцінки експерта; p_3 - середнє значення; p_4 - СКВ.
Затримка $delay (\tau)$	$E(C : I(p_1 * \delta(t - \tau)); (p_1); (p_2))$	τ - час затримки.

Побудуємо алгоритмічні моделі деяких типових систем.

1. Замкнений канал інтегруючого перетворення

Замкнений вимірювальний канал з інтегральним законом перетворення (рис.2.2) включає: сенсор, контролер, компаратор і інтерфейси.

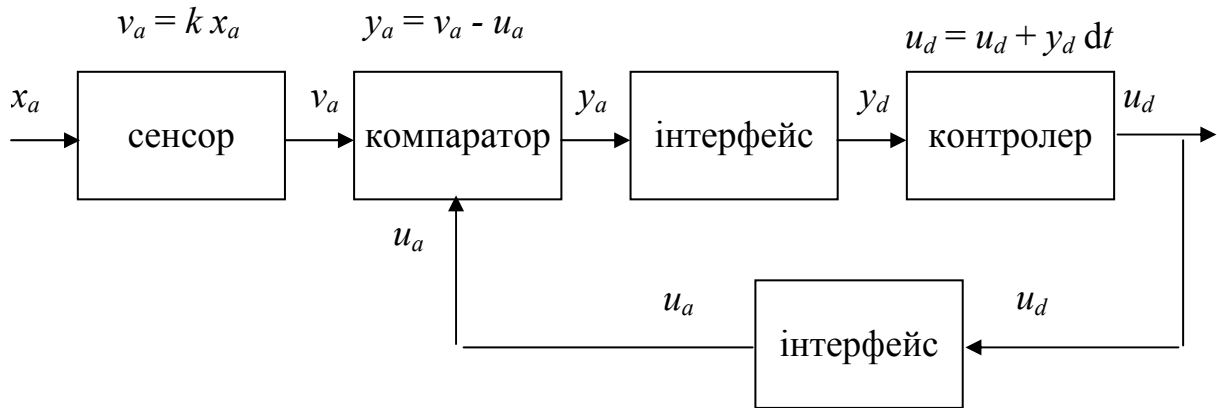


Рис. 2.2 - Функціональна схема замкнутого вимірювального каналу з інтегральним законом перетворення

У каналі виконується система перетворень

$$\begin{cases} v = kx \\ y = v - u \\ u = \int_0^t y \cdot dt \end{cases} \quad (2.4)$$

алгоритмічна модель з урахуванням інтерфейсних перетворень і форми сигналів (цифровий d /аналоговий a) буде мати вигляд:

$$M = A(B) \begin{matrix} E_1(k x_a; (x_a); v_a) \\ E_2(v_a - u_a; (v_a, u_a); y_a) \\ E_3(I_{AD}; (y_a); y_d) \\ E_4(u_d + y_d dt; (y_d); u_d) \\ E_5(I_{DA}; (u_d); u_a) \end{matrix} \quad | \quad 2 \quad A(E) \quad (2.5)$$

2. Алгоритмічна модель нечіткого оцінювання

Для збору даних в умовах дії погано формалізованих факторів використовують системи, керовані знаннями. Тоді до системи додається база знань, засоби її поповнення (навчання) та оцінювання факторів.

Розглянемо розв'язання цієї задачі на прикладі системи, в якій оцінювання здійснюється на основі нечіткої логіки.

Функціональна схема підсистеми оцінювання на базі нечіткої логіки наведена на рис.2.3. Схема складається з сенсорів для вимірювання частини параметрів об'єкта управління ОУ, системи прийняття рішень, моделі ОУ, бази знань. Система прийняття рішень включає три основних блоки - блок фазифікації, блок формування логічного рішення і блок дефазифікації.

У блоці фазифікації вхідні змінні описуються функціями належності (ФН) μ_x .

У блоці формування логічного рішення на основі матриці знань (бази

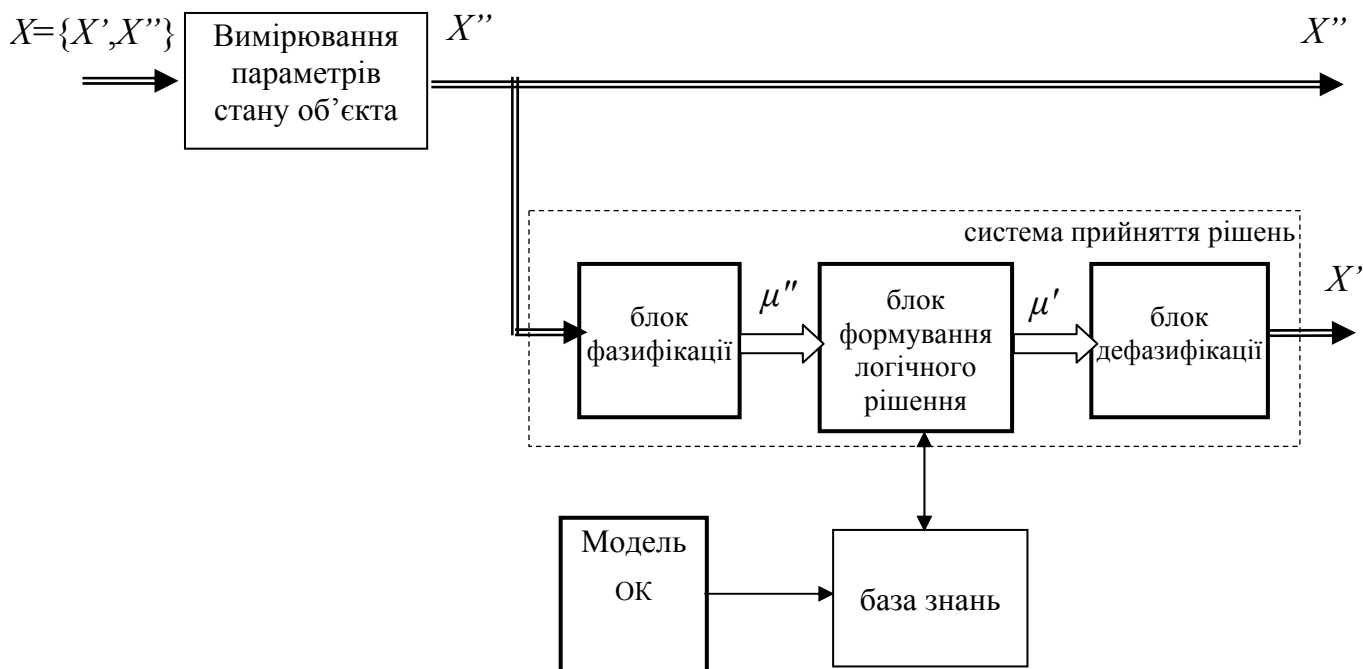


Рис.2.3 - Функціональна схема системи нечіткого оцінювання правил R_j) записуються лінгвістичні правила виду ЯКЩО $X''=(x_1, x_2, \dots x_n)$, ТО $X'=(x_{n+1}, x_{n+2}, \dots x_N)$.

Найчастіше використовується "мінімаксий" метод логічного рішення, в результаті якого визначається функція належності кожного з компонентів вектора X' за правилом

$$\forall U \in (u_{min}, u_{max}): \mu_U(u) = \max_{\forall R:u=U} \left[\min \right] \quad (2.6)$$

У блоці дефазифікації отримана результуюча функція належності для керуючого впливу на об'єкт керування перетвориться в числову величину, як правило, методом визначення "центра ваги" площини S результуючої фігури, що лежить під графіком результуючої ФН. Загальне правило розрахунку абсциси центра ваги ділянки площі, охопленою результуючою функцією в межах зміни перемінної u_1 від u_2 до, визначається по формулі

$$u_c = \frac{\int_{u_1}^{u_2} u \mu(u) du}{\int_{u_1}^{u_2} \mu(u) du} \quad (2.7)$$

Для формалізації алгоритмічної моделі системи нечіткого оцінювання введемо окремі позначення, показані у таблиці 2.2.

Алгоритмічна модель системи рис.2.3 подається нескінченим циклічним виконанням трьох паралельних процесів:

$$M_0 = A(B) \parallel [M_1 M_2] A(E)$$

де M_1 – процес оцінювання впливових факторів;

M_2 – процес навчання бази знань.

Відповідно до позначень таблиці 2.2 можна записати:

$$\begin{aligned} M_1 = & A(B) E_1(1,(),(i:int)) E_2(Im,(X),(X'':real,\varepsilon:real)) E_3(C_F,(X''),(\mu:function)) \\ & E_4(i+1,(i),(i:int)) w_5(i \leq n,(i)) _2 E_6(1,(),(j:int)) E_7(x'_{i min},(),(x'_{i 0}:real)) \\ & E_8(I_S,(x'_{i 0}=u_0,B),(L,f:array[n,m],e:real,x'_{i}:real)) E_9(C_L,(\mu_{x'}:array \\ & [n+1] \text{ of function},f:array[n,m],e:real,x'_{i}:real),(\mu_{x'}:real)) \\ & E_{10}(x'_{i 0}+dx'_{i},(x'_{i}), (x'_{i})) w_{11}(x'_{i} < x'_{i max},(x'_{i})) _7 \\ & E_{12}(C_D,(\mu_{x'}:function),(x'_{i}:real)) \\ & E_{13}(j+1,(j),(j:int)) w_{14}(j \leq N-n,(j)) _7 A(E); \end{aligned} \quad (2.8)$$

$$M_2 = A(B) E_1(S,(),(f:array[n])) A(E).$$

Позначення операцій алгоритмічної моделі системи на основі нечіткої логіки

Операція	Модель	Позначення		
		Вхід	Вихід	Операція
Фазифікація	$E(C_F, (x), (\mu: function))$	x - значення	μ - функція належності	C_F – функція належності обраного типу з центром ваги у точці x
Дефазифікація	$E(C_D, (\mu: function), (x: real))$	μ - функція належності	x – значення	$C_D : x = \int x \mu(x) dx / \int \mu(x) dx$ $X \quad \quad \quad X$
Пошук даних у базі знань	$E(I_S, (w, B: array[n+1, M]), (L: logic, x: array[n, m]))$	w – умова пошуку, B – база знань, що містить n атрибутів, 1 результат, M правил.	L – логічна змінна, отримує значення TRUE, якщо пошук успішний, і FALSE, якщо неуспішний x – масив результатів розміром $[n, m]$, n – розмірність вектора входів системи, m – кількість правил, що відповідають ключу пошуку	I_S – пошук всіх записів, які відповідають умові w
Вилучення правила з бази знань	$E(I_R, (x_0: array[n, 1], B: array[n, M]), (B: array[n, M-1]))$	x_0 – ключ видалення, B – база знань з M правил	B – база знань зменшена на 1 правило	I_R – видалення правила, яке відповідає умові $\bigcap_{i=1}^n (x_i = x_{0i})$
Запис правила у базу знань	$E(I_I, (x_0: array[n, 1], B: array[n, M]), (B: array[n, M+1]))$	x_0 – атрибути правила, що додається B – база знань з M правил	B – база знань збільшена на 1 правило	I_I – дописування правила до бази знань
Нечіткий висновок	$E(C_L, (\mu_x: array [n] of function, x: array[n, m], y: real), \mu_y: real)$	μ_x – n функцій належності вхідних даних x – масив атрибутів правил розміром $[n, m]$, n – розмірність вектора входів системи, m – кількість правил, що відповідають умові $y=y_i$	μ_y – значення функції належності для $y=y_i$	$C_L : \mu_y(y_i) = \max_m \left[\min_{j=1}^n \mu_{x_j}(x_j) \right]$

Алгоритми навчання в системах, керованих знаннями, досить різноманітні. Найпростіший алгоритм передбачає зменшення апріорної невизначеності оцінюваних параметрів і додавання до бази знань, яка втілює нечітку модель об'єкта, додаткових правил на проміжках великої кривизни моделі.

Модель навчання, побудована за таким алгоритмом, має вигляд

$$M_2 = A(B) E_1(I_M, (x:real, f:array[n], u:real, e:real), (x_0, f_0, u_0, e_0))$$

$$E_2(S, (), (y:real)) E_3((x-y)^2, (x,y), (e)) w_4(e < e_0, (e, e_0)) _7$$

$$E_5(I_S, (V=V_{min}, B), (L, f:array[n,m], e:real, u:real))$$

$$E_6(I_R, (f:array[n,m], e:real, u:real, B), (B)) _10$$

$$E_7(I_S, (V-V_{min} < E, E, B), (L, f:array[n,m], e:real, u:real)) w_8(L, ()) _10$$

$$E_9(I_L, (f:array[n,m], e:real, u:real, B), (B)) E_{10}() A(E)$$

3. Алгоритмічна модель чіткого оцінювання

Узагальнена структурна схема чіткого оцінювання наведена на рис.2.4.

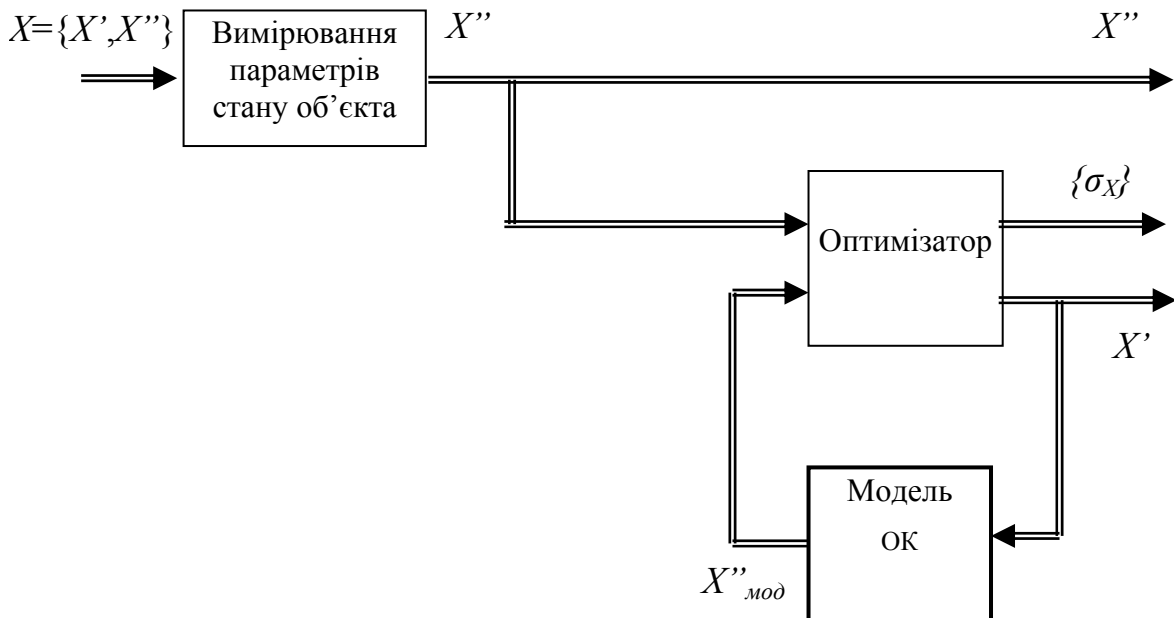


Рис.2.4. Функціональна схема системи чіткого оцінювання

Система складається з блоків вимірювання підмножини X'' параметрів об'єкта, моделі об'єкта і оптимізатора аналогічно рис. 2.3. Але пошук оптимального вектора оцінюваних параметрів X' здійснюється за допомогою алгоритму оптимізації, вибір якого залежить від вигляду моделі ОК (унімодальність або полімодальність, дискретність або неперервність, наявність або відсутність особливих точок, наявність та вид обмежень, розмірність векторів X' і X'').

Отже, алгоритмічна модель системи рис. 2.4 матиме вигляд:

$$M_1 = A(B) \ E_1(I, \emptyset, (i: \text{int})) \ E_2(\text{Im}, (X), (X'': \text{real}[n], \varepsilon_S : \text{real}[n])) \ E_3(i+1, (i), (i: \text{int})) \\ w_4(i \leq n, (i)) _2 \ M_2(\text{Iop}, (M_{OK}, X''), (X': \text{real}[k], \sigma_X : \text{real}[n+k])) \ A(E) \\ M_2 = A(B) \ E_1(C; (0, h); (X', h)) \ E_2(C: M_{OK}(X'); (X'); (X''_0)) \ E_3(C: \sigma(X'' - X''_0); \\ (X'', X''_0); (\sigma_0)) \ E_4(C: h \cdot \text{grad}(M_{OK}), (h), (dX')) \ E_4(C: (X'_0 + dX'); (X'_0, dX'); \\ (X'_0)) \ E_3(C: M_{OK}(X'); (X'); (X''_0)) \ E_5(C: \sigma(X'' - X''_0); (X'', X''_0); (\sigma)) \ w_4(|\sigma - \\ \sigma_0| > e) _4 \ A(E)$$

де M_{OK} – модель об'єкта;

N – рівняння перетворення сенсорів;

h – крок пошуку оптимуму;

M_2 – алгоритм пошукової оптимізації градієнтним методом.

2.2.2. Алгебра алгоритмічних моделей. Формалізуємо перетворення алгоритмічної моделі у вигляді алгебраїчної системи:

$$AS = (AM, OP) \tag{2.9}$$

де AM – множина алгоритмічних моделей; OP – множина операцій над ними.

Базовою операцією алгебри алгоритмічних моделей, яка забезпечує процес синтезу, є операція підстановки

$$P\{< \text{блок1} >, < \text{блок2} >\}, \tag{2.10}$$

що означає заміну <блока 1> на <блок 2>.

Підстановки повинні відповідати очевидним умовам

$$< \text{вхідні дані 1} > = < \text{вхідні дані 2} >, \tag{2.11}$$

$$< \text{результати 1} > \supseteq < \text{результати 2} >.$$

Послідовне виконання двох підстановок P_1 і P_2 назвемо добутком перестановок. Легко показати, що

- добуток перестановок має властивість асоціативності $P_1*(P_2*P_3)=(P_1*P_2)*P_3$,
- існує нейтральний елемент $1=P\{< \text{блок1} >, < \text{блок1} >\}$,
- існує обернений елемент $P^{-1}=P\{< \text{блок2} >, < \text{блок1} >\}$.

Така сукупність властивостей підстановок є підставою для твердження, що підстановки утворюють нескінченну групу [6]. Цей висновок є теоретичним підґрунтям використання алгебраїчних методів при оптимізації ПЗД на основі алгоритмічних моделей.

Операція підстановки еквівалентна множині операцій OP , яка складається з двох елементів:

- *paste* (B, n_1, n_2) - вставка блока B в алгоритмічну модель між елементами з номерами n_1 і n_2 ;
- *cut* (n_1, n_2) - вирізання блока з алгоритмічної моделі між елементами з номерами n_1 і n_2 ,

а також понять одиничної операції, яка не змінює моделі, і зворотної операції op^{-1} , яка задовольняє умову

$$op^{-1}(op(M)) = M. \quad (2.12)$$

З метою автоматизації формальних перетворень моделі введемо також додаткові операції над нею:

- next*(n) – функція визначення номера оператора, наступного за E_n ;
- prev*(n) - функція визначення номера оператора, попереднього до E_n ;
- arg*(n) - функція визначення аргументів оператора E_n ;
- res*(n) - функція визначення результатів оператора E_n ;
- last* - функція визначення номера останнього оператора;
- first* - функція визначення номера першого оператора.

2.3. Перетворення алгоритмічних моделей в умовах невизначеності

Використання запропонованої алгоритмічної моделі дозволяє формалізувати опис і перетворення структур ПЗД лише в умовах повної визначеності як даних, так і операцій над ними. Але в реальних системах такі

умови спостерігаються достатньо рідко. Тому є необхідним подальше узагальнення моделі на випадок невизначеності даних на основі методу узагальнюючих функцій, який дає можливість врахувати невизначеність різного типу, дозволяє оцінити на його основі характеристики системи, здійснити синтез оптимальних структур ПЗД, а в деяких випадках і оптимізацію ПЗД в процесі функціонування в реальному масштабі часу.

2.3.1. Отримання операторної форми алгоритмічної моделі.

Узагальнення алгоритмічної моделі на випадок невизначених даних здійсимо за допомогою описаних у п. 1.4.3 узагальнюючих функцій невизначеності.

Для перетворення моделі в умовах визначеності (R-форма) в узагальнену (G-форма) вихідну алгоритмічну модель записують у вигляді рядка символів, що утворюють певну математичну формулу в системі R з використанням позначень і правил, запропонованих у п. 2.2.1. Запис моделі перетворюється на G-форму за методикою:

1. Здійснюється декомпозиція моделі за виходами. В результаті загальна алгоритмічна модель подається у вигляді системи перетворень вектора контрольованих параметрів X'' на сукупність вихідних параметрів;

2. Послідовність запису операторів алгоритмічної моделі перетворюється з прямої («постфіксної») форми на обернену («префіксну») форму;

3. Невизначені вхідні змінні замінюються відповідними узагальнюючими функціями невизначеності;

4. Елементи алгоритмічної моделі замінюються операторами перетворення УФН у відповідності з табл. 2.3.

5. Вхідні дані операторів, які є проміжними результатами алгоритмічної моделі, замінюються на їх операторні вирази.

Подання моделі у вигляді системи операторних перетворень вимагає досить складної процедури декомпозиції багатовимірної системи, яка ґрунтується на побудові графу залежностей даних [63, 80] і розбитті цього графу на систему піддерев, коренями яких є вхідні дані ПЗД. Така процедура особливо складна у випадках наявності замкнених структур рекурсивного (зокрема, за наявності лічильників подій, інтеграторів тощо) та ітеративного (компенсаторів та регуляторів із зворотним зв'язком, процедур

покрокової оптимізації або ітеративного розв'язання рівнянь і систем рівнянь тощо) типів. Для спрощення операторної моделі багатовимірної системи введемо поняття векторного оператора.

Векторним оператором будемо називати оператор виду

$$\bar{\Phi}^{\{k_{\max}\}}(\{\varphi_i\}), \quad (2.13)$$

де $\{k_{\max}\}$ - розмірність вектора невизначених параметрів моделі;

$\{\varphi_i\}$ - вектор ядер операторів-компонент векторного оператора $\bar{\Phi}$.

Причому операторне рівняння

$$\{\beta(x_i \in X)\} = \bar{\Phi}^{\{k_{\max}\}}(\{\varphi_i\}) \cdot \prod_{j=1}^n \beta(x_j \in X''),$$

де $\{\beta(x_i \in X)\}$ - вектор УФН результатів перетворення;

$\beta(x_j \in X'')$ - УФН початкових даних,

еквівалентне системі операторних рівнянь

$$\begin{cases} \beta(x_1) = \Phi^{\{k_{\max}\}}(\varphi_1) \cdot \prod_{j=1}^{k_{\max}} \beta(x_j \in X'') \\ \dots \\ \beta(x_N) = \Phi^{\{k_{\max}\}}(\varphi_N) \cdot \prod_{j=1}^{k_{\max}} \beta(x_j \in X'') \end{cases}. \quad (2.14)$$

Твердження: Операторні перетворення

$$\beta(x_i) = \Phi^{k_i}(\varphi_i) \cdot \prod_{j=1}^{k_i} \beta(x_j \in Z_i), \quad (2.15)$$

де Z_i - множина вхідних даних, від яких залежить параметр x_i відповідно до графу залежностей,

$$i \quad \beta(x_i) = \Phi^{\{k_{\max}\}}(\varphi_i) \cdot \prod_{j=1}^{k_{\max}} \beta(x_j \in X'') \quad (2.16)$$

еквівалентні.

Справедливість цього твердження впливає з подання операторних перетворень у розгорнутому інтегральному вигляді

Операторні моделі основних елементів ПЗД

Елемент моделі	Алгоритмічна форма (R-форма)	Операторна форма (G-форма)	Порядок оператора	Примітка
1	2	3	4	5
Програмні елементи				
Перевірка умови і розгалуження	$w_1(a>0) \mid_{n_1} M_{n_2} \mid_{n_3} M_{n_1} M_{n_3}$	$\bar{\Phi} = \begin{cases} \Phi^{(1)}(h(a)) \cdot [\beta(a)], \\ \Phi^{(1)}(1-h(a)) \cdot [\beta(a)]. \end{cases}$	2	$h(a)$ – одинична функція Хевісайда.
Нелінійне перетворення одного параметра	$E(C:N(p_1); (p_1); (p_2))$	$\beta(p_2) = \Phi^{(1)}(N)[\beta(p_1)].$	1	
Обчислення функції m змінних	$E(C:N(p_1, \dots, p_m); (p_1, \dots, p_m); (p_{m+1}))$	$\beta(p_{m+1}) = \Phi^{(m)}(N)[\beta(p_1) \cdot \dots \cdot \beta(p_m)]$	m	
Ініціалізація констант	$E(C;(p_1); (p_2))$	$\beta(p_2) = \Phi^{(1)}(1)[\delta(p_1)].$	1	$\delta[p_2 - p_1] = \begin{cases} 0, \text{при } p_1 \neq p_2, \\ \infty, \text{при } p_1 = p_2. \end{cases}$ $\int_{-\infty}^{+\infty} \delta[p_2 - p_1] dp_2 = 1,$ p_1 – значення константи, p_2 – ім'я константи.

1	2	3	4	5
Цикл з рекурсією	$E_1(C; (x_0); (x_0)) E_2(C; (1); (i))$ $E(C; N(x_0); (x_0); (x)) E(C; (x); (x_0)) E_3(C; (i+1); (i); (i))$ $w_4(i \leq m) _ 2$	$\{\Phi^{(1)}(N)\}^m \prod_{i=1}^m \beta(x_i).$	m	$\forall \beta(x_i) = \beta(x_0).$
Цикл з перебором	$E_1(C; (1); (i)) M_2 E_3(C; (i+1); (i); (i)) w_4(i \leq m) _ 2$	$\bar{\Phi} = \{\Phi^{(n_{M_2})}(M_2)\}.$	$n_{M_2} \times m$	m – розмірність векторного оператора (кратність циклу)
Цикл з ітерацією	$E_1(C; (0, e); (x, e)) E_2(C; (x); (x_0)) E_3(C; N(x); (x); (x))$ $w_4(x-x_0 < e) _ 2$	$\{\Phi^{(1)}(N)\}^m \beta(e).$	m	m – кількість ітерацій за формулою Якобі $m = \frac{\ln[e \cdot (1 - \ a\) \cdot b]}{\ln \ a\ } - 1,$ $\ a\ $ - норма вектора коефіцієнтів моделі N , $ b $ - модуль вільного члена, e - допустима похибка.
Апаратні елементи				
Спрацьовування дискретного сигналізатора	$E(S: ent[p_1], (p_1); (p_2))$	$\Phi^{(1)}(* h(a))[\beta(p_1)].$	1	$h(a)$ – одинична функція Хевісайда, a – поріг спрацьовування.
Аналогова фільтрація	$E(C: I(x * g); (x, t); (y, t))$	$\Phi^{(n)}(g)[\beta(x)].$	$\frac{\tau}{T}$	τ – інтервал кореляції вхідного сигналу, T – стала часу фільтра.

Продовження, табл. 2.3.

1	2	3	4	5
Вимірювання	$E(S:N,g; (p_1); (p_2,p_3))$	$\Phi^{(2)}(+) \beta(p_3), \Phi^{(m)}(g)\Phi^{(1)}(N)[\beta(p_1)] .$	$3 + \frac{\tau}{T}$	$\beta(p_3) = \frac{1}{\sqrt{2\pi}\sigma_3} e^{-\frac{(p_3)^2}{2\sigma_3^2}},$ $\beta(p_1) = \delta[p_1 - p_{10}],$ <p>p_1 – вимірювана величина, p_2 – результат вимірювання, p_3 – похибка вимірювання, N – статична характеристика вимірювального перетворення, g – імпульсна перехідна функція вимірювального каналу, τ – інтервал кореляції вимірюваної величини, T – швидкодія вимірювального каналу.</p>
Аналого-цифрове перетворення	$E_1(C; (1); (i))$ $E_2(I: \text{ent}[\text{ent}[p_1/p_2]/k^i], (p_1, p_2);$ $(P_3: \text{int}[m])) E_3(C: (i+1); (i);$ $(i)) w_4(i \leq m) 2$	$\bar{\Phi} = \left\{ \Phi^{(1)} \left(\frac{1}{(p_2 k^i)} \right), i = 1 \dots m \right\}.$	$1 \times m$	p_2 – роздільна здатність АЦП, k – основа системи числення АЦП, m – розрядність АЦП.
Цифро-аналогове перетворення	$E_1(C; (1); (i)) E_2(C_h: [p_{1h} = p_{1h}$ $+ p_{2hi} * p_{3h} * k^i], (P_{2h}, p_{3h}, k);$ $(p_{1h})) E_3(C: (i+1); (i); (i))$ $w_4(i \leq m) 2$	$\Phi = \left\{ \Phi^{(m)} \left(\sum_{i=1}^m p_{2hi} p_{3h} k^i \right) \right\}.$	m	p_2 – роздільна здатність ЦАП, k – основа системи числення ЦАП, m – розрядність ЦАП.

Продовження, табл. 2.3.

1	2	3	4	5
Цифрова фільтрація	$E(C: \sum_{i=0}^n K_{x_i} x_{-i} + \sum_{j=1}^m K_{y_j} y_{-j}; (X[n], Y[m]); (y))$	$\Phi^{(n+m)}(+)\left[\prod_{i=0}^n \beta(x_{-i}) \prod_{j=1}^m \beta(y_{-j})\right]$	$n+m$	$K_{x_i} = \frac{(-1)^i \sum_{j=i}^n C_j^i \frac{a_j}{\Delta t^j}}{\sum_{j=0}^m \frac{b_j}{\Delta t^j}},$ $K_{y_i} = \frac{(-1)^{i+1} \sum_{j=i}^m C_j^i \frac{b_j}{\Delta t^j}}{\sum_{j=0}^m \frac{b_j}{\Delta t^j}}.$
Введення експертних даних	$E(I: Ex; (p_1, p_2); (p_3, p_4))$	$\Phi^{(2)}(\beta)[\delta(p_1), \delta(p_2)]$	2	p_1, p_2 – ліва та права границі оцінки експерта, β - прийнятий вигляд УФН для опису експертних даних.
Передавання даних без втрат	$E(T: (\tau); (p_1); (p_2))$	$1(p_2)[\beta(p_1)]$	1	g_τ – імпульсна перехідна функція ланки затримки, $g_\tau = \frac{1}{2\pi} \int_0^\infty e^{-p\tau} e^{pt} dp,$ τ – час затримки.

$$\begin{aligned} \beta(x_i) &= \Phi^{\{k_{\max}\}}(\varphi_i) \cdot \prod_{j=1}^{k_{\max}} \beta(x_j \in X'') = \\ &= \int_{D_{x_1}} \cdots \int_{D_{x_{k_{\max}}}} \varphi_i(Z_i) \cdot \prod_{j=1}^{k_{\max}} \beta(x_j \in X'') \cdot dx_1 \cdots dx_{k_{\max}}, \end{aligned} \quad (2.17)$$

Розіб'ємо множину X'' на підмножину Z_i потужністю k_i і її доповнення $(X'' \setminus Z_i)$. Оскільки ядро $\varphi_i(Z_i)$ не залежить від параметрів $x \in (X'' \setminus Z_i)$, то k_{\max} -кратний інтеграл можна розбити на дві частини у вигляді

$$\beta(x_i) = \int_{D_{X'' \setminus Z_i}} \cdots \int_{D_{x_{k_i+1}}} \prod_{j=k_i+1}^{k_{\max}} \beta(x_j \in \{X'' \setminus Z_i\}) \left[\int_{D_{x_1}} \cdots \int_{D_{x_{k_i}}} \varphi_i(Z_i) \cdot \prod_{j=1}^{k_i} \beta(x_j \in Z_i) \cdot dx_1 \cdots dx_{k_i} \right] dx_{k_i+1} \cdots dx_{k_{\max}}. \quad (2.18)$$

Враховуючи, що за означенням $\int_{D_x} \beta(x) dx = 1$, отримуємо

$$\beta(x_i) = \left[\int_{D_{x_1}} \cdots \int_{D_{x_{k_{\max}}}} \varphi_i(Z_i) \cdot \prod_{j=1}^{k_i} \beta(x_j \in Z_i) \cdot dx_1 \cdots dx_{k_i} \right] = \Phi^{k_i}(\varphi_i) \cdot \prod_{j=1}^{k_i} \beta(x_j \in Z_i), \quad (2.19)$$

що і необхідно було довести.

Використання векторних операторів перетворення УФН суттєво спрощує моделювання систем паралельної та циклічної дії, а також структурної невизначеності у ПЗД.

Структурна невизначеність включає невизначеність наявності зв'язку між підсистемами, а також невизначеність результату операції вибору послідовності дій.

Модель структурної невизначеності заснована на конструкції оператора w найбільш загального виду

$$w : \text{if } (a) \text{ then } n_1 \text{ else } n_2, \quad (2.20)$$

де a - логічна змінна (логічна умова);

$n_1 \in N$ - номер оператора алгоритму, до якого здійснюється перехід при істинному значенні a ;

$n_2 \in N$ - при хибному;

N - нумерована множина операторів алгоритму [185].

Виходячи з аксіоматики узагальнюючих функцій [28], модель такого оператора буде мати вигляд двокомпонентного вектора

$$\beta(n) = \{[\beta_a(a_1) \cdot \beta_n(n_1)], [\beta_a(a_2) \cdot \beta_n(n_2)]\}, \quad (2.21)$$

де $\beta_a(a) = k \cdot \delta[a - a_1] + (k - 1) \cdot \delta[a - a_2]$;

$$\beta_n(n) = \delta[a - n];$$

k – імовірність (можливість) істинного значення умови $a = \{a_1, a_2\}$.

2.3.2. Еквівалентність алгоритмічних моделей в умовах невизначеності. Дві алгоритмічні моделі називають функціонально еквівалентними, якщо при однакових вхідних даних вони дають однакові результати. Еквівалентним називають перетворення, в результаті якого одержують еквівалентну алгоритмічну модель.

Еквівалентні перетворення алгоритмічної моделі здійснюються на основі властивостей:

- $paste(B, n_1, n_2) cut(n_1, n_2) \equiv 1$;
- $cut(n_1, n_2) paste(B, n_1, n_2) \equiv 1$;
- $paste(B_1, n_1, n_2) paste(B_2, n_3, n_4) \equiv paste(B_2, n_3, n_4) paste(B_1, n_1, n_2)$,
якщо $(n_1, n_2) \cap (n_3, n_4) = \emptyset$;
- $cut(n_1, n_2) cut(n_3, n_4) \equiv cut(n_3, n_4) cut(n_1, n_2)$,
якщо $(n_1, n_2) \cap (n_3, n_4) = \emptyset$;
- $En_1(op, X, Y) En_2(op^{-1}, Y, X) \equiv 1$.

алгоритмічних моделей в умовах невизначеності.

В умовах невизначеності поняття еквівалентності має розмиті границі. З огляду на обмежену вірогідність результату роботи ПЗД в умовах невизначеності, можна говорити лише про еквівалентність їх алгоритмічних моделей із заданою вірогідністю або про ступінь еквівалентності алгоритмічних моделей. Розглянемо можливий підхід до оцінювання цієї вірогідності.

Нехай є дві системи S_1 і S_2 причому системам відповідають алгоритмічні моделі M_1 і M_2 , які визначають перетворення вхідних даних X'' на результат функціонування X' . Оскільки моделі систем відрізняються, то і результати функціонування будуть різними і до того ж невизначеними, отже

$$X_1' = M_1(X'') \text{ і } X_2' = M_2(X''). \quad (2.22)$$

В п. 1.4.3 визначені відносини порівняння невизначених даних в системі G . Охарактеризуємо результати X_1' і X_2' функціями невизначеності $\beta_1(X_1')$ і $\beta_2(X_2')$ відповідно, які показані на рис. 2.5. Вірогідність збігання результатів в умовах невизначеності

$$B(X_1' = X_2') = \int_{\Omega_{X_1'} \cap \Omega_{X_2'}} \beta_1(X') \beta_2(X') dX', \quad (2.23)$$

де $\Omega_{X_1'}$ і $\Omega_{X_2'}$ - області значень результатів X_1' і X_2' відповідно.

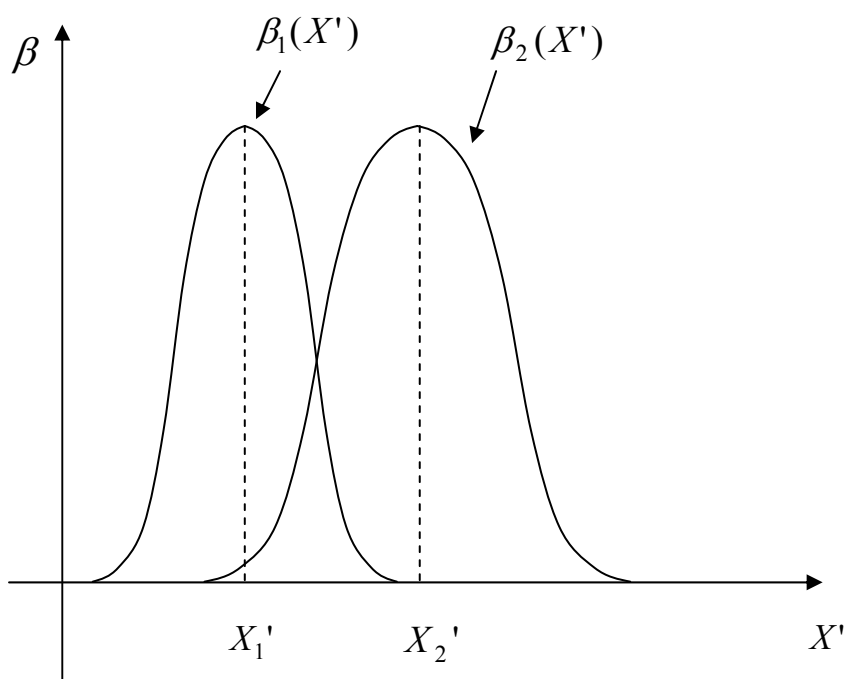


Рис. 2.5. Функції невизначеності результатів X_1' і X_2'

Виходячи з означення еквівалентності алгоритмічних моделей, можна сказати, що системи S_1 і S_2 в умовах невизначеності є еквівалентними з вірогідністю $B(X_1' = X_2')$. Введена міра еквівалентності дозволяє обґрунтувати метрику алгоритмічних моделей.

Результат роботи алгоритму можна означити на основі таких типів подання:

- 1) числове значення, яке належить деякому неперервному інтервалу можливих значень;
- 2) числове значення, яке належить деякій зліченній дискретній

множині можливих значень;

- 3) дія, яка здійснюється іншими технічними засобами системи;
- 4) зображення на екрані.

Для застосування означення 3 п. 1.4.3. для кожного виду результатів необхідно визначити метрику ρ' з урахуванням невизначеності. Враховуючи зручність застосування при розв'язанні оптимізаційних задач, використаємо евклідову метрику.

Метрику результатів 1-го типу означимо відповідно до виразу

$$\rho'_1 = \int_{-\infty}^{+\infty} z^2 \cdot \beta(z) \cdot dz, \quad (2.24)$$

де $z = x - y$.

Враховуючи незалежність результатів двох алгоритмів, отримуємо

$$\rho'_1 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - y)^2 \cdot \beta(x) \beta(y) \cdot dx \cdot dy. \quad (2.25)$$

Метрику результатів 2-го типу означимо відповідно до виразу

$$\rho'_2 = \sum_{i=1}^n \sum_{j=1}^n (x_i - y_j)^2 \cdot B_{x_i} \cdot B_{y_j}, \quad (2.26)$$

де n – потужність множини результатів;

B – вірогідність результату.

Вірогідність у загальному вигляді розраховується за формулою

$$B_{\xi} : \beta_{\xi} = \sum_{i=1}^n B_{\xi_i} \cdot \delta[\xi - \xi_i],$$

де $\delta[\xi]$ - дельта-функція.

Метрику результатів 3-го типу означимо аналогічно до виразу (2.26), вважаючи, що $\{y\}$ - множина можливих дій технічних засобів системи.

Метрика результатів 4-го типу вводиться окремо для двох випадків: коли зображення на екрані обирається з певної множини стандартних зображень – тоді означення метрики аналогічне виразу (2.24), і коли зображення має сталу структуру зі змінними параметрами (наприклад, графік функції) - тоді означення метрики аналогічне виразу (2.26), де x і y множини невизначених параметрів зображень двох алгоритмів.

Врахування невизначеності відкриває нові можливості оптимізації

систем на основі алгоритмічних моделей.

S_2 є менш витратним варіантом S_1 того ж призначення $c_{S1} > c_{S2}$, де c_{S1} , c_{S2} – вартість, відповідно, систем S_1 і S_2 . Тоді вибір ефективного варіанта ПЗД зводиться до оцінювання вартості ризику $U = (1 - B)/(c_{S1} - c_{S2})$.

Розглянемо приклад. Система обробки сигналів, наведена на рис. 2.6, може бути виконана у паралельному (рис. 2.6, а) і послідовному (рис. 2.6, б) варіантах.

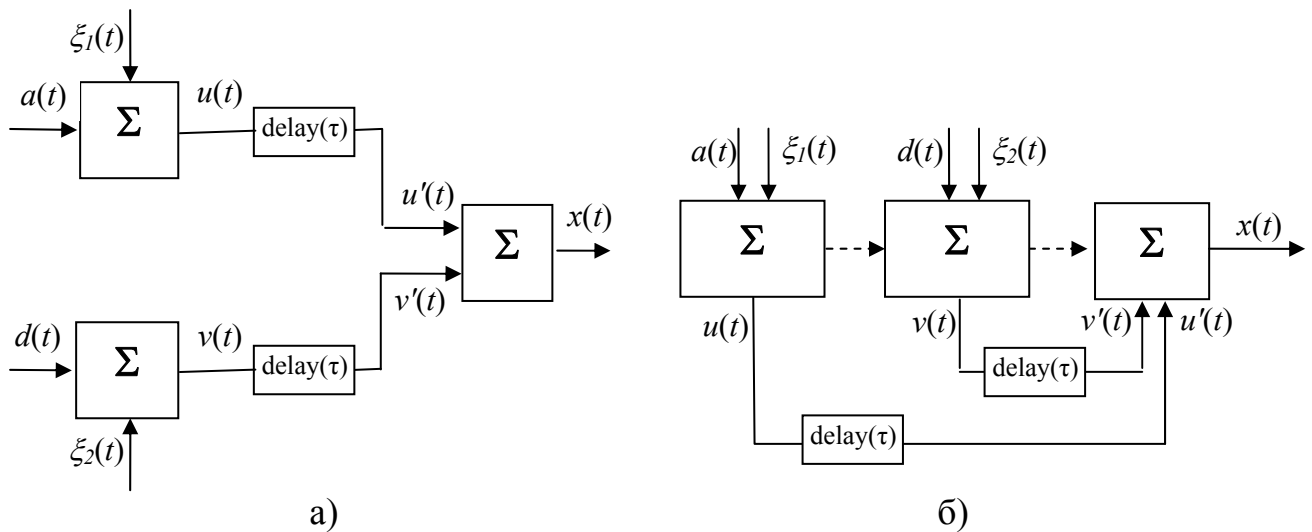


Рис. 2.6. Приклад системи обробки сигналів: а) паралельна, б) послідовна

Нехай сигнали, що надходять на входи систем мають вигляд:

$$a(t) = A \sin \omega t, \quad d(t) = D \sin \omega t, \quad (2.27)$$

де A і D – константи.

Алгоритмічна модель паралельного виконання матиме вигляд

$$M1 = A(B) \parallel [\text{Im}_1(a(t) / A \sin[\omega(t + \tau)]), \varepsilon_1 \quad \text{I}_2(f(a(t), \xi_1) / u(t) = a(t) + \xi_1) \\ \text{I}_3(u(t) / u'(t + \tau)) \quad \text{Im}_4(d(t) / D \sin[\omega(t + \tau)]), \varepsilon_2 \quad \text{I}_5(f(d(t), \xi_2) / \\ v(t) = d(t) + \xi_2) \quad \text{I}_6(v(t) / v'(t + \tau))] \text{I}_7(f(u'(t + \tau), v'(t + \tau)) / x(t)) A(E). \quad (2.28)$$

Алгоритмічна модель послідовного виконання має вигляд:

$$M2 = A(B) \text{Im}_1(a(t) / A \sin[\omega(t + \tau)]), \varepsilon_1 \quad \text{I}_2(f(a(t), \xi_1) / u(t) = a(t) + \xi_1) \quad \text{I}_3(u(t) / u'(t + \tau)) \\ \text{Im}_4(d(t + \tau) / D \sin[\omega(t + 2\tau)]), \varepsilon_3 \quad \text{I}_5(f(d(t + \tau), \xi_2) / v(t + \tau) = d(t + \tau) + \xi_2) \\ \text{I}_6(v(t + \tau) / v'(t + 2\tau)) \quad \text{I}_7(f(u'(t + \tau), v'(t + 2\tau)) / x(t)) A(E). \quad (2.29)$$

Алгоритмічні моделі (2.28) та (2.29) в операторній формі будуть мати вигляд:

$$\beta_1(x) = \Phi_7^{(2)}(+)[\Phi_6^{(n)}(g_\tau)[\Phi_5^{(2)}(+)[\beta(\xi_2), \Phi_4^{(2)}(+)[\beta(d), \beta(\varepsilon_2)]]], \Phi_3^{(n)}(g_\tau) [\Phi_2^{(2)}(+)[\beta(\xi_1), \Phi_1^{(2)}(+)[\beta(a), \beta(\varepsilon_1)]]].$$

$$\beta_2(x) = \Phi_7^{(2)}(+)[\Phi_6^{(n)}(g_{2\tau})[\Phi_5^{(2)}(+)[\beta(\xi_2), \Phi_4^{(2)}(+)[\beta(d), \beta(\varepsilon_2)]]], \Phi_3^{(n)}(g_\tau) [\Phi_2^{(2)}(+)[\beta(\xi_1), \Phi_1^{(2)}(+)[\beta(a), \beta(\varepsilon_1)]]].$$

На входи також поступає білий нормальний шум ξ_1 та ξ_2 .

В умовах визначеності (за відсутності шуму) результат отримуємо за допомогою R-моделі. Для схем рис. 2.3 (а, б) отримуємо:

$$\text{а) } A \sin[\omega(t + \tau)] + D \sin[\omega(t + \tau)] = (A + D) \sin[\omega(t + \tau)],$$

звідки $X_1' = (A + D)$;

$$\text{б) } A \sin[\omega(t + \tau)] + D \sin[\omega(t + 2\tau)] = \sqrt{A^2 + 2AD \cos(\omega\tau) + D^2} \sin[\omega t + \phi],$$

звідки $X_2' = \sqrt{A^2 + 2AD \cos(\omega\tau) + D^2}$.

Очевидно, в умовах визначеності алгоритмічні моделі (2.28) і (2.29) не еквівалентні, оскільки $X_1' \neq X_2'$.

В умовах невизначеності (за наявності шуму) результат перетворень отримуємо за допомогою G-моделі, яка дозволяє визначити функції невизначеності результатів $\beta(X_1'^*)$ і $\beta(X_2'^*)$:

$$\text{а) } X_1'^* = (A + D) + \xi_1 + \xi_2;$$

$$\text{б) } X_2'^* = \sqrt{A^2 + 2AD \cos(\omega\tau) + D^2} + \xi_1 + \xi_2;$$

За умови нормального білого шуму функції невизначеності результатів будуть гауссіанами з середніми значеннями відповідно X_1' і X_2' та дисперсіями $D_{Y^*} = \sigma_{\xi_1}^2 + \sigma_{\xi_2}^2$.

Вірогідність збігу результатів в умовах невизначеності

$$\begin{aligned} B(X_1'^* = X_2'^*) &= \int_{-(A+D)}^{+(A+D)} \left[\frac{1}{2\pi\sqrt{\sigma_{\xi_1}^2 + \sigma_{\xi_2}^2}} e^{-\frac{(Y-X_1')^2}{2(\sigma_{\xi_1}^2 + \sigma_{\xi_2}^2)}} \right] \left[\frac{1}{2\pi\sqrt{\sigma_{\xi_1}^2 + \sigma_{\xi_2}^2}} e^{-\frac{(Y-X_2')^2}{2(\sigma_{\xi_1}^2 + \sigma_{\xi_2}^2)}} \right] dX' = \\ &= \frac{1}{2\sqrt{2\pi} \cdot \sigma} e^{-\left(\frac{X_1' - X_2'}{2\sigma}\right)^2}. \end{aligned} \quad (2.30)$$

Залежність ступеня еквівалентності моделей (2.28) і (2.29) в умовах

невизначеності від часу виконання операцій і дисперсії шуму наведена на рис. 2.7. З рис. 2.7 видно, що при взятих для прикладу параметрах сигналів і швидкодії блока $0.15 < \tau < 0.25$ вірогідність співпадання результатів для систем (рис. 2.6, а і рис. 2.6, б) має максимум при сумарній дисперсії шуму $\sigma = 0.1$.

З аналізу еквівалентності моделей випливає, що за певних умов

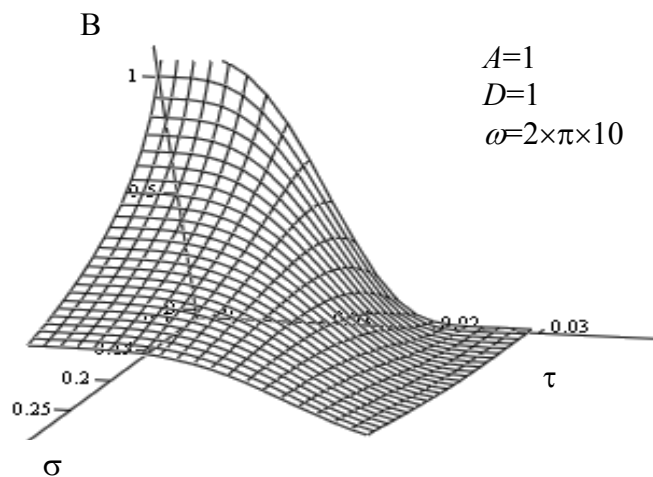


Рис.2.7. Вірогідність збігу результатів

послідовна система, яка може бути реалізована з меншою кількістю апаратних засобів, є еквівалентною паралельній системі, причому степінь еквівалентності може бути збільшений штучним введенням невизначеності.

Отже, на прикладі показано, що алгоритмічні моделі, які не еквівалентні в умовах визначеності, можуть бути еквівалентними у невизначених умовах, причому степінь еквівалентності може бути збільшений штучним введенням невизначеності.

2.4. Оцінювання характеристик підсистем збору даних

Ефективність ПЗД є узагальненням окремих характеристик підсистеми. Дослідимо способи оцінювання цих характеристик на основі алгоритмічної моделі ПЗД. Як базові показники для розрахунку характеристик підсистеми і перевірки достовірності моделі використані дані з [182, 200, 203].

Подання алгоритмічної моделі в операторній формі з врахуванням невизначеності дозволяє формалізувати оцінювання деяких важливих

характеристик ПЗД. Такими характеристиками є:

- складність системи;
- невизначеність результату контролю та керування;
- швидкодія системи;
- надійність системи.

2.4.1. Оцінювання складності. Як відзначено у розділі 1, існують різні означення поняття «складності» (складність функціонування системи, складність аналізу існуючої системи, складність синтезу нової системи, складність тестування системи, складність тиражування розробленої системи, складність репродукції існуючої системи) і, відповідно, різні методи оцінювання складності системи. Для інформаційної технології оптимізації структури ПЗД в умовах комбінованої невизначеності перевагу має оцінка складності функціонування, оскільки за певних обмежень решта характеристик складності є наслідками складності функціонування. Але існуючі методи оцінювання складності непридатні для підсистем збору даних, які містять як програмні, так і апаратні засоби, та функціонують в умовах невизначеності.

Як метод розв'язання поставленої задачі пропонується подавати опис ПЗД у вигляді алгоритмічної моделі, перетворювати алгоритмічну модель на узагальнену операторну модель і оцінювати складність операторної моделі. Операторна форма дозволяє сформулювати новий підхід до визначення складності системи. Оскільки кожен оператор, що описує дії вузлів системи в умовах невизначеності, має форму (1.11) і задовольняє умову асоціативності, то природною характеристикою складності є загальна кратність інтегрування операторної алгоритмічної моделі, тобто оцінка складності ПЗД в умовах невизначеності визначається сумарним порядком операторів моделі.

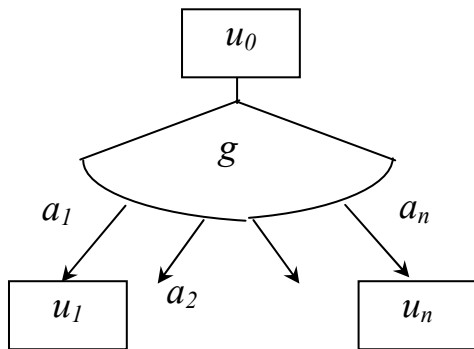
Очевидно, наявність розгалужень і циклів збільшує складність алгоритмічної моделі. Якщо перевірка умови g описується вектором узагальнюючих функцій розмірності n , то така алгоритмічна модель шляхом декомпозиції може бути перетворена на n моделей. Тоді алгоритмічна модель відповідно до схеми рис. 2.8, а матиме вигляд:

$$M = A(B) E_0(u_0) w_1(a_1)|_{(n+1)} w_2(a_2)|_{(n+2)} \dots w_n(a_n)|_{(n+n)} E_{n+1}(u_1) E_{n+2}(u_2) \dots E_{n+n}(u_n) A(E). \quad (2.31)$$

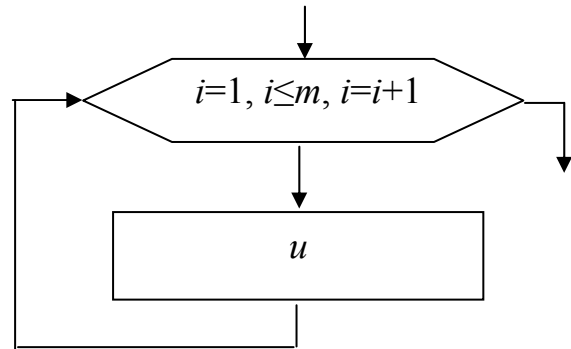
Оскільки порядок оператора перевірки умови дорівнює 2, то складність θ_S загальної моделі може бути знайдена за виразом

$$\theta_S = \theta_{S_0} + 2 \cdot n + \sum_{i=1}^n \theta_{S_i}, \quad (2.32)$$

де θ_{S_i} – складність підалгоритму u_i .



а)



б)

Рис. 2.8. Графічна інтерпретація складності алгоритму з розгалуженням (а) та циклічного алгоритму (б)

Алгоритмічна модель циклічної структури рис. 2.8, б має вигляд

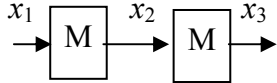
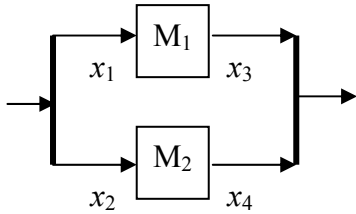
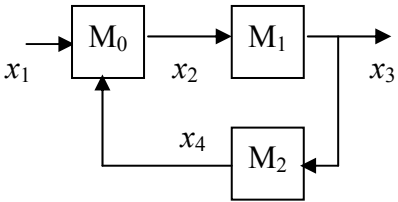
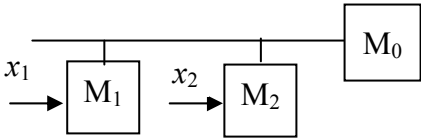
$$M = A(B) I_1((1), (1), (i)) E_2(u) I_3((i+1), (i), (i)) w_4(i \leq m)|_2 A(E). \quad (2.33)$$

Відповідна складність моделі

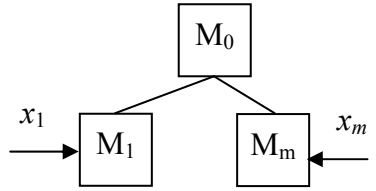
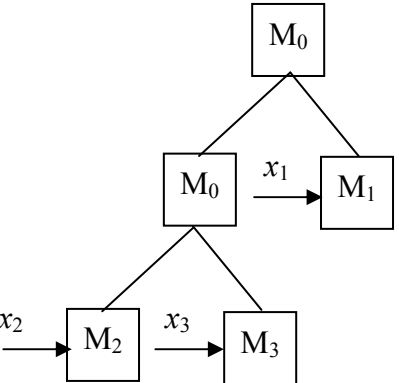
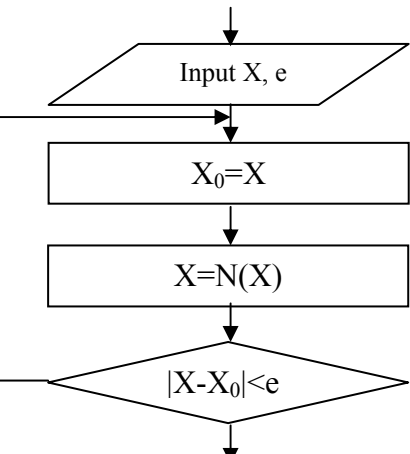
$$\theta_S = 4 + m \cdot \theta_{S_u}. \quad (2.34)$$

Оцінки складності типових структурних схем, отримані на основі підрахунку порядку операторної моделі, наведені у таблиці 2.4.

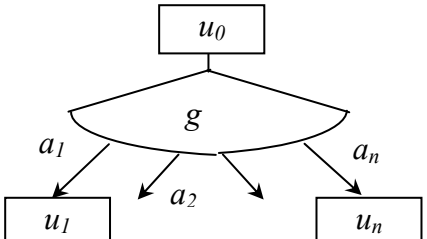
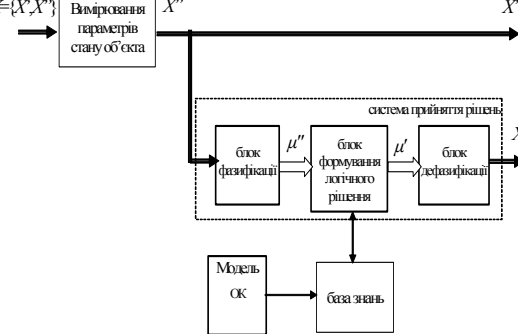
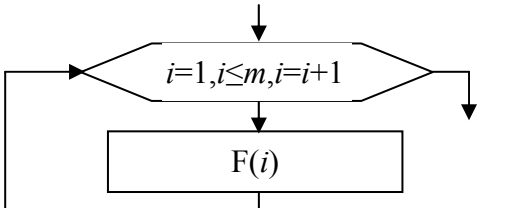
Прототип бази шаблонів

Тип структури/ алгоритму	Граф (схема)	Алгоритмічна модель	Операторна модель	Параметри	Складність
1	2	3	4	5	6
Послідовне з'єднання		$A(B) M_1 M_2 A(E)$	$\beta(x_3) = \Phi_2^{(n_2)}(M_2)$ $[\Phi_1^{(n_1)}(M_1)[\beta(x_1)]]$	n_1, n_2 – порядок операторів моделей підсистем	$S = n_1 + n_2$
Паралельне з'єднання		$A(B) \parallel [M_1 M_2] A(E)$	$\{\beta(x_3), \beta(x_4)\} = \overline{\Phi}^{(\max\{n_1, n_2\})}(M_1, M_2)$ $[\{\beta(x_1), \beta(x_2)\}]$	n_1, n_2 – порядок операторів моделей підсистем	$S = n_1 + n_2$
Замкнена структура		$A(B) M_0 M_1 M_2 _0 A(E)$	$\begin{cases} \beta(x_3) = \Phi^{(n_1)}(M_1) \Phi^{(2)}(M_0)[\beta(x_1), \beta(x_4)] \\ \beta(x_4) = \Phi^{(n_2)}(M_2)[\beta(x_3)] \end{cases}$	m – порядок передаточної функції лінеаризованої системи	$S = (n_1 + n_2 + 2) \cdot m$
Лінійна ПЗД		$A(B) M_0 \parallel [[M_{10} M_{ДТР 1}] [M_{20} M_{ДТР 1}]] A(E)$	$\{\beta(x_3), \beta(x_4)\} = \Phi^{(n_0)}(M_0)$ $[\overline{\Phi}^{(\max\{n_1, n_2\})}(M_1 M_{ДТР 1}, M_2 M_{ДТР 1})[\{\beta(x_1), \beta(x_2)\}]]$	$M_{ДТР 1}$ – протокол передавання даних у лінійній мережі	$S = n_1 + n_2 + n_0$

Продовження табл 2.4

1	2	3	4	5	6
Радіальна ПЗД		$A(B) M_{pm} [M_{DTP2} M_0] A(E)$ де $M_{pi} = A(B) \parallel [[M_i M_{DTP2}], [M_{p i-1} M_{DTP2}]] A(E)$	$\{\beta(x_3), \beta(x_4)\} = \Phi^{(n_0)}(M_0)$ $[\overline{\Phi}^{(n_i)}]_{(\{M_i M_{DTP1}\})}$ $[\{\beta(x_i)\}]$	M_{DTP2} – протокол передавання даних у радіальній мережі m – кількість радіальних підсистем	$S = n_0 + \sum_{i=1}^m n_i$
Ієрархічна ПЗД		$M_p = A(B) \parallel [[M_{10} M_{DTP2}], [M_p M_{DTP2}]] A(E)$ $M_R = A(B) M_p [M_{DTP2} M_R] A(E)$	$\{\beta(x_i), i=m+1 \dots 2m\} = \Phi^{(n_0)}(M_0 M_{DTR2})$ $[\Phi^{(n_p)}(M_p M_{DTR2})]^{r-1}$ $\overline{\Phi}^{(\max\{n_i\})}(\{M_i\})[\{\beta(x_i), i=1 \dots m\}]$	m – кількість контрольованих параметрів M_{DTP2} – протокол передавання даних у радіальній мережі d – коефіцієнт мультиплексування $r = \text{ent}[\log_d n] + 1$ - кількість рівнів ієрархії	$S = n_0 + \sum_{i=1}^m n_i + n_p^{r-1}$
Ітераційний алгоритм		$M = A(B) E_i(I_{inv}(), (X, e))$ $I_2(), (X), (X_0) I_3(F, (X), (X))$ $w_4(X-X_0 < e) _2 A(E)$	$\{\Phi^{(1)}(N)\}^m \beta(\bar{e})$	m – кількість ітерацій за формулою Якобі $m = \frac{\ln\ \varepsilon\ \cdot (1 - \ \alpha\) \cdot \ \beta\ }{\ln\ \alpha\ } - 1$ $\ \alpha\ $ - норма вектора коефіцієнтів моделі N ; $\ \beta\ $ - норма вектора вільних членів; $\ \varepsilon\ $ - норма вектора допустимих похибок n – розмірність вектора X	$m \cdot n$

Продовження табл. 2.4

1	2	3	4	5	6
Розгалуження		$M = A(B) E_0(u_0) w_1(a_1) _{(n+1)}$ $w_2(a_2) _{(n+2)} \dots w_n(a_n) _{(n+n)}$ $M_{n+1}(u_1) M_{n+2}(u_2) \dots M_{n+n}(u_n)$ $A(E)$	$\{\beta(x_i), i=1 \dots n\} =$ $\overline{\Phi}(\max[n_i]) (u_i) [\overline{\Phi}^{(2)} (\{a_i\})$ $[\Phi^{(n_0)}(u_0) \beta(x_0)]]$	$S_i = n_i - \text{складність}$ $\text{алгоритму } u_i$	$S = S_0 +$ $2 \cdot n +$ $\sum_{i=1}^n S_i$
Алгоритмічна модель нечіткого оцінювання		$M_1 = A(B) E_1(1, (i: \text{int}))$ $E_2(Im, (X), (X'': \text{real}, \varepsilon: \text{real}))$ $E_3(C_F, (X''), (\mu: \text{function}))$ $E_4(i+1, (i), (i: \text{int}))$ $w_5(i \leq n, (i)) _2 E_6(1, (j: \text{int}))$ $E_7(x'_{i \min}, (x'_{i0}: \text{real})) E_8(I_S,$ $(x'_i = u_0, B), (L, f: \text{array}[n, m],$ $e: \text{real}, x'_i: \text{real}))$ $E_9(C_L, (\mu_x: \text{array}[n+1] \text{ of}$ $\text{function}, f: \text{array}[n, m], e: \text{real},$ $x'_i: \text{real}), (\mu_{x_i}: \text{real}))$ $E_{10}(x'_{i0} + dx'_i, (x'_i), (x'_i))$ $w_{11}(x'_i < x'_{i \max}, (x'_i)) _7$ $E_{12}(C_D, (\mu_{x_i}: \text{function}), (x'_i: \text{real}))$ $E_{13}(j+1, (j), (j: \text{int}))$ $w_{14}(j \leq N-n, (j)) _7 A(E)$	$\beta(X'') =$ $= \overline{\Phi}^{(2)}(+)$ $[\beta(\varepsilon_S), \overline{\Phi}^{(n_g)}(g) \overline{\Phi}^{(1)}(N) [\beta(X''_0)]]$ $\beta(X') =$ $\overline{\Phi}^{(m+1)}(+)$ $[\{ \beta(\mu'_j), j=1 \dots m \}]$ $\Phi^{(2^{m_b}/m)}(\max)$ $\beta(\mu'_j) = \Phi^{(2m)}(\min) \Phi^{(1)}(\mu'_j)$ $[\beta(\mu'_j)]]$ $\beta(\mu'_j) = \Phi^{(1)}(\text{norm}) [\beta(x''_j)]]$	$m - \text{кількість}$ термів нечіткої шкали $m_b - \text{кількість}$ правил бази даних $\mu_j - \text{функція}$ належності $\text{нечіткої змінної } x_j$	$S = (2 + n_g +$ $1) \cdot n$ $+ (m+1)$ $\cdot k + 1 \cdot n +$ $\frac{2m_b}{m+1} + 2$
Цикл з параметром		$M = A(B) E_1(C; (1); (i)) M_F$ $E_3(C; (i+1); (i); (i)) w_4(i \leq m) _2$ $A(E)$	$\overline{\Phi} = \left\{ \Phi^{(n_{M_F})} (F) \right\}$	$m - \text{розмірність}$ векторного оператора (кратність циклу) $n_{M_F} = \max[n_{M_F i}]$	$S =$ $4 +$ $+ \sum_{i=1}^m n_{M_F i}$

1	2	3	4	5	6
<p>Замкнений канал інтегруючого перетворення</p>		<p> $M = A(B) E_1(k x_a; (x_a); v_a)$ $E_2(v_a - u_a; (v_a, u_a); y_a)$ $E_3(I_{AD}; (y_a); y_d)$ $E_4(u_d + y_d dt; (y_d); u_d)$ $E_5(I_{DA}; (u_d); u_a) _2 A(E)$ </p>	$\left\{ \begin{aligned} \beta(u_d) &= \Phi^{(2)}(u_d = u_d + y_d \cdot dt) \\ &[\Phi^{(1)}(y_d = M_{AD}(y_a))] \\ &\Phi^{(2)}(y_a = v_a - u_a)[\beta(u_a), \\ &\Phi^{(1)}(v_a = kx_a)[\beta(x_a)]] \\ \beta(u_a) &= \Phi^{(1)}(u_a = M_{DA}(u_d))[\beta(u_d)] \end{aligned} \right.$	<p>τ – стала інтегрування</p>	<p>$S=1 + \frac{\tau}{6 dt}$</p>
<p>Алгоритмічна модель чіткого оцінювання</p>		<p> $M_1 = A(B) E_1(1, 0, (i: int))$ $E_2(Im, (X), (X'': real[n], \varepsilon_S : real[n])) E_3(i+1, (i), (i: int))$ $w_4(i <= n, (i)) _2$ $M_2(Iop, (M_{OK}, X''), (X': real[k], \sigma_X: real[n+k])) A(E)$ $M_2 = A(B) E_1(C; (0, h); (X', h)) E_2(C: M_{OK}(X'); (X'); (X''_0)) E_3(C: \sigma(X'' - X''_0); (X'', X''_0); (\sigma_0))$ $E_4(C: h \cdot grad(M_{OK}), (h), (dX'))$ $E_4(C: (X'_0 + dX'); (X'_0, dX'); (X'_0)) E_3(C: M_{OK}(X'); (X'); (X''_0)) E_3(C: \sigma(X'' - X''_0); (X'', X''_0); (\sigma)) w_4(\sigma - \sigma_0 > e) _4 A(E)$ </p>	$\begin{aligned} \beta(X'') &= \bar{\Phi}^{(2)}(+ \\ &[\beta(\varepsilon_S), \bar{\Phi}^{(n_g)}(g) \bar{\Phi}^{(1)}(N) [\beta(X''_0)]] \\ \beta(X') &= (\bar{\Phi}^{(2)}(X' + dX)) \\ &[\bar{\Phi}^{(n_{MOK})}(X' = M_{OK}(X')) \\ &[\beta(X')], \\ &\bar{\Phi}^{(4)}(h \cdot grad M_{OK}) \\ &[\beta(X')]]^m \\ \beta(\sigma) &= \Phi^{(3n+1)} \\ &\left(\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x''_i - x''_{i0})^2} \right) \\ &[\beta(X'')] \end{aligned}$	<p> N – рівняння перетворення сенсорів h – крок пошуку оптимуму M_2 – алгоритм пошукової оптимізації градієнтним методом m – кількість кроків градієнтного пошуку </p>	<p> $S = (2 + n_g + 1) \cdot n + (2 + n_{MOK} + 4) \cdot k + 3n + 1$ </p>

Для чисельного аналізу впливу окремих структурних параметрів ПЗД на її складність визначимо складність «базової системи», тобто системи, яка здійснює збір даних про стан технологічного процесу розмірності N без урахування просторового розташування точок контролю, впливу сторонніх збурень, а отже і необхідності додаткової обробки сигналів, складного керування з метою оптимальної дискретизації та стиснення, обмеженої кількості входів і виходів пристроїв. Узагальнена структура такої ПЗД наведена на рис. 2.9.

Блок обробки даних базової системи здійснює обчислення невідомих параметрів стану технологічного процесу X' на основі вимірних значень параметрів X'' і моделі технологічного процесу F . Вважатимемо, що у базовій моделі система рівнянь F є не виродженою і має єдиний розв'язок. Модель в узагальненій структурі подана у формі Якобі

$$X' = F(X'', X'), \quad (2.35)$$

де F – лінійна форма, яка використовується для знаходження невідомих параметрів ітераційним методом.

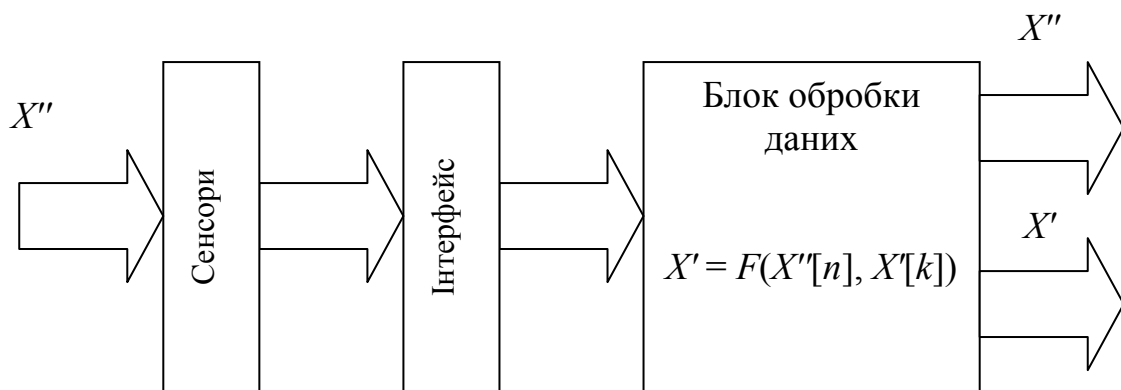


Рис. 2.9. Базова структура ПЗД

Відповідна базова алгоритмічна модель системи з урахуванням моделі ітераційного процесу, наведеної у таблиці 2.4, має вигляд:

$$M = A(B) I_1((0, (1), (i))) I_{m_2}(\tilde{x}''_i, \varepsilon / u''_i, \varepsilon) I_{p_3}(p_1/p_2, (u''_i), (x''_i)) I_4((i+1), (i), (i)) \\ w_5(i \leq n) _2 I_6((0, (1), (i))) I_{m_7}((0, 0.01), (x'_{i0}, e)) I_8((i+1), (i), (i)) w_9(i \leq m) _7 \\ I_{10}((0), (X'), (X'_0)) I_{11}(F, (X', X''), (X')) w_{12}(|X' - X'_0| < e) _10 A(E) \quad (2.36)$$

Складність алгоритмічної моделі визначається трьома циклічними блоками: (2-5) – циклічне вимірювання та перетворення у прийнятну для

обробки форму компонентів вектора параметрів X'' ; (7-9) – надання невідомим компонентам вектора X' і похибці ітераційного наближення e початкових значень; (11-13) – ітераційне розв’язування системи рівнянь моделі. Отже складність базової структури

$$\theta_S = [4 + (\theta_{Sim} + \theta_{Sip})n] + [4 + 2n] + [4 + (k + k(n + k - 1))m] = (\theta_{Sim} + \theta_{Sip} + 2 + km)n + (k^2m) + 12, \quad (2.37)$$

де θ_{Sim} – складність вимірювального перетворення одного каналу;

θ_{Sip} – складність програмно-апаратних засобів передавання результату вимірювань одним каналом;

n – кількість контрольованих параметрів (розмірність вектора X'');

k – кількість параметрів, які розраховуються (розмірність вектора X');

m – кількість ітерацій, яку можна оцінити за формулою Якобі [15]

$$m = \frac{\ln[\|\varepsilon\| \cdot (1 - \|\alpha\|) \cdot \|\beta\|]}{\ln\|\alpha\|} - 1, \quad (2.38)$$

$\|\alpha\|$ - норма (найчастіше середня квадратична) матриці коефіцієнтів моделі;

$\|\beta\|$ - норма вектора вільних членів;

$\|\varepsilon\|$ - норма вектора похибок.

Врахування реальних особливостей реалізації ПЗД ускладнює базову структуру.

Евристичні залежності складності ПЗД від параметрів її структури за умов не виродженої моделі технологічного процесу і наявності її однозначного розв’язку зображені на рис. 2.10. Загальна складність знайдена як сума програмної і апаратної складностей

$$\theta_S = \theta_{Snp} + \theta_{San}. \quad (2.39)$$

З графіків видно, що складність радіальної структури найменша, але вона може використовуватися лише для контролю $n \leq d$ параметрів.

Складність лінійної структури вища за радіальну через складність програмного забезпечення і необхідність спеціалізованих адаптерів з дешифруванням адрес і протоколів усунення колізій.

Ієрархічна структура характеризується найменшою швидкістю зростання складності, що найяскравіше проявляється при великих n . Графік розривний,

оскільки при вичерпанні можливостей мультиплексорів наявних рівнів подальше розширення системи потребує додавання нового рівня ієрархії, що стрибком збільшує складність, але зменшує швидкість її зростання. Кількість рівнів ієрархії

$$r = \text{ent}[\log_d n] + 1, \quad (2.40)$$

де $\text{ent}[*]$ - операція знаходження цілої частини,

d – коефіцієнт мультиплексування.

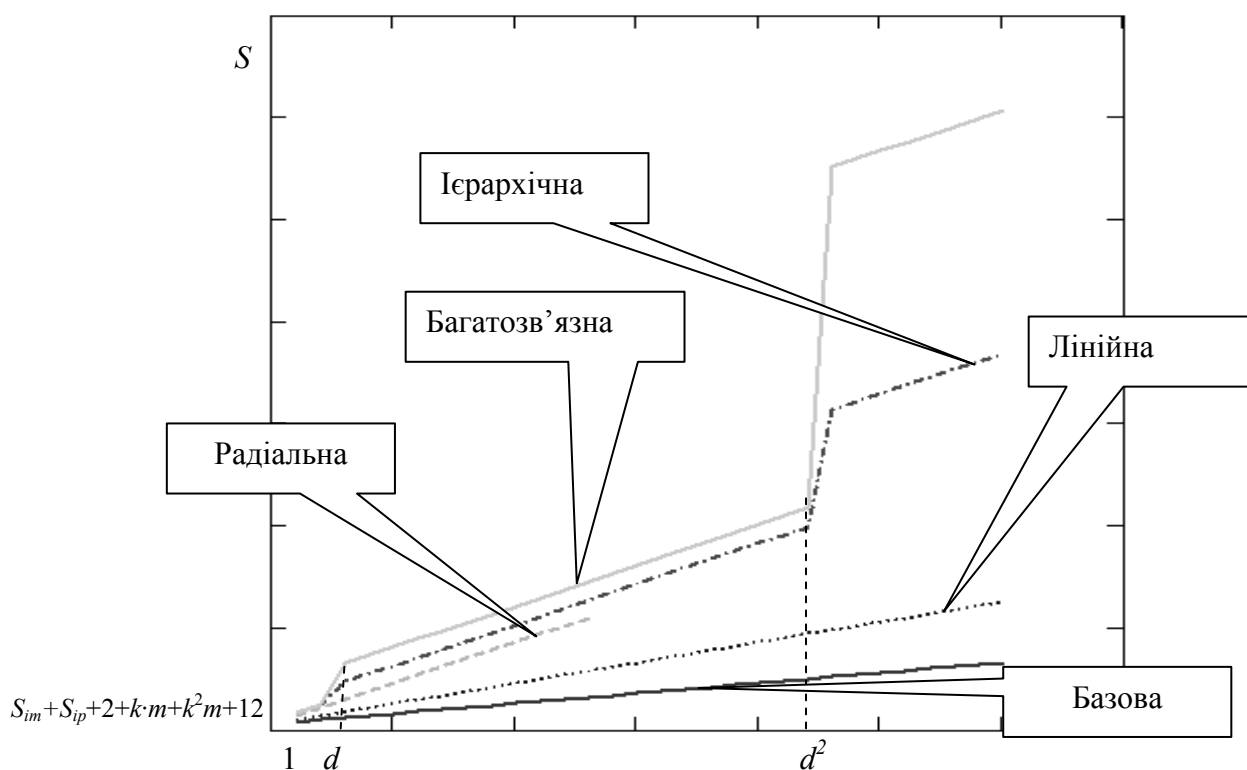


Рис. 2.10. Вплив кількості контрольованих параметрів на складність ПЗД

d – коефіцієнт мультиплексування;

k_n – коефіцієнт надлишковості багатозв'язної структури

Залежність (2.40) є обвідною відповідного графіка на рис. 2.7.

Графік багатозв'язної системи подібний до графіка ієрархічної, проте складність його вища через наявність надлишкових зв'язків.

Порівняємо результати оцінювання складності для деяких типових випадків за запропонованою методикою з відповідними оцінками за іншими (відомими) методиками. Результати порівняння наведені у табл.2.5. З наведених у таблиці результатів видно, що для простих перетворень і структур усі оцінки близькі з точністю до константи.

Оцінки складності за різними методиками*

Тип системи	За функцією Шеннона	За кількістю і степенем вузлів графа (метод Квайна)	Обчислювальна складність за кількістю мікрооперацій і переходів	Складність обладнання за кількістю елементів і зв'язків	Сумарна довжина шляхів у графі	За запропонованою методикою
Нелінійне перетворення визначеного даного	-	1	x^{**}	1	1	0
Нелінійне перетворення невизначеного даного	-	1	x^{**}	1	1	1
Просте лінійне бінарне перетворення (операція додавання визначених даних)	2	2	1	1	1	1
Операція додавання невизначених даних	2	2	1	1	1	2
Послідовне з'єднання	n_1+n_2	n_1+n_2+3	-	n_1+n_2+1	n_1+n_2+2	n_1+n_2
Паралельне з'єднання	n_1+n_2+1	n_1+n_2+3	-	n_1+n_2	n_1+n_2	n_1+n_2
Замкнена структура	n_1+n_2+2	n_1+n_2+4	-	n_1+n_2+4	$(n_1+n_2+2) \cdot m$	$(n_1+n_2+2) \cdot m$
Лінійна ПЗД	$n_1+n_2+n_0$	$n_1+n_2+n_0+3$	-	$n_1+n_2+n_0+2$	$n_1+n_2+n_0+2$	$n_1+n_2+n_0$
Радіальна ПЗД	n_0+m+ $\sum_{i=1}^m n_i$	n_0+m+ $\sum_{i=1}^m n_i$	-	n_0+m+ $\sum_{i=1}^m n_i$	n_0+m+ $\sum_{i=1}^m n_i$	n_0+ $\sum_{i=1}^m n_i$
Ієрархічна ПЗД	n_0+m+ $\sum_{i=1}^m n_i +$ n_p^{r-1}	n_0+2m+ $\sum_{i=1}^m n_i$	-	n_0+2m+ $\sum_{i=1}^m n_i$	n_0+ $\sum_{i=1}^m n_i +$ n_p^{r-1}	n_0+ $\sum_{i=1}^m n_i +$ n_p^{r-1}
Ітераційний алгоритм	$n+1$	$n+1$	$n+1$	-	$m \cdot n$	$m \cdot n$

Примітка: * розшифрування формул і зміст позначень відповідає таблиці 2.4.

** залежить від виду перетворення.

Оцінювання за функцією Шеннона, за кількістю мікрооперацій і за кількістю елементів не завжди може бути застосоване. Усі методики оцінювання, крім запропонованої, інваріантні до ступеня визначеності вхідних даних, що не відповідає особливостям умов оцінювання складності ПЗД.

2.4.2. Оцінювання невизначеності результатів збору даних. Ступінь невизначеності результатів роботи ПЗД - це одна з головних характеристик якості системи. Невизначеність залежить від багатьох факторів:

- структурних, алгоритмічних і параметричних характеристик ПЗД;
- інтенсивності завад;
- співвідношення динамічних характеристик сигналів, завад і перетворювачів ПЗД;
- нестабільності умов функціонування;
- надійності апаратно-програмного забезпечення.

Невизначеність результатів збору даних про стан технологічного процесу характеризується розкидом σ (СКВ) узагальнюючої функції невизначеності

$$\sigma = \int_X \sum_{i=1}^k (x_i - m_{xi})^2 \beta(\vec{x}) d\vec{x}, \quad (2.41)$$

де m_x - перший початковий момент невизначеного вектора результатів x ;

X – область зміни результатів,

k – розмірність вектора результатів,

або узагальненою ентропією

$$H = \sum_{i=1}^k \int_{X'} \beta(x_i) \ln \beta(x_i) dx_i, \quad (2.42)$$

де $X'[k]$ – параметри стану ПЗД, які визначають її ефективність.

Звичайно для зменшення невизначеності результатів застосовуються різні структурні, алгоритмічні і технологічні методи. Усі ці методи можна розділити на дві групи:

- методи, що використовують апріорну інформацію про дані і фактори, які впливають на невизначеність;

системи, при $k \neq n$ результат отримується методом оцінювання.

Оцінимо невизначеність остаточного результату за умови незалежності і нормальності даних. За цих умов відповідно до (2.42) загальна ентропія

$$H = \sum_{i=1}^N \ln(\sigma_i \sqrt{2\pi e}), \quad (2.52)$$

де $N = n + k$ - загальна розмірність вектора даних.

СКВ компонентів вектора $X''[n]$ визначаються похибкою їх вимірювання. СКВ компонентів вектора $X'[k=N-n]$ при $k=n$ отримаємо з розв'язку системи (2.51), який запишемо у вигляді

$$x_i = \frac{\Lambda_i}{\Lambda}, \quad i=n+1 \dots 2n, \quad (2.53)$$

де $\Lambda = |\alpha_{ij}|$;

$$\Lambda_i = \sum_{l=1}^n x_{0l} \mu_{il};$$

μ_{ij} - мінори матриці коефіцієнтів системи (2.51).

Тоді СКВ компонентів вектора X'

$$\sigma_i = \sqrt{\sum_{l=1}^n \left(\sigma_l \frac{\mu_{il}}{\Lambda} \right)^2}, \quad i=n+1 \dots 2n, \quad (2.54)$$

Таким чином, отримуємо вираз ентропії

$$H = \frac{N}{2} \ln \pi + \sum_{i=1}^n \ln \sigma_i + \sum_{i=n+1}^{2n} \ln \sqrt{2\pi e \sum_{l=1}^n \left(\sigma_l \frac{\mu_{il}}{\Lambda} \right)^2}. \quad (2.55)$$

При $k < n$, тобто $n > N/2$, система (2.51) міститиме k рівнянь, для отримання яких використовуються контрольовані параметри $\{x_1, \dots, x_k\}$. Решта контрольованих параметрів $\{x_{k+1}, \dots, x_n\}$ використовується для покращення результатів за допомогою алгоритму оцінювання. Алгоритм оцінювання мінімізує сумарну квадратичну нев'язку

$$\sigma_{\Sigma} = \sqrt{\sum_{i=k+1}^n \left[x_{i0} - \sum_{j=1}^k \alpha_{ij} x_{n+j} \right]^2}. \quad (2.56)$$

Для лінійної системи (2.44) нев'язка (2.56) може бути зменшена до нуля. Таким чином, загальна ентропія результату буде

$$H = \frac{N}{2} \ln \pi + \sum_{i=1}^n \ln \sigma_i + \sum_{i=n+1}^{2n} \ln \sqrt{2\pi e \sum_{l=1}^n \left(\sigma_l \frac{\mu_{il}}{\Lambda} \right)^2} - \ln \sqrt{2\pi e \sum_{i=N-n+1}^n \left[x_{i0} - \sum_{j=1}^{N-n} \alpha_{ij} x_{n+j} \right]^2}. \quad (2.57)$$

При $k > n$, тобто $n < N/2$, система (2.51) міститиме n рівнянь, для отримання яких використовуються усі контрольовані параметри. У цій системі кількість невідомих перевищуватиме кількість рівнянь. Для знаходження результатів алгоритм оцінювання мінімізує сумарну нев'язку

$$\sigma_{\Sigma} = \sqrt{\sum_{i=1}^n \left[x_{i0} - \sum_{j=1}^k \alpha_{ij} x_{n+j} \right]^2}. \quad (2.58)$$

Загальна ентропія результату буде

$$H = \frac{N}{2} \ln \pi + \sum_{i=1}^n \ln \sigma_i + \sum_{i=n+1}^{N-n} \ln \Omega_i - \ln \sqrt{2\pi e \sum_{i=1}^n \left[x_{i0} - \sum_{j=1}^{N-n} \alpha_{ij} x_{n+j} \right]^2}, \quad (2.59)$$

де $\{\Omega_i\}$ - інтервали початкової невизначеності параметрів.

Евристичні графіки залежності ентропії результату від кількості контрольованих параметрів зображено на рис. 2.11.

При кількості контрольованих параметрів $n = N/2$ на графіках спостерігається розрив.

Ентропія у послідовній структурі залежить також від співвідношення між інтервалом кореляції стану технологічного процесу τ і швидкодією підсистеми T . Очевидно, для послідовної структури

$$T = n \cdot t_0, \quad (2.60)$$

де t_0 – час перетворення одного вхідного параметра.

Тоді дисперсія стану технологічного процесу після закінчення усього циклу збору даних дорівнюватиме

$$D = D_0 [1 - r^2(nt_0)], \quad (2.61)$$

де D_0 – початкова (апріорна) дисперсія;

$r(t)$ – нормована кореляційна функція стану технологічного процесу.

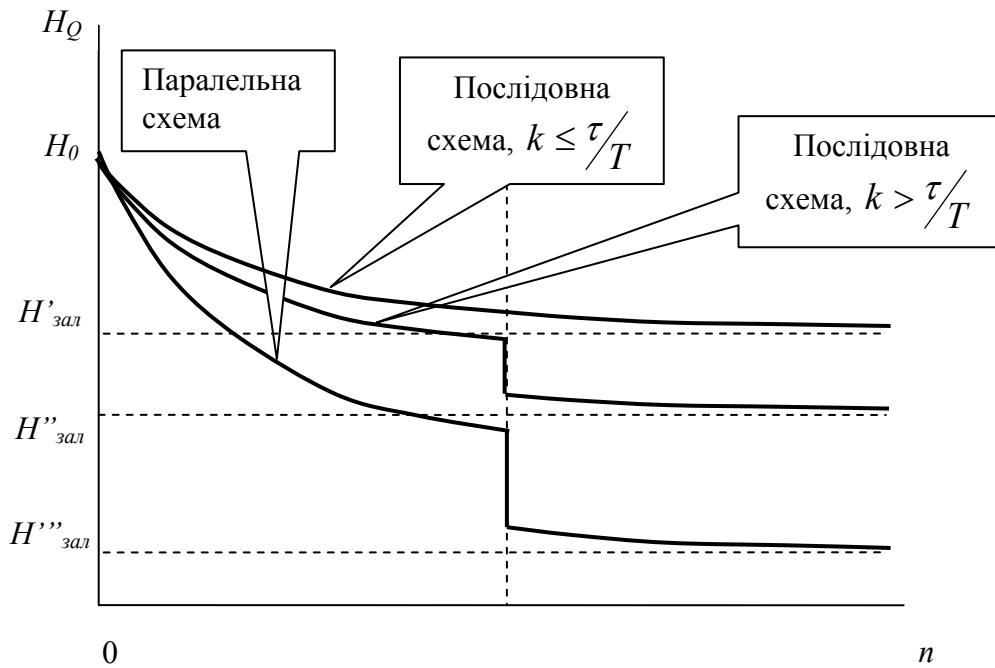


Рис. 2.11. Графіки залежності ентропії результату від кількості контрольованих параметрів

При $nt_0 > \tau$ отримуємо $D = D_0$, отже, застарілі вимірювання не дають додаткової інформації про стан технологічного процесу. Тобто, у цьому випадку фактична кількість контрольованих параметрів $n' = \frac{\tau}{t_0} < n$.

2.4.3. Оцінювання швидкодії. Під швидкодією ПЗД будемо розуміти характеристику, обернену до проміжку часу T від надходження на вхід керування ПЗД команди щодо визначення стану технологічного процесу до фактичного отримання вектора параметрів стану. Основні складові цього проміжку:

- час отримання значень контрольованих параметрів;
- час передавання цих значень до блоків обробки даних;
- час визначення даних, які задаються нечітко;
- час розрахунку даних, які визначаються опосередковано;
- час розрахунку даних, які визначаються шляхом оцінювання на основі моделі технологічного процесу.

В основі оцінювання швидкодії лежать аксіоми часової складності, наведені у табл. 1.8.

Загальний час T_{Σ} визначається за графом алгоритмічної моделі, вага ребер якого відповідає часу виконання кожної операції

$$T_{\Sigma} = L_{\max}, \quad (2.62)$$

де L_{\max} – найдовший шлях у графі від початкової вершини (отримання команди) до кінцевої (остаточне отримання вектора результатів) з урахуванням кратності циклічних структур.

Найскладнішою задачею автоматизованого оцінювання швидкодії ПЗД є визначення кратності таких циклічних структур, в яких вона залежить від значень даних. Для цього систематизуємо структури та їх ознаки у табл. 2.6. В цій таблиці у графі «Методика розрахунку швидкодії» подані розрахункові формули і методики оцінювання часу виконання перетворень для більшості типових структур.

Проаналізуємо зв'язок часової складності з складністю, оціненою за порядком операторної моделі відповідно до п. 2.4.1. Основні відмінності цих мір складності:

- при паралельному з'єднанні підсистем загальна часова складність визначається найбільшою складністю серед паралельних підсистем (аксіома 2 табл. 1.8), а у запропонованій мірі визначається сумарною складністю;

- послідовні однотипні перетворення (послідовні нелінійні перетворення або послідовні лінійні інтегро-диференціальні перетворення) можуть бути агреговані у одне перетворення. Часова складність агрегованого перетворення менше сумарної складності (аксіома 3 табл. 1.8), але у загальному випадку її визначення потребує аналізу змісту перетворень. Запропонована міра складності, виходячи з виду операторів (1.11), для нелінійних агрегованих перетворень дорівнює 1, для лінійних динамічних (інтегро-диференціальних) перетворень дорівнює сумі складностей. Запропонована міра більше відповідає потребам оцінювання швидкодії ПЗД, оскільки в них, на відміну від обчислювальних систем, швидкодія визначається переважно інерційними перетвореннями параметрів фізичних процесів у реальному масштабі часу, у порівнянні з якими витрати часу на більшу або меншу кількість нелінійних обчислювальних операцій впливають несуттєво;

- часова міра складності не залежить від визначеності чи невизначеності

даних, які перетворюються (обробляються) системою. Це характерно для обчислювальних систем, але нехарактерно для ПЗД. Запропонована міра, як впливає з табл.2.6, пропорційно залежить від кількості невизначених даних.

Враховуючи відзначені відмінності, запропонуємо методику оцінювання швидкодії ПЗД на основі алгоритмічної моделі з урахуванням її складності:

1. Алгоритмічна модель перетворюється на операторну форму.
2. Послідовні однотипні перетворення агрегуються.
3. Визначаються параметри складності: складність програмної і апаратної частин, складність послідовних і паралельних частин системи.
4. Визначається одиниця вимірювання швидкодії $\Delta t = \min_N \tau_i$, де τ_i - інтервал кореляції i -го вхідного процесу ПЗД.
5. Оцінюється швидкодія ПЗД з використанням методик оцінювання типових структур, наведених у табл. 2.6, і покрокової композиції типових структур у агрегати.

Отримана за такою методикою оцінка швидкодії нелінійно залежить від складності і часу виконання окремих операцій.

2.4.4. Оцінювання надійності. Методи оцінювання надійності комп'ютеризованих систем ґрунтовно розглянуті у літературі [26, 39, 34, 37, 57, 68, 109, 110, 131, 141, 151, 156, 174, 175, 176]. Зокрема виділяють структурну і функціональну, метрологічну і неметрологічну надійності.

Враховуючи функціональне призначення підсистем збору даних, можна зробити висновок, що узагальненою характеристикою надійності ПЗД є інформаційна надійність. Поняття інформаційної надійності відоме, але у нього переважно вкладають зміст метрологічної надійності [157]. Разом з тим ПЗД здійснює збір не тільки вимірюваної інформації, але й результатів спрацьовування сигналізаторів, даних, що надходять від операторів, тощо. Але головна особливість полягає у багатовимірності системи і наявності зв'язку між даними. За певних умов відмова деяких елементів ПЗД (як структурна, так і метрологічна) не приводить до втрати працездатності підсистеми, а лише до зменшення її ефективності. Такими умовами є:

Таблиця 2.6

Характеристики типових структур моделі ПЗД

Функціональне призначення		Тип структури	Ознаки структури при аналізі моделі	Характерні параметри	Методика розрахунку швидкодії
Програмна реалізація	Апаратна реалізація				
1	2	3	4	5	6
Просте функціональне перетворення	Перетворення роду сигналу і форми інформації	Послідовна	1. Відсутні переходи	l – кількість операцій моделі; t_i – час спрацьовування пристрою	$T = \sum_{i=1}^l t_i$
	Управління каналами	Послідовно-паралельна	1. Структура складається з кількох паралельних підмоделей, після яких йде цикл з параметром; 2. У циклі на кожному кроці виконуються операції з одним з виходів паралельної підструктури; 3. Кількість повторів циклу дорівнює кількості паралельних підструктур	$\{M_{\max}\}$ – підмножина операцій паралельних підструктур, які утворюють шлях максимальної довжини; $\{E_{ij}\}$ – множина операцій циклічної частини; t_i – час роботи підструктури (частини підсистеми); m_{ij} – кількість повторів циклу (фіксоване значення, з яким порівнюється змінна в умові)	1. Знайти шлях максимальної довжини серед паралельних підструктур; 2. Підрахувати час паралельного перетворення $T_n = \sum_{\forall M_i \in \{M_{\max}\}} t_i$ 3. Підрахувати час роботи циклічної підструктури $T_{ij} = m_{ij} \cdot \sum_{\{E_{ij}\}} t_j$ 4. Розрахувати загальний час роботи структури $T = T_n + T_{ij}$
	Перетворення даних у паралельних каналах	Паралельна	Відсутність залежності між даними у паралельних підмоделях	$\{E_{\max}\}$ – підмножина операцій, які утворюють шлях максимальної довжини; t_i – час спрацьовування пристрою	1. Знайти шлях максимальної довжини; 2. $T = \sum_{\forall i \in \{E_{\max}\}} t_i$
Перетворення за функцією, заданою на інтервалах		Послідовна з розгалуженнями	1. Наявні переходи вперед		
Логічний висновок (перетворення)					

1	2	3	4	5	6
Інтегрування, статистична обробка, фільтрація	Опитування сенсорів	Циклічна (замкнена) з параметром	<ol style="list-style-type: none"> 1. Наявний перехід назад; 2. Умовою переходу є перевірка перевищення змінною фіксованого значення; 3. Між переходом і точкою повернення ця змінна збільшується на 1. 	l – кількість операцій структури; $\{A_i\}$ – множина підструктур; T_i – час роботи підструктури (частини підсистеми); m_{Π} – кількість повторів циклу (фіксоване значення, з яким порівнюється змінна в умові)	Рекурсивна процедура: <ol style="list-style-type: none"> 1. Виділити усі циклічні підструктури; 2. У кожній циклічній підструктурі підрахувати час роботи всіх внутрішніх підструктур; 3. Підрахувати час роботи циклічної підструктури $T_i = m_{\Pi i} \cdot \sum_{\forall j \in \{A_i\}} T_j$ 4. Розрахувати загальний час роботи структури
Розв'язання нелінійних рівнянь і систем рівнянь	Передача даних в умовах завад	Циклічна (замкнена) з умовою	<ol style="list-style-type: none"> 1. Наявний перехід назад; 2. Умовою переходу є перевірка зменшення змінної нижче додатного граничного значення 	l – кількість операцій структури; $\{A_i\}$ – множина підструктур; T_i – час роботи підструктури (частини підсистеми); ε – похибка наближення; ε_0 – початкова похибка наближення; $\varepsilon_{\text{дон}}$ – гранична похибка наближення; m_{Π} – кількість повторів циклу	Рекурсивна процедура: <ol style="list-style-type: none"> 1. Виділити усі циклічні підструктури; 2. У кожній циклічній підструктурі підрахувати час роботи усіх внутрішніх підструктур; 3. Оцінити кількість повторів циклу $m_{\Pi} = \log \frac{\varepsilon_0}{\varepsilon_{\text{дон}}}$ 3. Підрахувати час роботи циклічної підструктури $T_i = m_{\Pi i} \cdot \sum_{\forall j \in \{A_i\}} T_j$ 4. Розрахувати загальний час роботи структури
Пошукова оптимізація	Складні вимірювальні перетворення*				

* Примітка: Маються на увазі вимірювання із замкненою схемою: компенсаційні, окремі види тестових, комутаційно-модуляційних, адаптивних, інваріантних тощо.

- використання алгоритму оптимального оцінювання стану технологічного процесу;

- відмови лише каналів паралельного визначення окремих даних.

За цих умов будемо характеризувати інформаційну надійність ймовірністю того, що невизначеність стану технологічного процесу не перевищує задану границю $P(H \leq H_{cp})$.

Оцінимо залежність інформаційної надійності від складності ПЗД. Для цього запишемо рівняння (2.55), (2.57), (2.59) у системі з рівнянням (2.37) і аналогічними рівняннями для випадків $n \neq N/2$ з урахуванням складності моделі чіткого оцінювання (табл. 2.4) $\theta_S = (2+n_g+1) \cdot n + (2+n_{\text{Мок}}+4) \cdot k + 3n + 1$.

При $n < N/2$

$$\left\{ \begin{array}{l} H = \frac{N}{2} \ln \pi + \sum_{i=1}^n \ln \sigma_i + \sum_{i=n+1}^{N-n} \ln \Omega_i - \ln \sqrt{2\pi e \sum_{i=1}^n \left[x_{i0} - \sum_{j=1}^{N-n} \alpha_{ij} x_{n+j} \right]^2}, \\ \theta_S = (\theta_{S_{im}} + \theta_{S_{ip}} + 2 + k \cdot m) \cdot n + [(2+n_g+1) \cdot n + (2+n_{\text{Мок}}+4) \cdot k + 3n + 1] + 12, \\ k = N - n. \end{array} \right. \quad (2.63)$$

При $n = N/2$

$$\left\{ \begin{array}{l} H = \frac{N}{2} \ln \pi + \sum_{i=1}^n \ln \sigma_i + \sum_{i=n+1}^{2n} \ln \sqrt{2\pi e \sum_{l=1}^n \left(\sigma_l \frac{\mu_{il}}{\Lambda} \right)^2}, \\ \theta_S = (\theta_{S_{im}} + \theta_{S_{ip}} + 2 + k \cdot m) \cdot n + (k^2 m) + 12, \\ k = N - n. \end{array} \right. \quad (2.64)$$

При $n > N/2$

$$\left\{ \begin{array}{l} H = \frac{N}{2} \ln \pi + \sum_{i=1}^n \ln \sigma_i + \sum_{i=n+1}^{2n} \ln \sqrt{2\pi e \sum_{l=1}^n \left(\sigma_l \frac{\mu_{il}}{\Lambda} \right)^2} - \ln \sqrt{2\pi e \sum_{i=N-n+1}^n \left[x_{i0} - \sum_{j=1}^{N-n} \alpha_{ij} x_{n+j} \right]^2}, \\ \theta_S = (\theta_{S_{im}} + \theta_{S_{ip}} + 2 + k \cdot m) \cdot n + [(2+n_g+1) \cdot n + (2+n_{\text{Мок}}+4) \cdot k + 3n + 1] + 12, \\ k = N - n. \end{array} \right. \quad (2.65)$$

Кількість контрольованих даних залежить від надійності каналів їх отримання. Якщо ймовірність відмови одного каналу \tilde{P}_1 , то ймовірність

відмови \tilde{n} каналів з n_0 повної кількості

$$\tilde{P}_n = C_{n_0}^{\tilde{n}} \tilde{P}_1^{\tilde{n}}, \quad (2.66)$$

де $C_{n_0}^{\tilde{n}}$ - кількість сполучень з n_0 по \tilde{n} .

Тоді ймовірність працездатності більше, ніж n_{zp} каналів

$$P(n > n_{zp}) = n_0 \sum_{i=0}^{n_0 - n_{zp}} \left(1 - C_{n_0}^i \tilde{P}_1^i \right). \quad (2.67)$$

Відповідно, розглядаючи систему рівнянь (2.63) – (2.65) як перетворення $n \xrightarrow{F_1} S, \quad n \xrightarrow{F_2} H$, знаходимо

$$P(H \leq H_{zp})(S) = \left[F_2 \left(F_1^{-1}(S) \right) \right]^{-1} P(n > n_{zp}). \quad (2.68)$$

Евристична залежність інформаційної надійності від складності ПЗД зображена на рис. 2.12.

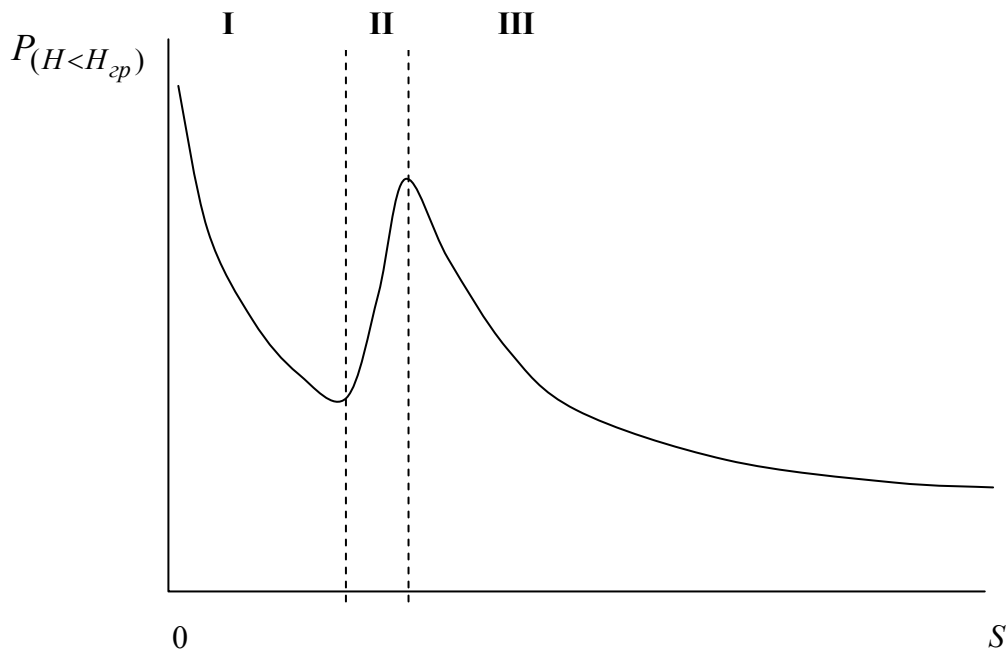


Рис. 2.12. Графік залежності надійності від складності ПЗД

Нелінійний характер залежності зумовлений декількома процесами. На інтервалі I збільшення складності відбувається за рахунок збільшення кількості каналів визначення параметрів технологічного процесу. Оскільки

для розв'язання задачі оцінювання необхідна певна мінімальна кількість даних, то при невеликій кількості каналів вимірювання оптимальне оцінювання недоцільне. Отже, збільшення складності призводить до зменшення надійності.

При подальшому збільшенні кількості вимірювальних каналів (інтервал II) починається застосування оптимального оцінювання. Це приводить до ускладнення програмного забезпечення, але ймовірність відмов програмного забезпечення значно менша за ймовірність апаратних відмов, отже, на цьому інтервалі збільшення складності приводить до збільшення надійності.

Подальше ускладнення системи (інтервал III) знову здійснюється за рахунок збільшення кількості апаратних засобів, що зменшує надійність ПЗД.

2.4.5. Комплексна оцінка ефективності ПЗД. Узагальнимо окремі характеристики ПЗД для отримання комплексного критерію ефективності, придатного для використання в процесі оптимізації структури підсистеми. Оскільки ПЗД відносять до інформаційних систем [73], то за основу узагальнення візьмемо інформаційно-вартісний показник [77]

$$Q = \frac{I_{\text{ц}}}{C(\theta_S, P_H, T_{\text{ц}}, H_{\text{зал}})}, \quad (2.69)$$

де $I_{\text{ц}}$ – кількість інформації про технологічний процес за один цикл;

$C(\theta_S, P_H, T_{\text{ц}}, H_{\text{зал}})$ - вартісна функція, яка залежить від параметрів:

$T_{\text{ц}}$ – тривалість циклу збору даних (швидкодія підсистеми);

$H_{\text{зал}}$ – залишкова ентропія (невизначеність стану технологічного процесу);

θ_S – складність;

P_H – інформаційна надійність.

Кількість інформації за означенням

$$I = H_0 - H_{\text{зал}}, \quad (2.70)$$

де H_0 – апіорна ентропія стану технологічного процесу.

Методики оцінювання залишкової невизначеності і швидкодії в

залежності від характеристик структури ПЗД розглянуті вище. Зупинимось на залежності вартості підсистеми від її структури.

Вартісна функція складається з трьох компонент:

- вартість розробки ПЗД;
- вартість виробництва ПЗД;
- вартість експлуатації ПЗД.

Величина складових вартості залежить не лише від технічних параметрів ПЗД і організаційних особливостей процесів її розробки, виробництва та експлуатації. Важливим фактором визначення вартості є внутрішній та світовий рівні цін на комплектуючі елементи і робочу силу, а також різні додаткові витрати як виробничого, так і невиробничого характеру. Тому більш зручною є відносна оцінка ефективності у порівнянні з деякою базовою ПЗД

$$q = \frac{Q}{Q_0} = \frac{I_u / I_0}{C / C_0}. \quad (2.71)$$

За базову ПЗД доцільно взяти достатньо просту систему масового застосування з відомими технічними даними і ціновими характеристиками. У подальшому такою базовою ПЗД обрана система охоронної сигналізації, яка відповідає всім зазначеним вимогам.

Залежність зазначених складових вартості від складності, швидкодії, надійності і невизначеності результату ПЗД не має формального аналітичного виразу. У практиці проектування ПЗД оцінювання складових вартості здійснюється на основі досвіду експертів. Узагальнимо такий досвід на основі нечіткої бази знань (експертні оцінки надані провідними фахівцями з розробки і впровадження АСУТП). Структурно-логічна схема нечіткого оцінювання вартісної функції зображена на рис. 2.13. Характеристики термів нечітких факторів вартості ПЗД наведені у таблиці 2.7, а мінімальний набір правил бази знань, який відповідає вимогам коректності бази, але потребує доповнення в процесі її використання, наведений у табл. 2.8-2.12.

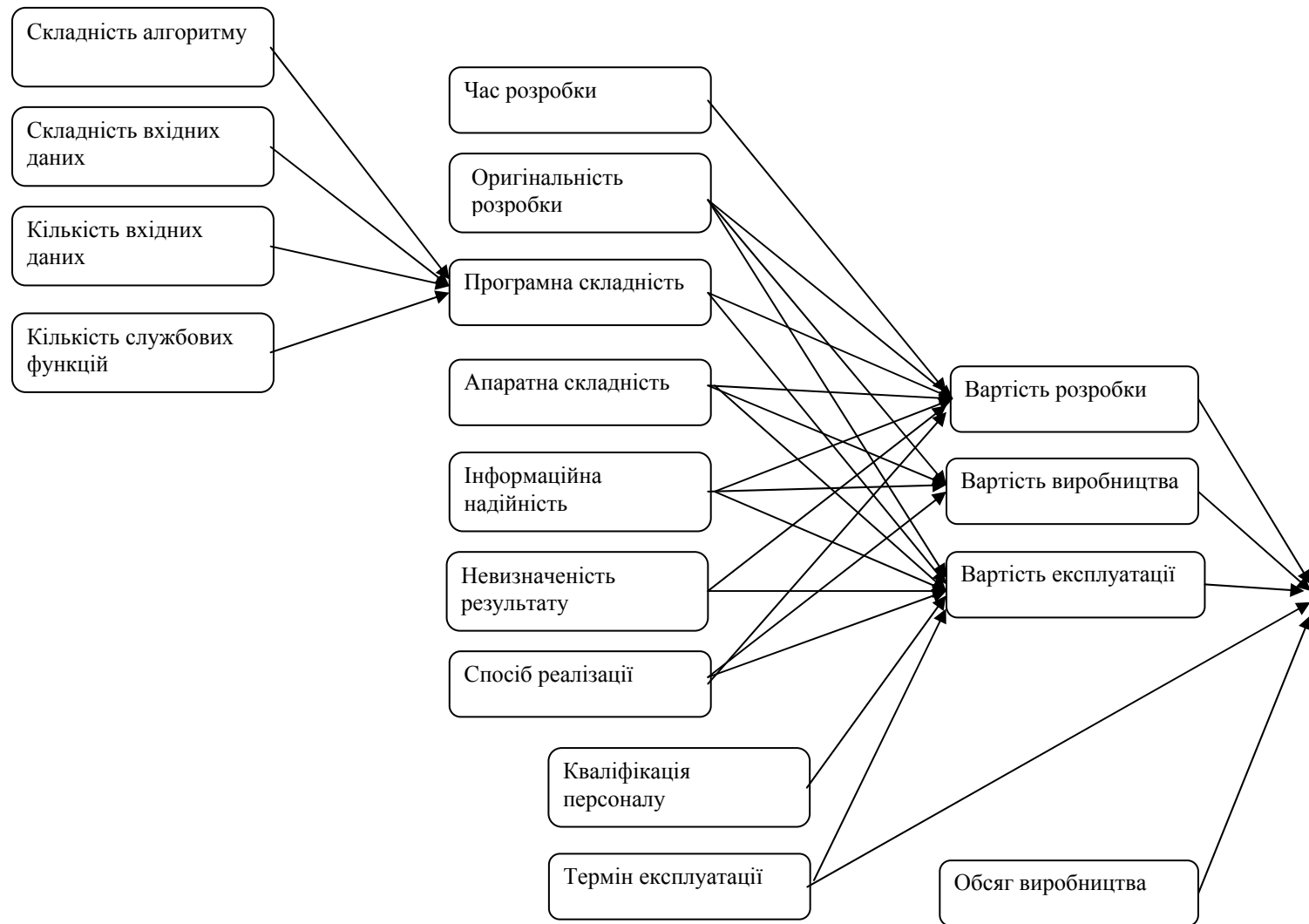


Рис. 2.13. Структурно-логічна схема оцінювання вартісної функції

Характеристики термів нечітких факторів вартості

Фактор	Тип	Множина значень (термів)	Максимальний розкид експертних оцінок
Вартість розробки відносно базової ПЗД	Числовий	$0 - 10^4$	10%
Вартість виробництва відносно базової ПЗД	Числовий	$0 - 10^4$	12%
Вартість експлуатації відносно базової ПЗД	Числовий	$0 - 10^4$	20%
Обсяг виробництва	Лінгвістичний	Одиничне, дрібносерійне, крупносерійне	0,5 терма
Час розробки	Лінгвістичний	Короткий, достатній, завеликий	1,5 терма
Оригінальність розробки	Лінгвістичний	Повністю оригінальна, частково оригінальна, адаптація відомої	1 терм
Програмна складність	Лінгвістичний	Мала, нижче середньої, середня, вище середньої, велика	1,5 терма
Апаратна складність	Числовий	$0-10^3$	5%
Інформаційна надійність	Числовий	0-1	10%
Невизначеність результату ($H_{зал}/H_{апр}$)	Числовий	0-1	10%
Спосіб апаратної реалізації	Лінгвістичний	Аналогова, дискретна, цифрова	1 терм
Кваліфікація персоналу	Лінгвістичний	Низька, середня, висока	1 терм
Термін експлуатації	Лінгвістичний	Короткий (до 1 року), середній (до 5 років), довгий (понад 5 років)	0,5 терма
Складність алгоритму	Лінгвістичний	Мала, середня, велика	1 терм
Складність вхідних даних	Лінгвістичний	Мала, середня, велика	1 терм
Кількість вхідних даних	Числовий	$1-10^3$	1
Кількість службових функцій	Числовий	$1-10^2$	1

Таблиця 2.8

Нечіткі правила оцінювання програмної складності

Складність алгоритму	Складність вхідних даних	Кількість вхідних даних	Кількість службових функцій	Програмна складність
Мала	Мала	1	1	Мала
Середня	Мала	20	1	Нижче середньої
Мала	Середня	20	1	Нижче середньої
Середня	Середня	10 ²	20	Середня
Середня	Велика	300	20	Вище середньої
Велика	Середня	300	20	Велика
Велика	Велика	>10 ³	10 ²	Велика

Таблиця 2.9

Нечіткі правила оцінювання вартості розробки

Час розробки	Оригінальність розробки	Програмна складність	Апаратна складність	Інформаційна надійність	Невизначеність результату	Спосіб апаратної реалізації	Вартість розробки
Завеликий	Адаптація відомої	Мала	0	0.5	0.9	Аналогова	10 ⁻¹
Завеликий	Адаптація відомої	Мала	1	0.75	0.5	Дискретна	1
Достатній	Частково оригінальна	Середня	10	0.9	0.2	Цифрова	10
Достатній	Частково оригінальна	Середня	10	0.95	0.1	Цифрова	10 ²
Короткий	Повністю оригінальна	Вище середньої	10 ²	0.98	0.05	Цифрова	10 ³
Короткий	Повністю оригінальна	Велика	10 ³	0.999	0.01	Цифрова	10 ⁴

Таблиця 2.10

Нечіткі правила оцінювання вартості виробництва

Оригінальність розробки	Апаратна складність	Інформаційна надійність	Спосіб апаратної реалізації	Вартість виробництва
Адаптація відомої	0	0.5	Аналогова	10 ⁻¹
Адаптація відомої	1	0.75	Дискретна	1
Частково оригінальна	10	0.9	Цифрова	10
Частково оригінальна	10	0.95	Цифрова	10 ²
Повністю оригінальна	10 ²	0.98	Цифрова	10 ³
Повністю оригінальна	10 ³	0.999	Цифрова	10 ⁴

Таблиця 2.11

Нечіткі правила оцінювання вартості експлуатації

Оригінальність розробки	Програмна складність	Апаратна складність	Інформаційна надійність	Невизначеність результату	Спосіб апаратної реалізації	Кваліфікація персоналу	Термін експлуатації	Вартість експлуатації
Адаптація відомої	Мала	0	0.5	0.9	Аналогова	Висока	Довгий	10^3
Адаптація відомої	Мала	1	0.75	0.5	Дискретна	Низька	Короткий	1
Частково оригінальна	Середня	10	0.9	0.2	Цифрова	Висока	Середній	10
Частково оригінальна	Середня	10	0.95	0.1	Цифрова	Середня	Середній	10^2
Повністю оригінальна	Вище середньої	10^2	0.98	0.05	Цифрова	Середня	Довгий	10^3
Повністю оригінальна	Велика	10^3	0.999	0.01	Цифрова	Низька	Довгий	10^4

Таблиця 2.12

Нечіткі правила оцінювання вартісної функції

Вартість розробки	Вартість виробництва	Вартість експлуатації	Термін експлуатації	Обсяг виробництва	Вартісна функція
10^{-1}	10^{-1}	10^{-1}	Довгий	Крупносерійне	10^{-1}
1	1	1	Короткий	Крупносерійне	1
10	10	10	Середній	Мілкосерійне	10
10^2	10^2	10^2	Середній	Мілкосерійне	10^2
10^3	10^3	10^3	Довгий	Одиничне	10^3
10^4	10^4	10^4	Довгий	Одиничне	10^4

При отриманні оцінки вартісної функції здійснюється фазифікація за допомогою трикутних функцій належності з півшириною основи, яка дорівнює розкиду експертних оцінок; нечітка макси-мінна композиція і дефазифікація методом знаходження центра ваги функції належності.

На рис. 2.14 показаний характер залежності вартості ПЗД (у відносних величинах щодо базової ПЗД) від співвідношення складності програмної і апаратної частин системи.

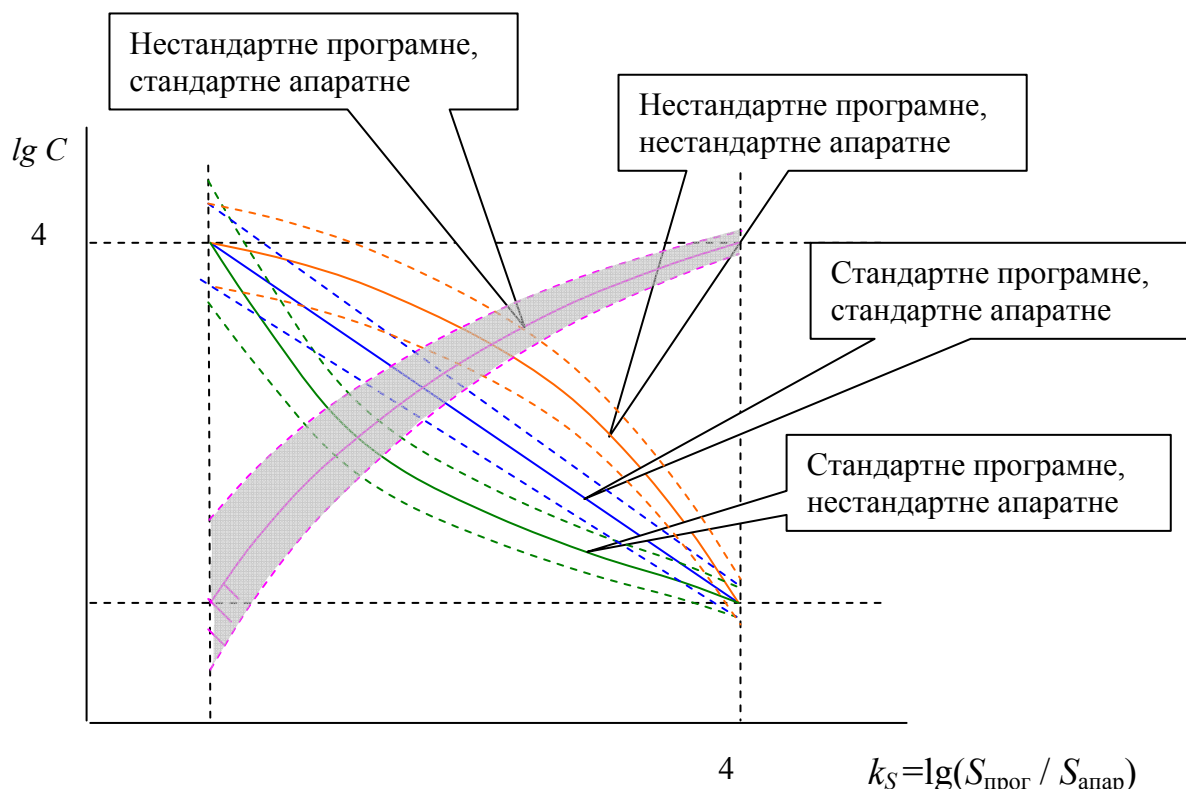


Рис. 2.14. Графік залежності вартості ПЗД від співвідношення складності програмної і апаратної частин системи

На рис. 2.14 розкид функцій належності наведений границями на α -рівні 0,1. Отримані залежності показують, що з достатньо високим рівнем вірогідності вартісна функція і критерій ефективності (2.69) в цілому можуть бути оцінені на основі показників програмної і апаратної складності і рівня оригінальності ПЗД.

2.4.6. Метрика у просторі структур ПЗД. Задача пошуку оптимальної структури підсистеми збору даних потребує ранжування простору структур. Таке ранжування дозволяє застосувати швидкі методи направлено пошуку

замість простого перебору варіантів. Для здійснення ранжування введемо норму і метрику на просторі структур.

Проаналізуємо, яка з характеристик $\{q\}$ структури ПЗД відповідає умовам введення норми [100]:

- 1) $\|q\| \geq 0$, причому $\|q\| = 0$ лише при $q = 0$;
- 2) $\|q_1 + q_2\| \leq \|q_1\| + \|q_2\|$ - нерівність трикутника;
- 3) $\|\alpha \cdot q\| = |\alpha| \cdot \|q\|$, де α - константа.

Легко бачити, що з усіх розглянутих характеристик зазначеним умовам відповідають дві характеристики: складність і вартість.

Вартість ПЗД є незручною для використання як норми, оскільки залежить не тільки від структури ПЗД, але й від цін на елементи і програмне забезпечення, які є змінними у часі і залежать від способу реалізації системи.

Складність ПЗД може бути використана за умови усунення неоднозначності залежності між складністю і вартістю, яка впливає з різного характеру залежності вартості від складності для апаратних і програмних засобів (див. рис. 2.14). Для усунення цього недоліку будемо розглядати структури у двовимірному просторі, координатами якого є складність програмного (q_n) і складність апаратного (q_a) забезпечення. Тоді норма окремого елемента ПЗД буде мати вигляд

$$q_i = \alpha q_n + \beta q_a, \quad (2.72)$$

де i – індекс елемента ПЗД;

α і β - коефіцієнти нормування, $\alpha, \beta > 0$.

Тоді метрика у нормованому двовимірному просторі елементів ПЗД однакового функціонального призначення буде мати вигляд

$$\rho_e = |q_1 - q_2| = \alpha |q_{n1} - q_{n2}| + \beta |q_{a1} - q_{a2}|, \quad (2.73)$$

де q_1 і q_2 – норми двох елементів.

Для переходу до метричного простору структур опишемо алгоритмічну модель матрицею перетворень даних. Визначимо матрицю перетворень за правилами:

1. Розмірність матриці перетворень $M[N'', N']$:

$$N'' = \text{card}[X'' \cup U], \quad (2.74)$$

$$N' = \text{card}[U \cup X'], \quad (2.75)$$

де $X''[n]$ - вектор контрольованих параметрів;

$X'[k]$ - вектор оцінюваних параметрів;

$U[m]$ - вектор проміжних параметрів;

card – оператор визначення розмірності.

2. Елементами матриці є оператори алгоритмічної моделі $E(op, p_1, p_2)$, де $p_1 = \{x'' \subset X''\}$ – вектор вхідних даних оператора, $p_2 = \{x' \subset X'\}$ – вектор вихідних даних оператора.

3. Оператор розташовується у рядках, які відповідають елементам x'' вектора X'' , і стовпцях, які відповідають елементам x' вектора X' .

4. Усі інші елементи матриці дорівнюють нулю.

З використанням матриці перетворень алгоритмічна модель системи може бути зображена у вигляді

$$[X'' \cup U] = [X'' \cup U] \cdot M, \quad (2.76)$$

де операція множення розуміється як «дія оператора», а операція додавання розуміється у теоретико-множинному сенсі об'єднання.

Відмінність між двома еквівалентними алгоритмічними моделями ПЗД з використанням матричного подання визначимо як модуль різниці

$$|\Delta M| = M_1 \text{ xor } M_2. \quad (2.77)$$

Охарактеризувавши елементи різницевої матриці нормами, отримуємо нормовану різницю структур $\Delta Q[N'', N']$.

Нормою різницевої матриці (2.77), виходячи з проведеного аналізу зв'язків між характеристиками ПЗД, природно прийняти сумарну складність елементів матриці.

Відповідно метрика у нормованому двовимірному просторі структур ПЗД буде мати вигляд

$$\rho_c = 1[N''] \cdot \Delta Q[N'', N'] \cdot 1^T[N'], \quad (2.78)$$

де $1[*]$ – одиничний вектор-стовпець відповідної розмірності;

T – знак транспонування.

3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ ПІДСИСТЕМ ЗБОРУ ДАНИХ В УМОВАХ НЕВИЗНАЧЕНОСТІ НА ОСНОВІ АЛГОРИТМІЧНИХ МОДЕЛЕЙ

Інформаційна технологія оптимізації ПЗД в умовах невизначеності включає комплекс методів оптимізації структур, прийомів їх подання та відповідне інформаційне забезпечення.

3.1. Концептуальна структура інформаційної технології

Застосування інформаційної технології оптимізації структури передбачається на двох стадіях життєвого циклу ПЗД: на стадії проектування і на стадії експлуатації. Якщо на стадії проектування всі операції технології можуть здійснюватися на локальному комп'ютері оператора, то на стадії експлуатації додаються задачі контролю стану структури ПЗД і її автоматизованої зміни. Для цього необхідні відповідні технічні засоби, які забезпечують апаратний інтерфейс інформаційної технології з ПЗД. Узагальнена структурна схема реалізації інформаційної технології з використанням позначень мови UML зображена на рис. 3.1.

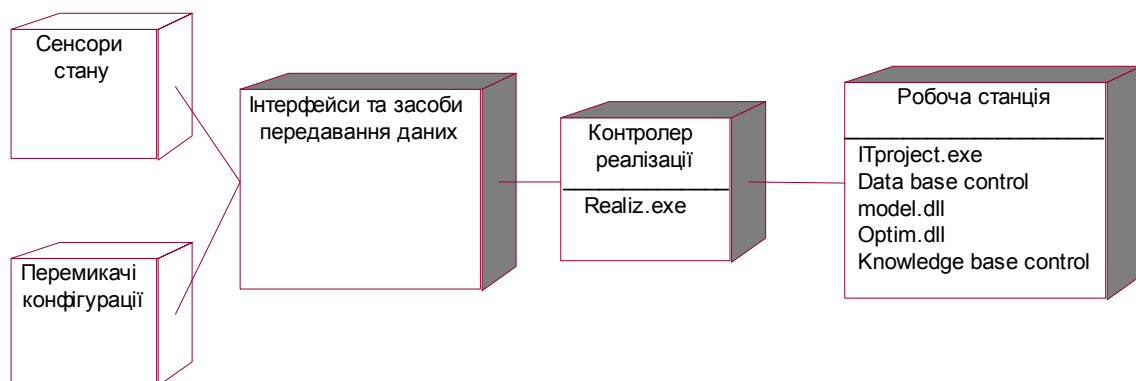


Рис. 3.1. Схема реалізації інформаційної технології на стадії експлуатації

Схема реалізації містить два головних елементи: робочу станцію, на якій оператор здійснює моделювання, пошук оптимальної структури і керування базами даних і знань, та контролер для взаємодії із ПЗД і реалізації оптимізованої структури. Взаємодія здійснюється через сенсори, які контролюють стан ПЗД, і перемикачі, які змінюють конфігурацію ПЗД. Зв'язок між контролером сенсорами і перемикачами здійснюється через

інтерфейси і засоби передавання даних.

Покажемо основні функції інформаційної технології у вигляді UML-діаграми варіантів використання, наведеної на рис. 3.2. На цій діаграмі з інформаційною технологією взаємодіють чотири зовнішніх агенти:

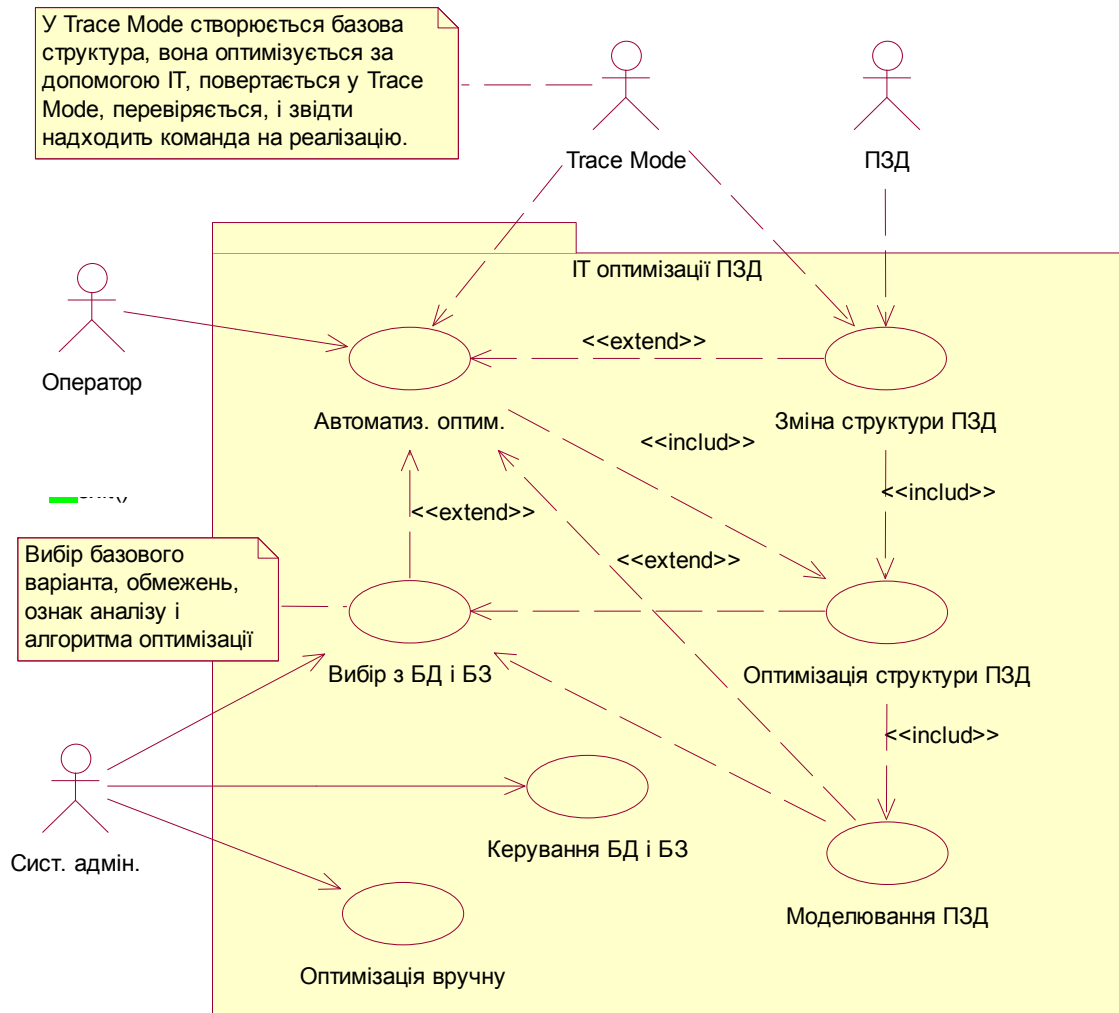


Рис. 3.2. Діаграма варіантів використання інформаційної технології

- оператор, який здійснює автоматизовану оптимізацію структури ПЗД;
- "системний адміністратор", оскільки постановка завдань і налаштування баз даних і знань виходять за рамки компетенції оператора;
- сама підсистема збору даних, якщо інформаційна технологія використовується на стадії експлуатації і оптимізація структури підсистеми здійснюється у автоматизованому режимі;
- система SCADA Trace Mode, яка використовується для реалізації і контролю результатів оптимізації.

На діаграмі показані також підфункції основних функцій системи.

Функції системи реалізуються за допомогою декількох основних класів програмних і апаратних засобів. Оскільки запропонована інформаційна технологія розглядається як розвиток технології, реалізованої у системі TraceMode, то основою для розробки структури класів є діаграма варіантів використання рис. 3.2., запропонований метод оптимізації структури систем на основі алгоритмічних моделей і базові класи системи TraceMode.

Розробку структури класів виконуємо у декілька етапів. На першому етапі покажемо класи, що реалізують основні варіанти використання і відносини між ними (рис. 3.3).

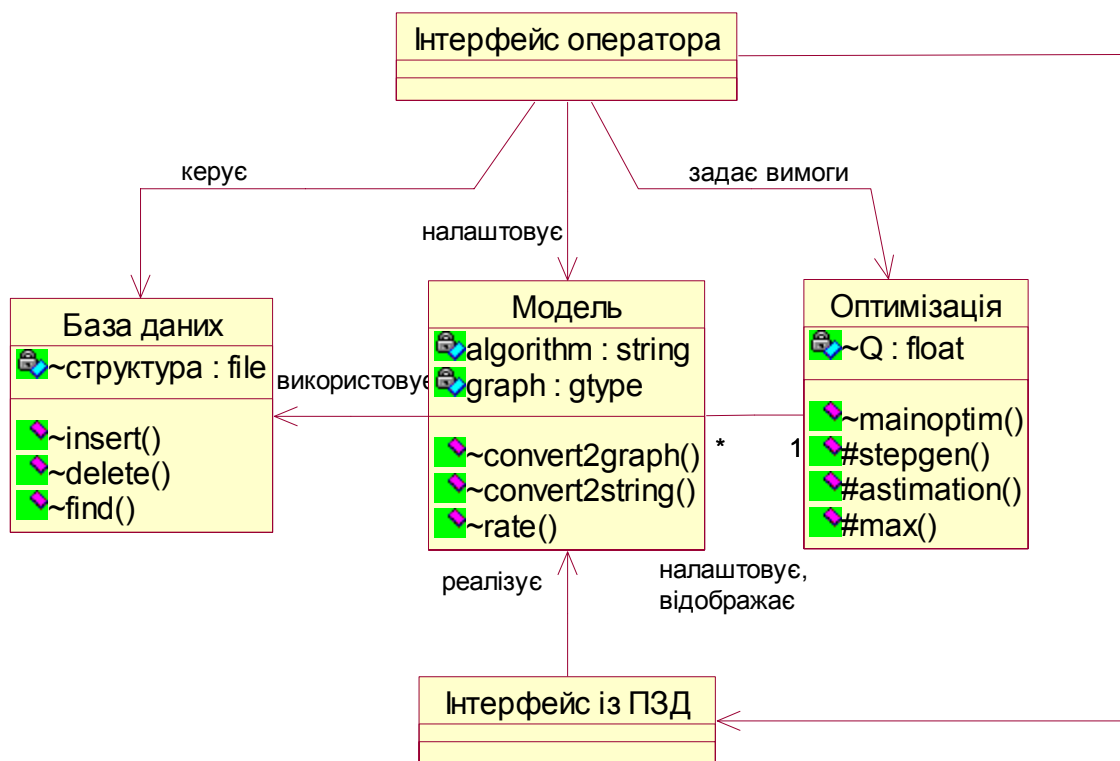


Рис. 3.3. Узагальнена діаграма класів

На другому етапі розглянемо детальніше кожен з класів.

База даних інформаційної технології складається з трьох основних сутностей: варіанти структур ПЗД, елементи ПЗД, алгоритми збору даних, а також база знань. Отже, клас «База даних» є клас-комполит, що показано на діаграмі рис. 3.4. Крім того, база даних має обов'язковий атрибут – опис її структури, що зберігається у файлі, і стандартний мінімальний набір операцій вставки, видалення і пошуку даних.

Модель відповідно до підрозділу 2.2 подається у вигляді алгоритму, записаного у формі рядка символів, і у вигляді графа. Покажемо це як агрегацію відповідних підкласів на рис. 3.5.

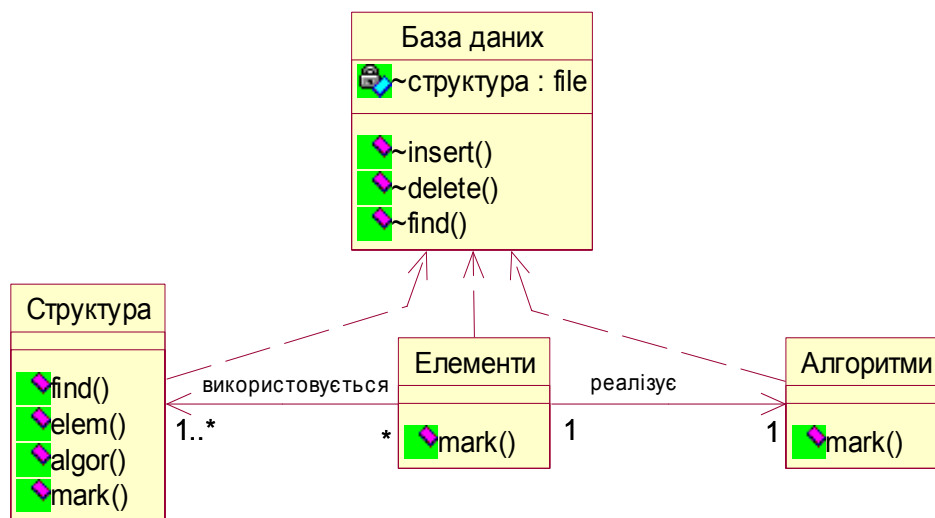


Рис. 3.4. Діаграма класу «База даних»

Клас «Модель» включає три основні операції: перетворення моделі у вигляді графу в модель у вигляді рядка символів і назад, а також операцію оцінювання характеристик моделі.

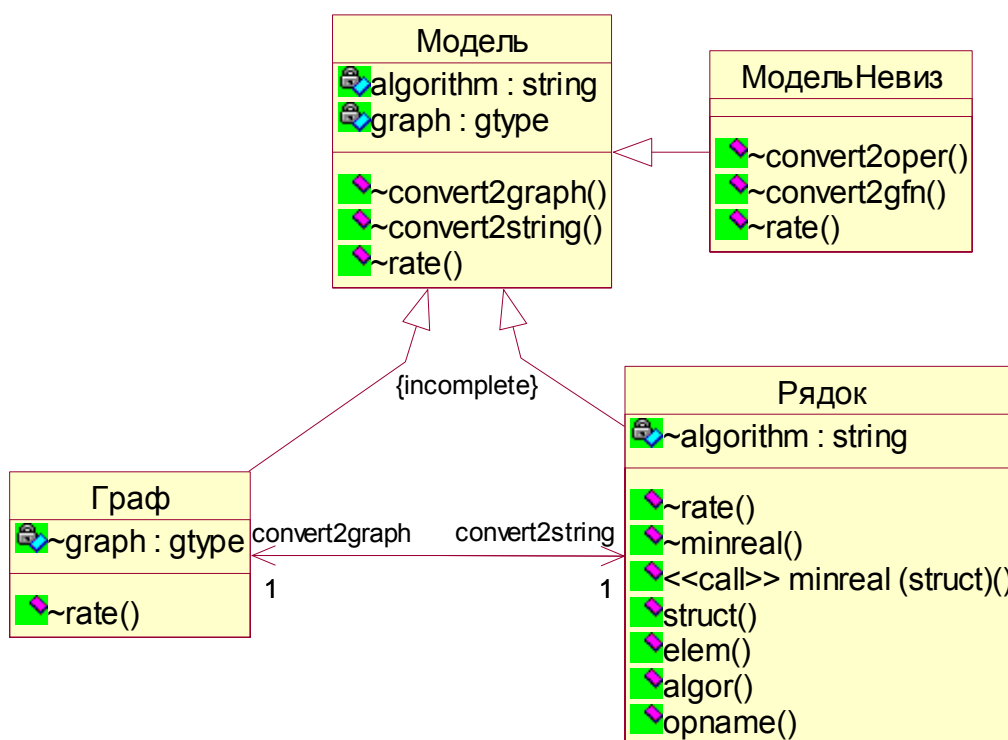


Рис. 3.5. Діаграма класу «Модель»

Модель може розглядатися не лише в умовах повністю визначених параметрів, але й в умовах невизначеності. Для цього створюється нащадок базового класу «Модель», в якому, відповідно до підрозділу 2.3, передбачені операції перетворення алгоритмічної моделі в операторний вигляд, детермінованого опису параметрів – в опис за допомогою узагальнюючих функцій невизначеності і перевизначена операція оцінювання з врахуванням нового способу опису. Атрибутами моделі є відповідно рядок символів і опис графу. Опис графу має нестандартний тип `grtype`, який є структурою, що об'єднує матрицю суміжності графу та опис потоків даних і операцій.

Клас «Оптимізація», показаний на рис. 3.6, включає базові операції генерування кроку пошуку, розрахунку значення критерію оптимізації Q і пошуку максимуму.

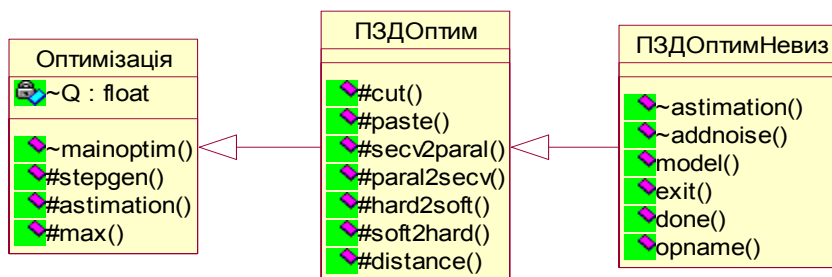


Рис. 3.6 Діаграма класу «Оптимізація»

Цей клас є абстрактним, оскільки в ньому не враховані особливості задання структурної оптимізації ПЗД на основі алгоритмічної моделі. Тому створимо на його основі конкретний клас-нащадок ПЗДОптим. У класі-нащадку визначимо операції еквівалентних перетворень алгоритмічної моделі, відповідно до підрозділу 2.3, за допомогою яких генеруються варіанти структур ПЗД. Оцінювання критерію оптимізації (2.71) відбувається з використанням виклику функції оцінювання характеристик моделі `rate()`.

Наступним етапом розробки класу «Оптимізація» буде створення нащадку ПЗДОптимНевиз, що враховує функціонування ПЗД в умовах невизначеності. При цьому змінюється спосіб (операція) оцінювання критерію. Перевизначена операція оцінювання критерію завдяки властивості поліморфізму використовує оцінки характеристик моделі класу «МодельНевиз».

Клас «Інтерфейс із ПЗД» подамо у вигляді композиції трьох підкласів, що реалізують основні функції.

Діаграма класу «Інтерфейс із ПЗД» показана на рис. 3.7. Клас виконує інтерпретацію оптимізованої моделі системи, порівнює її з результатами контролю стану існуючої структури, приймає рішення про зміни і реалізує ці зміни. Клас поданий як композиція, оскільки, вочевидь, підклас «Інтерпретація моделі» реалізовуватиметься програмно, клас «Контроль структури» - апаратний, а клас «Реалізація» - апаратний за участю персоналу.

І, нарешті, клас «Інтерфейс оператора» (рис. 3.8) є звичайним мультивіконним інтерфейсом.

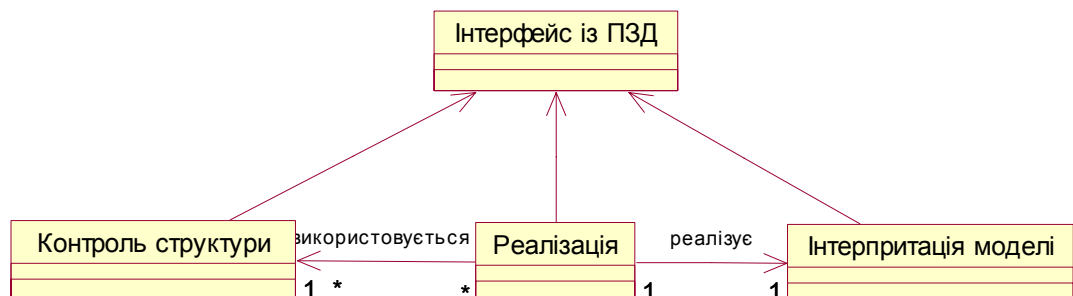


Рис. 3.7. Діаграма класу «Інтерфейс із ПЗД»

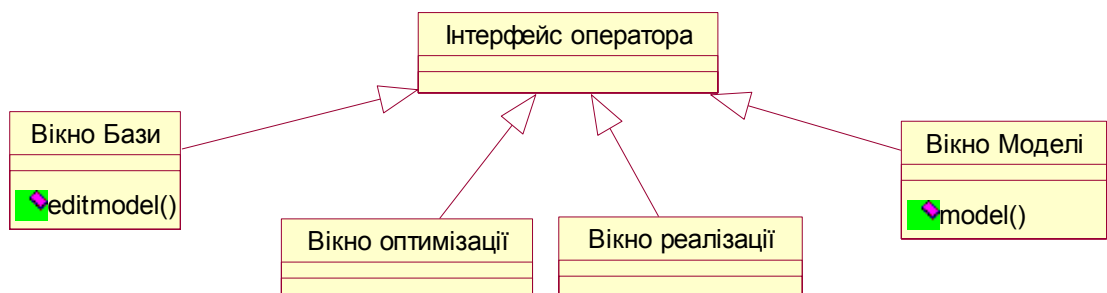


Рис. 3.8 Діаграма класу «Інтерфейс оператора»

3.2. Оптимізація підсистем збору даних

Клас «Оптимізація» реалізується за допомогою комбінації формальних і евристичних процедур.

Постановка задачі оптимізації: нехай дано модель ПЗД у вигляді системи рівнянь у векторній формі

$$X'[k] = F^{-1}(X''[n]).$$

Критерій ефективності у просторі структур ПЗД визначимо як функціонал від комплексу характеристик ПЗД

$$Q(C, \theta_S, P_H, I_u, T_u, H_{зал}) = \frac{I_u / I_0}{C / C_0}. \quad (3.1)$$

Параметром оптимізації є структура ПЗД, яку можна подати у вигляді комплексного показника

$$S : \{s_e, s_f, s_c, s_{inp}, s_{out}\}, \quad (3.2)$$

де $s_e(P_{21}, P_{22})$ - елементи ПЗД;

P_{21} – множина апаратно реалізованих операцій;

P_{22} – множина програмно реалізованих операцій;

$s_c(P_{11}, P_{12})$ - зв'язки між елементами;

P_{11} – множина послідовних операцій;

P_{12} – множина паралельних операцій;

s_f - функції елементів;

s_{out} - множина виходів;

s_{inp} - множина входів.

Обмеження, що накладаються на структуру ПЗД, визначимо у вигляді системи нерівностей

$$\begin{cases} 0 < C \leq C_{\max}, \\ 0 < T_u \leq T_{u\max}, \\ P_{H_{ГР}} \leq P_H \leq 1, \end{cases}$$

де C_{\max} - максимальна вартість підсистеми збору даних;

$T_{u\max}$ - максимальна швидкодія підсистеми;

$P_{H_{ГР}}$ - мінімальна інформаційна надійність, за якої ПЗД ефективно функціонує.

Оскільки в більшості випадків функціонування ПЗД відбувається в умовах невизначеності, розглянемо окремо оптимізацію в умовах визначеності та невизначеності.

3.2.1. Оптимізація в умовах повної визначеності. В умовах повної визначеності передбачається, що контролюються всі суттєві параметри об'єкта, склад вектора контрольованих параметрів не змінюється, а його компоненти є стаціонарними процесами.

Розв'язком задачі оптимізації моделі ПЗД є послідовність підстановок, яка деталізує і оптимізує базову модель

$$M=A(B) E0 (op_1; X; Y) A(E).$$

Оптимізація в процесі синтезу здійснюється на основі критерію ефективності (3.1).

Вираз критерію ефективності отримується з алгоритмічної моделі ПЗД за методикою, запропонованою у підрозділі 2.4.

Для дослідження впливу структурних характеристик на критерій ефективності охарактеризуємо вплив наявності зв'язку між підсистемами на результат функціонування системи трьома векторами:

- $D[m]$, елемент якого $d_{ij}(x)$ характеризує вплив наявності зв'язку між i -ю і j -ю підсистемами на результат функціонування системи (m – верхня границя кількості зв'язків між підсистемами; $x = 0$, якщо зв'язку немає і $x = 1$, якщо є);

- $T[m]$, елемент якого $t_{ij}(x)$ характеризує вплив наявності зв'язку між i -ю і j -ю підсистемами на час отримання результату функціонування системи;

- $L[m]$, елемент якого l_{ij} характеризує вартість з'єднання підсистем.

Відповідно до структурно-логічної схеми рис. 2.13, вартісна функція залежить від інформаційної надійності. Розглянемо вплив структурних особливостей ПЗД на цю залежність для окремих випадків отримання вартісної функції у критерії (3.1).

1. Вартісна функція паралельно з'єднаних підсистем, які забезпечують адитивний результат роботи системи. Матриця суміжності такої системи містить один ненульовий стовпець. Тоді

$$c_{nap} = E_0 \sum_{i=1}^n c_{P_{2i}} + \left[C_0 - \sum_{i,j} \int_0^1 d_{ij}(s_{ij}) \beta_{ij}(s_{ij}) ds_{ij} \right] + c_t \max_i \left[\int_0^1 t_{ij}(s_{ij}) (1 - \beta_{ij}(s_{ij})) ds_{ij} \right], \quad (3.3)$$

де $\{s_{ij}\} = S[n, n]$ – матриця з'єднань підсистем;

C_0 – вартість результату роботи системи;

E_0 – коефіцієнт окупності витрат на апаратні засоби, який дорівнює частині вартості апаратних засобів, що припадають на одне виконане завдання системи;

$\{c_{P_{2i}}\}$ – вектор вартості програмно-апаратних засобів i -ї підсистеми

$$c_{P_{2i}} = c_{P_{21i}} + c_{P_{22i}} + \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n l_{ij} s_{ij}. \quad (3.4)$$

2. Вартісна функція паралельно з'єднаних підсистем, які забезпечують мультиплікативний результат роботи системи

$$c_{nap} = E_0 \sum_{i=1}^n c_{P_{2i}} + \left[C_0 - \prod_{i,j} \int_0^1 d_{ij}(s_{ij}) \beta_{ij}(s_{ij}) ds_{ij} \right] + c_t \max_i \left[\int_0^1 t_{ij}(s_{ij}) (1 - \beta_{ij}(s_{ij})) ds_{ij} \right]. \quad (3.5)$$

3. Вартісна функція послідовно з'єднаних підсистем

$$c_{noc} = E_0 \sum_{i=1}^n c_{P_{1i}} + \left[C_0 - \prod_{i,j} \int_0^1 d_{ij}(s_{ij}) \beta_{ij}(s_{ij}) ds_{ij} \right] + c_t \sum_i \left[\int_0^1 t_{ij}(s_{ij}) (1 - \beta_{ij}(s_{ij})) ds_{ij} \right], \quad (3.6)$$

$$\text{де } c_{P_{1i}} = c_{P_{21i}} + c_{P_{22i}} + \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n l_{ij} s_{ij}. \quad (3.7)$$

В цілому перший доданок визначає апаратні витрати на виконання завдання системи. У загальному випадку у векторній формі

$$c_1 = E_0^T \times C, \quad (3.8)$$

де T – знак транспозиції.

В цілому другий доданок визначає втрати результату від невизначеності структури. У загальному випадку

$$c_2 = C_0 - \left| \int_0^1 D(s_{ij}) \times B(s_{ij}) ds_{ij} \right|, \quad (3.9)$$

де $|\bullet|$ - знак визначника.

В цілому третій доданок визначає втрати часу від невизначеності структури. У загальному випадку

$$c_3 = c_t \left\langle \int_0^1 T(s_{ij}) \times [1 - B(s_{ij})] ds_{ij} \right\rangle, \quad (3.10)$$

де $\langle \bullet \rangle$ - знак аналога визначника, в якому операція додавання замінюється на операцію \max , а операція множення – на операцію Σ .

Оптимізація здійснюється шляхом пошуку співвідношення послідовного або паралельного виконання операцій на основі зміни двох показників: програмна або апаратна реалізація операцій, послідовне або паралельне виконання операцій.

Нечітка база знань, яка відповідає схемі рис. 2.13 і використовує терми таблиці 2.7, складається з 3 груп правил: правила перетворення реалізації: програмна – апаратна, апаратна – програмна, послідовна – паралельна, паралельна – послідовна; правила оцінювання складності; правила оцінювання вартості. Форма таблиць бази знань з початковими даними, які повинні доповнюватися у процесі її використання наведена у табл. 3.1 – 3.4.

При заміні частини апаратних операцій на програмні до алгоритмічної моделі додаються операції інтерфейсних перетворень.

Заміна апаратних перетворень на програмні і навпаки здійснюється на підставі тверджень:

$$E_0(op_H, X_H, Y_H) = E_1(I_1, X_H, X_S) E_2(op_S, X_S, Y_S) E_3(I_2, Y_S, Y_H), \quad (3.11)$$

$$E_0(op_S, X_S, Y_S) = E_1(I_2, X_S, X_H) E_2(op_H, X_H, Y_H) E_3(I_1, Y_H, Y_S), \quad (3.12)$$

де H – позначення апаратно-орієнтованих операцій і форми даних;

S – позначення програмно-орієнтованих операцій і форми даних;

I_1 – апаратно-програмне інтерфейсне перетворення;

I_2 – програмно-апаратне інтерфейсне перетворення.

Отже, заміна апаратних перетворень $[n1 - n2]$ на програмні здійснюється за допомогою операцій

$$cut(n1, n2) paste(E_1(I_1, X_H, X_S) E_2(op_S, X_S, Y_S) E_3(I_2, Y_S, Y_H), prev(n1), next(n2)) \quad (3.13)$$

і аналогічно при оберненому перетворенні.

**База знань (структура і фрагменти змісту)
Перетворення реалізації: програмна – апаратна**

№	Функція	Модель	Ознаки	Варіанти реалізації	Оточення		Формальне перетворення
					Попереднє	Наступне	
1	2	3	4	5	6	7	8
1	Просте функціональне перетворення	$E_{n1}(C: op_s; (x_1); (x_2))$	1. Відсутні переходи	Аналогова	Програмне	Апаратне аналогове	$m=last;$ $m1=prev(n1); m2=next(n1);$ $cut(m1 m2 1);$ $paste(E_{n1}(), m1, m2);$ $paste(E_m(op_h; (p_{4h}, x_{2h})); m2+4; m2+5)$
2	Просте функціональне перетворення	$E_{n1}(C: op_s; (x_1); (x_2))$	1. Відсутні переходи	Дискретна	Програмне	Апаратне дискретне	$m=last;$ $m1=prev(n1); m2=next(n1);$ $m3=next(m2);$ $paste(E_{m+1}(T:(\tau); (arg(m2)\backslash res(n1)); (P_{2h}:int[m])))$ $E_m(op_h; P_{2h}, x_{2h}:int[m]) m2; m3)$ $cut(n1, m2);$ $paste(E_{n1}(), m1, next(m1))$
3	Просте функціональне перетворення	$E_{n1}(C: op_s; (x_1); (x_2))$	1. Відсутні переходи	Дискретна	Програмне	Програмне	$m=last;$ $m1=prev(n1);$ $m2=next(n1);$ $cut(m1, m2);$ $paste(E_{n1}(), m1, m2);$ $paste((E_{m+1}(T:(\tau); (x_{1s}); (P_{2h}:int[m])))$ $E_{m+6}(op_h; P_{2h}, P_{3h});$ $E_{m+11}(T:(\tau); (P_{3h}); (x_{2s})); n1; m2)$

1	2	3	4	5	6	7	8
4	Просте функціональне перетворення	$E_{n1}(C: op_s; (x_1); (x_2))$	1. Відсутні переходи	Аналогова	Програмне	Програмне	$m=last;$ $m1=prev(n1);$ $m2=next(n1);$ $cut(m1,m2);$ $paste((E_{n1}() E_{m+1}(T:(\tau); (x_{1s}); (P_{2h}:int[m])))$ $E_{m+2}(C_h; (1); (i)) E_{m+3}(C_h:[p_{2hi}*p_{3h}*k^i], (P_{2h},p_{3h},$ $k); (p_{4h})) E_{m+4}(C_h:(i+1); (i); (i))$ $w_{h\ m+5}(i \leq m) _{m+3}$ $E_{m+6}(op_h; p_{4h}, x_{2h}); E_{m+7}(C; (1); (i))$ $E_{m+8}(I:ent[ent[x_{2h}/p_{2h}]/k], (x_{2h},p_{2h}); P_{3h}:int[m])$ $E_{m+9}(C:(i+1); (i); (i))$ $w_{m+10}(i \leq m) _{m+8}$ $E_{m+11}(T:(\tau); (P_{3h}); (x_{2s}); m1; m2)$

Таблиця 3.2

Перетворення реалізації: послідовна – паралельна

№	Модель	Ознаки	Розташування у просторі	Варіанти реалізації	Оточення		Формальне перетворення
					попереднє	попереднє	
1	$E_{n1}(op_1, (x_1), (x_2))$ $E_{n2}(op_2, (x_3), (x_4))$	1. Відсутні переходи 2. Послідовні операції незалежні (множини вхідних даних E_{n2} і вихідних даних E_{n1} не перетинаються)	зосереджене	апаратна	Апаратне послідовне	Апаратне послідовне	$m=last;$ $m1=prev(n1);$ $m2=next(n2);$ $paste([E_{m+1}(I_3, (X\{arg(n_1),$ $arg(n_2)\}), arg(n_1)], m1, n1);$ $paste(][E_3(I_3, (X\{arg(n_1), arg(n_2)\}),$ $arg(n_2)) n1, n2);$ $paste(] E_4(I_4, (res(n_1), res(n_2)),$ $(X\{res(n_1), res(n_2)\})))$

Перетворення реалізації: апаратна – програмна

№	Функція	Модель	Ознаки	Розташування у просторі	Варіанти реалізації	Оточення		Формальне перетворення
						попереднє	наступнє	
1	Перетворення роду сигналу і форми інформації	$E_{n1}(op_H, (x_1), (x_2))$	1. Відсутні переходи 2. Виконуються виключно нелінійні операції	Зосереджене	цифрова	апаратне	апаратне	$m=last;$ $m1=prev(n1);$ $m2=next(n1);$ $cut(m1m21);$ $paste(E_{m+1}(I_1, (x_{H1}), (x_{S1})));$ $E_{n1}(ops, (x_{S1}), (x_{S2})); E_{m+2}(I_2, (x_{S2}), (x_{H2}), m1, m2)$
2	Управління каналами	$E_{n1}(x_3=x_2, x_1, (x_1, x_2), (x_3))$	1. Вхідна змінна використовується як індекс	Зосереджене	цифрова	апаратне	апаратне	$m=last;$ $m1=prev(n1);$ $m2=next(n1);$ $cut(m1m21);$ $paste(E_{m+1}(I_1, (x_{H1}), (x_{S1})));$ $E_{m+2}(I_1, (x_{H2}), (x_{S2})) E_{n1}(x_3=x_2, x_1, (x_1, x_2), (x_3)); E_{m+3}(I_2, (x_{S3}), (x_{H3}), m1, m2)$

Перетворення реалізації: паралельна – послідовна

№	Модель	Ознаки	Розташування у просторі	Варіанти реалізації	Оточення		Формальне перетворення
					попереднє	попереднє	
1	$ [E_{n1}(op_1, (x_1), (x_2)); E_{n2}(op_2, (x_3), (x_4Y))]$	1. Наявний символ	зосереджене	апаратна	Апаратне паралельне	Апаратне паралельне	$m1=prev(n1);$ $m2=next(n2);$ $cut(m1,n1);$ $cut(n1,n2);$ $cut(n2,m2)$

При розпаралелюванні виконання операцій до алгоритмічної моделі додаються операції синхронізації і збирання даних, що паралельно надходять.

Заміна послідовних операцій на паралельні здійснюється на підставі твердження

$$E_1(op_1, X1, Y) E_2(op_2, X2, Y) = || [E_3(I_3, (X1, X2), X1) E_1(op_1, X1, Y1)] [E_3(I_3, (X1, X2), X2) E_2(op_2, X2, Y2)] E_4(I_4, (Y1, Y2), Y), \quad (3.14)$$

де I_3 – операція розподілення вхідних даних;

I_4 – операція об'єднання результатів.

Процедура розпаралелювання використовує ГПЗ – граф перетворення сигналів і залежностей даних [80]. Для формалізації процедури в алгебрі AS визначаються відношення залежності \otimes , які і визначають граф $G(OP, X, Y)$:

$$G(OP, X, Y) : E_1(op_1, X1, Y1) \otimes E_2(op_2, X2, Y2), \quad \text{якщо} \\ (X1 \otimes X2) \cup (X1 \otimes Y2) \cup (X2 \otimes Y1). \quad (3.15)$$

Для відношень залежності виконується властивість транзитивності.

Аналіз залежностей може бути здійснений за допомогою динамічної структури типу двовимірний зв'язаний список (рис. 3.9), який формується за алгоритмом, наведеним на рис. 3.10.

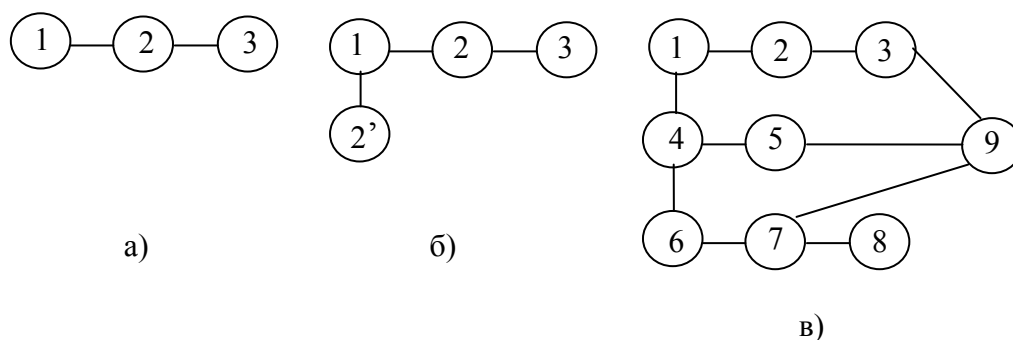


Рис. 3.9. Граф перетворень і залежностей

- а) елемент «3» додається до послідовного списку, який містить попереднє перетворення, або умову; б) елемент «2'» починає новий послідовний список; в) елемент «9» є об'єднувальним для відповідних послідовних

Згенеровані за допомогою перетворення (3.13) і ГПЗ варіанти формальних еквівалентних перетворень алгоритмічної моделі повинні бути

перевірені на можливість технічної реалізації. Для цього створимо базу знань для кожного типу перетворень. Схема бази знань показана на рис. 3.11.

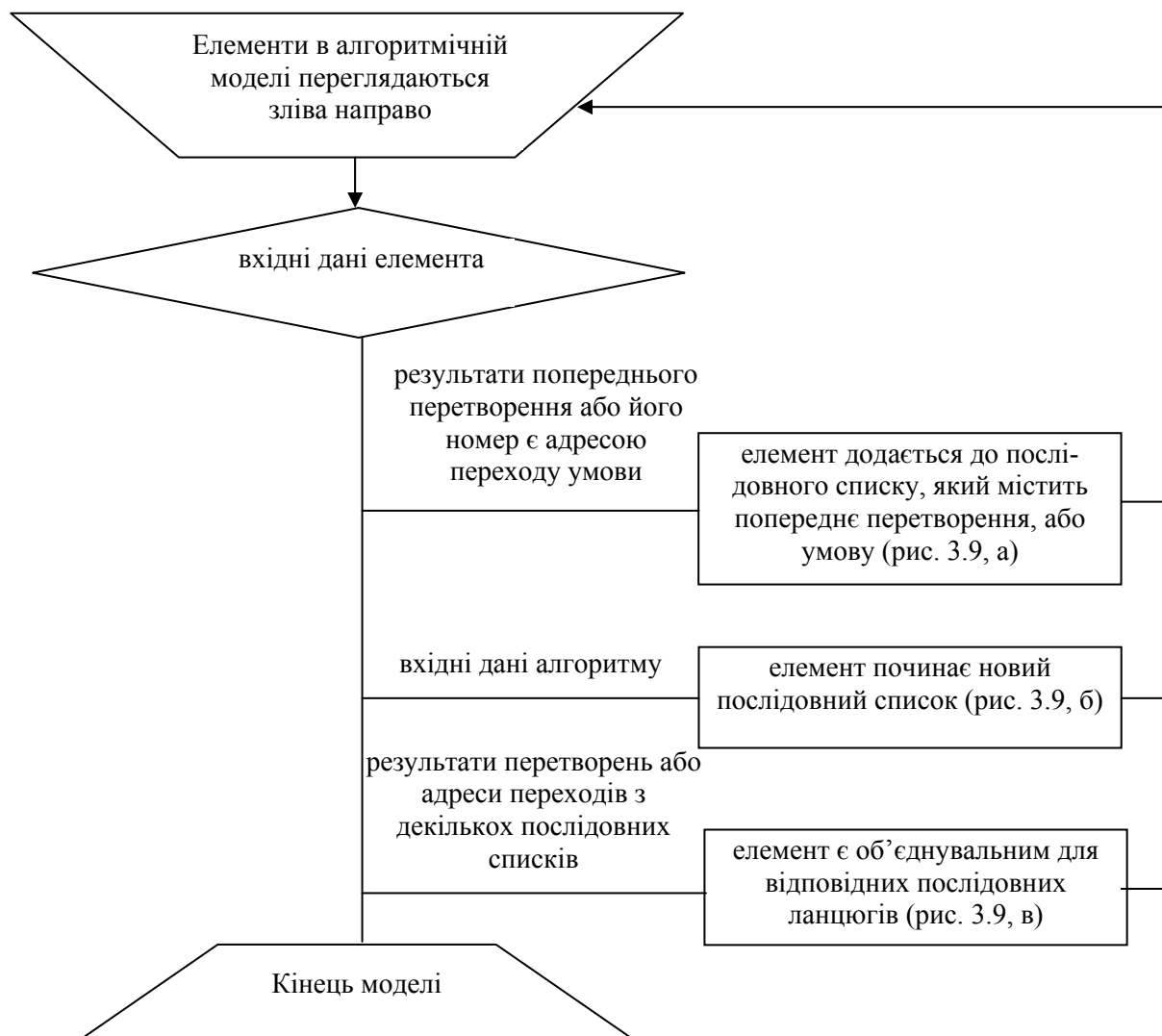


Рис. 3.10. Алгоритм синтезу структури моделі

Можливі перетворення апаратних операцій на програмні задаються базою знань, фрагмент якої наведений у табл.3.1, зворотне перетворення у табл. 3.2. Використання окремих реалізацій бази знань пов'язане з необхідністю виконання аналізу початкової структури ПЗД і виділення типових елементів на основі ознак, що наведені у колонках 4-7.

Аналогічні фрагменти бази знань для здійснення послідовно-паралельних перетворень і навпаки наведені у табл.3.3. і 3.4 відповідно.

Загальна методика оптимізації структури ПЗД в умовах визначеності передбачає:

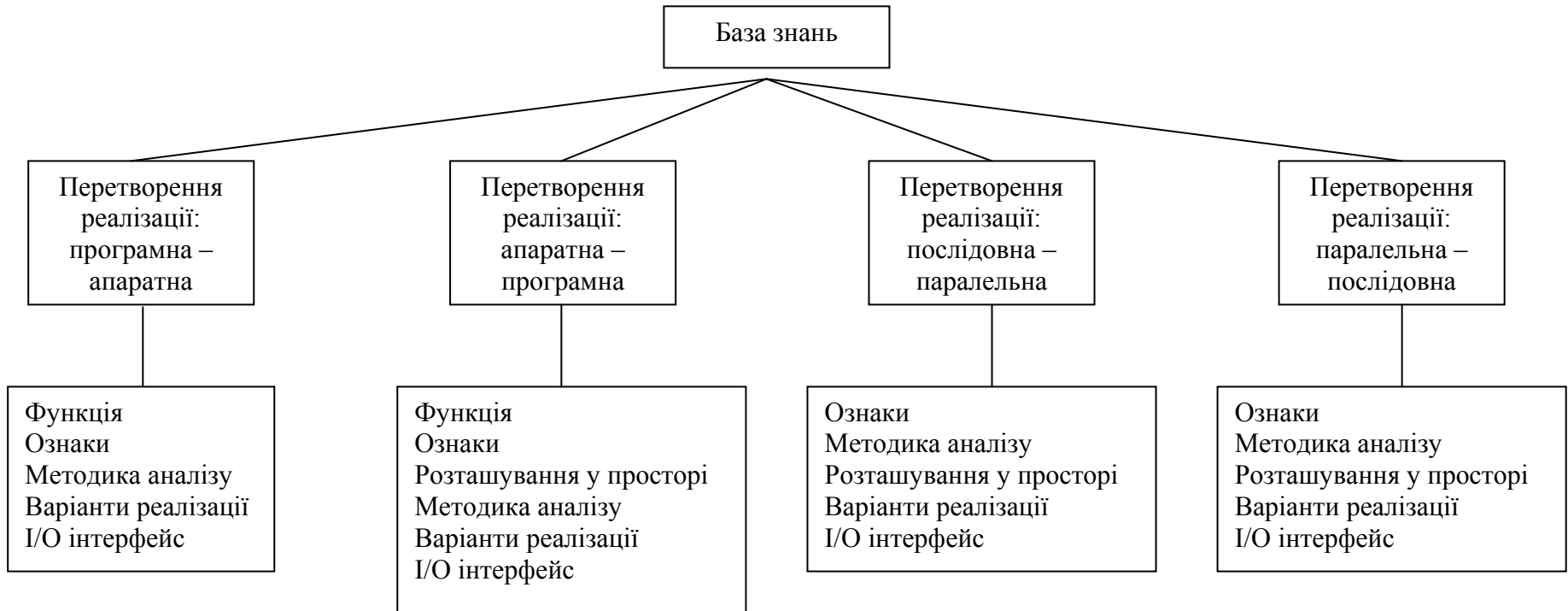


Рис. 3.11. Структура бази знань аналізу можливості технічної реалізації

1. На основі моделі технологічного процесу контролю і технічної документації визначаються варіанти векторів контрольованих параметрів, які повністю визначають стан процесу;

2. Для кожного варіанта складається базова алгоритмічна модель ПЗД. Для цього:

2.1. Модель технологічного процесу розв'язується відносно невідомих параметрів стану;

2.2. Розв'язок подається у вигляді алгоритмічної моделі (за методикою підрозділу 2.2 з використанням бази алгоритмів (рис. 3.4));

2.3. На основі алгоритмічної моделі будується граф залежності даних;

2.4. На основі графу залежностей з використанням бази типових структур (рис. 3.4) генеруються варіанти послідовно-паралельних структур виконання збору даних;

2.5. Обирається елементна база первинних засобів збору даних з використанням бази елементів (рис. 3.4);

2.6. Враховуючи фізичні характеристики сигналів-носіїв даних і розташування елементів для кожного варіанта послідовно-паралельного виконання ПЗД генеруються додаткові варіанти програмної та апаратної реалізацій і в алгоритмічну модель вставляються оператори інтерфейсу та передавання даних;

2.7. Оцінюються характеристики отриманих варіантів структур: складність ПЗД (за методикою п. 2.4.1), невизначеність стану технологічного процесу (за методикою п. 2.4.2), швидкодії (за методикою п. 2.4.3), надійності (за методикою п. 2.4.4);

2.8. Розраховується комплексний показник ефективності ПЗД;

2.9. Обирається базовий варіант моделі ПЗД, який характеризується найбільшим значенням показника ефективності;

3. За допомогою системи еквівалентних перетворень (підрозділ 2.3) генеруються варіанти модифікації структури ПЗД;

4. Використовуючи базу знань, відбираються варіанти структур, які можуть бути фізично реалізовані;

5. За допомогою введеної у п. 2.4.6. метрики отриманий набір варіантів ранжують за зростанням.

6. Для кожного варіанта структури виконують етапи 2.7 – 2.9.

Запропонована методика є основою методу оптимізації підсистеми збору даних.

3.2.2. Оптимізація в умовах невизначеності. В умовах невизначеності знайти розв'язок системи рівнянь моделі технологічного процесу, як це передбачено у кроці 2.1 методики, наведеної вище, неможливо. Тому основним способом визначення неконтрольованих параметрів технологічного процесу є, як зазначено у підрозділі 2.1, метод оцінювання. Алгоритмічні моделі оцінювання неконтрольованих параметрів для деяких типових випадків [71] наведені у п.2.2 і табл. 2.4.

Вхідними даними алгоритмів оцінювання є значення вимірних параметрів і експертні оцінки невимірних параметрів, а також апіорні оцінки їх невизначеності у вигляді с.к.в. В результаті виконання алгоритмів оцінювання отримуються оптимізовані значення параметрів і апостеріорні оцінки їх невизначеності [33, 39]. Якщо кількість вимірних параметрів надлишкова, тобто перевищує кількість невідомих у моделі об'єкта, то за умови правильності моделі апіорна невизначеність вимірних параметрів зменшується.

Розглянемо особливості і характеристики алгоритму оцінювання в умовах невизначеності параметрів моделі. Для цього подамо матрицю Якобі моделі (2.2) у вигляді суми визначеної і невизначеної складових

$$[A] = [A_0] + [\Delta_A], \quad (3.16)$$

де $[A_0]$ – матриця визначених складових;

$[\Delta_A]$ – матриця невизначених відхилень.

Тоді лінеаризована модель матиме вигляд

$$[X'] + [\Delta_{X'}] = ([X''] + [\Delta_{X''}]) \cdot ([A_0] + [\Delta_A]) \quad (3.17)$$

або

$$[\Delta_{X'}] = ([X''] + [\Delta_{X''}]) \cdot ([A_0] + [\Delta_A]) - [X']. \quad (3.18)$$

Якщо критерієм задачі оцінювання є мінімум сумарної середньої квадратичної похибки оцінювання, то, з урахуванням кореляції параметрів, цей критерій можна подати у вигляді

$$\sigma_{OЦ} = \sum_{i=1}^n (\Delta_{X_i})^2 = \{([X^n] + [\Delta_{X^n}] \cdot ([A_0] + [\Delta_A]) - [X^T] \cdot [r_{X^T}]) \cdot \{([X^n] + [\Delta_{X^n}] \cdot ([A_0] + [\Delta_A]) - [X^T]\}^T, \quad (3.19)$$

де $[r_{X^T}]$ - матриця коефіцієнтів кореляції.

Остаточна невизначеність результату оцінювання отримується підстановкою вектора X'_0 у рівняння (3.18).

Отже, задача оцінювання в умовах невизначеності параметрів моделі полягає у пошуку такого вектора X'_0 , який мінімізує критерій (3.19).

Оптимізація алгоритмічної моделі здійснюється за допомогою системи еквівалентних перетворень. В умовах невизначеності перетворення виконуються в операторній формі, яка отримується за методикою, запропонованою у п. 2.3.1.

У більшості досліджень невизначеність розглядалася як негативний фактор, вплив якого слід зменшувати, якщо це дозволяють технічні і економічні можливості [25, 28]. Але, як показано у п. 2.3.2, існують випадки, коли невизначеність може бути корисною для підвищення ефективності ПЗД. Тому врахування невизначеності відкриває нові можливості проектування і оптимізації ПЗД на основі алгоритмічних моделей.

Знайдемо умови, за яких додаткова невизначеність може покращувати ефективність системи. Розглянемо два типи систем: паралельну і послідовну (рис. 3.12). Ефективність систем будемо оцінювати за узагальненим критерієм (2.71).

Запишемо вартості систем рис. 3.12:

$$C_1 = c_1 + c_2 + c_3 + c_4 + c_5, \quad (3.20)$$

$$C_2 = c_1 + c_2 + c_3 + c_6 + c_5, \quad (3.21)$$

де c_i - вартості відповідних вузлів.

Очевидно, при вартості комутатора меншій за вартість блока проміжної обробки інформації $c_6 < c_4$ послідовна система рис. 3.12, б буде дешевшою за паралельну рис. 3.12, а.

Кількість інформації, отримання якої забезпечують системи (рис. 3.12),

$$I = H_{анр} - H_{анс} = - \int_{\Omega_Y} \beta(y) \cdot \log \beta(y) dy + \int_{\Omega_Y} \beta(y/x_1, x_2) \cdot \log \beta(y/x_1, x_2) dy, \quad (3.22)$$

де $H_{анр}$ - апіорна ентропія результату;

H_{anc} – апостеріорна ентропія результату.

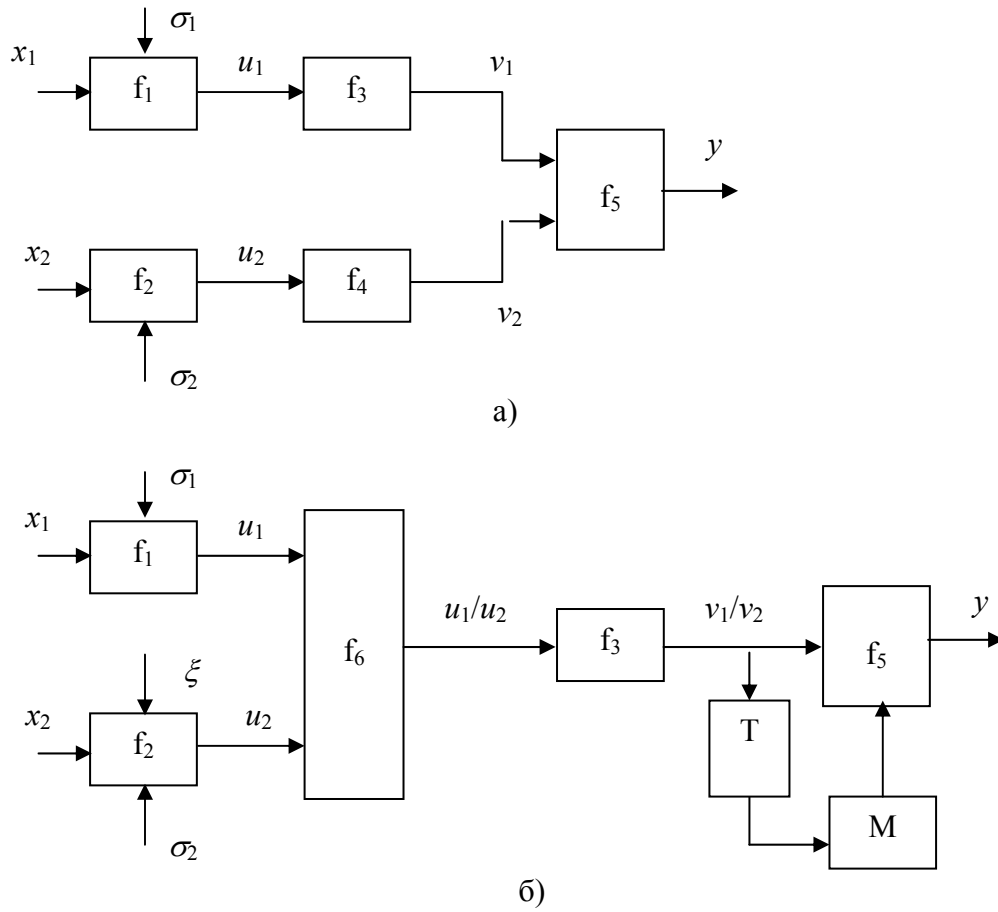


Рис. 3.12. Приклад системи обробки сигналів з додатково введеним штучним джерелом невизначеності
а) послідовна система; б) паралельна система

Або, за умови нормального розподілу, кількість інформації

$$I = \ln \sigma_{anp} - \ln \sigma_{anc} = \ln \frac{\sigma_{anp}}{\sigma_{anc}}. \quad (3.23)$$

Узагальнюючі функції невизначеності знайдемо за допомогою алгоритмічних моделей систем з наступним перетворенням їх на операторну модель

$$M_1=A(B) \parallel [E_1(S: kx, g, (x_1'), (x_1, \varepsilon_1)) \quad E_2(C: f_1(x_1+\sigma_1), (x_1, \sigma_1), (u_1(t))) \quad E_3(C: f_3(u_1(t)), (u_1(t)), (v_1(t))) \quad E_4(C:I(v_1(t)\cdot\delta(t-\tau));(v_1(t));(v_1(t+\tau)))] \quad [E_5(S: kx, g, (x_2'), (x_2, \varepsilon_2)) \quad E_6(C: f_2(x_2+\sigma_2), (x_2, \sigma_2), (u_2(t))) \quad E_7(C: f_4(u_2(t)), (u_2(t)), (v_2(t))) \quad E_8(C:I(v_2(t)\cdot\delta(t-\tau));(v_2(t));(v_2(t+\tau)))] \quad E_9(C: f_5(v_1(t+\tau)+v_2(t+\tau)), (v_1(t+\tau), v_2(t+\tau)), (y(t))) \quad A(E) \quad (3.24)$$

$$M_2=A(B) \parallel [E_1(S: kx, g, (x_1'), (x_1, \varepsilon_1)) \quad E_2(C: f_1(x_1+\sigma_1), (x_1, \sigma_1), (u_1(t)))] \quad [E_3(S: kx, g, (x_2'), (x_2, \varepsilon_2)) \quad E_4(C: f_2(x_2+\sigma_2), (x_2, \sigma_2), (u_2(t)))] \quad E_5(C: f_6(u_1(t)), (u_1(t)), (u_1'(t))) \quad E_6(C: f_3(u_1'(t)), (u_1'(t)), (v_1(t))) \quad E_7(C: I(v_1(t) \cdot \delta(t-\tau)); (v_1(t)); (v_1(t+\tau))) \quad E_8(C: f_6(u_2(t)), (u_2(t)), (u_2'(t))) \quad E_9(C: f_3(u_2'(t)), (u_2'(t)), (v_2(t))) \quad E_{10}(C: I(v_2(t) \cdot \delta(t-\tau)); (v_2(t)); (v_2(t+2\tau))) \quad E_{11}(C: f_5(v_1(t+\tau) + v_2(t+2\tau)), (v_1(t+\tau), v_2(t+2\tau)), (y(t))) \quad A(E) \quad (3.25)$$

Перетворимо ці алгоритмічні моделі на операторні

$$b_1(y) = F_9(2, f_5) [F_4(n, g_\tau) [F_3(1, f_3) [F_2(2, +)[\delta(0), F_1(2, kx) [b(x_1')]]]]] , [F_8(n, g_\tau) [F_7(1, f_4) [F_6(2, +)[\delta(0), F_5(2, kx) [b(x_2')]]]]]], \quad (3.26)$$

$$b_2(y) = F_{11}(2, f_5) [F_7(n, g_\tau) [F_6(1, f_3) [F_5(1, f_6) [F_2(2, +)[\delta(0), F_1(2, kx) [b(x_1')]]]]]] , [F_{10}(n, g_{2\tau}) [F_9(1, f_3) [F_8(1, f_6) [F_4(2, +)[\delta(0), F_3(2, kx) [b(x_2')]]]]]]]. \quad (3.27)$$

Оцінимо кількість інформації за умов:

- час, виділений на перетворення одного вхідного сигналу у послідовній системі, - T_{II} ;
- похибки σ є нормальними з нульовим середнім;
- шум ξ є білим нормальним;
- вхідні сигнали x_1 і x_2 є стаціонарними нормальними процесами з автокореляційними функціями $R_{11}(\tau)$ і $R_{22}(\tau)$ і взаємнокореляційною функцією $R_{12}(\tau)$;
- усі перетворення лінійні (табл. 3.5).

Таблиця 3.5

Опис сигналів відповідно до рис. 3.12

паралельна та послідовна схеми апріорно	паралельна схема апостеріорно	послідовна схема апостеріорно
$u_1 = a_1 x_1 + b_1$ $u_2 = a_2 x_2 + b_2$ $v_1 = a_3 u_1 + b_3$ $v_2 = a_4 u_2 + b_4$ де $a_4 = a_3, b_4 = b_3$ $y = a_5 v_1 + b_5 v_2$	$u_1 = a_1 x_1 + b_1 + \sigma_1$ $u_2 = a_2 x_2 + b_2 + \sigma_2$ $v_1 = a_3 u_1 + b_3$ $v_2 = a_4 u_2 + b_4$ де $a_4 = a_3, b_4 = b_3$ $y = a_5 v_1 + b_5 v_2$	$u_1(t) = a_1 x_1(t) + b_1 + \sigma_1$ $u_2(t) = a_2 x_2(t) + b_2 + \sigma_2 + \xi$ $v_1(t - T_{II}) = a_3 u_1(t - T_{II}) + b_3$ $v_2(t) = a_3 u_2(t) + b_3$ $y = a_5 v_1(t - T_{II}) + b_5 v_2(t)$

1) Апріорна дисперсія

$$\sigma_{анр} = \sqrt{(a_3 a_1 \sigma_{x1})^2 + (a_3 a_2 \sigma_{x2})^2 - 2 a_3^2 a_1 a_2 \sigma_{x1} \sigma_{x2} r_{12}}. \quad (3.28)$$

2) Апостеріорно для паралельної схеми:

- Математичне сподівання вихідного сигналу буде таке ж, як апіорне значення

$$m_{y1} = a_5 a_3 a_1 m_{x1} + b_5 a_3 a_2 m_{x2} + a_5 a_3 b_1 + b_5 a_3 b_2 + b_5 b_3. \quad (3.29)$$

Дисперсія з урахуванням додаткової штучної невизначеності:

$$\sigma_{y1} = \sqrt{(a_3 a_1 \sigma_{x1} + a_3 \sigma_1)^2 + (a_3 a_2 \sigma_{x2} + a_3 \sigma_2)^2 - 2(a_3 a_1 \sigma_{x1} + a_3 \sigma_1)(a_3 a_2 \sigma_{x2} + a_3 \sigma_2) r_{12}}. \quad (3.30)$$

- Апостеріорно для послідовної схеми:

$$m_{y2} = a_5 (a_3 (a_1 m_{x1} + b_1) + b_3) + b_5 (a_3 (a_2 (m_{x2} + r_{12} \frac{\sigma_{x2}}{\sigma_{x1}} (x_1 - m_{x1})) + b_2) + b_3), \quad (3.31)$$

$$\sigma_{y2} = \sqrt{(a_3 a_1 \sigma_{x1} + a_3 \sigma_1)^2 + (a_3 a_2 \sigma_{x2} \sqrt{1 - r_{12}^2} + a_3 \sigma_2 + a_3 \sigma_\xi)^2 - 2(a_3 a_1 \sigma_{x1} + a_3 \sigma_1)(a_3 a_2 \sigma_{x2} \sqrt{1 - r_{12}^2} + a_3 \sigma_2 + a_3 \sigma_\xi) r_{12}}. \quad (3.32)$$

Вірогідність еквівалентності алгоритмічних моделей послідовної і паралельної систем (отримання від послідовної системи такого ж результату, як і від паралельної)

$$B = \int_{m_{y1} - \sigma_1}^{m_{y1} + \sigma_1} \beta(y_2) dy_2 = \int_{m_{y1} - \sigma_1}^{m_{y1} + \sigma_1} \frac{1}{\sigma_{y2} \sqrt{2\pi}} e^{-\frac{(y_2 - m_{y2})^2}{2\sigma_{y2}^2}} dy_2. \quad (3.33)$$

Відповідно ентропія

$$H_{анс 2} = -B \cdot \ln B - (1 - B) \cdot \ln(1 - B). \quad (3.34)$$

Аналіз залежності кількості інформації, яку забезпечують системи (рис. 3.12) в умовах невизначеності, від часу виконання операцій і дисперсії додаткового шуму показує, що при взятих для прикладу параметрах сигналів і швидкодії блока $0.15 < \tau < 0.25$ вірогідність співпадання результатів для систем (рис. 3.12, а, б) має максимум при сумарній дисперсії шуму $\sigma = 0.1$.

З аналізу ефективності системи випливає, що за певних умов послідовна система, яка може бути реалізована з меншою вартістю апаратних засобів, є

еквівалентною паралельній системі, причому ступінь еквівалентності може бути збільшений штучним введенням невизначеності.

Таким чином, загальна методика оптимізації структури ПЗД в умовах невизначеності передбачає:

1. На основі моделі технологічного процесу контролю і технічної документації визначаються варіанти векторів контрольованих параметрів, які дозволяють оцінити стан процесу оптимально з точки зору критерію (2.71);

2. До вектора контрольованих параметрів додається штучна компонента з невизначеною дисперсією;

3. Для кожного варіанта складається базова алгоритмічна модель ПЗД. Для цього:

3.1. Розв'язується задача оцінювання за критерієм (3.19) відносно невідомих параметрів стану;

3.2. Розв'язок подається у вигляді алгоритмічної моделі (за методикою підрозділу 2.2 з використанням бази алгоритмів (рис. 3.4);

3.3. На основі алгоритмічної моделі будується граф залежності даних;

3.4. На основі графу залежностей з використанням бази типових структур (рис. 3.4) генеруються варіанти послідовно-паралельних структур виконання збору даних;

3.5. Обирається елементна база первинних засобів збору даних з використанням бази елементів (рис. 3.4);

3.6. Враховуючи фізичні характеристики сигналів-носіїв даних і розташування елементів для кожного варіанта послідовно-паралельного виконання ПЗД генеруються додаткові варіанти програмної та апаратної реалізацій і в алгоритмічну модель вставляються оператори інтерфейсу та передавання даних;

3.7. Оцінюються характеристики отриманих варіантів структур: складність ПЗД (за методикою п. 2.4.1), невизначеність стану об'єкта (за методикою п. 2.4.2), швидкодії (за методикою п. 2.4.3), надійності (за методикою п. 2.4.4);

3.8. Розраховується комплексний показник ефективності (2.71) як функція від дисперсії штучного шуму і знаходиться максимум цієї функції;

3.9. Обирається базовий варіант моделі ПЗД, який характеризується найбільшим значенням максимуму показника ефективності;

4. За допомогою системи еквівалентних перетворень (п. 2.3.2) генеруються варіанти модифікації структури ПЗД;

5. Використовуючи базу знань, відбираються варіанти структур, які можуть бути фізично реалізовані;

6. За допомогою введеної у п. 2.4.6. метрики отриманий набір варіантів ранжують за зростанням.

7. Для кожного варіанта структури виконують етапи 3.7 – 3.9 і з усіх варіантів обирають структуру з найбільшим значенням максимуму показника ефективності.

Запропонована методика дозволяє автоматизувати пошук оптимальної структури ПЗД в умовах невизначеності.

3.3. Модель розвитку підсистеми збору даних

Необхідність розвитку ПЗД може бути спричинена багатьма обставинами. Виділимо серед них такі:

- ускладнення технологічного процесу (розширення виробництва, яке контролюється; поява нового технологічного обладнання);
- підвищення вимог до повноти і якості контролю;
- оновлення елементної бази ПЗД.

Відповідно, розвиток ПЗД (у випадку ускладнення технологічного процесу) відбивається на основних складових її моделі, зокрема збільшується розмірність вектора контрольованих параметрів

$$X''[n_0 + n_{\Delta}] = X''_0[n_0] \cup X''_{\Delta}[n_{\Delta}], \quad (3.35)$$

де $X''_0[n_0]$ – вектор контрольованих параметрів початкової системи;

$X''_{\Delta}[n_{\Delta}]$ – вектор додаткових контрольованих параметрів і розмірність вектора оцінюваних параметрів технологічного процесу

$$X'[k_0 + k_{\Delta}] = X'_0[k_0] \cup X'_{\Delta}[k_{\Delta}], \quad (3.36)$$

де $X'_0[k_0]$ – вектор параметрів початкової системи, які оцінювалися;

$X'_{\Delta}[k_{\Delta}]$ – вектор додаткових оцінюваних параметрів.

Загальний вектор параметрів є об'єднанням цих векторів

$$\begin{aligned} X[N] &= X''[n_0] \cup X''[n_{\Delta}] \cup X'[k_0] \cup X'[k_{\Delta}] = (X''[n_0] \cup X'[k_0]) \cup (X''[n_{\Delta}] \cup X'[k_{\Delta}]) = \\ &= X_0[N_0] \cup X[N_{\Delta}] \end{aligned} \quad (3.37)$$

В результаті загальна модель (2.2) матиме вигляд

$$\begin{cases} X'_0[k_0] = F_0^{-1}(X''_0[n_0]), \\ X'_{\Delta}[k_{\Delta}] = F_{\Delta}^{-1}(X''_0[n_0] \cup X''_{\Delta}[n_{\Delta}]), \end{cases} \quad (3.38)$$

де друге рівняння характеризує додаткові функціональні задачі ПЗД.

Граф залежностей даних буде складатися з двох графів: графу базової системи і додаткового графа, побудованого на основі залежності F_{Δ}^{-1} . Ці графи мають спільну частину, тому виділимо підграф доповнення базової системи

$$G_{\Delta}(OP, X'', X_0) = (G_0(F_0^{-1}, X''_0, X'_0) \cup G(F_{\Delta}^{-1}, X''_0 \cup X''_{\Delta}, X'_{\Delta})) \setminus G_0(F_0^{-1}, X''_0, X'_0). \quad (3.39)$$

На основі отриманого графа доповнення будується оптимальна алгоритмічна модель доповненої ПЗД.

3.4. Алгоритм оптимізації підсистеми збору даних

Алгоритм оптимізації ПЗД деталізує методику, сформульовану у підрозділі 3.2.

Зобразимо алгоритм оптимізації ПЗД у вигляді алгоритму, який показує діяльність об'єкта :ПЗДОптимНевиз, що реалізує алгоритм пошуку оптимальної структури ПЗД. Алгоритм зображений на рис. 3.13.

Алгоритм призначений для автоматизованої оптимізації структури ПЗД на стадії експлуатації. Тут, на відміну від стадії проектування, передбачається, що певна ПЗД вже існує, її структуру називатимемо поточною. Цій структурі відповідає алгоритмічна модель, яку теж називатимемо поточною.

Необхідність оптимізації поточної структури виникає у двох випадках:

- у випадку розвитку ПЗД через ускладнення технологічного процесу, підвищення вимог до даних, оновлення елементної бази;
- у випадку усунення наслідків відмов обладнання, приведення структури у відповідність до спрощення технологічного процесу.

В результаті поточна модель відрізнятиметься від базової моделі, тобто від тої моделі, яка була оптимізована і реалізована на стадії проектування. Виявлення такої різниці є умовою початку процедури оптимізації.

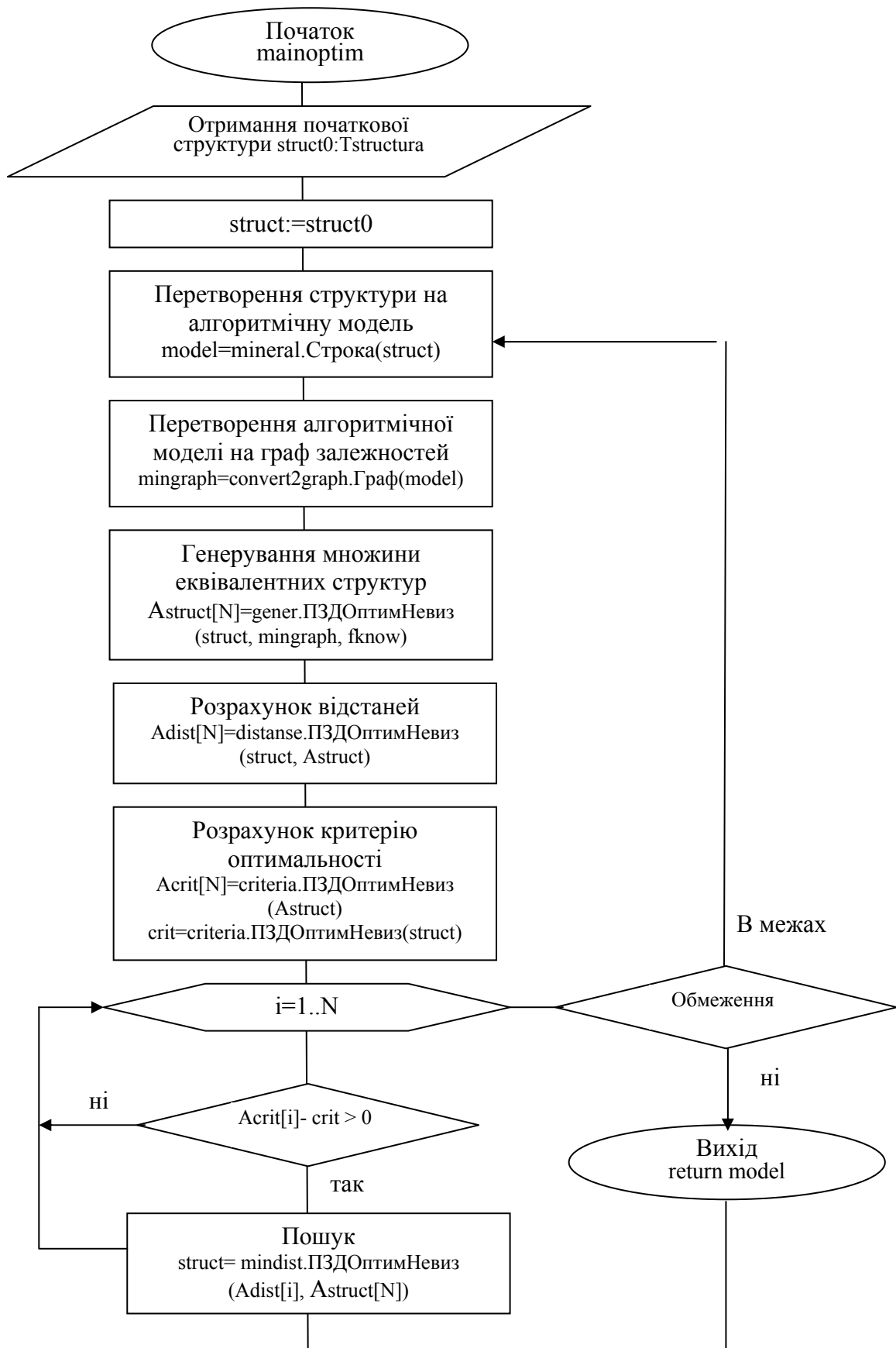


Рис. 3.13. Алгоритм оптимізації ПЗД

Головною частиною алгоритму оптимізації є покроковий пошук оптимального варіанта структури ПЗД. Оскільки критерій оптимальності у загальному випадку нелінійний, а кожна зміна структури ПЗД пов'язана з витратами, то для мінімізації витрат бажано зменшити кроки пошуку. Для цього застосуємо метрику у нормованому просторі структур за п. 2.4.6 і розрахуємо норму поточної моделі. Отримані значення змінних :Модель і :Норма є початковими для пошуку оптимальної структури.

Генерування множини еквівалентних структур зображено у вигляді чотирьох піддіяльностей перебору варіантів (рис. 3.14), одна з яких показана на рис. 3.15.

У цій піддіяльності генерування здійснюється шляхом заміни двох послідовних операцій на паралельні. У інших піддіяльностях крок генерування будується аналогічно, але використовує заміни паралельних операцій на послідовні, програмних – на апаратні і навпаки. Можливість реалізації відповідної заміни і правила її виконання перевіряються по базі знань (див. методику п. 3.2.1 і рис. 3.10).

З отриманих варіантів відкидаються такі, що вже розглядалися, а з тих, що залишилися, обирається найкращий за заданим критерієм оптимізації. Отриманий варіант береться за поточний і процес повторюється до тих пір, поки після генерування і відсіювання не залишиться нерозглянутих варіантів.

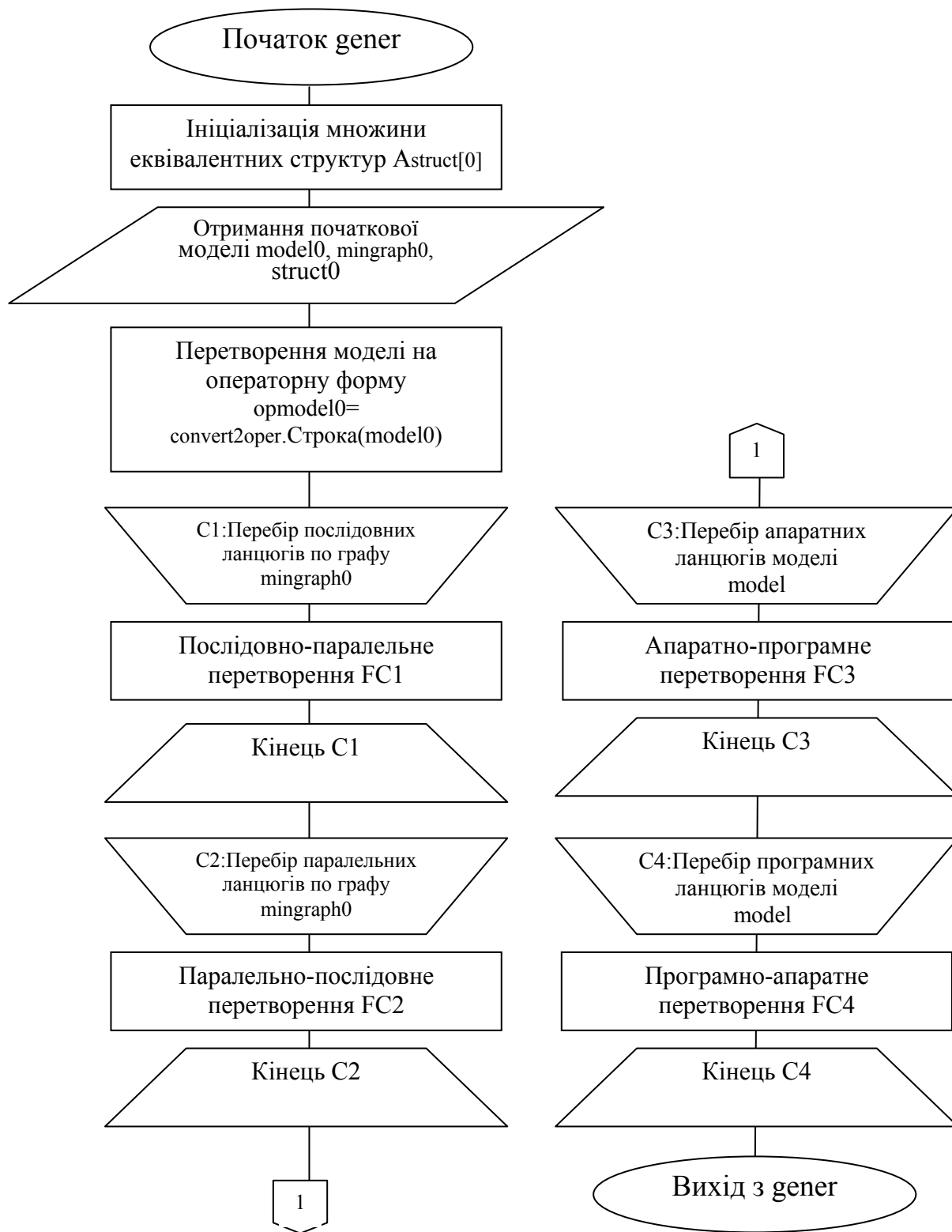


Рис. 3.14. Генерування множини еквівалентних структур

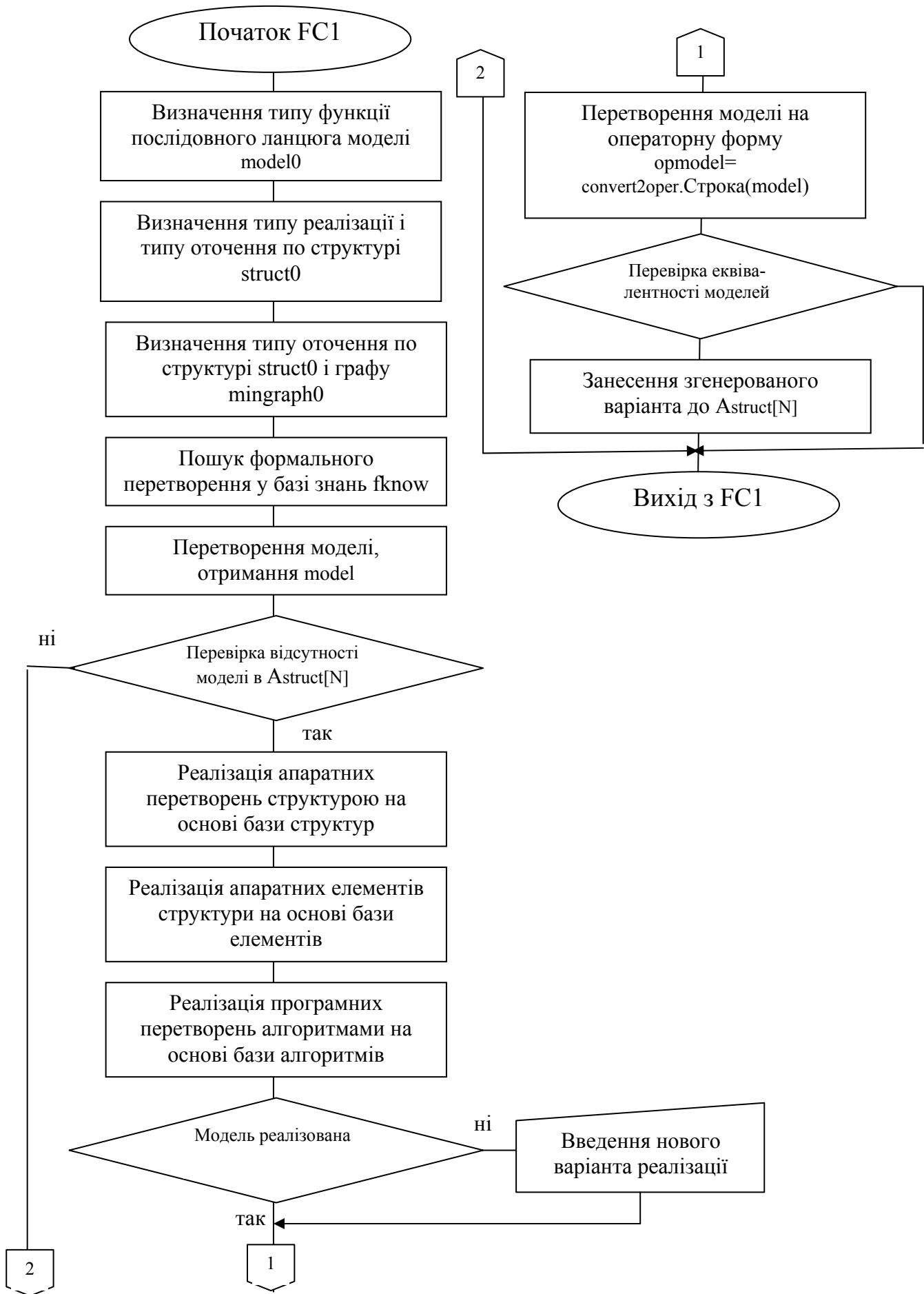


Рис. 3.15. Послідовно-паралельне перетворення FC1

4. ПРАКТИЧНА РЕАЛІЗАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Результати моделювання та розроблені алгоритми, отримані в попередніх розділах, є основою інформаційної технології оптимізації структур ПЗД в умовах невизначеності.

Інформаційна технологія реалізована у вигляді пакету програм і даних «Checkout Subsystem Optimizer», які розвивають інформаційну технологію автоматизованого проектування, реалізовану у системі Trace Mode. Розвинута технологія дозволяє здійснювати оптимізацію структури ПЗД АСУТП на основі інформаційно-вартісного критерію ефективності.

Перед застосуванням ІТ та розробкою проекту в SCADA-системі, необхідно вивчити технологічний процес і скласти проектну документацію по його автоматизації. Основою для цього є функціональна схема автоматизації технологічного процесу. В розробці проектної документації приймають участь технолог, інженер контрольно-вимірювальних приладів та апаратури та розробник АСУТП на базі SCADA-системи. Ці фахівці, вивчивши технологічний процес, заповнюють таблиці з такими даними: дані по технологічним параметрам об'єкту, дані по сигналам (вид сигналу, тип сигналу, коефіцієнт множення, дрейф нуля); дані по об'єктам АСУТП (ім'я об'єкту, ім'я каналу, число біт (для дискретних каналів) та інш. Ці дані використовуються для розробки базової функціональної схеми структури підсистеми збору даних.

Наступним етапом є застосування ІТ оптимізації структури ПЗД до базової структури ПЗД і оцінювання інформаційно-вартісного показника ефективності. При цьому на основі розробленої функціональної схеми базової ПЗД створюється алгоритмічна модель за допомогою бази даних елементарних структур, алгоритмів та елементів ПЗД. Далі вона перетворюється на операторну модель для врахування невизначеності. Використовуючи створену методику оптимізації операторна модель ПЗД оптимізується. Ця процедура ґрунтується на генеруванні кроку пошуку, генеруванні варіантів структур ПЗД на основі еквівалентних перетворень моделі, оцінки критерію ефективності кожної з структур та вибору оптимальної з них, враховуючи метрику у просторі структур, розглянуту в 2 розділі.

4.1. Розробка програмного забезпечення інформаційної технології оптимізації підсистем збору даних АСУТП

Задача програмного забезпечення «Checkout Subsystem Optimizer» – на основі початкової інформації щодо підсистеми збору даних про стан технологічного процесу та наявної апаратної та програмної бази формалізувати початкову структуру ПЗД у вигляді алгоритмічної моделі (АМ), графічно зобразити та оптимізувати цю структуру з точки зору інформаційно-вартісного критерію. Програмне забезпечення дозволяє оцінювати деякі характеристики ПЗД на основі АМ та перетворювати АМ на операторну модель, що враховує невизначеність.

Перелік основних функцій системи:

- Створення моделі структури на основі існуючих баз даних елементів, підсистем, алгоритмів;
- Збереження проекту ПЗД у вигляді алгоритмічної моделі, що описує структуру ПЗД, у файл і відкриття збереженого файлу;
- Редагування синтезованої алгоритмічної моделі (структури ПЗД): додавання, видалення, переміщення підсистем (компонентів структури ПЗД);
- Перетворення алгоритмічної моделі на операторну і навпаки;
- Додавання нових записів у існуючі БД елементів, структур та алгоритмічних моделей.
- Оцінювання характеристик алгоритмічної моделі синтезованої структури на основі операторної моделі;
- Оптимізація базової структури ПЗД на основі розглянутого критерію оптимізації та обмежень.

Деякі функції програмного забезпечення показані за допомогою UML-діаграм.

Діаграма послідовності для процесу оптимізації структури ПЗД зображена на рис. 4.1. Об'єкти, які беруть участь в процесі оптимізації, є екземплярами конкретних класів. Головний фокус керування належить анонівному об'єкту :ПЗДОптимНевиз. Життєвий цикл даного об'єкту зображений як рекурсивний, оскільки процес дискретної оптимізації – ітераційний, а ітерація є окремим випадком рекурсії. Рекурсія починається з виклику функції пошуку нової структури ПЗД, яка має найменшу різницю з поточною структурою і задовольняє обмеження.

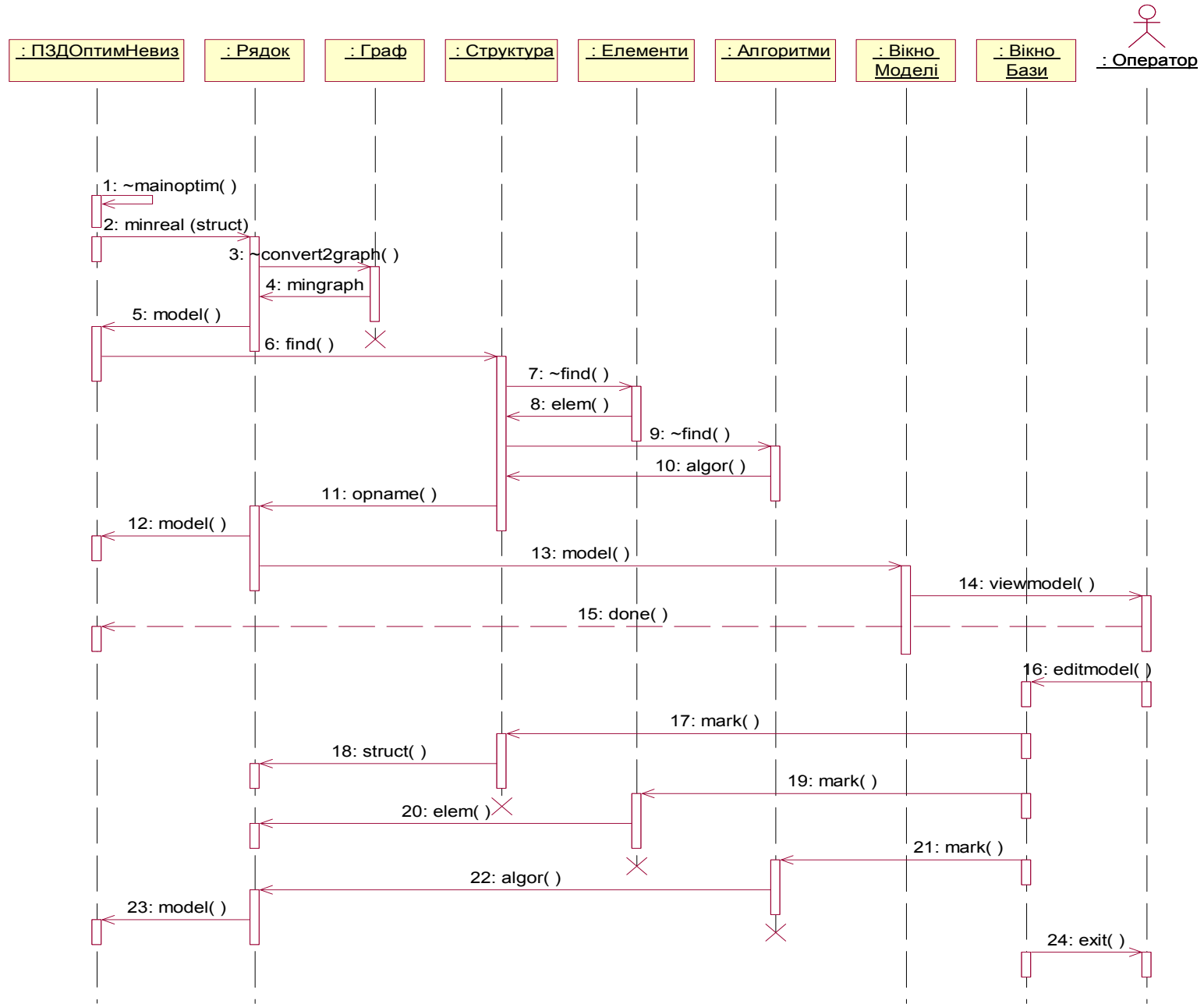


Рис. 4.1. Діаграма послідовності для процесу оптимізації

На першому кроці рекурсії поточною структурою є реально існуюча структура ПЗД, тобто задається початкове значення змінної `struct`, яка містить опис структури ПЗД: `struct=struct0` (див. діаграму кооперації рис. 4.2) Далі об'єкт `:ПЗДОптимНевиз` звертається до об'єкту `:Строка` для отримання моделі ПЗЖ. Об'єкт `:Строка` в свою чергу використовує об'єкт `:Граф`, з допомогою якого генеруються варіанти розпаралелювання виконання операцій збору і обробки даних.

На основі отриманої моделі `model` у вигляді текстового рядка і графа об'єкт `:ПЗДОптимНевиз` генерує декілька варіантів структур `struct`, які відрізняються від вихідної структури на одну елементарну зміну і задовольняє обмеження. При генеруванні варіантів структур перевіряється еквівалентність їх алгоритмічних моделей. Врахування умов невизначеності розширює поняття еквівалентності, що дозволяє збільшувати кількість можливих варіантів.

Для цього об'єкт `:ПЗДОптимНевиз` звертається до об'єкту `:Структура`, який перевіряє можливість реалізації зміни структури у вигляді комбінації варіантів структур, які зберігаються в базі даних. Об'єкт `:Структура` повертає вже готове рішення – з вказівкою можливих технічних засобів мінімальної вартості і відповідних алгоритмів обробки даних. Для цього він звертається відповідно до об'єктів `:Елемент` і `:Алгоритм`. Структура перетворюється об'єктом `:Строка` в модель, яка повертається основному об'єкту `:ПЗДОптимНевиз` і відображується у вікні інтерфейсу оператора об'єктом `:Вікно Моделі`.

Залежності між компонентами показані на діаграмі кооперації рис. 4.2. Для реалізації цих залежностей більшість компонентів має інтерфейси, які містять функції, що на діаграмі кооперації викликаються повідомленням із стереотипом `<<call>>`.

Розглянемо варіант використання інформаційної технології для оптимізації структури системи в автоматизованому режимі (з участю оператора). У цьому режимі вікно бази даних як активний об'єкт використовується оператором лише у разі потреби скоректувати результати автоматичного підбору варіантів. Взаємодіючими об'єктами є екземпляри конкретних класів. Активними об'єктами є вікно інтерфейсу оператора, через яке здійснюється управління процесом оптимізації структури ПЗД, і об'єкт, що здійснює контроль структури системи.

Логічним початком процесу оптимізації структури служить повідомлення від об'єкту «Контроль структури» про зміну умов роботи ПЗД (вихід з ладу окремих елементів, підключення нового устаткування і так далі).

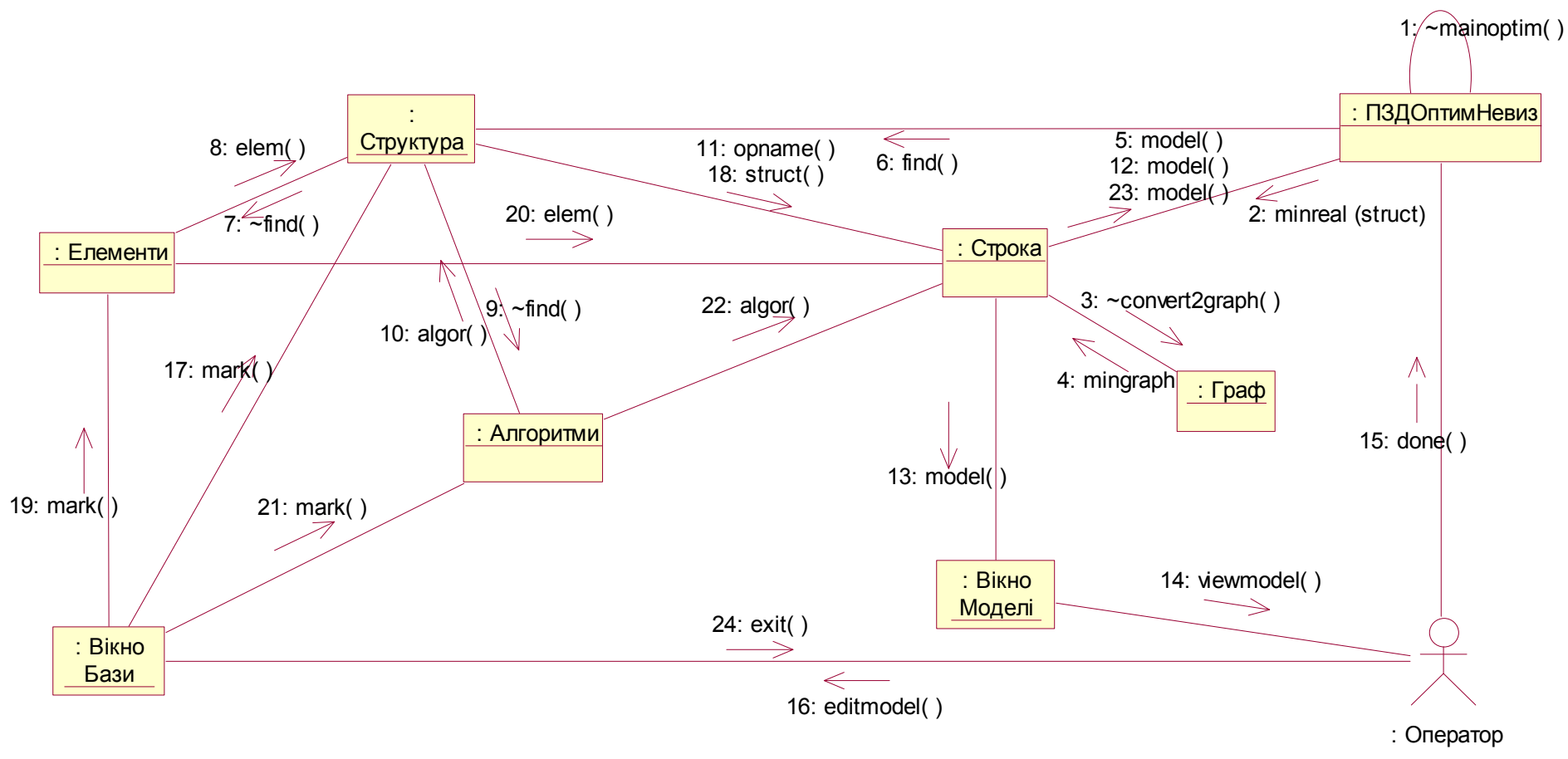


Рис. 4.2. Діаграма кооперації

Повідомлення викликає відображення відповідних змін у вікні реалізації. Прореагувавши на це повідомленням 2, оператор запускає через «Вікно оптимізації» процес оптимізації структури. Для цього за допомогою повідомлення 3 викликається основний алгоритм оптимізації, що реалізовується об'єктом «ПЗДОптимНевиз».

В процесі оптимізації покроково змінюється структура ПЗД і отримується варіант, що забезпечує максимум критерію оптимізації з врахуванням обмежень. Початковою точкою цього процесу служить поточна структура `struct0`. Для цього використовується модель системи у формі рядка символів. В процесі вибору кроку зміни структури розглядаються варіанти послідовного/паралельного і програмного/апаратного виконання процесів. Для цього використовується граф залежності сигналів об'єкту «Граф» і варіанти структур, алгоритмів і технічних засобів «Бази даних». При необхідності оператор може втручатися в процес вибору варіантів в «Базі даних» і виконання кроку зміни через «Вікно Бази» і «Вікно Моделі».

Діаграма станів об'єкта :Строка, життєвий цикл якого показаний на діаграмі послідовності, наведена на рис. 4.3. Аналіз життєвого циклу об'єкта :Строка на діаграмі послідовності показує, що цей об'єкт має 4 етапи активності, між якими розташовані проміжки очікування. Під час перших двох проміжків активності за допомогою об'єкта :Строка вирішується завдання одержання моделі вихідної ПЗД з використанням наявних у ній технічних і програмних засобів (`task=1`). При цьому об'єкт :Строка спочатку переходить у стан "Створення алгоритмічної моделі" і при виході із цього стану звертається до об'єкта :Граф для одержання від нього графа залежності даних. Потім об'єкт :Строка переходить у стан "очікування", а потім у стан "одержання відповіді" від об'єкта :Граф. Після одержання графа об'єкт переходить у стан "Створення моделі", у якому формується комплексна оптимізована модель, що включає інформацію про програмно-апаратні засоби й залежності даних. Під час третього проміжку активності (`task=2`) об'єкт :Строка відразу переходить у стан "Створення моделі", оскільки граф залежності вже сформований. Вихідні дані (структура системи, склад апаратних і програмних засобів) при потраплянні в цей стан об'єкт :Строка одержує від об'єкта :Структура. Під час четвертого проміжку активності (`task=3`) об'єкт :Строка до переходу в стан "Створення моделі" попадає в стан "Збір структури", оскільки вихідні дані про структуру системи і її компоненти надходять роздільно (після вибору їх оператором "вручну" з бази даних).

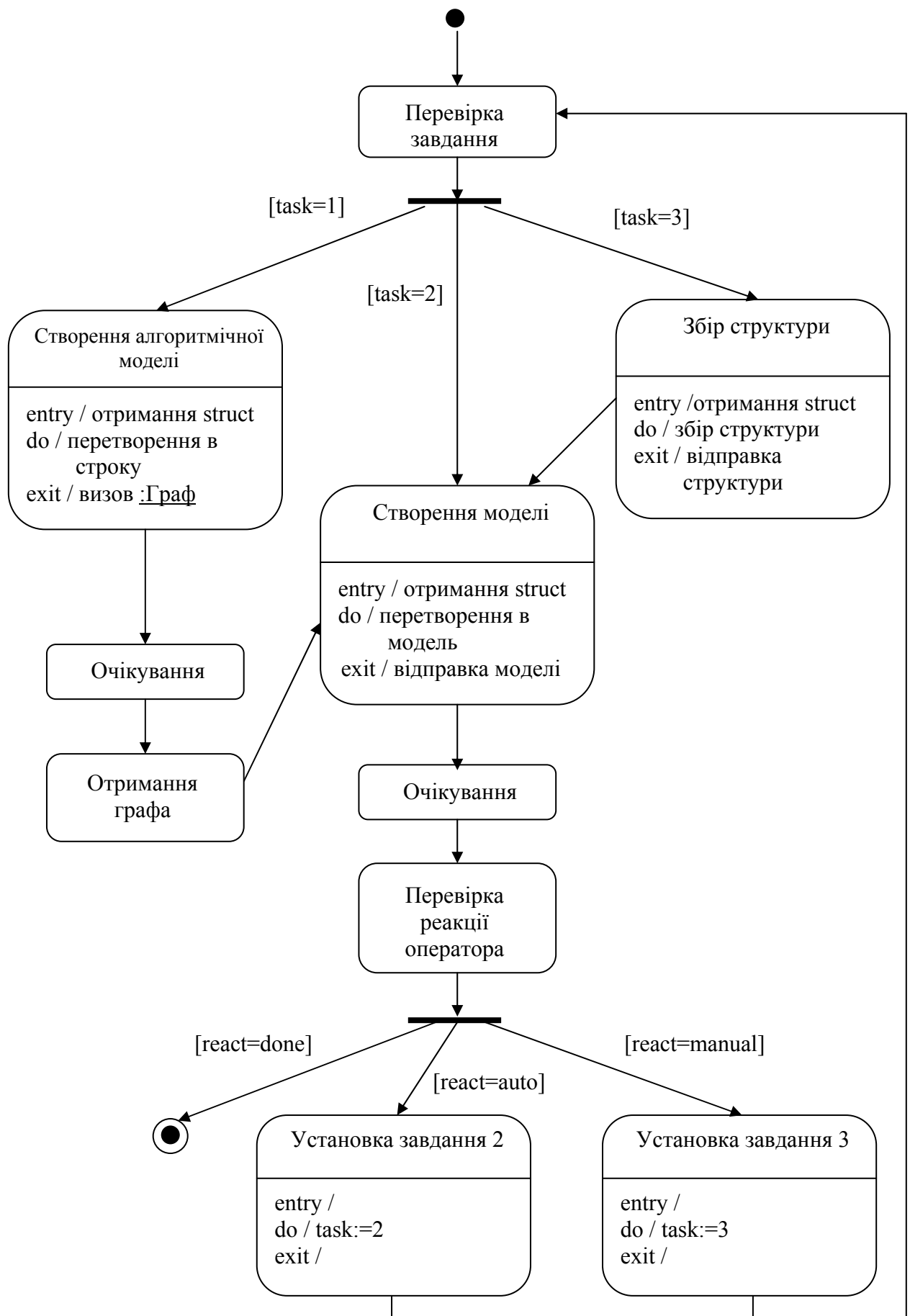


Рис. 4.3. Діаграма станів об'єкта :Строка

Наприкінці життєвого циклу перевіряється реакція оператора, що визначає, чи закінчений процес моделювання, або його необхідно продовжити.

Діаграма компонентів програмного забезпечення інформаційної технології оптимізації структури ПЗД показана на рис. 4.4.

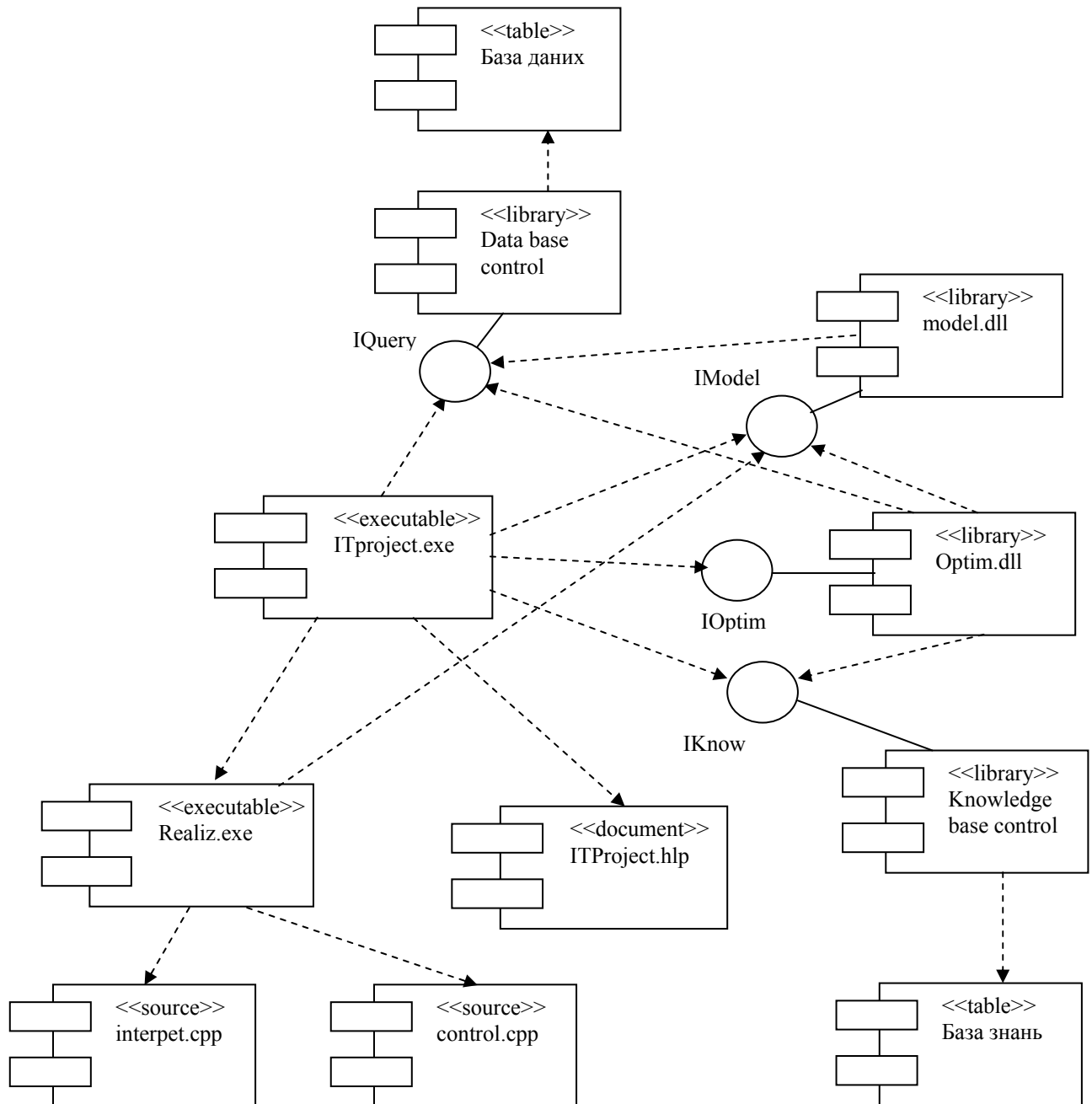


Рис. 4.4. Діаграма компонентів програмного забезпечення інформаційної технології

Кожен з компонентів реалізує окремий клас з усіма його підкласами, які відповідають діаграмі класів. Діалоговий віконний інтерфейс оператора реалізується виконуваним файлом ITproject.exe, решта у вигляді динамічних

бібліотек і файлів даних.

Для створення БД елементів та алгоритмів ПЗД використовувалось середовище Microsoft SQL Server Management Studio Express. Загальна діаграма бази даних зі зв'язками між відношеннями наведена на рис. 4.5.

Головне меню програми «Checkout Subsystem Optimizer» має структуру, наведену в табл. 4.1.

Інтерфейс програмного забезпечення інформаційної технології з вікном для редагування схеми ПЗД в створеному проекті наведено на рис. 4.6.

Для редагування алгоритмічних моделей структур ПЗД та оцінки їх характеристик в програмному забезпеченні «Checkout Subsystem Optimizer» було створено редактор «Editor model», де для опису алгоритму роботи ПЗД використовується запропонована в 2 розділі система алгоритмічних моделей (АМ) програмних та апаратних засобів ПЗД [114, 119].

Для введення операторів алгоритмічних моделей, їх параметрів розроблено інтерфейс вводу-виводу алгоритмічних моделей. Розроблений інтерфейс є багатовіконним і дозволяє змінювати та видаляти оператори в алгоритмічній моделі, передбачена можливість переходу по операторах та їх додавання. Інтерфейс користувача даного модулю показано на рис. 4.7. Редактор АМ виконує такі функції:

- редагування алгоритмічної та операторної моделей;
- збереження алгоритмічної та операторної моделей на диск;
- додавання операторів в алгоритмічну модель у вигляді окремої форми для введення параметрів обраного оператора (рис. 4.8);
- заміна одного оператора на інший;
- видалення одного або усіх операторів в алгоритмічній моделі;
- додавання нового оператора в базу даних «DataCollectionSystem», структура якої наведена нижче;
- видалення оператора з бази даних;
- редагування оператора в базі даних;
- надання інформації про кількість операторів в моделі і порядковий номер поточного оператора;
- підрахунок складності операторної моделі.

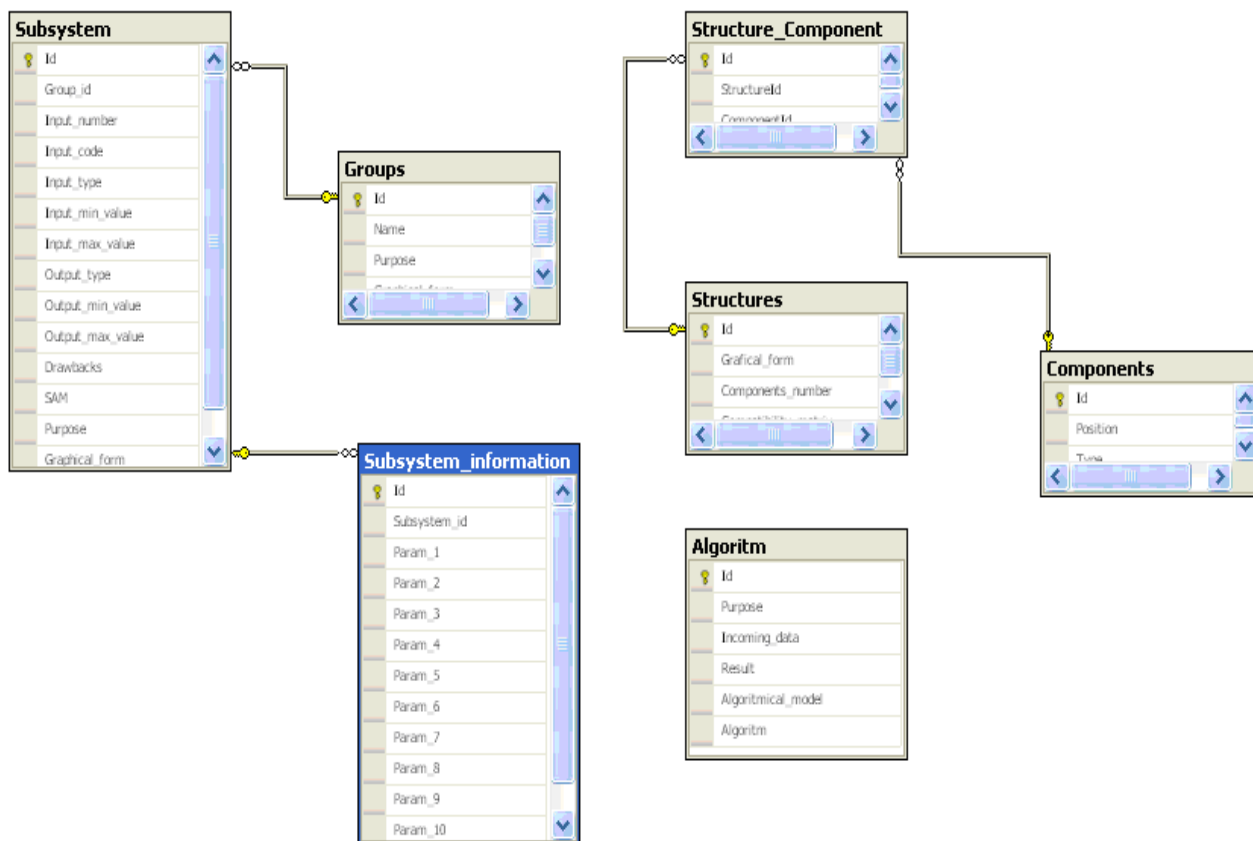


Рис. 4.5. Схема даних бази даних

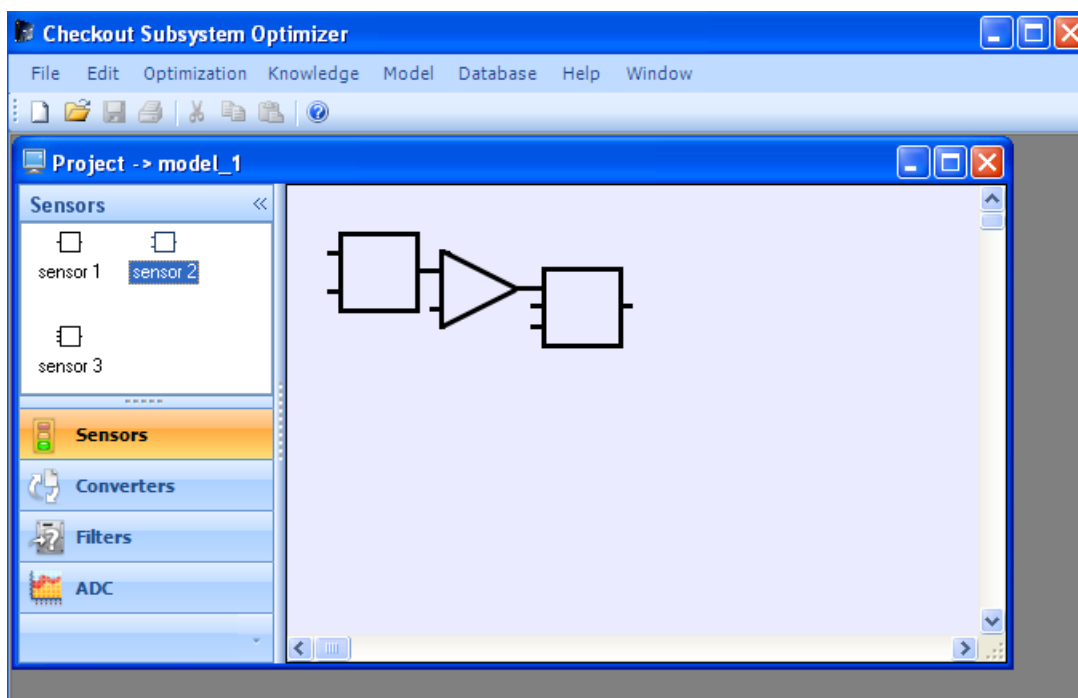


Рис. 4.6. Робочий інтерфейс програми

Меню програми «Checkout Subsystem Optimizer»

Пункт меню	Зміст
1. Project	Ініціалізація проекту
New	Створення, збереження і завантаження алгоритмічної моделі
Open	
Save	
Save as	
Export to Trace Mode	Зв'язок даних з Trace Mode
Import from Trace Mode	
Make report	Створення звіту
2. Edit	Редагування алгоритмічної моделі
Clear project	Видалення проекту
Add (Subsystem або Connection)	Додавання, видалення та переміщення підсистеми чи зв'язку
Delete (Subsystem або Connection)	
Move (Subsystem або Connection)	
3. Model	
Algorithm	Перетворення алгоритмічної моделі на операторну та графову форму
Operator	
Graph	
Calculate	Оцінювання параметрів моделі.
4. Optimization	Оптимізація моделі
Conditions	Введення обмежень та критерію оптимізації
Proceed	
5. Database	Робота з Базою Даних елементів, структур та алгоритмів
Elements	Вибір елементів-підсистем, збережених структур збору даних та алгоритмів
Structures	
Algorithms	
6. Knowledge	Робота з Базою знань
Structure analyses	Аналіз структур та зв'язків
Realization analyses	
7. Help	Допомога у користуванні програмою
About	Інформація про програму та довідкова інформація
Index	

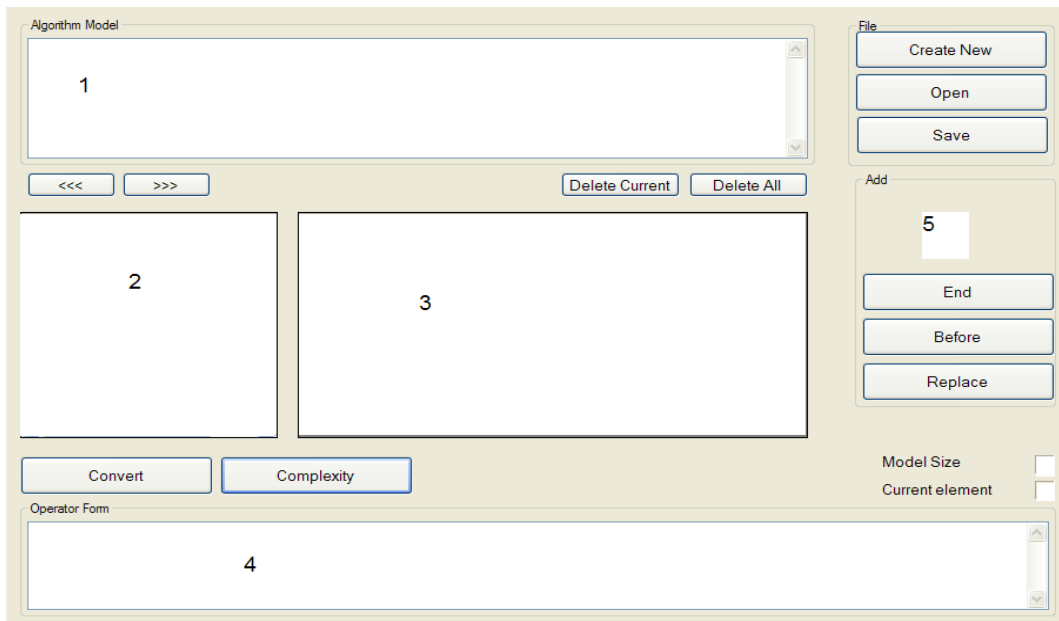


Рис. 4.7. Інтерфейс модуля «Editor model»

1 – поле призначене для введення алгоритмічної моделі; 2 – поле призначене для вибору оператора з таблиці БД; 3 – поле призначене для вибору операції з таблиці БД; 4 – поле призначене для відображення операторної моделі; 5 – поле призначене для відображення номера оператора, який вибраний в даний

Рис. 4.8. Вікно додавання нового оператора

База даних «*DataCollectionSystem*» містить три відношення Realizations, Operators, Operations.

Відношення Operators містить список усіх стандартних операторів, і до

кожного з них введена алгоритмічна та операторна модель заміщення.

Відношення Operations містить список операцій, які проводяться над певним технологічним процесом. Також воно містить посилку на відношення Realizations, де вказується чи це програмна чи апаратна реалізація операції.

Алгоритм перетворення алгоритмічної моделі в операторну форму

(формування ланцюга послідовностей для перетворення в операторну форму починається з нижнього правого кута матриці інциденції):

1. Ввести алгоритмічну модель.
2. Підрахувати кількість операторів, для того, щоб визначити розмір матриці суміжності.
3. Переглянути усі оператори та сформуванати матрицю суміжності, на якій показано, як пов'язані між собою оператори моделі. За кількістю одиниць – визначаємо кількість дуг, які з'єднують всю модель.
4. Формуємо матрицю дуг, на якій показано, послідовність сполучення операторів, тобто, вихідні дані якого з операторів є вхідними для іншого.
5. Формуємо матрицю інциденції, на якій показано всю необхідну інформацію про сполучення операторів.
6. Вводимо допоміжну матрицю кількості рядків якої така ж сама як і в матриці інциденції, а кількість стовпців дорівнює трьом. Перший стовпець відображатиме чи знайдені в рядку тільки -1, другий стовпець відповідає за кількість знайдених -1, третій рядок – за те, чи вся стрічка дорівнює нулю.
7. В циклі перевіряємо усі стрічки матриці інциденції і якщо рядок містить тільки -1, то підраховуємо їх кількість та помічаємо відповідний рядок першого стовпця 1, а другий – кількістю знайдених мінус одиниць. Якщо рядок містить тільки нулі, то помічаємо відповідний рядок третього стовпця 1, якщо ж рядок містить мішані дані, то записуємо лише кількість мінус одиниць до другого стовпця того ж рядка.
8. В циклі починаємо формувати ланцюг послідовностей операторної моделі. Кінцями вершин будуть ті рядки, які містили тільки мінус одиниці в рядку, або тільки нулі – тоді ланцюг складатиметься тільки з одного оператора. Якщо ж рядок містить як мінус одиниці так і одиниці, то це означає, що на графі в цю вершину входять багато дуг і вона є тим проміжним ланцюжком переходу на інші рівні. Формування ланцюга завершується тоді, коли в рядку де була знайдена одиниця – не буде

знайдено мінус одиниці. Якщо ж дані в рядку перемішані і там є декілька мінус одиниць, то після використання однієї з них, ми повинні її обнулити, щоб дати можливість продовження ланцюга в альтернативних напрямках.

9. Зі сформованих ланцюгів послідовностей формуємо строку операторної форми, використовуючи формат даних взятий із БД «DataCollectionSystem».

Розроблена програма [122] дозволяє відкрити для редагування та зберігати створену алгоритмічну модель системи у файл, який може імпортуватись головною програмою «Checkout Subsystem Optimizer» для подальшої оптимізації базової структури та визначення найкращого варіанту оптимальної структури за методикою наведеною в третьому розділі.

4.2. Експериментальні дослідження ефективності інформаційної технології

Ефективність розвинутої інформаційної технології перевірена на практиці в процесі оптимізації підсистеми збору даних системи управління технологічним процесом виготовлення консервів.

Технологічний процес (ТП) теплової обробки консервів є одним з найважливіших етапів виготовлення консервів, оскільки від нього залежать якісні показники та смак кінцевої продукції. Тому важливою є задача підвищення ефективності технологічного процесу теплової обробки консервів, зокрема оптимізації структури ПЗД та швидкого збору контрольованих та керованих параметрів.

ТП теплової обробки консервів в першу чергу залежить від температурного графіку роботи автоклавів, в яких проходить процес. Більшість сучасних автоклавів працюють на основі пари, яка нагріває воду в автоклаві. Тому важливо контролювати такі параметри, як тиск, температура та щільність пари і води в автоклавах та на виході з котла, який подає пар. Крім того в процесі теплової обробки в автоклаві обов'язково протоколюється температурний графік роботи автоклаву і встановлені норми витрат теплоносія й повернення конденсату та режимні графіки [2, 85, 137]. Одним з важливих аспектів виготовлення

консервів є проблема економії теплоносія.

АСУ ТП теплової обробки консервів передбачає збір та облік великої кількості контрольованих та регульованих параметрів, що зумовлює необхідність розв'язання задачі оптимізації структури підсистеми збору даних з метою економії енерговитрат та зменшення кількості браку продукції.

Розв'язання цієї задачі може бути досягнуте за рахунок впровадження додаткових давачів, регуляторів, контролерів, блоку управління, збільшення частоти збору даних, розпаралелювання деяких процесів в ПЗД [85, 170].

АСУТП теплової обробки консервів функціонує в умовах невизначеності, що зменшує ефективність цих систем. Великий вплив на втрати від браку продукції здійснюють різноманітні відхилення від планових режимів в роботі системи, пов'язаних з виходом з ладу обладнання, відключенням світла, невчасним ручним керуванням (запізнення реагування), малою частотою контролю та збору даних, невизначеністю певних параметрів.

Контроль за технологічним процесом здійснюється оператором (за допомогою показів манометрів і термометрів), що призводить до збільшення невизначеності результату функціонування АСУТП.

В існуючій системі керування за допомогою пристроїв підсистеми збору даних вся необхідна поточна інформація про стан технологічного процесу передається в інформаційно-керуючу частину системи. Вимірювання температури здійснюється за допомогою терморезистивних давачів. Вимірювання тиску здійснюється безперервно за допомогою давачів тиску з уніфікованим сигналом від 4 до 20 мА. Сигнали з усіх давачів передаються на уніфіковані входи контролера. Для зв'язку між контролером та модулем розширення виходів та між контролером та панеллю оператора, яка відображує інформацію про технологічний процес використовується інтерфейс RS-485. Керування виконавчими пристроями для збільшення точності регулювання здійснюється з допомогою клапанів мембранного виконавчого механізму (МВМ). Оскільки виконуючі механізми пневматичні, для перетворення електричного сигналу (4-20 мА) в пневматичний (0,02-0,1 МПа) в системі використовуються електропневматичні перетворювачі (ЕПП).

Забезпечення автоклавів паром здійснюється за допомогою парового котла. У системі паропостачання до автоклаву передбачене регулювання подачі теплоносія на водопідігрівник. Оператор здійснює контроль температури, тиску, щільності пару на виході з котла та на вході в автоклав. Крім цього контролюється рівень витрат холодної води та газу на вході в котел. Оператор здійснює керування цими параметрами для забезпечення належного проходження технологічного процесу при мінімізації витрат ресурсів та сумарних витрат та втрат від браку продукції.

Структура АСУТП варки консервів показана на рис. 4.9.

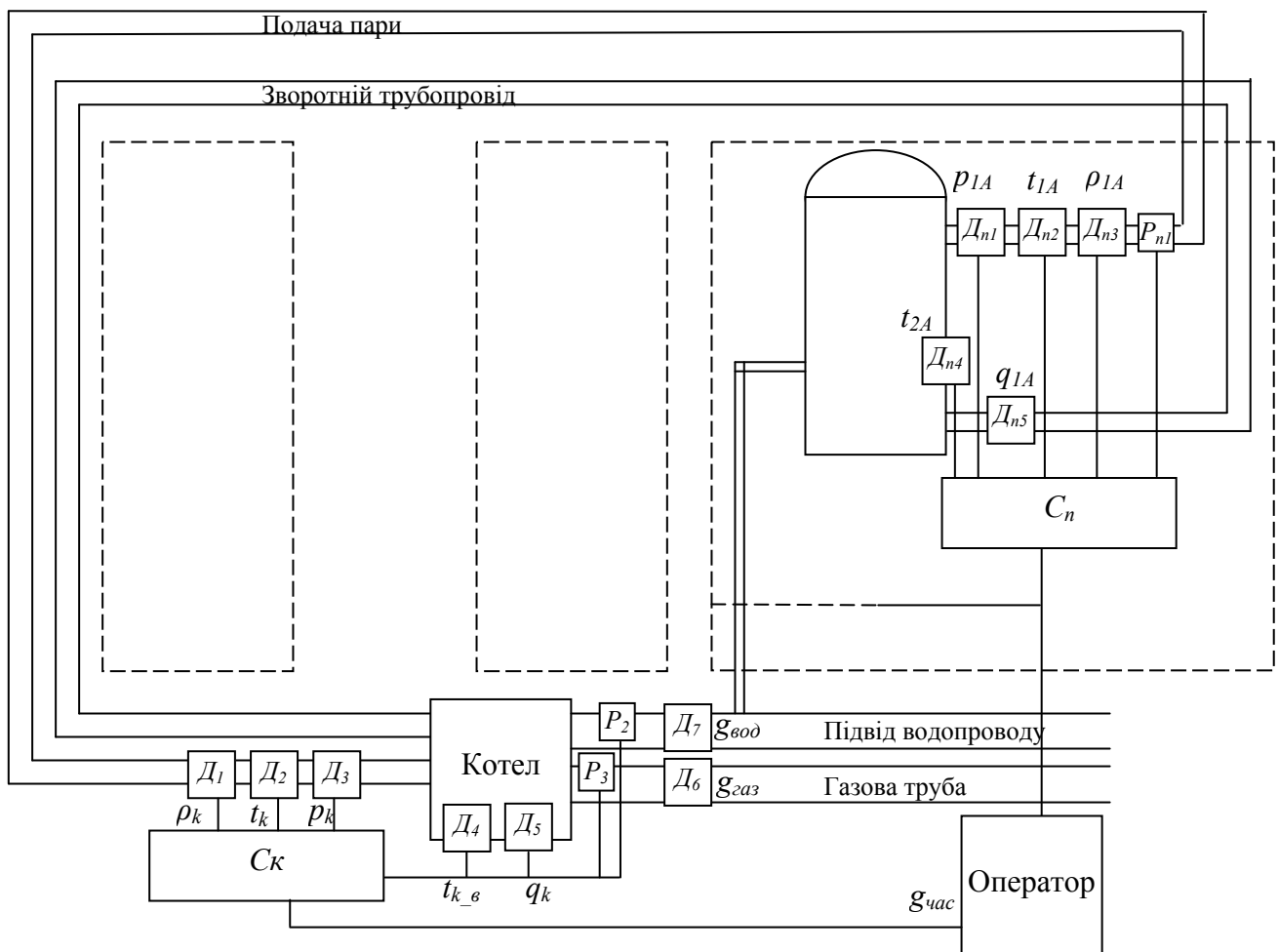


Рис. 4.9. Структурна схема АСУ ТП теплової обробки консервів

При виробництві рибних консервів можна виділити такі витрати:

- а) витрати води – $g_{вод}$;
- б) витрати газу (тепла) – $g_{газ}$;
- в) витрати часу – $g_{час}$.

Керування здійснюється за допомогою контролерів C_i , які

управляють регуляторами подачі пари в автоклави та регулятором подачі газу для нагрівання води в котлі. На входи контролерів поступає інформація з давачів $D_{1n} - D_{5n}$ (де n – кількість автоклавів) та $D_1 - D_5$ (давачі котла) про параметри, які необхідні для забезпечення обліку та керування системою. По кожному вводу в автоклав параметри система контролює такі параметри: p_{1A} – тиск пари на вході в автоклав; t_{1A} – температуру пари на вході в автоклав; ρ_{1A} – щільність пари на вході в автоклав; t_{2A} – температуру пари в автоклаві; q_{1A} – кількість конденсату на виході з автоклаву, який повертається в систему по зворотному трубопроводу.

Окрім цього, оператор контролює такі параметри на виході з котла: p_k – тиск пари; t_k – температуру пари; ρ_k – щільність пари; q_k – кількість води в котлі; $t_{k_вод}$ – температуру води в котлі.

Крім технологічних параметрів для забезпечення програмного керування система контролює: $g_{вод}$ – витрати холодної води з мережі, $g_{газ}$ – витрати газу, $g_{час}$ – витрати часу за допомогою таймера.

На рис. 4.10 наведено структуру ПЗД, яка використовується для контролю процесу теплової обробки.

Наявну структуру ПЗД використано як базову. Відповідно до інформаційної технології створено алгоритмічну модель ПЗД

$$A(B) \ M_0 \ | \ 0 \ A(E),$$

$$M_0 = A(B) \ || \ [M_1 \dots M_6 M_7 M_8 M_9] \ A(E),$$

де $M_1 - M_5$ – моделі окремих підсистем збору даних на виході автоклавів; M_6 – модель окремої підсистеми збору даних на вході в котел; M_6, M_8, M_9 – моделі вимірювання $g_{вод}$ – витрати холодної води з мережі, $g_{газ}$ – витрати газу, $g_{час}$ – витрати часу за допомогою таймера;

$$\begin{aligned} M_1 = M_2 = M_3 = M_4 = M_5 = M_i = A(B) \ || \ [E_1(S: kx+k_0, ke^{-t/T}/T, (p_{1Ai}), (I_{11}, \varepsilon_1)) \\ || \ [E_2(S: \alpha_0(t^{1/2}+1)(t-t_0), ke^{-t/T}/T, (t_{1Ai}), (u_{11}, \varepsilon_2)) \ E_3(C: k_1u_{11}, (u_{11}), (I_{12}))] \\ [E_4(S: \alpha_0(t^{1/2}+1)(t-t_0), ke^{-t/T}/T, (t_{2Ai}), (u_{12}, \varepsilon_3)) \ E_5(C: k_2u_{12}, (u_{12}), (I_{13}))] \\ [E_6(S: kx, ke^{-t/T}/T, (q_{1Ai}), (f_1, \varepsilon_4)) \ E_7(C: k_3f_1, (f_1), (I_{14}))] \\ E_8(C: f(I_{11}, I_{12}, I_{13}, I_{14}), (I_{11}, I_{12}, I_{13}, I_{14}), (Q_{ii})) \\ E_9(C: I(Q_{ii} \cdot \delta(t-\tau)); (Q_{ii}); (Q_{ii}'))] \ A(E) \end{aligned}$$

Давачі

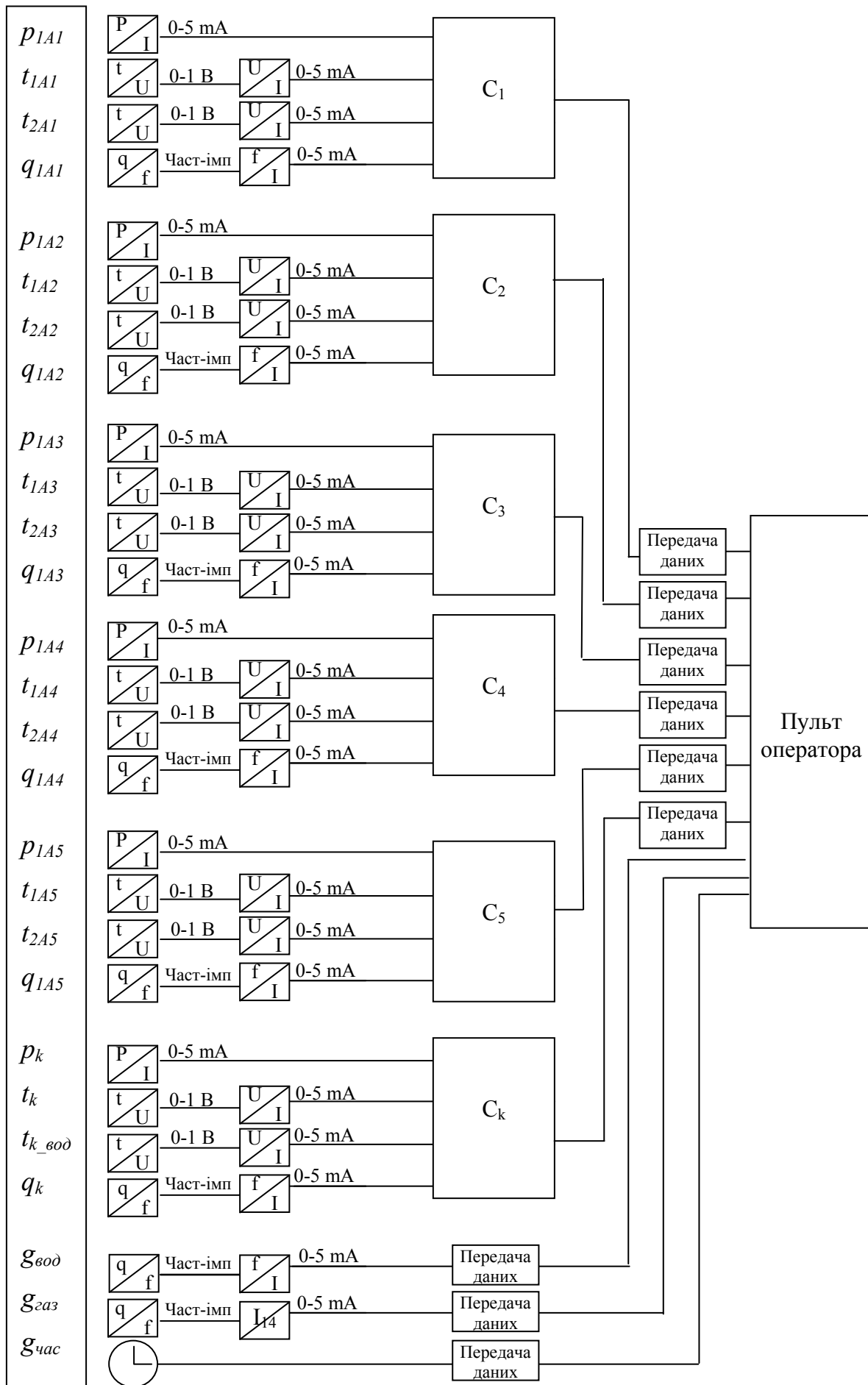


Рис. 4.10. Структура підсистеми збору даних

$$\begin{aligned}
M_6 = A(B) \quad & || [E_1(S: kx+k_0, ke^{-t/T}/T, (p_k), (I_{11}, \varepsilon_1)) \\
& || [E_2(S: \alpha_0(t^{1/2}+1)(t-t_0), ke^{-t/T}/T, (t_k), (u_{11}, \varepsilon_2)) E_3(C: k_1u_{11}, (u_{11}), (I_{12}))] \\
& [E_4(S: \alpha_0(t^{1/2}+1)(t-t_0), ke^{-t/T}/T, (t_{k_год}), (u_{12}, \varepsilon_3)) E_5(C: k_2u_{12}, (u_{12}), (I_{13}))] \\
& [E_6(S: kx, ke^{-t/T}/T, (q_k), (f_1, \varepsilon_4)) E_7(C: k_3f_1, (f_1), (I_{14}))] \\
& E_8(C: f(I_{11}, I_{12}, I_{13}, I_{14}), (I_{11}, I_{12}, I_{13}, I_{14}), (Q_k)) \\
& E_9(C: I(Q_k \cdot \delta(t-\tau)); (Q_k); (Q_k'))] A(E)
\end{aligned}$$

$$\begin{aligned}
M_7 = A(B) \quad & [E_1(S: kx, ke^{-t/T}/T, (q_{k1}), (f_1, \varepsilon_1)) E_2(C: k_1f_1, (f_1), (I_{11})) \\
& E_3(C: I(I_{11} \cdot \delta(t-\tau)); (I_{11}); (I_{11}'))] A(E)
\end{aligned}$$

$$\begin{aligned}
M_8 = A(B) \quad & [E_1(S: kx, ke^{-t/T}/T, (q_{k2}), (f_2, \varepsilon_2)) E_7(C: k_2f_2, (f_2), (I_{12})) \\
& E_3(C: I(I_{12} \cdot \delta(t-\tau)); (I_{12}); (I_{12}'))] A(E)
\end{aligned}$$

$$M_9 = A(B) \quad E_1(S: 1x, ke^{-t/60}/60, (T_1), (T, \varepsilon)) E_2(C: I(T \cdot \delta(t-\tau)); (T); (T')) A(E)$$

На основі алгоритмічної моделі складено граф залежності даних (рис. 4.11)

Для оптимізації алгоритмічної моделі ПЗД АСУТП було використано інформаційну технологію «Checkout Subsystem Optimizer». На рис. 4.12 представлено вікно редактора алгоритмічних моделей «Editor model», фрагмент алгоритмічної моделі ПЗД та його перетворення на операторну форму.

Відповідно до інформаційної технології на основі моделі та нечіткого оцінювання вартісної функції розраховано інформаційно-вартісний критерій оптимізації. Вікно розрахунку показано на рис. 4.13.

Для підвищення ефективності роботи підсистеми збору даних, зокрема, для зменшення невизначеності результату та покращення інших показників ПЗД використано алгоритм оцінювання на основі моделі об'єкта управління.

Модель об'єкта управління складається з моделей котла, що надає пару, та автоклава, який її споживає, і являє собою рівняння теплового балансу.

Для автоклава рівняння теплового балансу:

$$\begin{aligned}
\Delta Q &= q_{ex} - q_{вих} = c \cdot m_n \cdot \Delta t^\circ, \\
(P_{ex} - P_{вих})d\tau &= c_n \cdot m_n \cdot dt^\circ,
\end{aligned}$$

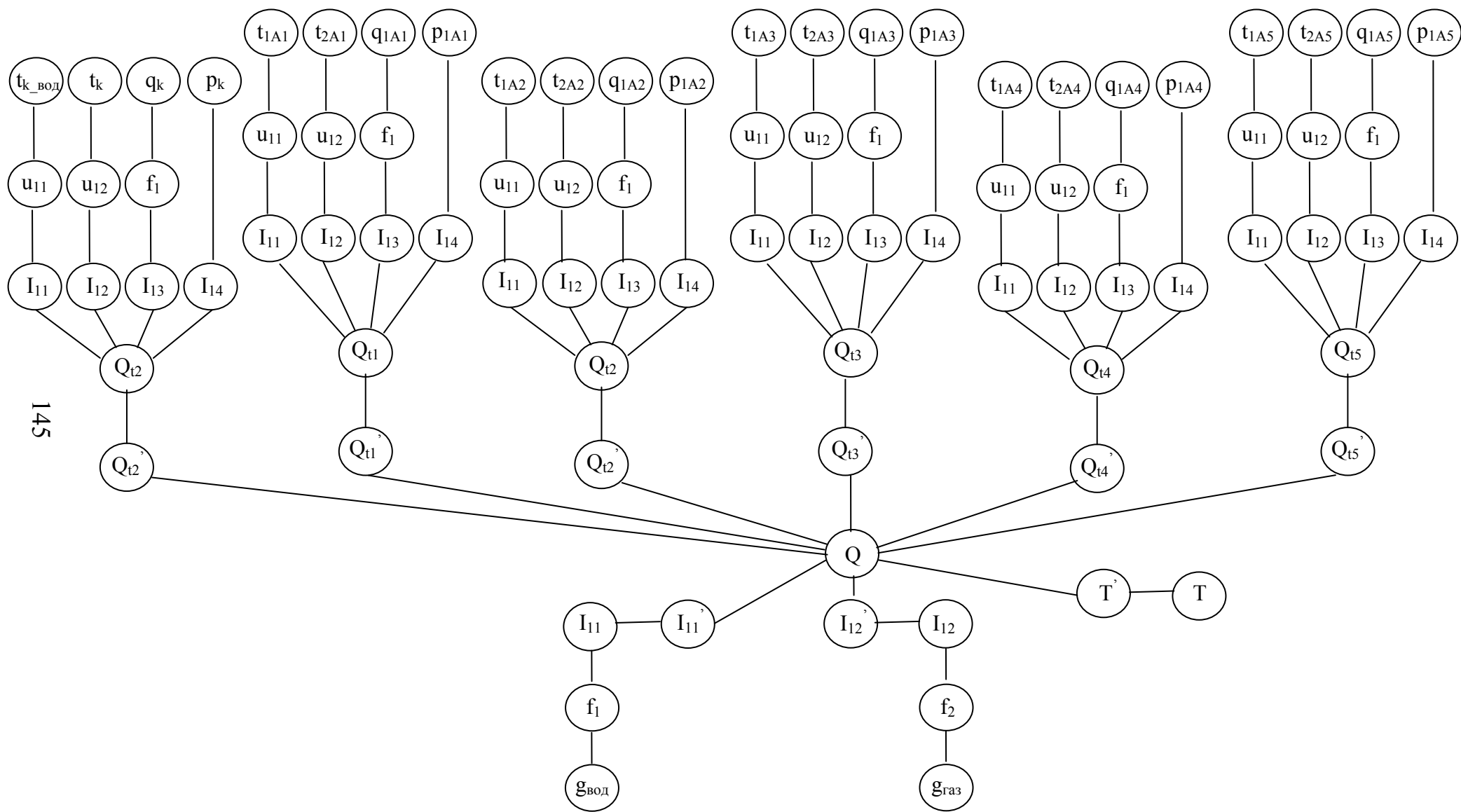


Рис. 4.11. Граф залежності даних

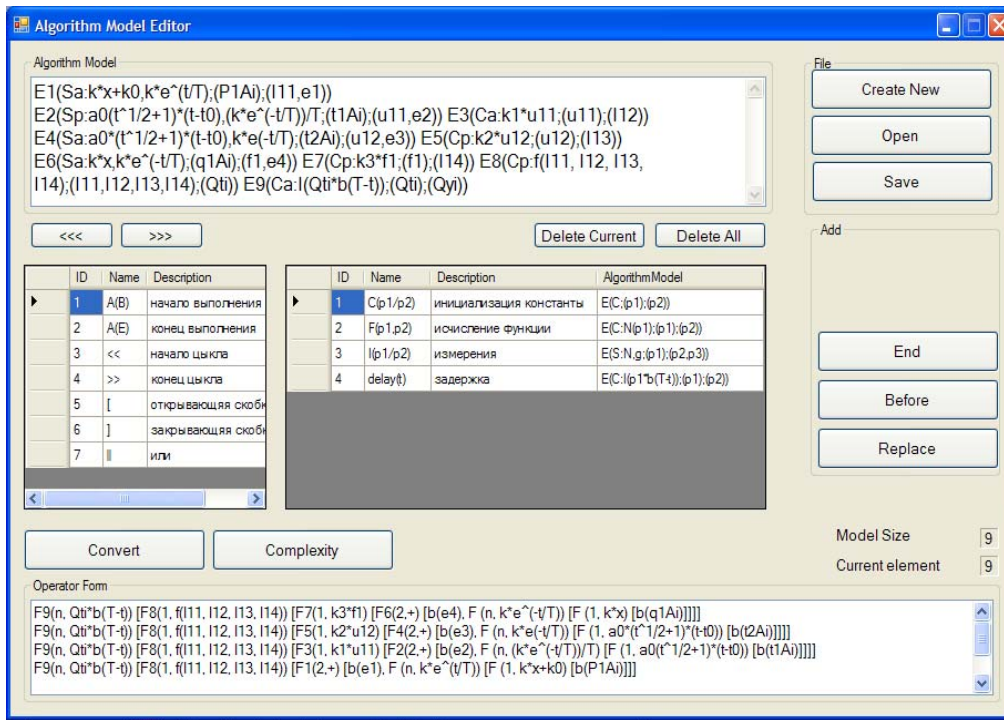


Рис. 4.12. Фрагмент алгоритмічної моделі ПЗД

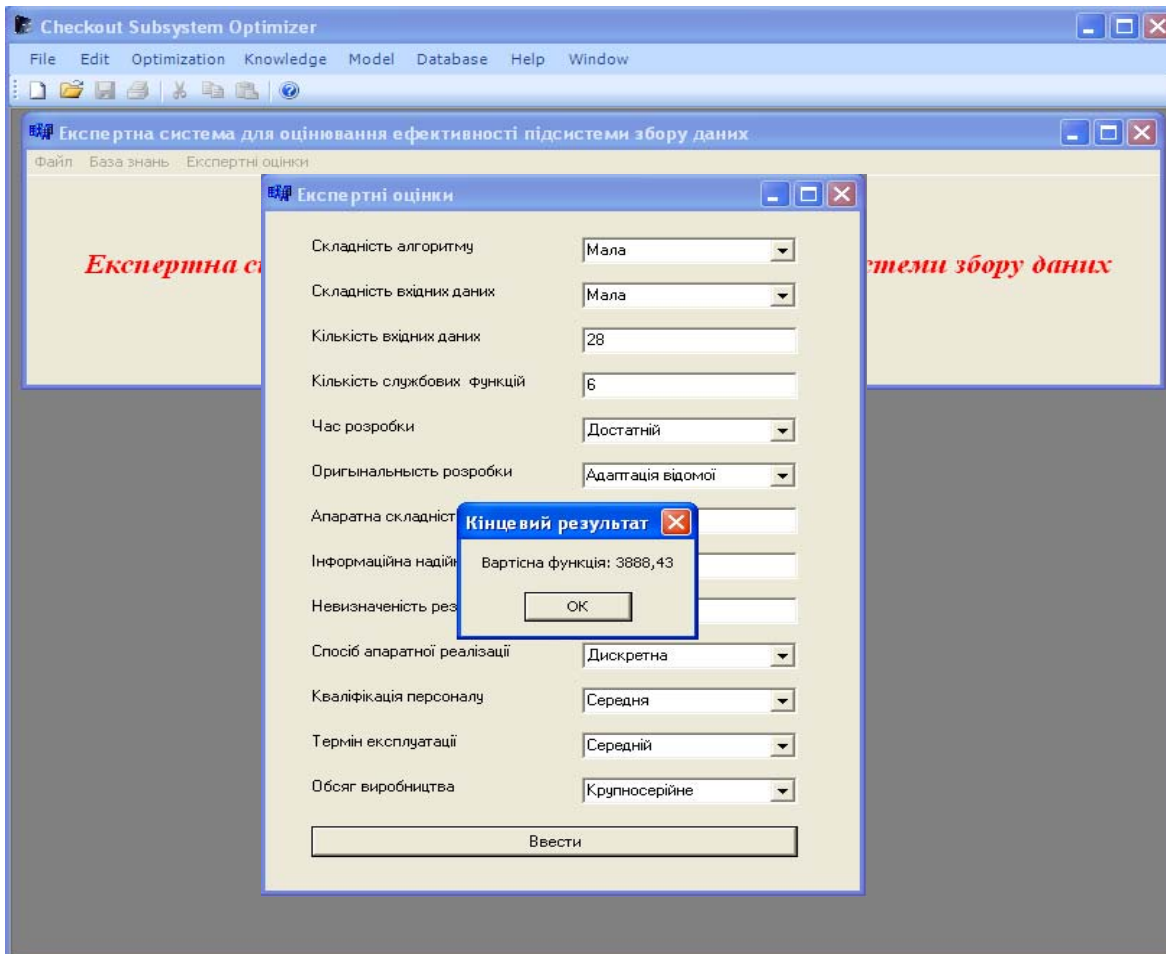


Рис. 4.13. Розрахунок вартісної функції існуючої структури ПЗД

$$P_{\text{вх}} = c_n \cdot V_n \cdot \gamma_n \cdot t_n^{\circ},$$

$$P_{\text{вих}} = c_v \cdot g_{\text{вод}} \cdot t_v^{\circ}.$$

де ΔQ - кількість тепла, що йде на нагрівання автоклаву;

$P_{\text{вх}}$ та $P_{\text{вих}}$ - вхідна та вихідна теплова потужність;

m_n - маса пара;

Δt - зміна температури пара;

$c_{\text{п}}$, $c_{\text{в}}$ - питома теплоємність пари та води;

V_n , γ_n - об'єм та насиченість пара;

$g_{\text{вод}}$ - витрати води;

τ - інтервал часу.

Рівняння теплового балансу котла матиме вигляд:

$$Q_k = \eta_{\text{котла}} \cdot g_{\text{газ}} \cdot \chi,$$

$$Q_k = c_v \cdot g_{\text{вод}} \cdot [t_k(p_n) - t_v] + \lambda_n \cdot g_{\text{вод}},$$

де Q_k - кількість тепла, що йде на нагрівання котла;

$\eta_{\text{котла}}$ - ККД котла;

$g_{\text{газ}}$ - витрати газу (об'єм газу);

χ - питома теплота згорання газу;

$[t_k(p_n) - t_v]$ - різниця температур пари і води в барабані котла;

p_n - тиск пара.

Відповідно до розробленої ІТ на основі графу залежностей (рис. 4.11) з використанням бази даних типових структур були згенеровані варіанти послідовно-паралельних структур виконання збору даних та варіанти їх програмної та апаратної реалізації і перевірена можливість їх реалізації по БЗ. Для кожного з варіантів структури ПЗД було оцінено характеристики та розраховано комплексний показник ефективності. На рис. 4.14 наведена оцінка вартісної функції максимального показника ефективності для структури ПЗД, яка показана на рис. 4.15. Цей варіант структури має найбільший інформаційно-вартісний показник та найменше відрізняється від вже існуючої ПЗД, що зменшує вкладені кошти на її реконструкцію.

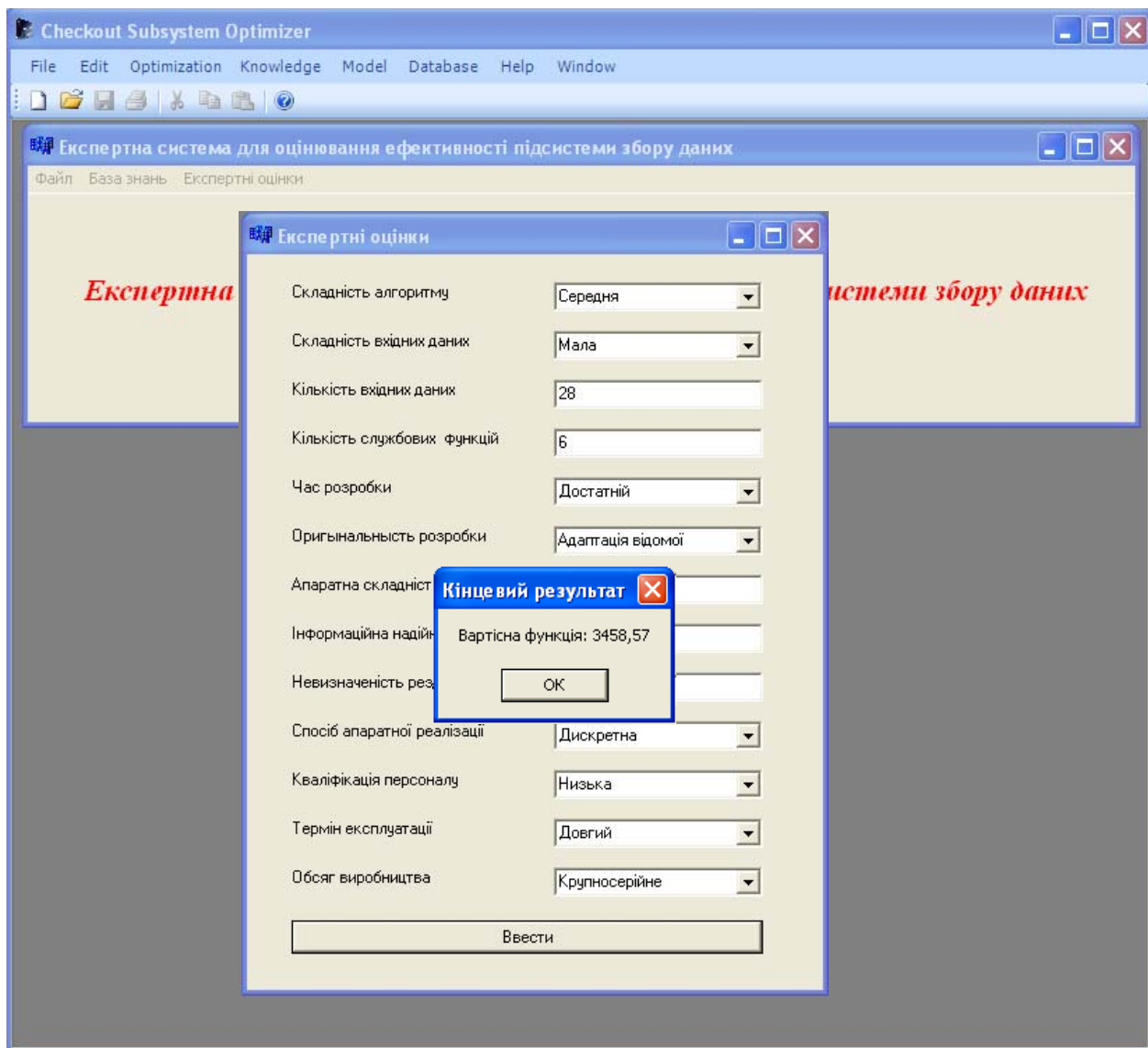


Рис. 4.14. Розрахунок вартісної функції оптимальної структури ПЗД

В оптимізованій структурі ПЗД введено центральний блок управління (ЦБУ), який здійснює облік води і тепла по кожному вводу, облік втрат води і тепла, контроль аварійних ситуацій, керування параметрами споживання. Окрім того, оскільки АСУ ТП теплової обробки консервів досить інерційна система, то для здешевлення ПЗД в схемі зменшено кількість контролерів, які тепер послідовно можуть обробляти інформацію з датчиків автоклавів. Оскільки датчики для обліку витрат холодної води з мережі, газу та часу знаходяться в одному приміщенні з котлом, в оптимальній структурі ПЗД ці параметри контролюються тим самим контролером, що і вимірює параметри води котла.

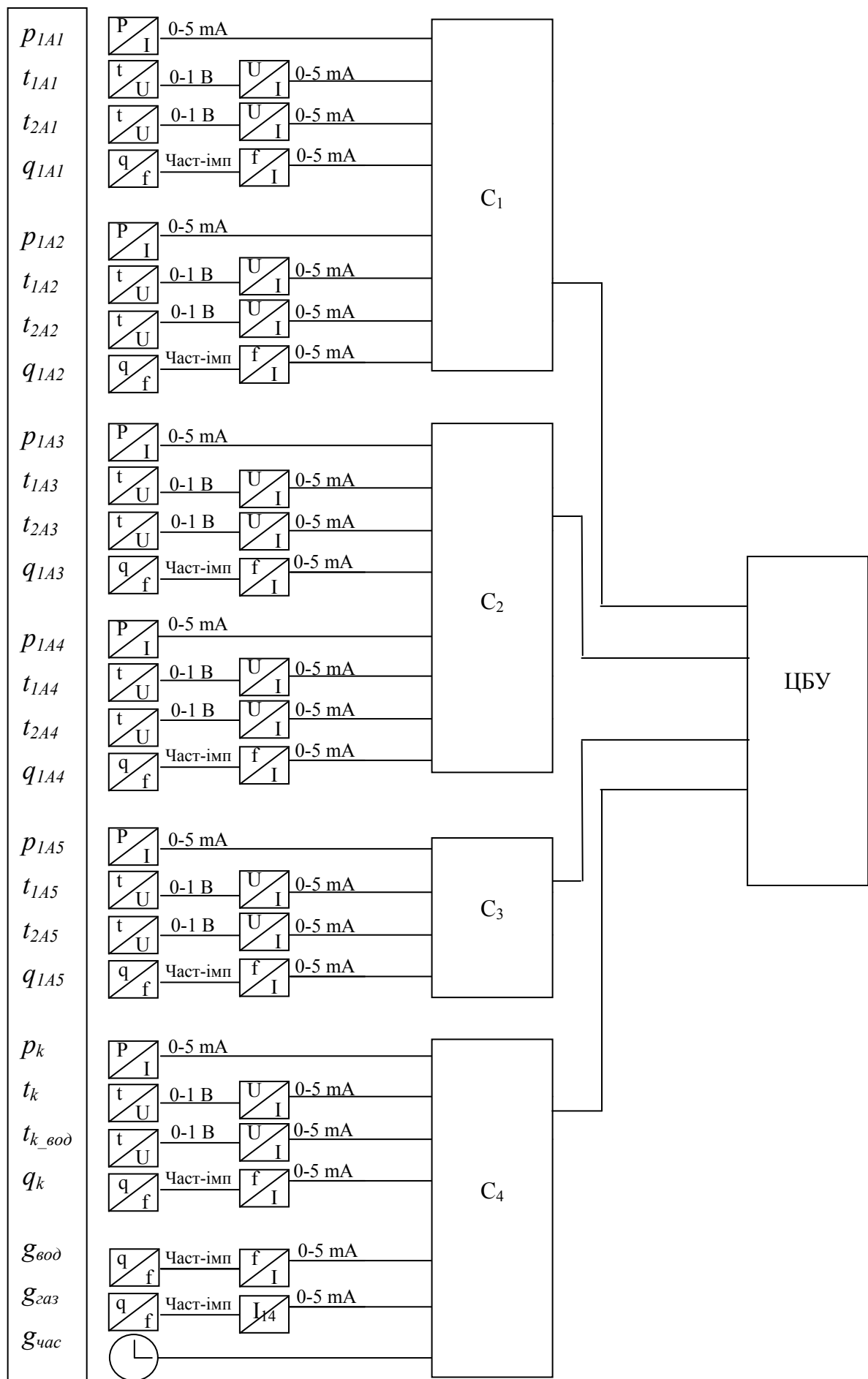


Рис. 4.15. Оптимальна структура ПЗД АСУТП

ПІСЛЯМОВА

В результаті проведених досліджень розвинуто інформаційну технологію оптимізації структури підсистем збору даних (ПЗД) АСУТП з метою підвищення їх ефективності в умовах комбінованої невизначеності.

Моделювання і проектування структури підсистеми збору даних доцільно здійснювати на основі алгоритмічної моделі. При проектуванні ПЗД необхідно враховувати невизначеність функціонування систем.

Розроблений метод моделювання ПЗД в умовах невизначеності базується на формальній системі алгоритмічних моделей і використовує операторне подання перетворень невизначених даних. Застосування цього методу дозволяє здійснювати оптимізацію структур ПЗД в умовах комбінованої стохастичної та нечіткої невизначеності. Для спрощення операторної моделі багатовимірної системи введено поняття векторного оператора.

Вдосконалений метод оцінювання характеристик підсистем збору даних в умовах комбінованої невизначеності ґрунтується на алгоритмічній моделі. Цей метод дозволяє розширити область застосування та підвищити достовірність оцінок характеристик ПЗД, на основі яких визначається критерій оптимізації.

Запропонований підхід до перетворення структурних схем підсистем збору даних ґрунтується на понятті про еквівалентні перетворення алгоритмічної моделі в умовах невизначеності, що дозволяє розширити множину варіантів структур схем підсистем збору даних. Цікавим результатом є висновок, що алгоритми, які не є еквівалентними у визначених умовах, можуть бути еквівалентними в умовах невизначеності, причому степінь еквівалентності може бути збільшена штучним введенням невизначеності.

Розроблені методи, моделі, алгоритми та методики реалізовані у вигляді програмного забезпечення «Checkout Subsystem Optimizer», яке розвиває можливості існуючих систем моделювання, аналізу і проектування АСУ ТП.

Дослідження ефективності розвинутої інформаційної технології показало її перспективність у розв'язанні задачі підвищення ефективності ПЗД в умовах невизначеності.

ЛІТЕРАТУРА

1. Автоматизация технологических процессов и производств [электронный ресурс] // Учебное пособие. Ч. 1 Конспект лекций. – Ангарск, 2005. – Режим доступа: http://window.edu.ru/window_catalog/files/r62090/АТР_Conspsect.pdf.
2. Автоматизированные системы управления технологическими процессами: справочник / [Грищенко А. З., Грищук В. П., Денисенко В. М. и др.] ; под ред. Б. Б. Тимофеева. – К. : Техніка, 1983. – 351 с.
3. Ажогин В. В. Машинное проектирование оптимальных систем управления пространственно-распределенными динамическими объектами / В. В. Ажогин, М. З. Згуровский. – К. : Вища шк., 1985. – 170 с.
4. Акулич И. Л. Математическое программирование в примерах и задачах: [учеб. пособие для студентов экон. спец. вузов] / И. Л. Акулич. – М. : Высш. шк., 1986. – 319 с.
5. Александров А. Г. Оптимальные и адаптивные системы / А. Г. Александров. – М. : Высш. шк., 1989. – 263 с.
6. Александров П. С. Введение в теорию групп / П. С. Александров. – М. : Наука, 1980. – 144 с.
7. Алексеев В. М. Оптимальное управление / В. М. Алексеев, В. М. Тихомиров, С. В. Фомин. – М. : Наука, 1979. – 430 с.
8. Алтунин А. Е. Модели и алгоритмы принятия решений в нечетких условиях: Монография [электронный ресурс] / А. Е. Алтунин, М. В. Семухин. – Тюмень. : Издательство Тюменского государственного университета, 2000. – 352 с. – Режим доступа: <http://www.plink.ru/tnm/gl65.htm>.
9. Ахо А. Структуры данных и алгоритмы / А. Ахо, Д. Ульман, Д. Хопкрофт. – М. : Вильямс, 2003. – 384 с.
10. Ашерев А. Т. Ергономіка інформаційних технологій: оцінка, проектування, експертиза : [навч. посіб.] / А. Т. Ашерев, Г. І. Сажко. – Х. : УПА, 2005. – 243 с.
11. Багриновський К. А. Нові інформаційні технології / К. А. Багриновський, Е. Ю. Хрустальєв. – Х. : ЭКО, 2003. – 411 с.

12. Барковский В. В. Методы синтеза систем управления / В. В. Барковский, В. Н. Захаров, А. С. Шаталов. – М. : Машиностроение, 1981. – 277 с.
13. Батищев Д. И. Методы оптимального проектирования / Д. И. Батищев. – М. : Радио и связь, 1984. – 248 с.
14. Беллман Р. Принятие решений в расплывчатых условиях / Р. Беллман, Л. Заде ; под ред. И. Ф. Шахнова. — М. : Мир, 1976. — С. 172 — 215.
15. Бердышев В. И. Численные методы приближения функций / В. И. Бердышев, Ю. Н. Субботин. – Свердловск. : Средне-Уральское книжное издательство, 1979. – 120 с.
16. Блюмин С. Л. Модели и методы принятия решений в условиях неопределенности / С. Л. Блюмин, И. А. Шуйкова. – Липець : ЛЭГИ, 2001. – 138 с.
17. Бондарь Ю. В. Об одном методе оптимального использования алгоритмической избыточности / Ю. В. Бондарь, И. В. Сафонов // Автоматика и вычислительная техника. – 1975. – № 3. – С. 26–29.
18. Борисов А. Н. Принятие решений на основе нечетких моделей: Примеры использования / А. Н. Борисов, О. А. Крумберг, И. П. Федоров. – Рига : Зинатне, 1990. – 184 с.
19. Боровська Т. М. Імовірнісна модель для прогнозування розвитку розподілених систем / Т. М. Боровська, С. П. Бадьора // Вісник Вінницького політехнічного інституту. – 2006. – № 1. – С. 45-52.
20. Бурков В. Н. Теория графов в управлении организационными системами / В. Н. Бурков, А. Ю. Заложнев, Д. А. Новиков. – М. : Синтег, 2001. – 124 с.
21. Васин Н. Н. Системы сбора информации на железнодорожном транспорте: Учебное пособие. / Н. Н. Васин, В. П. Мохонько. – Самара : СамИИТ, 2001. – 120 с.
22. Вигман Б. А. Стохастические модели контроля / Вигман Б. А. // Управляющие системы и машины. – 1973. – № 2. – С. 112–115.
23. Вирт Н. Алгоритмы + структуры данных = программы / Н. Вирт ; пер. с англ. Л. Ю. Иоффе. – М. : Мир, 1985. – 406 с.
24. Володарський Є. Т. Метрологічне забезпечення вимірювань і контролю : [навч. посіб.] / Є. Т. Володарський, В. В. Кухарчук, В. О. Поджаренко, Г. Б. Сердюк. – Вінниця : Велес, 2001. – 219 с.

25. Воцинин А. П. Оптимизация в условиях неопределенности / А. П. Воцинин, Г. Р. Сотиров. – Изд-во МЭП (СССР), “Техника” (НРБ), 1989. – 224 с.
26. Гамм А. З. Наблюдаемость электроэнергетических систем / А. З. Гамм, И. И. Голуб. – М. : Наука, 1990. – 200 с.
27. Гвоздик М. И. Оптимизация организационно-технических систем ВМФ. Методы. Алгоритмы. Программы. / М. И. Гвоздик, В. Г. Евграфов, Е. Б. Цой. – СПб. : ВВМУРЭ, 1997. – 223 с.
28. Глонь О. В. Моделювання систем керування в умовах невизначеності. Монографія / О. В. Глонь, В. М. Дубовой. – Вінниця : УНІВЕРСУМ-Вінниця, 2004. – 170 с.
29. Глушков В. М. Алгебра. Языки. Программирование / В. М. Глушков, Г. Е. Цейтлин, В. М. Ющенко. – К. : Наукова думка, 1989. – 376 с.
30. Глушков В. М. К вопросу о минимизации программ и схем алгоритмов / В. М. Глушков // Кибернетика. – 1966. – № 5. – С. 1-3.
31. Глушков В. М. О применении метода формализованных технических заданий к проектированию программ обработки структур данных / В. М. Глушков, Ю. В. Капитонова, А. А. Летичевский // Программирование. – 1978. – № 6. – С. 5-12.
32. Годлевский А. Б. Об одном случае специальной проблемы функциональной эквивалентности дискретных преобразователей [Электронный ресурс] / А. Б. Годлевский // Кибернетика. — 1974. — № 3. — С. 32-36. Режим доступа:
<http://www.contrterror.tsure.ru/site/magazine7/pdf>.
33. Голубева Т. О. Використання методу оцінювання в задачах інваріантного управління / Т. О. Голубева, В. М. Дубовой // Вісник ВПІ. – 2007. – № 1. – С. 5-9.
34. Губинский А. И. Надежность и качество функционирования эргатических систем / А. И. Губинский. – Л. : Наука, 1982. – 270 с.
35. Губинский А. И. Эргономическое проектирование судовых систем управления / А. И. Губинский, В. Г. Евграфов. – Л. : Судостроение, 1977. – 224 с.
36. Дейт К. Введение в системы баз данных / К. Дейт ; пер. з англ. Ю. С. Слепцова. – М : Издательский дом «Вильямс», 2001. – 1072 с.

37. Дружинин Г. В. Надежность автоматизированных производственных систем / Г. В. Дружинин. – М. : Энергоатомиздат, 1986. – 480 с.
38. Дубовой В. М. Разработка операторного метода моделирования динамики информационных потоков в распределенных измерительных информационных системах: дис. доктора техн. наук: 05.13.02/ Дубовой Владимир Михайлович. — Винница, 1997. – 378 с.
39. Дубовой В. М. Контроль та керування в мережах теплопостачання. Монографія / В. М. Дубовой, В. В. Кабачій, Ю. М. Паночишин. – Вінниця : УНІВЕРСУМ-Вінниця, 2005. – 190 с.
40. Дубовой В. М. Обработка результатов косвенных измерений при нечетко заданных параметрах / В. М. Дубовой, О. В. Глонь // Научные труды КГПУ. – 2000. – № 2. – С. 262–265.
41. Дубовой В. М. Властивості моделей інформаційних систем в умовах невизначеності / В. М. Дубовой, О. В. Глонь // Матеріали III Міжнародної конференції ІОН-2002. Том 2. – Вінниця : УНІВЕРСУМ-Вінниця, 2002. - С. 410-412.
42. Дубовой В. М. Использование обобщенной вычислительной модели в интеллектуальных системах управления / В. М. Дубовой, О. В. Глонь // Вісник Технологічного університету Поділля. – 2002. – № 3, Т. 1 (41). – С. 122-125.
43. Дубовой В. М. Моделирование систем контролю та керування : [навч. посіб.] / В. М. Дубовой. – Вінниця : ВНТУ, 2005. – 175 с.
44. Дядюнов А. Н. Адаптивные системы сбора и передачи аналоговой информации. Основы теории / А. Н. Дядюнов, Ю. А. Онищенко, А. И. Сенин. – М. : Машиностроение, 1988. – 288 с.
45. Емеличев В. А. Лекции по теории графов. В. А. Емеличев, О. И. Мельников, В. И. Сарванов. — М.: Наука, 1990.— 384 с.
46. Жабін В. І. Прикладна теорія цифрових автоматів [Електронний ресурс] / В. І. Жабін, І. А. Клименко, В. В. Ткаченко, Є. В. Красовська, А. О. Антипов // Електронний посібник. – Режим доступу:
<http://ot.iit.nau.edu.ua/ptca/index.html>
47. Зайченко Ю. П. Многокритериальные задачи принятия решений в нечётких условиях и методы их решения / Ю. П. Зайченко // Системні дослідження та інформаційні технології. – 2002. – № 2 – С. 53-62.

48. Захаров И. П. Неопределенность измерения: общие подходы к составлению бюджета неопределенности / И. П. Захаров // Украинський метрологічний журнал. – 2004. – № 2 – С. 10-15.
49. Захаров И. П. Теория неопределенности в измерениях / И. П. Захаров, В. Д. Кукуш. – Харьков : Консум, 2002. – 256 с.
50. Зингер Н. С., Куцык Б. С. Обеспечение достоверности данных в автоматизированных системах управления производством. – М.: Наука, 1974. – 136 с.
51. Иванов С. В. Параллельные алгоритмы моделирования комплексных сетей / С. В. Иванов, И. И. Колыхматов, А. В. Бухановский // Параллельные вычислительные технологии (ПаВТ, 2008): междунауч. конф., 28 янв. - 1 февр. 2008 г.: матер. конф. – Челябинск : Изд. ЮУрГУ, 2008. – С. 395-402.
52. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения: ГОСТ 34.003-90. — [Чинний від 1992-01-01]. — М. : Издательство стандартов, 1991. — 68 с.
53. Информационно-управляющие человеко-машинные системы: Исследования, Проектирование, Испытания: Справочник / [Адаменко А. Н., Губинский А. И. и др.] – М. : Машиностроение, 1993. – 528 с.
54. Інформаційно-пошуковий сайт Вікіпедія [Електронний ресурс]. – Режим доступу: www.wikipedia.com.ua
55. Калужнин Л. А. Об алгоритмизации математических задач / Л. А. Калужнин // Проблемы кибернетики. – 1959. – № 2. – С. 51-67.
56. Карпова Н. А. О некоторых свойствах функций Шеннона [Електронний ресурс] / Н. А. Карпова // Математические заметки. – 1970. – Т. 8, № 5. – С. 663-674. – Режим доступу: http://www.mathnet.ru/php/getFT.phtml?jrnid=mzm&paperid=7014&what=fullt&option_lang=rus
57. Карповский Е. Я. Надежность программной продукции / Е. Я. Карповский, С. А. Чижов. – К. : Техника, 1990. – 160 с.
58. Касти Дж. Большие системы. Связность, сложность и катастрофы / Дж. Касти ; пер. с англ. – М. : Мир, 1982. – 216 с.

59. Кветний Р. Н. Основи моделювання та обчислювальних методів. Навчальний посібник / Р. Н. Кветний. – Вінниця : ВНТУ, 2007. – 150 с.
60. Кириллов Н. П. Концепция построения алгоритмических моделей процессов принятия решения при управлении сложными техническими системами / Н. П. Кирилов // «Информатизация и связь». – 2003. – № 1-2. – 116 с.
61. Клиначёв Н. В. Теория систем автоматического регулирования [Электронный ресурс] / Н. В. Клиначёв // Учебно-методический комплекс. – Режим доступа : <http://model.exponenta.ru/lectures/0060.htm>
62. Князева М. А. Концепция банка знаний в области оптимизации программ для поддержки научных исследований, образования и профессиональной деятельности / М. А. Князева, А. С. Клещев. – Владивосток : ИАПУ ДВО РАН, 2003. – 28 с.
63. Князьков В. С. Введение в теорию графов [Электронный ресурс] / В. С. Князьков, Т. В. Волченская. – Режим доступа: www.intuit.ru
64. Ковалюк Д. О. Про один алгоритм прийняття рішень в умовах ризику / Д. О. Ковалюк, О. Д. Никитенко // Конференція студентів та молодих вчених „Політ”: 4-а Міжнар. наук. конф., 15-16 квітня, 2004 року. : тези. – К., 2004. – С. 45.
65. Ковалюк О. О. Аксиоматична основа прийняття рішень в умовах комбінованої невизначеності / В. М. Дубовой, О. О. Ковалюк // Оптико-електронні інформаційно-енергетичні технології. – 2006. – № 1 (11). – С. 70-75.
66. Ковалюк О. О. Порівняльний аналіз методів прийняття рішень в умовах невизначеності / В. М. Дубовой, О.О. Ковалюк // „Автоматика – 2004”: 11-а Міжнар. конф.: тези. – К., 2004. – С. 39.
67. Ковалюк О. О. Прийняття рішень в розподілених системах за умов комбінованої невизначеності / В. М. Дубовой, О. О. Ковалюк // Інформаційні технології та комп’ютерна інженерія. – 2004. – № 1. – С. 8-11.
68. Козлов Б. А. Справочник по расчету надежности аппаратуры радиоэлектроники и автоматики / Б. А. Козлов, И. А. Ушаков. – М. : Сов. радио, 1975. – 472 с.
69. Коллинз Г. Структурные методы разработки систем: от стратегического

- планирования до тестирования / Г. Коллинз, Блэй Дж. - М. : Финансы и статистика, 1986. - 263 с.
70. Котов В. Е. Теория схем программ / В. Е. Котов, В. К. Сабельфельд. – М. : Наука, 1991. – 248 с.
 71. Красовский А. А. Справочник по теории автоматического управления / А. А. Красовский. – М. : Наука. Гл. ред. физ.-мат. лит., 1987. – 712 с
 72. Краус М. Измерительные информационные системы (характеристические функции, критерии качества, оптимизации) / М. Краус, Э. Вошин ; пер. с нем. Я. М. Малкова. – М. : Мир, 1975. – 310 с.
 73. Краус М. Сбор данных в управляющих вычислительных системах / М. Краус, Э. Кучбах, О. Г. Вошни ; пер. с нем. – М. : Мир, 1987. – 294 с.
 74. Крилов І. В. Інформаційні технології: теорія і практика / І. В. Крилов. - К. : Центр, 2002. – 410 с.
 75. Криницкий Н. А. Равносильные преобразования алгоритмов и программирование [Электронный ресурс] / Н. А. Криницкий — М. : Советское радио, 1970. — 304 с. Режим доступа:
<http://alib.com.ua/bs.php4?bs=Knigoxod&id=2780257>
 76. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – М. : Мир, 1978 – 432 с.
 77. Кузьмин И. В. Оценка эффективности и оптимизации АСКУ / И. В. Кузьмин. – М. : Сов. радио, 1997.– 296 с.
 78. Кузьмин И. В. Основы моделирования сложных систем / Под ред. И. В. Кузьмина. – К. : Вища школа, 1981. – 359 с.
 79. Кулик В. Т. Алгоритмизация объектов управления / В. Т. Кулик. - К. : Наукова думка, 1968. – 364 с.
 80. Кун С. Ю. Матричные процессоры на СБИС / С. Ю. Кун ; пер. с англ. – М. : Мир, 1991. – 672 с.
 81. Кунцевич В. М. Адаптивное управление: алгоритмы, системы, применение / В. М. Кунцевич. – К. : Вища шк., 1988. – 64 с.
 82. Кунцевич В. М. Управление в условиях неопределенности: гарантированные результаты в задачах управления и идентификации / В. М. Кунцевич. – Киев : Наук. думка, 2006. – 264 с.
 83. Куссуль Н. М. Робастні методи оцінювання та ідентифікації в умовах невизначеності / Н. М. Куссуль // Український метрологічний журнал. –

2004.– № 3. – С. 15-23.

84. Ладанюк А. П. Основы системного анализа : [навч. посіб.] / А. П. Ладанюк. – Вінниця : Нова книга, 2004. – 176 с.
85. Ладанюк А. П. Автоматизація технологічних процесів і виробництв харчової промисловості : [підручник] / А. П. Ладанюк, В. Г. Трегуб та ін. – К. : Аграрна освіта, 2001. – 224 с.
86. Лежнюк П. Д. Оцінка якості оптимального керування критеріальним методом. Монографія / П. Д. Лежнюк, В. О. Комар. – Вінниця : УНІВЕРСУМ-Вінниця, 2006. – 108 с.
87. Лежнюк П. Д. Аналіз чутливості оптимальних рішень в складних системах критеріальним методом. Монографія / П. Д. Лежнюк. – Вінниця : УНІВЕРСУМ-Вінниця. – 2003. – 131 с.
88. Летичевский А. А. Практические методы распознавания эквивалентности дискретных преобразователей и схем программ / А. А. Летичевский // Кибернетика. — 1973. — № 4 — С. 15—26.
89. Лигун Л. Экономическая эффективность разработки проекта АСУ ТП В Трейс Моуд 5 [Электронный ресурс] / Л. Лигун // Электроэнергетика. Статьи по электроэнергетике, электрическим сетям, оборудованию электрических подстанций и высоковольтных линий электропередач. – Режим доступа : <http://forca.ru/stati/sdtu/ekonomicheskaya-effektivnost-razrabotki-proekta-asu-tp-v-treis-moud-5.html>
90. Лидовский В. В. Теория информации : [учеб. пособ.] / В. В. Лидовский. – М. : Компания Спутник+, 2004. – 111 с.
91. Ляпунов А. А. О логических схемах программ / А. А. Ляпунов // Проблемы кибернетики. – 1958. – № 1. — С. 46 — 74.
92. Макаров И. М. Линейные автоматические системы (элементы, теории, методы расчета, справочный материал) : [учеб. пособ. для вузов] / И. М. Макаров, Б. М. Менский. – М. : Машиностроение, 1982. – 504 с.
93. Максименко И. Н. Объектно-ориентированный подход к анализу автоматизированных систем теплоснабжения с изменяемой структурой / И. Н. Максименко // Вісник ВПШ. – 2006. - № 6. – С. 34-38.
94. Маликов В. Т. Исследование и оптимизация информационных характеристик устройств контроля / В. Т. Маликов, В. М. Дубовой, Р. Н. Кветный. – К. : Знание, 1983. – 213 с.

95. Маликов В. Т. Анализ измерительных информационных систем / В. Т. Маликов, В. М. Дубовой, Р. Н. Кветный. – Ташкент : ФАН, 1984. – 176 с.
96. Малышев Н. Г. Нечеткие модели для экспертных систем в САПР / Н. Г. Малышев, Л. С. Берштейн, А. В. Боженюк. – М. : Энергоатомиздат, 1991.– 136 с.
97. Марков А. А. Теория алгоритмов / А. А. Марков, Н. М. Нагорный. - М. : ФАЗИС, 1996. – 496 с.
98. Марков Н. Г. Автоматизированные системы сбора и регистрации сейсмической информации / Н. Г. Марков. - М. : Недра, 1992. - 219 с.
99. Маркушевич Н. С. Автоматизированная система диспетчерского управления / Н. С. Маркушевич. – М. : Энергоатомиздат, 1986. – 136 с.
100. Математическое моделирование систем и объектов. Нормированное пространство [Электронный ресурс]. - Режим доступа: http://de.ifmo.ru/--books/0051/6/6_3/63_normpr_1.htm
101. Мелихов А. Н. Ситуационные советующие системы с нечеткой логикой / А. Н. Мелихов, Л. С. Берштейн, С. Я. Коровин. – М.: Наука. Гл. ред. физ.-мат. лит., 1990.– 272 с.
102. Мищенко В. А. Теория эквивалентных преобразований алгоритмов в САПР СБИС / В. А. Мищенко, А. А. Прихожий. – Минск : Навука і тэхніка, 1991. - 263 с.
103. Мітюшкін Ю. І. Soft Computing: ідентифікація закономірностей нечіткими базами знань. Монографія. / Ю. І. Мітюшкін, Б. І. Мокін, О. П. Ротштейн – Вінниця : УНІВЕРСУМ-Вінниця, 2002. – 145 с.
104. Мокін Б. І. Новий метод моделювання секвенційного опису комп'ютеризованої системи / Б. І. Мокін, В. Б. Мокін, С. О. Жуков // Наукові праці ВНТУ. – 2008. – № 2. - Режим доступу: <http://www.nbu.gov.ua/e-journals/VNTU/2008-2/2008-2.htm>
105. Мокін Б. І. Властивості слабких операцій в теорії нечітких множин / Б. І. Мокін, В. В. Камінський, С. Ш. Каців // Вісник Вінницького політехнічного інституту. – 2001. – № 5. – С. 106-113.
106. Мокін Б. І. Слабкі множини та їх застосування до розв'язання задач прийняття рішень в умовах невизначеності даних / Б. І. Мокін, В. В. Камінський // Вісник Вінницького політехнічного інституту. –

2004. – № 3. – С. 102-108.
107. Мороз О. В. Оптимальне управління економічними системами в умовах невизначеності та ризику. Монографія / О. В. Мороз, А. В. Матвійчук. – Вінниця : УНІВЕРСУМ-Вінниця, 2003. – 177 с.
108. Морозов В. П. Система управління проектами на основі алгоритмічних моделей САПФІР АС / В. П. Морозов // «Информатизация и связь». – 2003. – № 1-2. – 116 с.
109. Надежность алгоритмов управления / [Е. Я. Карповский, В. В. Сагач, А. А. Чернецкий]. – К.: Техніка, 1983. – 112 с.
110. Надежность технических систем / [Ю. К. Беляев, В. А. Богатырев, В. В. Болотин и др.] ; под ред. И. А. Ушакова. – М. : Радио и связь, 1985. – 608 с.
111. Нечеткие множества в моделях управления и искусственного интеллекта / [А. Н. Аверкин, И. З. Батыршин, А. Ф. Блишун и др.] ; под ред. Д. А. Поспелова – М. : Наука, 1986. – 312 с.
112. Никитенко О. Д. Алгоритмічна модель в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // Конференція професорсько-викладацького складу, співробітників та студентів університету з участю працівників науково-дослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області., 23-25 бер., 2005 р. : тези доповідей. – Вінниця. – 2005. - С. 134.
113. Никитенко О.Д. Алгоритмічна модель системи, керованої знаннями / В. М. Дубовой, О. Д. Никитенко // Автоматика-2006: XIII Міжнар. конф. з автоматичного управління, 25-26 верес. 2006 р: матер. конф. – Вінниця : УНІВЕРСУМ-Вінниця, 2007. – С. 382-385.
114. Никитенко О. Д. Визначення вимог до структури підсистеми керування вимірювально-обчислювальної системи / В. М. Дубовой, О. Д. Никитенко // Вісник Хмельницького національного університету. – 2005. – № 4. Ч1., Т1 (68). – С. 115-118.
115. Никитенко О. Д. Еквівалентність алгоритмічних моделей в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // „ІНТЕРНЕТ – ОСВІТА – НАУКА – 2008” : Шоста міжнар. конф., 7-11 жовт. 2008 р.: матер. конф. – Вінниця : УНІВЕРСУМ-Вінниця, 2008. – Том 2. – С. 510-513.

116. Никитенко О. Д. Еквівалентність невизначених алгоритмів [Електронні дані] / В. М. Дубовой, О. Д. Никитенко, О. В. Глонь // Наукові праці ВНТУ. – Вінниця, 2009. – № 2. – Режим доступу:
http://www.nbuuv.gov.ua/e-journals/VNTU/2009_2_ua/2009-2.files/uk/09vmdoia_ua.pdf.
117. Никитенко О. Д. Еквівалентність невизначених алгоритмів [Електронні дані] / В. М. Дубовой, О. Д. Никитенко // Контроль і управління в складних системах – 2008 : VIII Міжнар. конф., 21—24 жовт. 2008 р. : тези доповідей. - Вінниця, 2008. – Режим доступу:
http://www.vstu.vinnica.ua/mccs2008/materials/subsection_1.2.pdf
118. Никитенко О. Д. Застосування алгоритмічної моделі до оптимізації інформаційно-обчислювальних систем в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // Контроль і управління в складних системах - 2005: VII Міжнар. конф., 24-27 жовт. 2005 р. : тези доповідей. – Вінниця, 2005. – С. 34
119. Никитенко О. Д. Застосування алгоритмічної моделі до оптимізації інформаційно-обчислювальних систем в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // Вісник Вінницького політехнічного інституту. – 2005. – № 6. – С. 9-13.
120. Никитенко О.Д. Звіт про науково-дослідну роботу "Розробка рекомендацій щодо проектування оптимальної структури та алгоритму системи керування технологічним процесом варки рибних консервів" (заключний) / О. Д. Никитенко, В. М. Дубовой. – Вінниця, 2008. – 74 с.
121. Никитенко О. Д. Звіт про науково-дослідну роботу «Розробка математичної моделі та методу прийняття рішень в умовах комбінованої невизначеності» (заключний) / В. М. Дубовой, О. Д. Никитенко та ін. – Вінниця, 2007. – 138 с.
122. Никитенко О.Д. Комп'ютерна програма для оптимізації алгоритмів прийняття рішень в умовах невизначеності в системах керування «EditorModel» / В. М. Дубовой, О. Д. Никитенко // Свідоцтво про реєстрацію авторського права на твір № 17351 від 24.07.2006
123. Никитенко О. Д. Об'єктно-орієнтований аналіз розвитку складних систем управління / В. М. Дубовой, О. Д. Никитенко // Вісник Вінницького політехнічного інституту. – 2007. – № 3. – С. 74 -77.

124. Никитенко О. Д. Оптимізація структури системи на основі алгоритмічної моделі / В. М. Дубовой, О. Д. Никитенко // Вісник Хмельницького національного університету. – 2007. – № 3, Т.1. – С. 97-100.
125. Никитенко О. Д. Спеціальні розділи математики / В. М. Дубовой, О. Д. Никитенко ; навч. посіб. – Вінниця : ВНТУ, 2007. – 165 с.
126. Никитенко О. Д. Формалізація операцій над алгоритмічними моделями систем керування в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // Інтелектуальні системи прийняття рішень та прикладні аспекти інформаційних технологій (ISDMIT'2006): міжнар. наук. конф., Євпаторія, 15-18 трав. 2006 р. : тези доповідей. – Херсон : Видавництво Херсонського морського інституту, 2005. – Том 4. – С. 176.
127. Никитенко О. Д. Формалізація перетворень алгоритмічних моделей систем керування в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // Вісник Хмельницького національного університету. Технічні науки. – 2006. – № 5. – С. 54-57.
128. Никитенко О. Д. Формалізація синтезу алгоритмічних моделей систем керування в умовах невизначеності / В. М. Дубовой, О. Д. Никитенко // Системні технології. – 2006. – № 6 (47). – С. 121-130.
129. Никитенко О. Д. Формальна система представлення невизначених алгоритмічних моделей / В. М. Дубовой, О. Д. Никитенко // Інформаційні технології та комп'ютерна інженерія. – 2005. – № 1. – С. 49-52.
130. Никитенко О. Д. Штучна невизначеність як фактор підвищення ефективності систем / В. М. Дубовой, О. Д. Никитенко // «Автоматизація: проблеми, идеи, решения»: междунар. науч.-техн. конф., 8-12 сент. 2008 г.: матер. конф. – Севастополь: Изд-во СевНТУ, 2008. – С. 164-167.
131. Новицкий Н. Н. Оценивание параметров трубопроводных систем методом приведенной линеаризации / Н. Н. Новицкий // Изв. Ан СССР. Энергетика и транспорт. – 1990. – № 6. – С.122-129.
132. Новоселов А. А. Лекции по теории риска [Электронный ресурс] / А. А. Новоселов. – Режим доступа: <http://anov.narod.ru/lectures.htm>.
133. Носов В. А. Основы теории алгоритмов и анализа их сложности / В. А. Носов. – М. : Наука, 1992. – 139 с.
134. Орловский С. А. Проблемы принятия решений при нечеткой исходной информации / С. А. Орловский. – М. : Наука, 1981. – 208 с.

135. Палагин А. В. Системная интеграция средств компьютерной техники / А. В. Палагин, Ю. С. Яковлев. Монография. – Винница : Універсум-Вінниця, 2005. – 680 с.
136. Перестюк М. О. Задачі оптимального керування / М. О. Перестюк, О. М. Старицький, О. В. Капустян. – К. : «ТВіМС», 2004. – 55 с.
137. Перцовский М. И. Системы промышленной и лабораторной автоматизации: методы и средства построения [Электронный ресурс] / М. И. Перцовский // Научно-технический журнал «Мир компьютерной автоматизации». – 2000. – № 3. – Режим доступа: <http://www.mka.ru/>
138. Петров В. В. Информационная теория синтеза оптимальных систем контроля и управления (Непрерывные системы) / В. В. Петров, А. С. Усков. – М. : Энергия, 1975. – 232 с.
139. Пивоваров А. Н. Методы обеспечения достоверности информации в АСУ. – М. : Радио и связь, 1982. – 144 с.
140. Подловченко Р. И. Недетерминированные схемы алгоритмов / Р. И. Подловченко // Докл. АН СССР. – 1972. – Т. 207, № 4. – С. 789-792.
141. Половко А. М. Основы теории надежности / А. М. Половко, С. В. Гуров. – СПб. : ВНУ, 2006. – 704 с.
142. Пономарев В. М. Алгоритмические модели в задачах исследования систем / В. М. Пономарев // Алгоритмы и системы автоматизации исследований и проектирования. – М. : Наука. – 1980. – С. 4-8.
143. Попович П. Р. Эргономическое обеспечение деятельности космонавтов / П. Р. Попович, А. И. Губинский, Г. М. Колесников. – М. : Машиностроение, 1985. – 272 с.
144. Поспелов Г. С. Искусственный интеллект - основа новой информационной технологии / Г. С. Поспелов. - М. : Мир, 1988. – 280 с.
145. Проектування інформаційних систем : [посібник / за ред. В. С. Пономаренка]. – К. : Академія, 2002. – 486 с.
146. Пугачов В. С. Основы статистической теории автоматических систем / В. С. Пугачов. – М. : Машиностроение, 1974. – 400 с.
147. Распараллеливание алгоритмов обработки информации. Том 1 / Под ред. А. Н. Свенсона. – К. : Наук. думка, 1985. – 280 с.
148. Репин В. Г. Статистический синтез при априорной неопределенности и адаптация информационных систем / В. Г. Репин, Г. П. Тартаковский. –

- М. : Советское радио, 1977. – 432 с.
149. Рогов С. Ф. О математических методах теории принятия решений [Электронный ресурс] / С. Ф. Рогов. – Режим доступа: www.mfua.ru/temp/file/konf_1/1_6.doc
 150. Ротштейн А. П. Проектирование бездефектных человеко-машинных технологий / А. П. Ротштейн, П. Д. Кузнецов – К. : Техніка, 1992. – 180 с.
 151. Ротштейн А. П. Нечеткая надежность алгоритмических процессов / А. П. Ротштейн, С. Д. Штовба. – Винница : Континент–ПРИМ, 1997. – 142 с.
 152. Ротштейн О. П. Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі / О. П. Ротштейн. – Вінниця : УНІВЕРСУМ–Вінниця, 1999. – 320 с.
 153. Ротштейн О. П. Евристична оптимізація розстановки контрольних точок в технологічних процесах при багатовимірному просторі типів дефектів / О. П. Ротштейн, С. Д. Штовба, С. Б. Дубіненко, О. М. Козачко // Вісник Вінницького політехнічного інституту. – 2004. – №1. – С. 54-62.
 154. Ротштейн О. П. Діагностика на базі нечітких відношень в умовах невизначеності / О. П. Ротштейн, Г. Б. Ракитянська. Монографія - Вінниця: УНІВЕРСУМ-Вінниця, 2006. - 275 с.
 155. Сайт Википедия. – Режим доступу: www.wikibooks.org/wiki
 156. Сейдж Э. П. Теория оценивания и ее применение в связи и управлении / Э. П. Сейдж, Д. Мелса ; пер. з англ. – М. : Связь, 1976. – 496 с.
 157. Сергеев А. Г. Метрология : [Учебник] / А. Г. Сергеев. – М. : Логос, 2005 - 272 с.
 158. Системы диспетчерского управления и сбора данных (SCADA-системы) [Электронный ресурс] // Журнал «Мир компьютерной автоматизации». – 1999. - № 3. – Режим доступа: <http://ankey.ru/tech/scada/intro.htm>
 159. Системы параллельной обработки / [под ред. Д. Ивенса]. – М. : Мир, 1985. – 416 с.
 160. Советов Б. Я. Большие системы сбора и передачи информации : [Учеб. пособ.] / Б. Я. Советов. – Л., 1974. – 255 с.
 161. Советов Б. Я. Моделирование систем. Практикум : [Учеб. пособ.] / Б. Я. Советов, С. А. Яковлев. - М. : Высш. шк., 1999. - 224 с.
 162. Соколов В. В. Подход к оценке сложности систем / В. В. Соколов //

- Академия информационных технологий. Публикации. Оценка сложности системы. – Режим доступа: http://ait.org.ua/p/pub_podhod.html
163. Справочник Novell Netware 4, Глава 10. Протоколы передачи данных, С. Б. Орлов, по заказу ИИЦ "Попурри", 1994. – Режим доступа: http://www.citforum.ru/operating_systems/nw4/gl10.shtml
 164. Техника проектирования систем автоматизации технологических процессов / [под ред. Л. И. Шипетина]. – М. : Машиностроение, 1976. – 496 с.
 165. Трауб Дж. Ф. Информация, неопределенность, сложность / Дж. Ф. Трауб. – М. : Мир, 1988. – 183 с.
 166. Трахтенброт Б. А. Алгоритмы и вычислительные автоматы / Б. А. Трахтенброт. – М. : Советское радио, 1974. – 200 с.
 167. Трахтенброт Б. А. Сложность алгоритмов и вычислений [спецкурс для студентов НГУ] / Б. А. Трахтенброт. – Новосибирск : НГУ, 1967. – 257 с.
 168. Управление ГПС: Модели и алгоритмы / [под общ. ред. С. В. Емельянова]. – М. : Машиностроение, 1987. – 365 с.
 169. Успенский В. А. Теория алгоритмов: основные открытия и приложения / В. А. Успенский, А. Л. Семенов. - М. : Наука, 1987. – 288 с.
 170. Цапенко М. П. Измерительные информационные системы: Структуры и алгоритмы, системотехническое проектирование / М. П. Цапенко. - М. : Энергоатомиздат, 1985. - 439 с.
 171. Цапенко М. П. Содержательные логические схемы алгоритмов измерительных систем / М. П. Цапенко // Измерения. Контроль. Автоматизация. – 1982. – № 4. – С. 3-8.
 172. Чернов В. Г. Устройство ввода-вывода аналоговой информации для цифровых систем сбора и обработки данных / В. Г. Чернов. - М. : Машиностроение, 1988. - 183 с.
 173. Чернорудский И. Г. Методы принятия решений / И. Г. Чернорудский. – СПб: БХВ – Петербург, 2005. – 416 с.
 174. Штовба С. Д. Забезпечення надійності алгоритмічних процесів [Електронні дані] / С. Д. Штовба // Наукові праці ВНТУ (Кусс-2008). – Вінниця, 2009. – Режим доступу: http://www.nbu.gov.ua/e-journals/VNTU/2009_2_ua/2009-2.files/uk/09sdsscm_ua.pdf
 175. Штовба С. Д. Матричні моделі надійності алгоритмічних процесів за

- сумісності помилок різних типів [Електронні дані] / С. Д. Штовба // Вісник Вінницького політехнічного інституту. – Вінниця, 2007. – №4. – Режим доступу: <http://visnyk.vstu.vinnica.ua/2007/4/pdf/07ssdprt.pdf>
176. Юсупов Р. М. Элементы теории идентификации технических объектов / Р. М. Юсупов. – М. : Мир, 1974. – 170 с.
177. Юхимчук С. В. Математичні моделі ризику для систем підтримки прийняття рішень / С. В. Юхимчук, А. О. Азарова. Монографія. – Вінниця : УНІВЕРСУМ-Вінниця, 2003. – 188 с.
178. Янов Ю. И. О логических схемах алгоритмов / Ю. И. Янов // Проблемы кибернетики ; [Сб. статей]. – М. : Физматгиз, 1958. – Вып. 1. – С. 75-127.
179. Яхьяева Г. Э. Основы теории нечетких множеств [Электронный ресурс] / Г. Э. Яхьяева. – Режим доступа: www.intuit.ru
180. Barabási A.-L. Emergence of scaling in random networks / A.-L.Barabási, R. Albert // Science. — 1999. —Vol. 286.— P. 509-512
181. Bosgra O. H. Design Methods For Control Systems [Электронный ресурс] / Bosgra O. H., Kwakernaak H., Meinsma G. // Dutch Institute of Systems and Control, 2003. – 25 p. Режим доступа: <http://wwwhome.math.utwente.nl/meinsmag/dmcs/dmcs0405.pdf>
182. Albatros. Каталог продукції, випуск 11.0, Москва. Режим доступу: <http://www.albatros.ru>
183. Alfa-Prom. Автоматизация вертикальных автоклавов [Электронный ресурс] // Альфа-Пром. Промышленная автоматизация и энергосбережение. Пищевая промышленность, автоклавы. – Режим доступа: http://www.alfa-prom.ru/art_4_aut_vert_avtoklavov.html
184. Caicong Wu Decision-making modeling method based on artificial neural network and data envelopment analysis / Wu Caicong, Chen Xiuwan, Yang Yinsheng // Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International Volume 4. – 2004. – P. 2435 – 2438.
185. Dubovoy V. Algorithmic models of systems in conditions of uncertainty / V. Dubovoy, O. Nikitenko // „ІНТЕРНЕТ – ОСВІТА – НАУКА – 2004”: Четверта міжнар. конф., 5-9 жовт. 2004 р.: матер. конф. – Вінниця : УНІВЕРСУМ-Вінниця, 2004. – Том 2. – С. 538-541.
186. Dubovoy V. Decision making in combined uncertainty conditions/

- V. Dubovoy, O. Kovalyuk // Збірник Четвертої міжнародної конференції „ІНТЕРНЕТ – ОСВІТА – НАУКА – 2004”. Том 2. – Вінниця : УНІВЕРСУМ-Вінниця, 2004. – С. 535-537.
187. Erdős P. On the evolution of random graphs / P. Erdős, A. Rényi // Publications of the Mathematical Institute of the Hungarian Academy of Sciences. – 1960. – Vol. 5. – P. 17-61
 188. Fuzzy logic [Електронний ресурс] / Wikipedia. The Free Encyclopedia. – Режим доступа: http://en.wikipedia.org/wiki/Fuzzy_logic
 189. Frank O. Markov graphs / O. Frank, D. Strauss // Journal of the American Statistical Association. – 1986. – Vol. 81. – P. 832-842.
 190. Gkantsidis C. The markov chain simulation method for generating connected power law random graphs / C. Gkantsidis, M. Mihail, E. Zegura // Fifth Workshop on Algorithm Engineering and Experiments, January 11, 2003, Baltimore, Maryland, USA, Proceedings. SIAM. – 2003. – P. 16-25
 191. Holland P. W. An exponential family of probability distributions for directed graphs / P. W. Holland, S. Leinhardt // Journal of the American Statistical Association. – 1981. – Vol. 76. – P. 33-65.
 192. Kahneman D. Judgment under uncertainty: Heuristics and Biases / D. Kahneman, P. Slovic, A. Tversky // Cambridge: Cambridge University Press, 1982. – P. 1124-1131.
 193. Kahneman D. Prospect Theory: an Analysis of Decisions under Risk / D. Kahneman, A. Tversky // Econometrica. – 1979. – № 47. – P. 313-327.
 194. Kosko B. Fuzzy Systems as Universal Approximators / B. Kosko // IEEE Trans. on Computers. – 1994. - Vol. 43. – №11. – P. 1329 -1333.
 195. Lee T. S. Asynchronous, distributed, decision-making systems with semi-autonomous entities: a mathematical framework / T. Lee, S. Ghosh, A. Neerode // Systems, Man and Cybernetics, Part B, IEEE Transactions on Volume 30, Issue 1, Feb. – 2000. - P.229 – 239.
 196. Molloy M. A critical point for random graphs with a given degree sequence / M. Molloy, B. Reed // Random Structures Algorithms. – 1995. – Vol. 6. – P. 161-179.
 197. Newman M. E. J. The Structure and Function of Complex Networks / Newman M.E.J // SIAM Review. – 2003. – Vol. 45, N. 2. – P. 167-256.
 198. Ozen T. Modelling the variation in human decision making / T. Ozen,

- J. Garibaldi, S. Musikasuwan // Fuzzy Information. Processing NAFIPS '04. IEEE. – 2004. – Vol. 2. – P. 617–622.
199. Price D. J. A general theory of bibliometric and other cumulative advantage processes / D.J.Price // Journal of the American Society for Information Science. – 1976. – Vol. 27. – P. 292-306.
200. Prosoft. Передові технології автоматизації. Каталог продукції 8.0. – Режим доступу: <http://www.prosoft.ru>
201. Rapoport A. Contribution to the theory of random and biased nets / A. Rapoport // Bulletin of Mathematical Biophysics. – 1957. – Vol. 19. – P. 257-277.
202. SCADA-системи, или муки выбора [Электронный ресурс] // Средства и системы компьютерной автоматизации. – Режим доступа: <http://www.asutp.ru/?p=600055>
203. The Measurement and Automation. National Instruments. – Режим доступу: <http://www.ni.com>
204. Yager R. R. Fuzzy Decision Making Including Unequal Objectives / R. R. Yager // Fuzzy Sets and Systems. – 1978. – P. 87-95.

Наукове видання

**Володимир Михайлович Дубовой
Олена Дмитрівна Никитенко**

**ОПТИМІЗАЦІЯ ПІДСИСТЕМ ЗБОРУ ДАНИХ АСУТП
В УМОВАХ КОМБІНОВАНОЇ НЕВИЗНАЧЕНОСТІ**

Монографія

Редактор

Оригінал-макет підготовлено О. Д. Никитенко

Видавництво ВНТУ «УНІВЕРСУМ-Вінниця»
Свідоцтво Держкомінформу України
серія ДК № 746 від 25.12.2001
21021, м. Вінниця, Хмельницьке шосе, 95
ВНТУ, ГНК, к. 114
Тел. (0432) 59-85-32

Підписано до друку 28.05.2008 р.
Формат 29,7×42¼ Папір офсетний
Гарнітура Times New Roman
Друк різнографічний Ум. др. арк. 10,68
Наклад 100 прим. Зам № 2008-071

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького національного технічного університету
Свідоцтво Держкомінформу України
серія ДК № 746 від 25.12.2001 р.
21021, м. Вінниця, Хмельницьке шосе, 95
ВНТУ, ГНК, к. 114
Тел. (0432) 59-81-59