


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
приладобудівний факультет
кафедра комп'ютерно-інтегрованих технологій виробництва приладів

До захисту допущено

Завідувач кафедри

 Наталія Стельмах
«16» грудня 2024 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Комп'ютерно-інтегровані системи та
технології в приладобудуванні»
зі спеціальності 174- Автоматизація, комп'ютерно-інтегровані технології та
робототехніка
на тему: «Розроблення методів та алгоритмів автоматизованого контролю та
діагностування технологічних процесів»


Виконав:

Студент II курсу, групи ПБ-31мп
Нестеренко Андрій Вячеславович



Науковий керівник:

Доцент кафедри комп'ютерно-інтегрованих
Технологій виробництва приладів к.т.н., доцент
Філіппова Марина В'ячеславівна



Консультант з розробки стартап-проєкту:

д.е.н., проф., завідувач кафедри
економічної кібернетики КПІ ім. Ігоря Сікорського
Бояринова Катерина Олександрівна



Рецензент:

Професор кафедри комп'ютерно-інтегрованих
Оптичних та нвігаційних систем д.т.н., професор
Микитенко Володимир Іванович



Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент 

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Приладобудівний факультет

Кафедра комп'ютерно-інтегрованих технологій виробництва приладів


Рівень вищої освіти – другий (магістерський)

Спеціальність – 174- Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувач кафедри

 Наталія СТЕЛЬМАХ

«16» вересня 2024 р.

ЗАВДАННЯ

на магістерську дисертацію студенту



Нестеренку Андрію Вячеславовичу

1. Тема дисертації «Розроблення методів та алгоритмів автоматизованого контролю та діагностування технологічних процесів», науковий керівник дисертації Філіппова Марина В'ячеславівна, канд. техн. наук, доцент, затверджені наказом по університету від «07» листопада 2024 р. № 4987-с
2. Термін подання студентом завершеної дисертації 20 грудня 2024 року
3. Об'єкт дослідження: технологічний процес дрібносерійного багатоменклатурного виробництва.
4. Вихідні дані: технологічний процес дрібносерійного багатоменклатурного виробництва.
5. Перелік завдань, які потрібно розробити: 1. Аналіз існуючих систем моніторингу та класифікація відповідно до концепції розумного виробництва. 2. Розробка методики діагностування технологічного процесу дрібносерійного багатоменклатурного виробництва. 3. Розроблення структури цифрової моделі технологічного процесу. 4. Розробка алгоритмів та методів діагностування технологічного процесу. 5. Розробка моделі інформаційної системи, проведення імітаційного моделювання.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація за результатами проведених досліджень


7. Орієнтовний перелік обов'язкових публікацій: тези доповідей на науково-технічних, наукових конференціях, стаття у фаховому виданні категорії Б..

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стартап-проекти	Завідувач кафедрою економічної кібернетики, д.е.н., проф. Бояринова К.О.		

9. Дата видачі завдання 16 вересня 2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Аналіз існуючих систем моніторингу та класифікація відповідно до концепції розумного виробництва	27.09.2024	
2.	Розробка методики діагностування технологічного процесу дрібносерійного багатомономенклатурного виробництва	11.10.2024	
3.	Розроблення структури цифрової моделі технологічного процесу	25.10.2024	
4.	Розробка алгоритмів та методів діагностування технологічного процесу	08.11.2024	
5.	Розробка моделі інформаційної системи, проведення імітаційного моделювання	22.11.2024	
6.	Розробка стартап проекту	06.12.2024	
7.	Оформлення пояснювальної записки і презентації	18.12.2024 р.	
8.	Представлення магістерської дисертації до захисту	20.12.2024 р.	

Студент

Науковий керівник




Андрій НЕСТЕРЕНКО

Марина ФІЛІПОВА

РЕФЕРАТ

Обсяг роботи: 122

Кількість ілюстрацій: 46

Кількість таблиць: 29

Кількість джерел з переліком посилань: 22

Актуальність теми

Технічні системи піддаються впливу багатьох зовнішніх і внутрішніх факторів, які є причинами повільного псування виробів і можливих збоїв у роботі систем. Можна вважати, що всі несправності технічних систем, які виникають при нормальній роботі, є небажаними і шкідливими. Несправності або знижують якість виробленої продукції, або, в гіршому випадку, остаточно переривають виробничий процес, що призводить до вимірних економічних втрат. Значення втрат іноді настільки величезне, що зростаючий інтерес до ефективних методологій діагностики зрозумілий. Найбільш серйозні труднощі з розробкою хорошої діагностичної методології для технічних систем: зростаюча складність їх структур і різноманітність використовуваних технологій. Ці два фактори дуже ускладнюють процес діагностики навіть для добре підготовлених спеціалістів, збільшують час, необхідний для діагностики та відновлення, а також ускладнюють діагностичні процедури.

Зважаючи на вищезазначені факти, розробка ефективних методологій діагностики стає дуже важливим завданням. Ефективна діагностична методологія повинна мати можливість підтримувати діагностичні дії людини або навіть проводити діагностику досить автоматично.

Мета і завдання дослідження

Метою дисертаційної роботи є розроблення методів та алгоритмів автоматизованого контролю та діагностування технологічних процесів з метою покращення швидкості діагностичних процедур, за допомогою побудови ієрархічних структур систем, підвищення точності процесу діагностування за рахунок верифікації етапів діагностики.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Проаналізувати існуючі підходи при створенні ефективних систем діагностики, зокрема з використанням засобів штучного інтелекту.
2. Представити формальну концепцію загальної моделі для моделювання ієрархічних структур складних систем для використання в діагностиці.
3. Розробити методологію ієрархічної діагностики.
4. Розробити алгоритми виправлення несправностей.
5. Провести моделювання роботи алгоритму.
6. Розробити стартап проєкт

Об'єкт дослідження

Системи моніторингу та діагностування технологічних процесів

Предмет дослідження

Метод ієрархічного діагностування несправностей обладнання, алгоритм корекції систем діагностування

Методи дослідження

Моделювання в середовищі Matlab/Simulink, математичні методи аналізу систем, методи штучного інтелекту для аналізу та верифікації діагностичних процедур.

Наукова новизна одержаних результатів

Розроблено алгоритм діагностування технологічного процесу з використанням ієрархічних процедур (процедур з ефектом «фокусу») на основі застосування аналізу залишків, що дозволяє скоротити час виконання діагностичних процедур та підвищити точність пошуку несправностей.

Практичне значення одержаних результатів

Розроблено ієрархічний алгоритм аналізу несправностей; розроблено метод точкового аналізу з функцією фокусу; проведено моделювання та дослідження розробленого алгоритму на процесі електрогенерації.

Апробація результатів дослідження

Нестеренко А. Алгоритми автоматизованого контролю та діагностування технологічних процесів / Нестеренко А. // VII МСНТК, 25-26 квітня 2024 року. — Т. : ТНТУ, 2024. — С. 267–268.

Нестеренко А. Автоматизоване діагностування технологічних процесів методами ієрархічного моделювання та діагностичних процедур // Контроль і управління в складних системах (КУСС-2024). ВНТУ. Вінниця. – 2024.

Автоматизована система діагностування технологічних процесів / Нестеренко А.В., Філіппова М.В. // Збірник праць XX Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні». 04-05 грудня 2024 р. – К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2024.

Публікації

Нестеренко А. Алгоритми автоматизованого контролю та діагностування технологічних процесів / Нестеренко А. // VII МСНТК, 25-26 квітня 2024 року. — Т. : ТНТУ, 2024. — С. 267–268.

Нестеренко А. Автоматизоване діагностування технологічних процесів методами ієрархічного моделювання та діагностичних процедур // Контроль і управління в складних системах (КУСС-2024). ВНТУ. Вінниця. – 2024.

Автоматизована система діагностування технологічних процесів / Нестеренко А.В., Філіппова М.В. // Збірник праць XX Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні». 04-05 грудня 2024 р. – К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2024.

Нестеренко А.В., Філіппова М.В. Застосування систем діагностування для виявлення несправностей в системах керування технологічними процесами. Вимірювальна та обчислювальна техніка в технологічних процесах. № 4 (2024). (Подано до друку).

Ключові слова: діагностування, моделювання, несправність, керування, перехідний процес

ABSTRACT

Volume of work: 122

Number of illustrations: 46

Number of tables: 29

Number of sources with a list of references: 22

Actuality of theme

Technical systems are exposed to many external and internal factors that are the causes of slow deterioration of products and possible failures in the operation of systems. It can be assumed that all malfunctions of technical systems that occur during normal operation are undesirable and harmful. Malfunctions either reduce the quality of the manufactured products, or, in the worst case, completely interrupt the production process, which leads to measurable economic losses. The value of losses is sometimes so enormous that the growing interest in effective diagnostic methodologies is understandable.

The most serious difficulties in developing a good diagnostic methodology for technical systems: the increasing complexity of their structures and the diversity of technologies used. These two factors greatly complicate the diagnostic process even for well-trained specialists, increase the time required for diagnosis and recovery, and also complicate diagnostic procedures.

Given the above facts, the development of effective diagnostic methodologies becomes a very important task. An effective diagnostic methodology should be able to support human diagnostic actions or even perform diagnostics fairly automatically.

The purpose and tasks of the research

The purpose of the dissertation is to develop methods and algorithms for automated control and diagnostics of technological processes in order to improve the speed of diagnostic procedures by building hierarchical structures of systems, increasing the accuracy of the diagnostic process by verifying the diagnostic stages.

To achieve the goal, it is necessary to solve the following tasks:

1. Analyze existing approaches to creating effective diagnostic systems, in particular using artificial intelligence tools.

2. Present a formal concept of a general model for modeling hierarchical structures of complex systems for use in diagnostics.
3. Develop a methodology for hierarchical diagnostics.
4. Develop algorithms for troubleshooting.
5. Conduct simulation of the algorithm.
6. Develop a startup project

Object of study

Systems for monitoring and diagnosing technological processes

Subject of research

Method of hierarchical diagnostics of equipment malfunctions, algorithm for correcting diagnostic systems

Research methods

Modeling in the Matlab/Simulink environment, mathematical methods of system analysis, artificial intelligence methods for analyzing and verifying diagnostic procedures.

Scientific novelty of the obtained results

A diagnostic algorithm for a technological process using hierarchical procedures (procedures with a “focus” effect) has been developed based on the use of residual analysis, which allows reducing the execution time of diagnostic procedures and increasing the accuracy of fault finding.

Practical significance of the obtained results

A hierarchical fault analysis algorithm has been developed; a point analysis method with a focus function has been developed; the developed algorithm has been simulated and studied in the power generation process.

Approbation of research results

Nesterenko A. Algorithms for Automated Control and Diagnostics of Technological Processes / Nesterenko A. // VII International Scientific and Technical Conference, April 25-26, 2024. — Ternopil: TNTU, 2024. — pp. 267–268.

Nesterenko A. Automated Diagnostics of Technological Processes by Methods of Hierarchical Modeling and Diagnostic Procedures // Control and Management in Complex Systems (CMCS-2024). VNTU. Vinnytsia. – 2024.

Nesterenko A.V., Filippova M.V. Automated System for Diagnosing Technological Processes // Proceedings of the XX All-Ukrainian Scientific and Practical Conference of Students, Postgraduates, and Young Scientists "Efficiency and Automation of Engineering Solutions in Instrumentation". December 4-5, 2024. – Kyiv: IPM, Igor Sikorsky KPI. – 2024.

Publications

Nesterenko A. Algorithms for Automated Control and Diagnostics of Technological Processes / Nesterenko A. // VII International Scientific and Technical Conference, April 25-26, 2024. — Ternopil: TNTU, 2024. — pp. 267–268.

Nesterenko A. Automated Diagnostics of Technological Processes by Methods of Hierarchical Modeling and Diagnostic Procedures // Control and Management in Complex Systems (CMCS-2024). VNTU. Vinnytsia. – 2024.

Nesterenko A.V., Filippova M.V. Automated System for Diagnosing Technological Processes // Proceedings of the XX All-Ukrainian Scientific and Practical Conference of Students, Postgraduates, and Young Scientists "Efficiency and Automation of Engineering Solutions in Instrumentation". December 4-5, 2024. – Kyiv: IPM, Igor Sikorsky KPI. – 2024.

Nesterenko A.V., Filippova M.V. Application of Diagnostic Systems for Detecting Faults in Technological Process Control Systems. Measuring and Computing Technology in Technological Processes. No. 4 (2024). (Submitted for publication).

Keywords: diagnostics, modeling, fault, control, transient process

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	12
ВСТУП.....	13
1. АНАЛІЗ НАПРАЦЮВАНЬ В ГАЛУЗІ ПРОВЕДЕННЯ ДІАГНОСТИЧНИХ ПРОЦЕДУР	14
1.1 Аналіз методів та факторів, які визначають якість та надійність проведення діагностичних процедур.....	14
1.2. Проблемний аналіз сфери діагностування.....	17
1.3. Огляд діагностичних методик обладнання технологічного процесу	19
1.3.1. Діагностичні матриці	20
1.3.2. Древа несправностей	22
1.3.3. Системи підтримки істини.....	23
1.3.4. Експертні системи.....	24
1.3.5 Міркування на основі випадків	25
1.3.6. Обмеження та абстракція обмежень	26
1.3.7. Ієрархічні діагностичні матриці.....	29
1.4 .Причинно-наслідкові графіки І/АБО/НІ.....	31
1.4.1. І/АБО/НІ графи. Основні визначення	32
1.4.2. Причинно-наслідкова діагностична методика.....	36
Висновки до розділу 1.....	40
2 СИТЕМИ ВИЯВЛЕННЯ ТА УСУНЕННЯ НЕСПРАВНОСТЕЙ.....	41
2.1 Базова модель діагностування DX.....	41
2.2. Діагностика на основі моделі FDI.....	47
2.3. Ієрархічна діагностика в гетерогенних середовищах.....	51
Висновки до розділу 2.....	61
3 РОЗРОБКА ДІАГНОСТИЧНИХ АЛГОРИТМІВ ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ.....	62
3.1. Діагностичний алгоритм	62
3.1.1. Алгоритм локальної діагностики	62
3.1.2. Глобальний діагностичний алгоритм.....	67

3.2. Діагностичний процес	72
3.2.1. Виявлення несправностей	72
3.2.2. Локалізація проблем.....	73
3.2.3. Верифікація діагнозів	74
3.3. Приклади використання ієрархічного діагностування	79
3.3.1. Графічне зображення діагнозів	79
3.3.2. Концептуальна система для ієрархічного моделювання	80
3.3.3. Приклад діагностики системи трьох ємностей.....	82
3.3.4 Застосування систем діагностування для виявлення несправностей при діагностування технологічного процесу енергогенерації	86
Висновки до розділу 3.....	98
4. РОЗРОБКА СТАРТАП ПРОЄКТУ «РОЗРОБЛЕННЯ МЕТОДІВ ТА АЛГОРИТМІВ АВТОМАТИЗОВАНОГО КОНТРОЛЮ ТА ДІАГНОСТУВАННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ»	99
4.1 Опис та технологічний аудит ідеї стартап-проекту	99
4.2 Аналіз ринкових можливостей запуску стартап проекту.....	101
4.3. Розроблення ринкової та маркетингової програми стратегії проекту	107
4.4 Економічна ефективність стартап проєкту.....	113
Висновки до розділу 4.....	118
ВИСНОВКИ	119
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	120

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

FIS – (Fault Information System) система інформації про несправності;

IS – (Information System) інформаційна система;

FTA – (Fault tree analysis) Дерево аналізу несправностей;

FDI – (Fault Detection and Isolation) Виявлення та усунення несправностей;

ATSM – (Assumption-based Truth Maintenance System) системи підтримки істини, засновані на припущеннях;

NEC – (necessary) необхідний;

SUF – (sufficient) достатній;

MAY – (maybe) можливий;

ARR – (analytical redundancy relation) аналітичне відношення надмірності.

ВСТУП

Історія діагностики, ймовірно, така ж стара, як історія людства. Найпростіше термін «діагностика» можна визначити як дослідження систем з метою виявлення та локалізації їх несправностей. Спочатку був лише один об'єкт дослідження: сама людина. Діагностичні процедури зводилися до використання експертних знань фахівця. Але з розвитком технічної цивілізації область діагностики розширилася і почала охоплювати технічні системи та технологічні установки.

Технічні системи піддаються впливу багатьох зовнішніх і внутрішніх факторів, які є причинами повільного псування виробів і можливих збоїв у роботі систем. Можна вважати, що всі несправності технічних систем, які виникають при нормальній роботі, є небажаними і шкідливими. Несправності або знижують якість виробленої продукції, або, в гіршому випадку, остаточно переривають виробничий процес, що призводить до вимірних економічних втрат. Значення втрат іноді настільки величезне, що стає зрозумілим зростаючий інтерес до ефективних методологій діагностики.

Найбільш серйозні труднощі з розробкою діагностичної методології для технічних систем: зростаюча складність їх структур і різноманітність технологій, які використовуються. Ці два фактори дуже ускладнюють процес діагностики навіть для добре підготовлених фахівців, збільшують час, необхідний для діагностики та відновлення, а також ускладнюють діагностичні процедури.

Зважаючи на вищезазначені факти, розробка ефективних методологій діагностики стає дуже важливим завданням. Ефективна діагностична методологія повинна мати можливість підтримувати діагностичні дії людини або навіть проводити діагностику досить автоматично.

В даній магістерській дисертації розглянуто методи та алгоритми автоматизованого контролю та діагностування технологічних процесів

1. АНАЛІЗ НАПРАЦЮВАНЬ В ГАЛУЗІ ПРОВЕДЕННЯ ДІАГНОСТИЧНИХ ПРОЦЕДУР

1.1 Аналіз методів та факторів, які визначають якість та надійність проведення діагностичних процедур.

Типову діагностичну процедуру можна розділити на наступні три основні етапи:

виявлення збоїв у роботі системи;

локалізація виявлених збоїв у роботі системи;

виявлення несправності, оцінка розміру та типу локалізованого порушення поведінки.

Загалом, ідея виявлення несправностей базується на порівнянні поточної поведінки системи з очікуваною поведінкою системи. Якщо є помітні відмінності, то можна припустити, що система не працює належним чином. Іншими словами, спостережувана поведінка системи не узгоджується з очікуваною.

Наступним кроком є усунення несправності; локалізовано частини системи, відповідальні за спостережувані порушення. Неправильна поведінка точно описується під час ідентифікації несправності. Ізоляцію несправності та ідентифікацію несправності іноді об'єднують і називають діагностикою несправності.

Існуючі методи діагностики можна розділити за кількома різними критеріями. Найбільш загальні з них: тип діагностичних знань і тип пошукової стратегії [1]. Нижче діагностичні методології згруповані відповідно до типу діагностичних знань, які описують поведінку системи або зв'язки між симптомами та причинами дефектів:

1. Системи описуються добре розробленою моделлю. Цей клас систем можна діагностувати найбільш точно і повно, але лише дуже обмежена кількість систем має модель, яка є достатньо хорошою для такої діагностики. Найбільш типові рішення, що застосовуються на практиці, зазвичай базуються на:

- Аналітичних моделях, таких як:
 - фізичні рівняння;
 - лінійні моделі вводу-виводу (передаточна функція);
 - лінійні рівняння стану;
 - спостерігачі станів і фільтри Калмана.
- Моделях, заснованих на знаннях.

2. Системи описуються тільки загальними правилами. Знання отримані від експертів у галузі. Прикладні рішення базуються головним чином на інженерії знань та обчислювальному інтелекті:

- випадкові графи,, діагностичні матриці;
- банкомати, наприклад;
- діагностичні дерева та графи,
- класичні експертні системи на основі правил штучного інтелекту.

3. Системи, які не описані жодною моделлю чи правилами; усі наші знання про системи базуються лише на спостереженнях за їх поведінкою. Прикладні рішення:

- розпізнавання образів;
- нейронні мережі;
- нечіткі нейронні мережі.

Найважливішими для нас є підходи, похідні від методів логічного міркування. Зокрема моделі, засновані на знаннях, і причинно-наслідкові методології. Це так, оскільки точні математичні моделі доступні лише для обмеженої кількості систем, тоді як методи інженерії знань і обчислювального інтелекту в принципі застосовні до будь-якої системи довільної складності (хоча рівень точності може змінюватися залежно від складності системи).

Моделі, засновані на знаннях, для діагностики використовують невідповідності між спостережуваною та очікуваною поведінкою системи, де очікувана поведінка передбачена відповідно до знань про модель системи. Цей тип підходів також називають діагностикою на основі послідовності. Такі діагностичні системи не потребують етапу отримання експертних знань або навчання і можуть діагностувати навіть дуже нові системи, для яких експертні

діагностичні знання, засновані на попередньому досвіді, не існують. Для таких систем для діагностики можуть бути використані існуючі моделі їх коректної поведінки, розроблені при проектуванні та моделюванні таких систем.

Основна ідея представлених подальших діагностичних моделей базується на компонентах, де система визначається як набір компонентів і відносин між ними [2]. Діагностика такої складової системи проводиться в два етапи: виявлення несправності та усунення несправності. Модель системи має кілька обраних входів і виходів. Значення вхідних даних системи вимірюються та використовуються як вхідні дані для моделі. Значення вихідних даних моделі розраховуються та порівнюються з результатами спостереження системи. Якщо є значні розбіжності, то виявлено несправність системи. Загальна ідея такого модельного підходу представлена на рисунку 1.1.

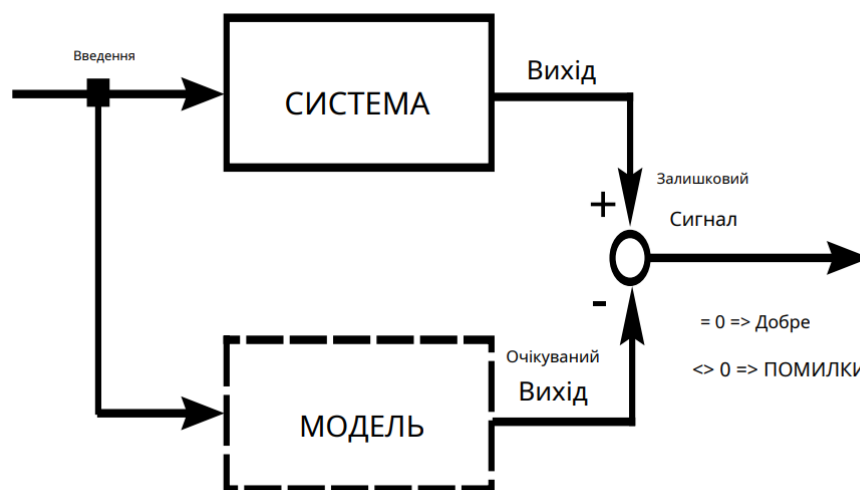


Рисунок – 1.1. Загальна ідея діагностики на основі послідовності.

У разі виявлення неправильної поведінки необхідно виконати ізоляцію несправності. Ідентифікуються набори компонентів, які не можуть працювати належним чином (так звані конфліктні набори). Конфліктні набори включають компоненти, які беруть участь у генеруванні виходів системи за якими були виявлені невідповідності. Елементи наборів конфліктів є природними кандидатами на несправність (принаймні один з них), і потенційний діагноз повинен включати принаймні один компонент із кожного набору конфліктів.

З іншого боку, причинно-наслідкові методології моделюють причинно-наслідкові зв'язки, які описують правильну та неправильну поведінку досліджуваних систем. Основні логічні функтори (І, АБО, НІ) зазвичай використовуються для моделювання зв'язків між симптомами та причинами несправностей. Простота цього підходу є його найбільшою перевагою, але іноді цей формалізм є занадто простим для опису більших систем.

Експертні логічні графи є прикладом каузальної методології. Вони являють собою графічні зображення причинно-наслідкових зв'язків [3]. Експертний граф визначається як ациклічний орієнтований граф із вузлами, що моделюють симптоми, і дугами, що відносяться до причинно-наслідкових зв'язків. Типи вузлів у таких графах можуть бути: AND (кон'юнкція), OR (диз'юнкція) і NOT (заперечення). Вузли AND представляють логічний кон'юнктивний вплив вузлів-попередників на змодельований симптом, вузли OR представляють диз'юнктивний вплив, тоді як дуги NOT представляють негативний (зворотний) вплив. Діагнози генеруються за допомогою простих правил для поширення станів вузлів через експертний граф у поєднанні з пошуком логічних значень вхідних вузлів, які передбачають спостережувану неправильну поведінку. Вузол AND представлений на малюнках у вигляді вузла з додатковою дугою під вузлом, OR є чистим вузлом графа, а дуга NOT представлена дугою з чорною крапкою.

1.2. Проблемний аналіз сфери діагностування.

Діагностування сучасних технологічних систем стає дедалі складнішим і складнішим завданням. Головним чином це пов'язано з їхньою складністю - сучасні технологічні системи складаються з численних компонентів, які взаємодіють і рекурсивно включають інші компоненти.

Діагностичні описи на основі згаданого в попередньому розділі методу діагностики зазвичай плоскі. Це означає, що всі компоненти аналізуються одночасно як потенційне джерело несправностей. Коли кількість компонентів збільшується, то збільшується і складність діагностичного обґрунтування, і час

розрахунку, незалежно від необхідної точності діагнозів і потенційного впливу нових компонентів на спостережувані симптоми. Однорівнева діагностична процедура генерує багато потенційних діагнозів, які неможливо перевірити за короткий час.

Проблему збільшення складності можна вирішити шляхом ієрархізації моделі, де позиція у вертикальній ієрархії представляє певний рівень розглянутих деталей. Ієрархічні підходи для вирішення діагностичних проблем розглядалися лише декількома авторами [4], теоретичний опис ієрархічних підходів, ієрархічний підхід, заснований на субсумпції між пунктами першого порядку або ієрархічна діагностика на основі обмежень, але жоден із цих підходів не виявився застосовним на практиці. Нещодавній підхід також представлений у [5] у контексті ПШ, заснований на використанні визначеної експертом діагностичної матриці та її декомпозиції на матриці підсистем.

У цій роботі розглядається інший, модельно-ієрархічний підхід. Він базується на прямому моделюванні ієрархії компонентів системи, яка може рекурсивно включати інші компоненти (рис. 1.2). Процедура діагностики може стосуватися певного рівня представлення компонентів. Система, яка підлягає діагностиці, структурована в ієрархії компонентів відповідно до її структури та функціональності.

Зауважимо, що деякі компоненти складної системи можуть мати гарний діагностичний опис, а деякі з них можуть мати лише частковий опис або навіть не мати жодного, наприклад, деякі компоненти надходять від інших виробників або представляють зовсім інший спосіб роботи, ніж інші. Якщо ми подивимося на представлену раніше класифікацію, то побачимо, що існують підходи, що випливають із класичної теорії керування, а також багато підходів, що належать до класу «м'яких обчислень». Отже, необхідно знайти спосіб одночасного дослідження різних діагностичних моделей для генерації діагнозів для настільки різних компонентів в тій самій ієрархічній моделі.

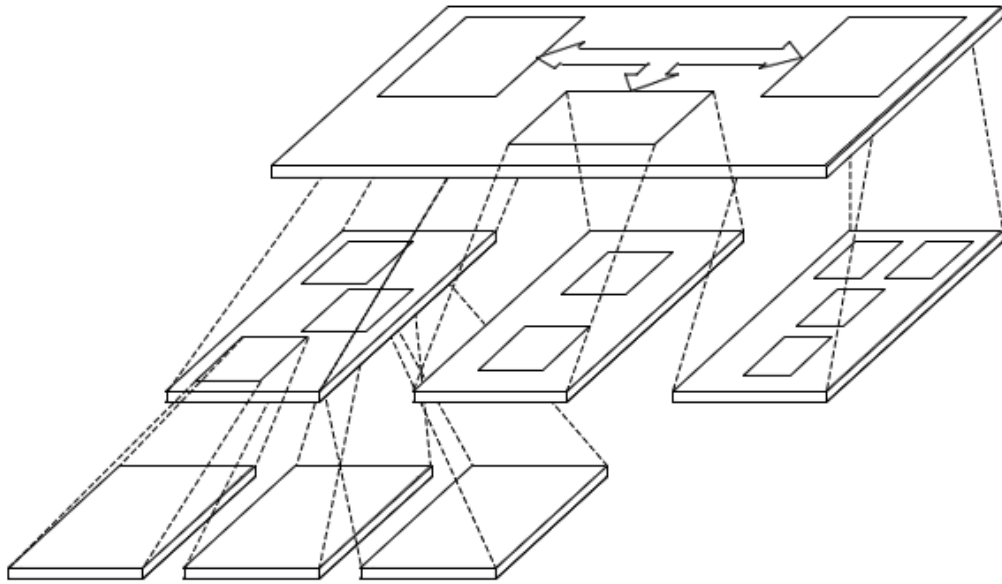


Рисунок – 1.2. Абстрактний приклад ієрархічної моделі.

Для реалізації роботи необхідно вирішити наступні завдання:

Провести аналіз існуючих підходів штучного інтелекту до діагностики, особливо підходів на основі узгодженості та причинно-наслідкових зв'язків.

Визначити формальну концепцію загальної моделі для моделювання ієрархічних структур.

Розробка методології ієрархічної діагностики.

Розробити концептуальне програмне забезпечення, що підтримує ієрархічне моделювання.

1.3. Огляд діагностичних методик обладнання технологічного процесу

Діагностичні методології виникають майже з усіх областей сучасної інформ У цьому розділі представлені лише деякі підходи до діагностики, вибрані з багатьох інших. Критерії відбору включають в основному тип діагностичних процедур і те, наскільки діагностичний висновок близький до методів, що використовуються в класичній логіці. Представлені підходи в основному є підходами ізоляції несправностей автоматики та теорії управління.

1.3.1. Діагностичні матриці

Двійкові діагностичні матриці [6] становлять інструменти, що належать до групи заходів із виявлення несправностей. У цьому підході причини несправності визначаються на основі залишкових сигналів від алгоритму виявлення несправностей і знання про їх зв'язок із реальними причинами несправностей.

Ознайомимося з цим підходом детальніше. Позначимо через F набір можливих несправностей системи,

$$F = \{f_k : k = 1, 2, \dots, K\},$$

і нехай S – набір діагностичних (залишкових) сигналів, які є виходами алгоритмів виявлення,

$$S = \{s_j : j = 1, 2, \dots, J\}.$$

Діагностичне відношення R_{fs} , будучи стрижневим поняттям у представленому підході, визначається на декартовому добутку множин F і S , тобто

$$R_{fs} \in F \times S.$$

Відношення має матричну форму; матричний елемент $r(f_k, s_j)$ визначається таким чином:

$$r(f_k, s_j) \equiv v_j(f_k) = \begin{cases} 0 & \leftrightarrow \langle f_k, s_j \rangle \text{ not } \in R_{FS} \\ 1 & \leftrightarrow \langle f_k, s_j \rangle \in R_{FS} \end{cases}$$

Скажемо також, що набір діагнозів складається наступним чином:

$$DGN = \{f_k \in F : \forall_{j: s_j \in S} [v_j(f_k) = v_j]\}$$

Діагностичні матриці можуть бути визначені на основі залишкових рівнянь або експертних знань. Двійкова діагностична матриця для прикладу системи представлена в таблиці 1.1.

Базовий підхід із значеннями 0 і 1 можна розширити до включення більш детальних характеристик діагностичних сигналів. У цьому випадку значення

діагностичних сигналів можуть приймати більше двох значень (наприклад -1, 0, +1).

Таблиця 1.1. Бінарна діагностична матриця для системи з трьома резервуарами

S/F													
	1				1	1	1	1					
	1	1	1						1			1	
		1	1	1					1	1			1
			1	1						1	1		1

Потім можливе проведення процесу діагностики на основі системи інформації про несправності (FIS). Інформаційна система про несправності базується на інформаційній системі IS і визначається як чотиризалежна:

$$FIS = \langle F, S, V_s, r \rangle$$

де $V_s = \bigcup_{j:s_j \in S} V_j$ є набором усіх діагностичних сигналів, V_j , є доменом діагностичних сигналів s_j і r є функцією, що сигнал набуває співвідношення:

$$r: F \times S \rightarrow V_s$$

і

$$\forall f \in F, s \in S r(f, s) \in V_s$$

Кожній парі (f, s) присвоєно одне значення або набір значень. У першому випадку така система називається простою, а в другому – грубою. Приклад діагностичної матриці для грубої системи представлено в таблиці 1.2.

Основним недоліком діагностичних матриць є те, що їхні залишкові сигнали є дуже специфічними для типу системи, що діагностується, і матриці потрібно модифікувати, коли деякі частини системи змінюються.

Крім того, локалізація несправності для цієї методології в основному базується на експертних знаннях і може бути неоднозначною або неповною в деяких випадках.

Таблиця 1.2. Приклад системи інформації про несправності [5]

S/F	f_1	f_2	f_3	f_4	f_5	f_6	f_7
s_1	1	0	1	0	0	1	{0, 1}
s_2	0	-1	0	+1	-1	0	{0, +1, -1}
s_3	-1	+1	+1, -1	0	+1	+1	{0, +1, -1}
s_4	0	1, 2	0, 1	0	1, 2	1, 2	{0, 1, 2}
s_5	+1	0	+1	+1	0	+1, -1	{0, +1, -1}

1.3.2. Древа несправностей

Підхід дерева аналізу несправностей (FTA), розвинувся в аерокосмічній промисловості на початку 60-х років; в даний час це один з основних методів аналізу безпеки систем [7]. Він може передбачити найбільш ймовірні причини збою системи в разі поломки системи. Метою побудови дерева відмов є моделювання умов системи, які можуть призвести до небажаної події.

Дерево несправностей — це модель, яка графічно та логічно представляє різні комбінації можливих подій, як несправностей, так і нормальних, що відбуваються в системі та призводять до головної події. Приклад дерева несправностей представлено на рис. 1.3.

Термін «подія» означає динамічну зміну стану, яка відбувається з елементом системи. Подія несправності – це ненормальний стан системи. Нормальна подія — це подія, яка має статися. Використовуються різні символи подій:

Прямокутник визначає подію, яка є виходом логічного вентиля та залежить від типу логічного вентиля та вхідних даних логічного вентиля.

Коло визначає основну притаманну несправність елемента системи при експлуатації в межах його проектних специфікацій.

Ромб символізує невдачу, окрім основної невдачі, яка навмисно не розвивається далі.

Подія перемикання являє собою подію, яка, як очікується, відбудеться або ніколи не відбудеться через проектування та нормальні умови, такі як зміна фази в системі.

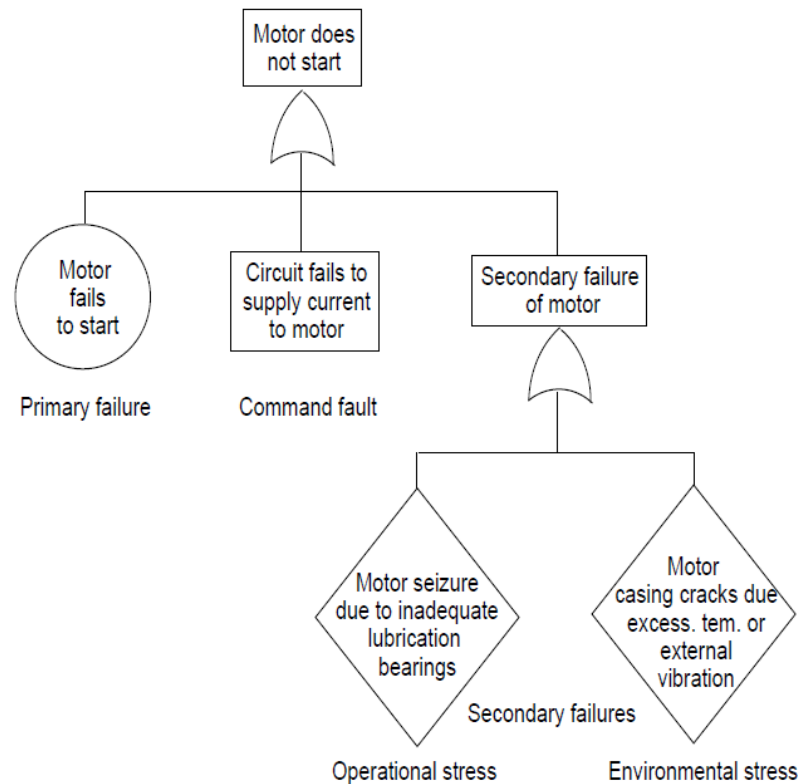


Рисунок – 1.3. Приклад дерева несправностей [4]

Основними логічними воротами для побудови дерева відмов є вентиля АБО та І. Вентилі АБО описують ситуацію, коли вихідна подія буде існувати, якщо існує одна або більше входних подій. Вентиль І описує логічну операцію, яка вимагає співіснування всіх входних подій для отримання вихідної події.

1.3.3. Системи підтримки істини

Системи підтримки істини (TSM) та системи підтримки правдивості на основі припущень (ATSM) використовувалися для вирішення проблем. TMS призначені для пошуку одного рішення, ATMS призначені для пошуку всіх рішень.

Контекст ATMS — це набір, сформований на основі припущень узгодженого середовища в поєднанні з усіма вузлами, які можна вивести з цих

припущень. Середовище, що характеризує контекст, — це набір припущень, з яких можна вивести вузол контексту. Мітка ATMS – це набір середовищ, пов'язаних із кожним вузлом.

Базовою структурою даних є вузол ATMS. Він містить інформацію про розв'язувач проблем, з яким він пов'язаний, обґрунтування, створені для нього розв'язувачем проблем, і мітку, обчислену для нього банкоматом. Усі вузли обробляються однаково та відрізняються лише індивідуальним шаблоном середовищ міток і обґрунтувань.

Існує чотири типи вузлів: передумови, припущення, передбачувані вузли та похідні вузли. ATMS пов'язує кожну інформацію з її контекстом. Якщо дані знаходяться в контексті, то вони також є в кожній супермножині (неузгоджені супермножини ігноруються). Три основні дії ATMS: створення вузла ATMS для даної системи вирішення проблем, створення припущення та додавання обґрунтування до вузла.

Мови якісного міркування та мови обмежень передбачають вибір між альтернативами. Вони використовують поширення обмежень для прогнозування поведінки пристрою на основі моделей компонентів і вимірювань схеми. Коли виникає несправність схеми, деякі її компоненти не працюють належним чином. Таким чином, в якийсь момент, оскільки правильна модель компонента не описує фактично несправний компонент, прогнози стануть суперечливими. Припускається, що окремі компоненти функціонують правильно. Протиріччя означає, що якесь припущення порушується. Таким чином, несправність локалізована в певному наборі компонентів. Найкраще наступне вимірювання – це те, яке надає максимальну інформацію щодо ще неперевірених припущень. Ця програма вимагає, щоб припущення, що дозволяють зробити діагностичний висновок, були явно доступними.

1.3.4. Експертні системи

Експертні системи зазвичай засновані на правилах з великою базою знань, включаючи правила «якщо-то». Експертні діагностичні знання кодифіковані в

цих правилах, пов'язуючи симптоми з основними несправностями. Подальша діагностика здійснюється шляхом прямого або зворотного міркування на основі спостережень і правил.

До основних етапів розробки експертної системи належать:

- отримання знань;
- вибір представлення знань;
- кодування знань у базі знань.

Цей підхід, подібний до дерев несправностей і ATMS, потребує етапу отримання знань і сильної перевірки отриманих знань (неповнота, непослідовність). Крім того, кількість правил швидко зростає зі складністю системи, що аналізується. Такі діагностичні системи дають збій, коли зустрічається нова, не визначена в базі знань умова, тобто знання, зібрані в правилах, описують радше досвід щодо конкретного пристрою, ніж загальні знання про структуру чи поведінку.

Призначення та перевірка систем на основі правил може підтримуватися спеціальним програмним забезпеченням. Серед них існує багато підходів до графічного представлення правил і залежностей.

1.3.5 Міркування на основі випадків

Основна ідея аргументації на основі випадків полягає в тому, що якщо дві проблеми подібні, то можна вирішити першу проблему, використовуючи адаптоване рішення другої проблеми. Отже, це форма міркування за аналогією. Набір раніше вирішених випадків становить базу знань. Коли виникає певна проблема, у базі знань шукають подібну ситуацію в минулому. Якщо така приблизно схожа ситуація знайдена, то її минуле рішення адаптується для поточної проблемної ситуації.

Обґрунтування на основі випадку успішно використовується в діагностиці. Для цього підходу потрібна фаза вивчення випадку, а форма знань може змінюватися залежно від діагностованої системи.

1.3.6. Обмеження та абстракція обмежень

Ієрархічна діагностика, заснована на обмеженнях, де абстракція досягається шляхом згортання значень, видалення змінних і спрощення рівнів. Абстракція системи будується там на основі обмежень між незалежними змінними (стани системи) і залежними змінними (спостереження входу-виходу). Більш абстрактні рівні мають більш загальні обмеження. Оператор — це ієрархічний оператор, який відображає змінні між двома рівнями. Ідея дворівневої моделі показана на рис. 1.4. Розрізняють абстрактний рівень і детальний рівень.

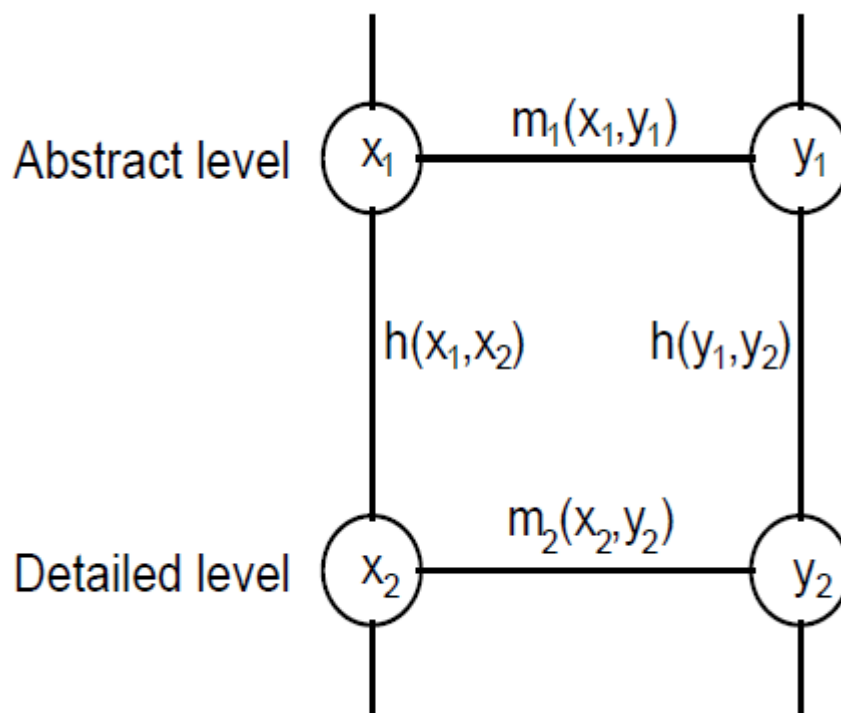


Рисунок – 1.4. Відношення абстракції [4].

Приклад. Модель вентилів АБО має абстрактний рівень і детальний рівень. Абстрактний рівень включає логічну диз'юнкцію двох входів. Значення входів і виходів логічний «0» або логічна «1». Обмеження на цьому рівні:

$$org1(1,1,1)$$

$$org1(1,0,1)$$

$$org1(0,1,1)$$

$$org1(0,0,0)$$

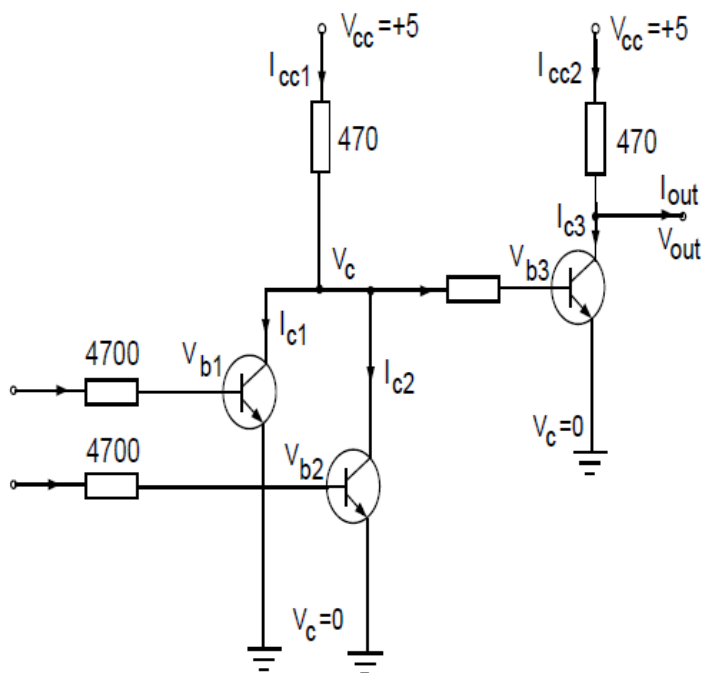


Рисунок – 1.5. Вентиль АБО, реалізований на трьох pnp транзисторах.

Детальний рівень являє собою електричне коло з деякими транзисторами та деякими пасивними елементами. Використовуючи нотацію, подібну до Пролога, модель схеми можна задати так:

```

org2(vi(Vin1, Iin1), vi(Vin2, Iin2), vi(Vout, Iout)) ←
Vcc = 5, Ve = 0,
resistor(Vin1, Vb1, Iin1, 4700),
transistor(Vb1, Vc, Ve, Iin1, Ic1, Ie1),
resistor(Vin2, Vb2, Iin2, 4700),
transistor(Vb2, Vc, Ve, Iin2, Ic2, Ie2),
resistor(Vcc, Vc, Icc1, 470),
Icc1 = Ic1 + Ic2 + Ib3,
resistor(Vc, Vb3, Ib3, 4700),
transistor(Vb3, Vout, Ve, Ib3, Ic3, Ie3),
resistor(Vcc, Vout, Icc2, 470),
Icc2 = Ic3 + Iout, 0 ≤ Iout, Iout ≤ 0.006.

transistor(Vb, Vc, Ve, Ib, Ic, Ie) ← %active
Vb = Ve + 0.7, %Vbe = 0.7
Vc ≥ Vb,
Ic = 100 × Ib, %Beta = 100
Ib ≥ 0,
Ie = Ic + Ib.

```

```

transistor( $V_b, V_c, V_e, I_b, I_c, I_e$ )  $\leftarrow$  %saturated
 $V_b = V_e + 0.7, \%V_{be} = 0.7$ 
 $V_c = V_e + 0.3, \%V_{cesat} = 0.3$ 
 $I_b \geq 0,$ 
 $I_c \geq 0,$ 
 $I_e = I_c + I_b.$ 

```

```

transistor( $V_b, V_c, V_e, I_b, I_c, I_e$ )  $\leftarrow$  %cutoff
 $V_b < V_e + 0.7, \%V_{be} = 0.7$ 
 $I_b = 0,$ 
 $I_c = 0,$ 
 $I_e = 0.$ 

```

```

resistor( $V_1, V_2, I, R$ )  $\leftarrow$ 
 $R > 0,$ 
 $V_1 - V_2 = I \times R.$ 

```

Ієрархічні зв'язки між змінними на окремих рівнях визначаються наступними двома пунктами:

```

 $h(0, vi(V, I)) \leftarrow 0 \leq V, V < 0.7.$ 
 $h(1, vi(V, I)) \leftarrow 2 \leq V, V \leq 5.$ 

```

Процес діагностики можна здійснити за допомогою наступного пункту:

```

org1( $In1, In2, Out$ )  $\leftarrow$ 
 $h(In1, VI_{in1}),$ 
 $h(In2, VI_{in2}),$ 
 $h(Out, VI_{out}),$ 
org2( $VI_{in1}, VI_{in2}, VI_{out}$ ).

```

Також необхідні дві умови узгодженості на цю абстракцію:

$$C1: \forall x_2, y_2 \ m_2(x_2 y_2) \rightarrow \neg \exists x_1 h(x_1, x_2) \vee \exists y_1 h(y_1, y_2)$$

$$C2: \forall x_2, y_2 \ (\exists x'_1, y'_1 \ m_2(x_2 y_2) \wedge h(x'_1, x_2) \wedge h(y'_1, y_2))$$

$$\rightarrow \exists x_1, y_1 \ m_1(x_1 y_1) \wedge h(x_1, x_2) \wedge h(y_1, y_2)$$

Умова $C1$ обмежує неповноту, введenu ієрархічними операторами, і забороняє випадки, коли детальна незалежна змінна x_2 з абстракцією x_1 відображається на детальну залежну змінну y_2 без абстракції. Умова необхідна, оскільки абстрактний рівень не включає деякі змінні або значення змінних, які знаходяться на детальному рівні. Таким чином, y_1 має існувати тоді, коли існує x_1 для збереження послідовності.

1.3.7. Ієрархічні діагностичні матриці

Діагностована система ділиться на h рівнів. Змінною M_h позначимо кількість підсистем на i -му рівні. Підсистема n на рівні описується наступним тривимірним виразом:

$$O_n^h = \langle F_n^h, S_n^h, R_{FS}^{h/n} \rangle,$$

де: F_n^h - це підмножина несправностей для підсистеми n в рівні h ;

S_n^h - є підмножиною діагностичних сигналів, що виявляють несправності підсистеми n в рівні h ;

$R_{FS}^{h/n}$ - рівень є їх діагностичним відношенням.

Наведена вище ієрархічна структура має такі властивості:

- підмножини діагностичних сигналів окремі у всіх підсистемах:

$$S_n^g \cap S_n^h = \emptyset, m, n = 1, \dots, M_h, m \neq n, g, h = 1, \dots, H,$$

- відношення несправності–діагностичні сигнали також окремі:

$$R_{FS}^{g/m} \cap R_{FS}^{h/n} = \emptyset, m, n = 1, \dots, M_h, m \neq n, g, h = 1, \dots, H,$$

- підмножини несправностей, виявлених у підсистемах на одному рівні ієрархії, є окремими:

$$F_m^h \cap S_n^h = \emptyset, m, n = 1, \dots, M_h, m \neq n, g, h = 1, \dots, H.$$

Ієрархічна структура діагностування описується графом:

$$GHSD = \langle O, L \rangle.$$

Підсистеми представлені вузлами графа

$$O = \{O_n^h: n = 1, \dots, M_h, h = 1, \dots, H\},$$

дуги визначаються як

$$L = \{l_{m,n}^{g,h}: m, n = 1, \dots, M_h, m \neq n, g, h = 1, \dots, H, g \neq h\}.$$

Дуги з'єднують підсистеми, які мають спільні елементи множини несправностей

$$l_{m,n}^{g,h} \in L \leftrightarrow [F_m^g \cap F_n^h \neq \emptyset],$$

спільну частину цих множин позначають

$$F_{m,n}^{g,h} = F_m^g \cap F_n^h, g \neq h.$$

Початкова діагностика підсистеми рівень h визначається як:

$$DGN_n^{h*} = \{f_k \in F_n^h: \forall_{j: S_j \in S_n^h} [v_j(f_k) = v_j]\}$$

Остаточний діагноз відрізняється, коли виявлені несправності належать лише підсистемі, і інший, коли виявлені несправності належать також іншим підсистемам. Множина F_n^{hW} містить лише помилки, виявлені в підсистемі:

$$F_n^{hW} = F_n^h - \bigcup_{l_{m,n}^{g,h}} F_{m,n}^{g,h}$$

Якщо $DGN_n^{h*} \subseteq F_n^{hW}$ (несправності не підтверджені), то остаточний діагноз дорівнює первинному діагнозу $DGN_n^h = DGN_n^{h*}$;

Якщо $DGN_n^{h*} \text{ not } \subset F_n^{hW}$ (деякі помилки підтверджено), потім початковий діагноз може бути уточнений або верифікований на основі діагнозів в інших підсистемах:

$$-F_{m,n}^{g,h} \neq \emptyset \wedge DGN_n^{h*} \cap DGN_m^{g*} = \emptyset \rightarrow DGN_{\frac{n}{m}}^{\frac{h}{g}} = \{f_k \in F_n^{hW}\}$$

$$-F_{m,n}^{g,h} \neq \emptyset \wedge DGN_n^{h*} \cap DGN_m^{g*} = \emptyset \rightarrow DGN_{\frac{n}{m}}^{\frac{h}{g}} = DGN_n^{h*} \cap DGN_m^{g*}$$

Діагноз для всієї системи - це сукупність остаточних діагнозів, отриманих від усіх підсистем.

Найбільшою перевагою цього підходу є децентралізація процесу діагностики (виявлення несправностей) на кількох комп'ютерних блоках, що призводить до значного скорочення часу розрахунку. Крім того, блоки стають незалежними в локальній діагностиці та можуть використовуватися без центральної структури.

Найбільші недоліки такі ж, як і у однорівневих діагностичних матриць. Крім того, якщо несправність може бути виявлена в двох підсистемах, то її діагностичні сигнали повинні бути завжди активними, коли виникає несправність. Але діагностичні сигнали в реальних системах можуть бути компенсованими або зашумленими, тому несправність може бути виявлена в першій підсистемі і не виявлена в другій.

1.4 .Причинно-наслідкові графіки І/АБО/НІ

Випадкові зв'язки моделюють вплив причин на спостережувані реакції системи. Цей вид діагностичного опису також можна розглядати як один із модельних підходів, де модель базується на експертних знаннях. Загальна ідея виявлення відмови базується на понятті очікуваної поведінки. Основне припущення, що лежить в основі, полягає в тому, що експерти мають більш точне уявлення про те, яку поведінку системи вони можуть очікувати,

тобто вони можуть кваліфікувати спостережувану поведінку як нормальну та ненормальну. Крім того, передбачається, що вони здатні розпізнавати різні типи ненормальної поведінки та класифікувати спостережувані за певними категоріями. Таким чином, поняття очікуваної поведінки включає як очікувану нормальну поведінку, так і різні види очікуваної ненормальної поведінки. Поведінка описується реляційними моделями.

Графічне представлення причинно-наслідкових зв'язків, графіків І/АБО/НІ, описано, наприклад, [8].

1.4.1. І/АБО/НІ графи. Основні визначення

Загалом причинно-наслідкові графіки є графічним зображенням причинно-наслідкових зв'язків. Причинно-наслідковий зв'язок між вузлами n і n' це вплив того факту, що n відбувається на те, що відбувається факт n' .

Може існувати декілька видів причинно-наслідкових впливів, тобто різні види причинно-наслідкових зв'язків. Три наведені нижче причинно-наслідкові зв'язки відносяться до «сили» причинності та неповноти наших знань; вони такі:

Симптом n викликає симптом n' завжди, коли виникає перший; крім того, виникнення n' обов'язково спричинено n . Ми називаємо цей тип причинності достатнім і необхідним (NEC).

Симптом n викликає симптом n' завжди, коли виникає перший, але також існує кілька можливих різних симптомів, що викликають n' . Ми будемо називати цей тип причинно-наслідкового зв'язку достатньою (SUF).

Поява симптому n може спричинити появу симптому n' , однак є випадки, коли n' не слідує n . Ми називаємо цей найслабший тип причинно-наслідкового зв'язку, наскільки це можливо (MAY).

Зв'язок NEC є найсильнішим, він дає нам найточнішу інформацію, але такий вплив дуже рідко зустрічається в реальних проблемах. Зазвичай корисними є два останні випадки.

Тепер буде введено формальне визначення причинних графів:

Якщо N це набір симптомів і Ψ набір функцій, визначених на них симптомів, то причинно-наслідковий графік є структурою $G = (N, \psi)$.

Визначення причинних графів І/АБО/НІ можна сформулювати так:

Нехай N позначають набір симптомів, $N = D \cup V \cup M$, де M - це набір симптомів прояву, D - це набір елементарних діагнозів або розладів, V - це набір заздалегідь визначених проміжних симптомів. Далі позначимо E^* набір співвідношень, що визначають причинно-наслідкові залежності, і E^- набір відносин, що визначають бінарні негативні залежності. Причинно-наслідковий граф І/АБО/НІ є структурою (N, E^*, E^-) , що задовольняє таким умовам:

Множиною вузлів графа є множина $N = D \cup V \cup M$ (для простоти, ми не розрізняємо вузли графа та не позначаємо їх симптомами).

Причинно-наслідкові зв'язки, задані E^* і E^- визначають дуги в графі.

Існує набір кінцевих вузлів M і набір початкових вузлів D в графі.

$$D \cap V = \emptyset.$$

$$D \cap M = \emptyset.$$

$$M \cap V = \emptyset.$$

Кожен вузол має принаймні одну дугу, що йде до цього вузла та/або виходить з цього вузла (граф зв'язаний).

В графі немає петель.

У причинно-наслідкових графах є три типи вузлів, їх графічне представлення показано на рис. 1.6.

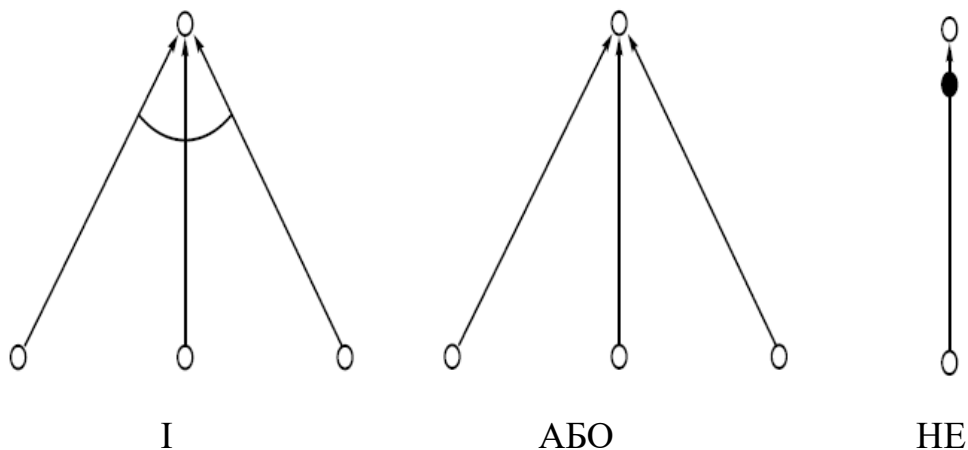


Рисунок – 1.6. Базові вузли.

Вузол АБО - логічна диз'юнкція попередників. Якщо n_i є вузлом АБО, таким що $(n_1, n_i), \dots, (n_{i-1}, n_i)$, є всі пари, що належать E^2 мають як другий аргумент; тоді формула, призначена для n_i є

$$n_i = n_1 \vee n_2 \vee \dots \vee n_{i-1}.$$

Вузол І - логічна кон'юнкція попередників. Якщо n_i є AND-вузол, такий що $(n_1, \dots, n_{i-1}, n_i) \in E^i$, то формула визначення n_i є

$$n_i = n_1 \wedge n_2 \wedge \dots \wedge n_{i-1}.$$

Вузол НЕ - логічне заперечення попередників. Якщо $(n_1, n_2) \in E^-$, формула яка визначає n_2 має форму:

$$n_2 = \neg n_1.$$

Приклад. Граф на рис. 3.2 б представляє причинно-наслідкові зв'язки для простого логічного елемента І. Структура графа базується на таблиці істинності, представлений на рис. 3.2а. Вентилі мають два входи: $aa1$ і $ba1$; і один вихід $ma1$. Симптоми включають:

$ma1 \in M$ це заявлений симптом - вихід вентиля.

$\{va11, va12, va13, va14, va15, va16\} \in V$ це проміжні симптоми.

$da1 \in D$ це елементарний діагноз - несправність вентиля І.

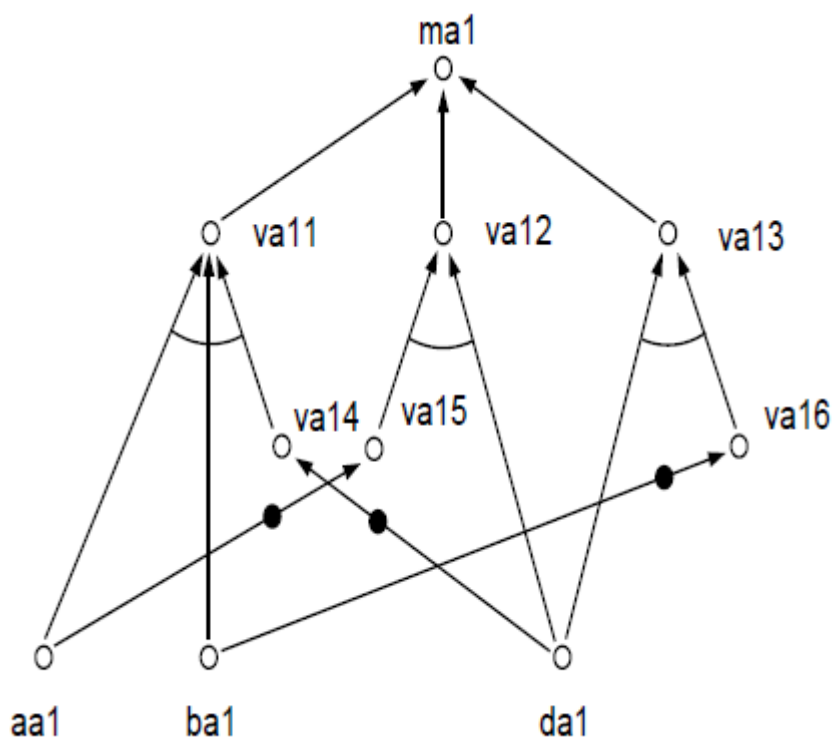
вузли $aa1$ і $ba1$ є потенційні спостереження OBS.

Припущення, що є основою для моделювання причинно-наслідкових графіків, включають наступне:

Кінцева, заздалегідь задана кількість компонентів, що підлягають діагностиці; вжиті контрольні дії та умови експлуатації враховуються як можливі причини ненормальної поведінки. Компоненти, можливі дії та умови повинні бути апріорі відомі, а несправна поведінка системи може бути діагностована лише шляхом призначення несправної поведінки одному або декільком із цих компонентів, застосування певних дій та виникнення умов експлуатації.

	aa1, ba1		va11	
da1	00	01	11	10
0	0	0	1	0
1	1	1	0	1
			va13	va12

a)



б)

Рисунок – 1.7. Графи AND/OR/NOT для логічного I.

Слід діагностувати спостережуваний стаціонарний стан динамічної системи, а не неправильну зміну стану.

Типи аномальної поведінки, що підлягають діагностиці, мають бути в певному сенсі «визначеними».

Тобто повинні бути надані формули, що описують якісні ситуації, що стосуються очікуваних несправностей.

Мають бути доступні знання про причинно-наслідкові залежності між симптомами.

1.4.2. Причинно-наслідкова діагностична методика

Діагностична проблема для причинно-наслідкових графів визначається наступним чином:

Діагностична проблема визначається (G, M, OBS) , де $t \subseteq M$ це набір симптомів разом із їхніми логічними значеннями, що підлягають поясненню (тобто той, що вказує на аномальну поведінку), і OBS це набір симптомів, які вважаються «істинними» чи «хибними» і, таким чином, надають допоміжну інформацію для діагностичних міркувань.

Кожен вузол $n \in N$ характеризується станом, логічним значенням. Стан вузла належить набору $\{true, false, unknown\}$.

Нехай $G = (N, E^+, E^-)$ буде І/АБО/НІ причинний граф, де $N = n_1, n_2, \dots, n_n$. Нехай $Q = \{q^1, q^2, \dots, q^n\}$ будуть поточні логічні значення вузлів. Будь-який набір $s = (n^1, q^1), (n^2, q^2), \dots, (n^j, q^j)$ буде називатися набором станів.

Визначивши частковий стан, можна отримати найбільш повну інформацію про статус симптомів шляхом поширення цього стану. Кінцевий стан, у якому подальше поширення неможливе, буде позначено s^* . Правила розповсюдження - це, в основному, логічні міркування, однак залежно від інтерпретації причинності можливі різні модифікації.

Загалом, усі види міркувань можна розділити на три основні категорії:

1. Дедукція, аналітичний процес, заснований на застосуванні загальних правил до окремих кейсів, з виведенням результату.
2. Індукція, синтетичне міркування, яке виводить правило з випадку та результату.
3. Абдукція, інша форма синтетичного висновку, але випадок на основі правила та результату.

Діагностична процедура для причинно-наслідкових графіків І/АБО/НІ є процедурою абдукції, абдукція — це зворотні міркування, і ми можемо записати це правило наступним чином

$$\frac{\beta, \alpha \rightarrow \beta}{\alpha}$$

Міркування на основі правила дозволяє висувати потенційні гіпотези, які пояснюються висновками. Два основних правила абдукції для вузлів АБО та І:

$$\frac{n, n_1 \vee n_2 \vee \dots \vee n_i \rightarrow n}{n_k}$$

$$\frac{\neg n, n_1 \vee n_2 \vee \dots \vee n_i \rightarrow n}{\neg n_k}$$

де $k \in \{1, 2, \dots, i\}$

Наведені нижче правила точно визначають процедури прямого та зворотного розповсюдження для причинних графів AND/OR/NOT.

Пряме поширення станів графа:

1. АБО вузол true: якщо принаймні один із попередників вузла АБО є true, тоді значення вузла АБО встановлюється як true.
2. І вузол false: якщо принаймні один із попередників вузла І є false, тоді значення вузла І встановлюється як false.
3. НЕ вузол true: якщо попередник вузла НЕ false, тоді значення вузла НЕ встановлюється як true.
4. НЕ вузол false: якщо попередник вузла НЕ є true, тоді значення вузла НЕ встановлюється як false.
5. АБО вузол false: якщо всі попередники вузла АБО false, тоді значення цього вузла АБО встановлюється як false.
6. І вузол true: якщо всі попередники вузла І є true, тоді значення цього вузла І встановлюється як true.

Просте зворотне розповсюдження станів графа:

1. АБО вузол false: якщо вузол АБО є false, то значення всіх його попередників встановлюються як false.

2. І вузол true: якщо вузол І є true, то значення всіх його попередників встановлюються як true,

3. НЕ вузол true: якщо вузол НЕ має значення true, то значення його попередника встановлюється рівним false.

4. НЕ вузол false: якщо вузол НЕ є false, то значення його попередника встановлюється рівним true.

Зворотне поширення станів графа з вибором попередників:

1. АБО вузол true: якщо вузол АБО є true, тоді значення одного його вибраного попередника встановлюється як true.

2. І вузол false: якщо вузол І має значення false, тоді значення одного з його вибраних попередників встановлюється значенням false.

Вибір дозволяє знайти всі можливі причини станів вузлів АБО та І. Попередників можна вибирати у попередньо визначеному порядку або за допомогою іншої інформації, наприклад статистичних або експертних знань. Ці два правила реалізують правила абдуктивного висновку вказані вище.

Наведені вище правила визначають принципи поширення стану. Щоразу, коли застосовується правило, генерується нове значення симптому; далі він поміщається в набір, що представляє поточний стан. Якщо якийсь симптом виявляється таким, що приймає два неузгоджених значення, початковий стан для поширення вважається неузгодженим і більше не враховується.

Діагноз, створений на основі причинно-наслідкового графу, визначається наступним чином:

Якщо G - причинний графік І/АБО/НІ, що визначає зв'язок між симптомами в аналізованій системі M , це набір проявів, які необхідно пояснити, і OBS це набір спостережень, тоді діагноз - це будь-який мінімальний і послідовний набір D початкові симптоми та їх значення, такі, що:

D пояснює M , тобто припускаючи, що G уточнює теорію домену, $G \cup D = M$.

D узгоджується із спостереженнями (та їх наслідками), тобто якщо S^* позначає максимальний стан, отриманий шляхом поширення станів симптомів

$D \cup OBS$, визначених понад G , то S^* є відповідним (жоден симптом не є істинним і хибним одночасно).

D є мінімальним.

Причинно-наслідкові графи І/АБО/НІ дозволяють користувачеві моделювати поведінку діагностованих систем. Моделі можуть бути однорівневими або багаторівневими з використанням вертикальної та горизонтальної ієрархізації.

Висновки до розділу 1

В розділі проведено аналіз методів та факторів, які використовуються в системах діагностування та засновані на знаннях. Оглянуто діагностичні методики, їх переваги та недоліки.

Розглянуто системи міркування на основі випадків, ієрархічні діагностичні матриці, причинно-наслідкові графи.

Описана причинно-наслідкова методика для систем діагностики.

2 СИТЕМИ ВИЯВЛЕННЯ ТА УСУНЕННЯ НЕСПРАВНОСТЕЙ

Діагностичні знання в модельних підходах мають форму явної моделі діагностованої системи. Модель містить опис правильної поведінки діагностованої системи. Якщо модель чітко визначена, то реальні спостереження системи повинні узгоджуватися з її прогнозованою поведінкою, інакше ми повинні припустити, що система не працює належним чином. Розташування невідповідностей у моделі є основною інформацією для діагностичної процедури.

Існує два основні підходи до діагностики на основі моделі: FDI (виявлення та усунення несправностей) і DX (скорочення взято від назви DX International Workshop on Principles of Diagnosis). Підходи FDI ближчі до підходів, які випливають із теорії контролю та статистичного аналізу. DX ближче до області комп'ютерних наук і штучного інтелекту.

Ця презентація підходів, заснованих на моделях DX та FDI, значною мірою базується на роботах авторів Cordier та Dague [9].

Приклад. Подальші міркування щодо обох типів підходів будуть проілюстровані простою арифметичною одиницею. Схема (рис. 2.1) складається з трьох помножувачів m_1 , m_2 , m_3 та двох суматорів a_1 , a_2 . Значення вхідних даних $A=3$, $B=2$, $C=2$, $D=3$ та $E=3$. Значення виходів $F=23$ і $G=12$. Отже, існує невідповідність між вимірним значенням виходу F і його прогнозованим значенням.

2.1 Базова модель діагностування DX

Діагностична структура Рейтера базується на узгодженості між правильною поведінкою системи та її спостережуваною ненормальною поведінкою [9]. Система працює правильно, якщо між цими двома способами поведінки немає невідповідностей.

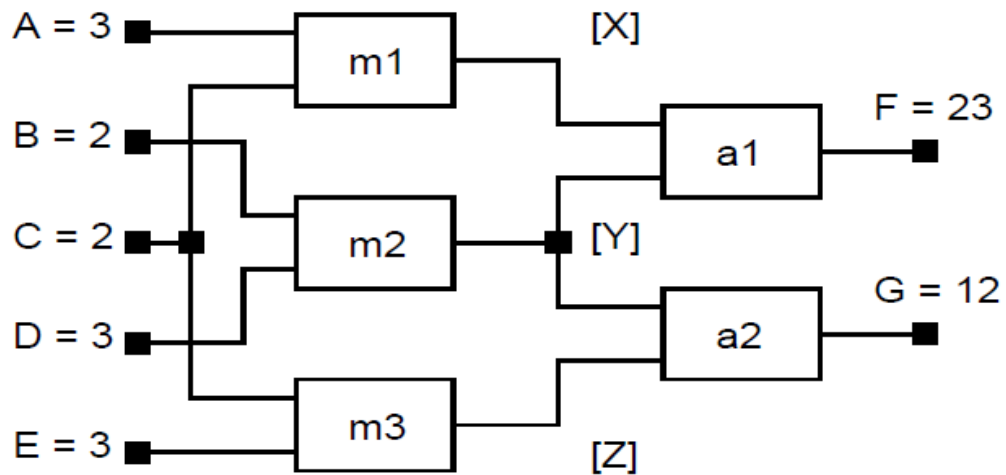


Рисунок – 2.1. Проста арифметична схема системи.

Якщо виникають якісь невідповідності, то деякі частини системи (компоненти) несправні. Система визначається таким чином:

Система - це пара $(SD, COMPONENTS)$, де:

- 1) SD , опис системи, є набором речень першого порядку.
- 2) $COMPONENTS$, компоненти системи, це кінцевий набір констант.

Приклад.

Повернемося до схеми, представленої на рис. 2.1. Набір компонентів такий:

$$COMPONENTS = \{a1, a2, m1, m2, m3\},$$

а опис системи:

$$ADD(x) \wedge \neg AB(x) \rightarrow Output(x) = Input1(x) + Input2(x).$$

$$MULT(x) \wedge \neg AB(x) \rightarrow Output(x) = Input1(x) * Input2(x).$$

$$ADD(a1), ADD(a2), MULT(m1), MULT(m2), MULT(m3),$$

$$Output(m1) = Input1(a1), Output(m2) = Input2(a1),$$

$$Output(m2) = Input1(a2), Output(m2) = Input2(a2),$$

$$Input(m1) = Input(m3).$$

Процедура діагностики базується на невідповідності між розрахунковими та спостережуваними значеннями входів і виходів компонентів. Визначення спостережень наведено нижче:

OBSERVATION система - це скінченний набір речень першого порядку.
Будемо писати $SD(COMPONENTS, OBS)$ для системи з спостереженням OBS .

Приклад.

Спостереження для арифметичної одиниці:

$$Input1(m1) = 3, Input2(m1) = 2,$$

$$Input1(m2) = 2, Input2(m2) = 3,$$

$$Input1(m3) = 3, Input2(m3) = 3,$$

$$Output1(a1) = 23, Output(a2) = 12.$$

Конфлікт і сети попадань

Опис системи SD визначає нормальну («правильну») поведінку компонентів системи. Ненормальна поведінка компонента буде позначатися попередньо визначеним предикатом AB . Якщо набір $\{\neg AB(c_1), \dots, \neg AB(c_n)\}$ представляє припущення, що всі компоненти системи працюють правильно, тому $SD \cup \{\neg AB(c_1), \dots, \neg AB(c_n)\}$ представляє коректну поведінку системи (всі компоненти працюють коректно). Через невідповідності система не працює належним чином. Отже, формула, задана

$$SD \cup \{\neg AB(c_1), \dots, \neg AB(c_n)\} \cup OBS$$

є непослідовним.

Діагноз — це припущення про те, що певні компоненти системи поведуться ненормально. Ця гіпотеза має узгоджуватися з тим, що відомо про систему, і зі спостереженнями.

Діагноз для $(SD, COMPONENTS, OBS)$ - це мінімальний набір $\Delta \subseteq COMPONENTS$, який

$$SD \cup OBS \cup \{AB(c) | c \in \Delta\} \cup \{\neg AB(c) | c \in COMPONENTS - \Delta\}$$

є послідовним.

Стабільність системи можна відновити, лише припустивши, що деякі її компоненти несправні. Сукупність таких компонентів становить діагноз.

Приклад. Приклад діагностики для нашої системи $\{m1\}$; це означає, що ненормальна поведінка цього елемента пояснює спостережувану неправильну поведінку пристрою.

Знайти всі потенційні діагнози нелегко, але їх можна розрахувати на основі конфліктні набори і сети попадань.

Налаштування конфлікту ($SD, COMPONENTS, OBS$) набір компонентів $\{c_1, \dots, c_k\} \subseteq COMPONENTS$ такий, що

$SD \cup OBS \cup \{\neg AB(c_1), \dots, \neg AB(c_k)\}$ є непослідовним.

Конфліктна множина – це будь-яка множина, яка має принаймні один несправний компонент, тобто конфліктна множина для системи є мінімальною, якщо жодна належна їй підмножина не є конфліктною множиною.

Приклад. Повернемося до прикладу системи. Прогноз, що $Output(a1)=12$ залежить від правильної роботи $a1, m1$ і $m2$ (рис. 2.2). Оскільки $Output(a1)=23$. Вихідний принаймні один із компонентів $a1, m1$ і $m2$ має бути несправним. Таким чином набір $\{a1, m1, m2\}$ є набором конфліктів. Оскільки жодна з його підмножин не є конфліктною множиною, конфлікт $\{a1, m1, m2\}$ є мінімальним конфліктним набором.

Таким чином, діагноз - це набір, який бере принаймні один компонент з кожного набору конфліктів, тобто це набір влучень усіх наборів конфліктів.

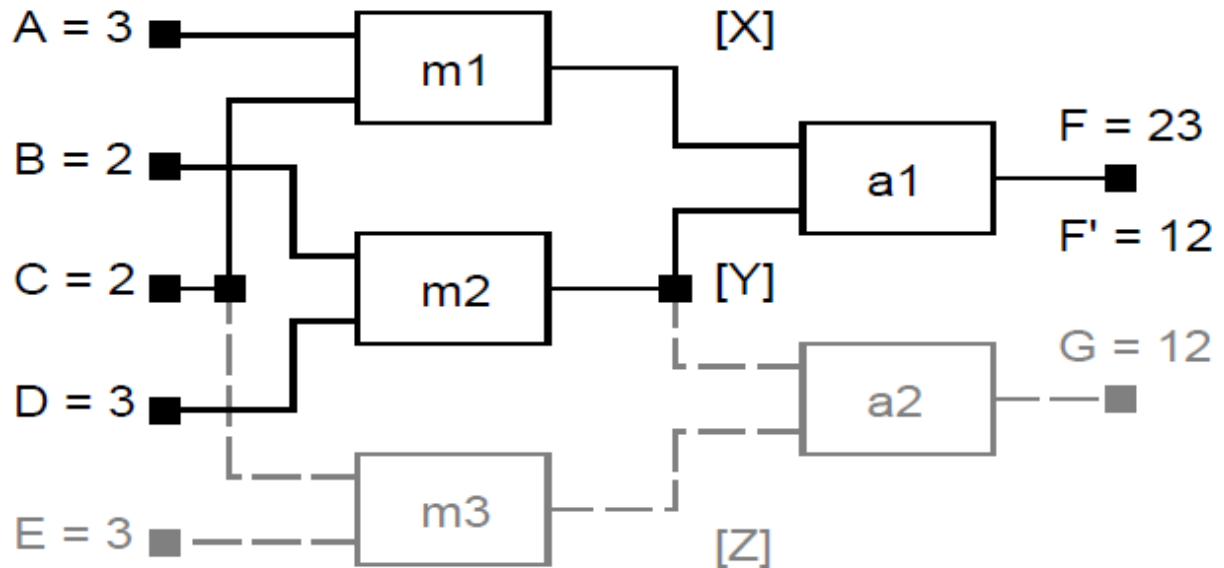


Рисунок – 2.2. Набір конфліктів $\{a1, m1, m2\}$.

Припустимо, C є набором множин.

Набір попадань для C - це такий набір $H \cap \bigcup_{S \in C} S$, що $H \cap S \neq \emptyset$ для кожного $S \in C$. Набір збігів для C є мінімальним, якщо жодна належна його підмножина не є набором збігів для C .

Тоді множиною збігів є будь-яка множина, що має непорожній перетин з кожним із конфліктних множин і складається лише з елементів конфліктних множин.

$\Delta \subseteq COMPONENTS$ це діагноз для $(SD, COMPONENTS, OBS)$, якщо Δ є мінімальним набором збігів для набору конфліктних наборів для $(SD, COMPONENTS, OBS)$.

Приклад. Для арифметичної системи мінімальні набори конфліктів: $\{a1, m1, m2\}$, $\{a1, a2, m1, m3\}$.

Конфлікт $\{a1, m1, m2\}$ є очевидним (рис. 2.2). Можна легко обчислити виходи множників $m1$ і $m2$ рівних 6. Отже, результат суматора $a1$ має дорівнювати 12, але спостереження дорівнює 23, і виникає конфлікт.

Конфлікт $\{a1, a2, m1, m3\}$ обчислюється за допомогою зворотного обчислення (рис. 2.3). Вихід множників $m1$ і $m3$ рівні 6. Отже, ми знаємо обчислене значення змінної Z і спостережуване значення змінної G . Тепер значення змінної Y обчислюється за допомогою суматора $a2$ та дорівнює 6.

Вихід суматора $a1$ має дорівнювати 12, але спостереження дорівнює 23 і є конфлікт.

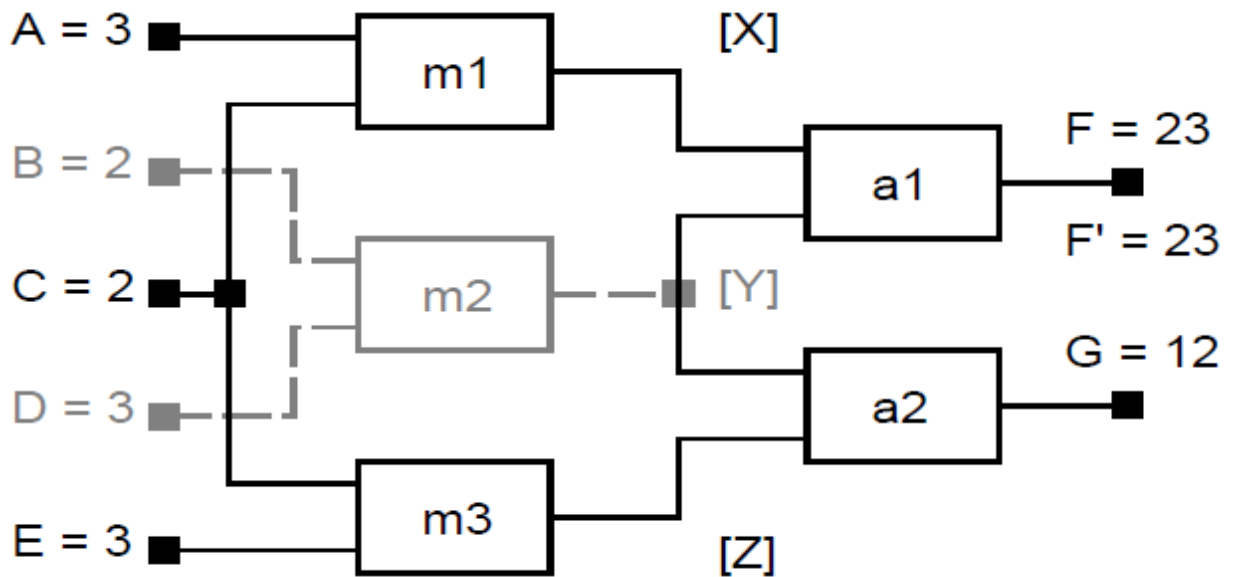


Рисунок – 2.3. Набір конфліктів $\{a1, a2, m1, m3\}$.

Можна розрахувати наступні мінімальні набори попадань (рис. 2.4):

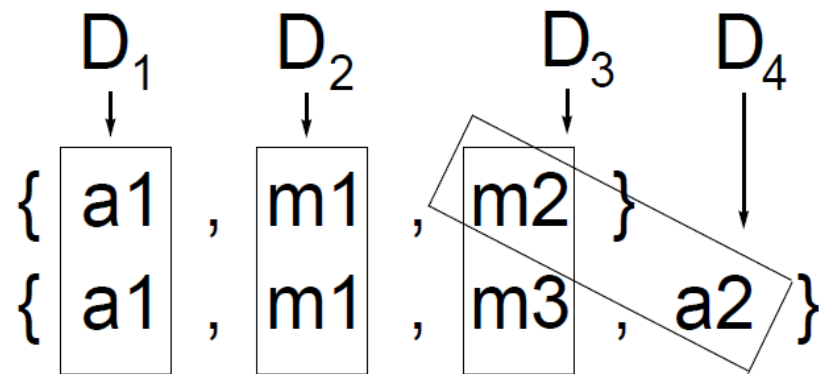


Рисунок – 2.4. Мінімальні набори попадань.

Кожен із наборів попадань є діагнозом, оскільки припущення, що його компоненти несправні, дозволяє відновити послідовність системи.

Основним недоліком підходів DX є висока обчислювальна складність, через що діагностика великих систем стає дуже важкою.

Процедура діагностики на основі моделі є поетапною процедурою. На першому етапі розраховуються діагнози, на другому етапі перевіряються та усуваються деякі локалізовані несправності. Після цього, якщо нові отримані

спостереження все ще не узгоджуються з розрахунками, процедуру повторюють.

2.2. Діагностика на основі моделі FDI

Основні визначення.

Поведінкова модель є основним елементом підходів до FDI. Модель, у більшості випадків, походить від структури системи та показує зв'язки між компонентами системи та їхньою моделлю поведінки.

Визначення. Модель системи SM визначається як модель поведінки BM (*Behavior model*), тобто набір відношень, що визначають поведінку системи, разом із моделлю спостереження OM (*Obs model*), тобто набір відношень між змінними X системи та спостережуваними змінними O , отриманими давачами.

Приклад. Елементарними компонентами є суматори $a1, a1$, множники $m1, m2, m3$. Модель системи SM задається такими рівняннями:

BM :

$$-Rm1: x = a * c,$$

$$-Rm2: y = b * d,$$

$$-Rm3: z = c * e,$$

$$-Ra1: f = x + y,$$

$$-Ra2: g = y + z,$$

OM :

$$-RSa : a = A,$$

$$-RSb : b = B,$$

$$-RSc : c = C,$$

$$-RSd : d = D,$$

$$-RSe : e = E,$$

$$-RSf : f = F,$$

$$-RSg : g = G.$$

Визначення. Діагностичну проблему визначає модель системи SM , набір спостережень OBS , що присвоюють значення спостережуваним змінним, і набір помилок F (тобто a набір діагнозів). Несправність можна розглядати як набір несправних компонентів.

Приклад. $OBS = \{A = 2, B = 2, C = 3, D = 3, E = 2, F = 23, G = 12\}$.
Набір поодиноких несправностей є $SF = \{F_{a1}, F_{a2}, F_{m1}, F_{m2}, F_{m3}\}$. Сукупність несправностей є $F = 2^{SF}$ і кількість можливих несправностей рівна $m = 2^{|SF|} - 1$.

Визначення. Структура системи визначається через бінарну програму $s: SM \times V \rightarrow \{0,1\}$, де $V = X \cup O$ є набір змінних. Елемент двійкового застосування $s(rel, v) = 1$, якщо і тільки якщо v з'являється у відношенні rel .

Відносини резервування.

Відносини резервування є основою виявлення та локалізації несправностей у підходах FDI.

Визначення. Аналітичне відношення надлишковості (ARR) – це відношення SM , що містить лише спостережувані змінні, і тому їх можна оцінити з OBS . Відзначається $r=0$, де r називається залишком ARR . Для даного OBS , зазначається примірник залишку $val(r, OBS)$, скорочено як $val(r)$ коли є двозначним. Таким чином $val(r, OBS) = 0$, якщо спостереження задовольняють ARR .

ARR можна отримати з моделі системи шляхом усунення невідомих змінних.

Приклад. Повна відповідність призводить до таких ARR :

$$ARR1: r_1 = 0 \text{ where } r_1 \equiv F - A \times C - B \times D$$

$$ARR2: r_2 = 0 \text{ where } r_2 \equiv G - B \times D - C \times E$$

$$ARR3: r_3 = 0 \text{ where } r_3 \equiv F - G - A \times C + C \times E$$

Тепер буде введено визначення сигнатури несправності.

Визначення. Дано набір $R = \{ARR_1, \dots, ARR_n\}$ із n ARR і набір $F = \{F_1, \dots, F_m\}$ з m помилок, сигнатуру помилки задає двійковий вектор $FS_j = [s_{1j}, \dots, s_{nj}]^T$ в якому s_{ij} задається: $(ATT_i, F_j) \rightarrow s_{ij} = 1$, якщо деякі компоненти, що беруть участь у F_j беруть участь у ARR_i $(ATT_i, F_j) \rightarrow s_{ij} = 0$ інакше.

Вхід $s_{ij}=0$ означає появу несправності F_j не впливає ARR_i ($val(r_i) = 0$).

Інакше коли $s_{ij}=1$ то несправність F_j впливає ARR_i ($val(r_i) \neq 0$).

Визначення. За наявності набору R з n ARR сигнатури набору F з m несправностей разом складають так звану матрицю сигнатур.

Введені тут матриці підписів дуже схожі на діагностичні матриці, представлені попередньо. Основна філософія та сама, але термінологія в деяких місцях відрізняється.

Таблиця 2.1. Матриця підписів

	F_{a1}	F_{a2}	F_{m1}	F_{m2}	F_{m3}
ARR_1	1	0	1	1	0
ARR_2	0	1	0	1	1
ARR_3	1	1	1	0	1

Приклад. У нашому прикладі матриця сигнатур для набору одиничних несправностей відповідає реагуючи на компоненти $a1$, $a1$, $m1$, $m2$ і $m3$ наведені в таблиці 2.1.

Якщо ми хочемо діагностувати кілька несправностей, то необхідно розширити матрицю (табл. 2.1). 26 додаткових стовпців ($2^m - 1$) мають $[1,1,1]^T$ підписи, за винятком $F_{a1,m1}$ який має $[1,0,1]^T$ підпис, і $F_{a2,m3}$, який має $[0,1,1]^T$ підпис.

Діагностичні набори в підході FDI надаються в термінах помилок, врахованих у матриці підписів. Генерація діагностичних наборів базується на інтерпретації стовпців матриці сигнатур і полягає в порівнянні сигнатур спостереження з сигнатурами несправностей. Це порівняння визначено як проблема прийняття рішення.

Визначення. Підпис спостереження OBS є двійковий вектор $OS = [OS_1, \dots, OS_n]^T$, де $OS_i = 0$ якщо $val(r_i, OBS) = 0$.

Першим кроком (завдання виявлення) є створення сигнатури спостереження, тобто визначення того, дорівнює залишкове значення нулю чи ні, за наявності шуму та перешкод. Це, як правило, формулюється як статистична проблема прийняття рішень, використовуючи доступні моделі шуму та завад.

Приклад. З OBS , як зазначено, $OS = [1,0,1]^T$ У випадку $i = 10$, $i OS = [1,1,0]^T$ в випадку $f=10$ і $g=14$, $OS = [1,1,1]^T$.

Другим кроком (задача ізоляції) є порівняння поточної сигнатури спостереження з сигнатурами несправності. Рішення цієї проблеми прийняття рішень полягає у визначенні критерію узгодженості таким чином:

Визначення. Підпис спостереження $OS = [OS_1, \dots, OS_n]$ є послідовний із сигнатурою помилки $FS_j = [s_{1j}, \dots, s_{nj}]^T$ $FS_j = [s_{1j}, \dots, s_{nj}]^T$, якщо і тільки якщо $OS_i = s_{ij}$ для всіх i .

Визначення. Діагностичні набори задаються несправностями, сигнатури яких узгоджуються з сигнатурою спостереження.

Приклад. Діагностичні набори, отримані для таких сигнатур спостереження:

$$OS = [1,0,1]^T \leftrightarrow F_{A1} \text{ or } F_{M1} \text{ or } F_{\{A1,M1\}}$$

$$OS = [1,1,0]^T \leftrightarrow F_{M2}$$

$$OS = [1,1,1]^T \leftrightarrow \text{any multiple fault except } F_{\{A1,M1\}} \text{ and } F_{\{A2,M3\}}$$

Діагнози, засновані на підході FDI для арифметичної системи $\{a1\}$, $\{m1\}$ і $\{a1, m1\}$. Та сама система, яка діагностується за допомогою методології DX, має діагнози $\{a1\}$, $\{m1\}$, $\{a2, m2\}$ і $\{m2, m3\}$, що означає підрахунок діагнозів FDI. Причина нестачі двох діагнозів є сильним обмежуючим припущенням, зробленим у підході DFI, що кілька недоліків ніколи не можуть компенсувати одна одну, тобто *ARR* є сильно незалежними. У багатьох заявках на FDI часто прийняте припущення полягає в тому, що може виникнути лише одна помилка. Існує багато розширень і практичних застосувань цього підходу.

2.3. Ієрархічна діагностика в гетерогенних середовищах

Сучасні технологічні системи зазвичай складаються з багатьох спеціалізованих частин, які рекурсивно складаються з інших. Такі частини будуть далі називатися компонентами, а система, побудована з багатьох таких компонентів, буде називатися складною системою. Природа компонентів може бути функціональною або структурною. Функціональний компонент розуміється як частина складної системи, визначена її роллю (поведінкою), тоді як структурний компонент є частиною складної системи, визначеною її формою. Компонентні підходи до діагностики базуються на структурному розподілі більших систем.

Діагностика складних систем стикається з двома основними труднощами. Перший полягає в частій відсутності загального роду діагностичного опису для всієї системи. Деякі групи компонентів можуть мати різний характер роботи або можуть виготовлятися різними виробниками, що тягне за собою відмінності в найкраще підігнаному або доступному діагностичному описі між групами компонентів. Тоді непросто вибрати хороший загальний діагностичний опис, який задовольняє всі компоненти та не втрачає надто багато інформації.

Друга проблема – складність діагностичного опису. Належний діагностичний опис складної системи повинен містити всі компоненти, які можна змінити під час процедури ремонту. Коли складна система дійсно

велика, тоді діагностична процедура для такого діагностичного опису є дорогою з обчислювальної точки зору та може створити багато потенційних діагнозів.

Запропонована методологія діагностики складних систем намагається вирішити згадані труднощі кількома способами.

Компоненти можна описати тут двома згаданими раніше видами представлень знань на основі логіки: моделями та експертними графами. Опис на основі моделі дозволяє більш точно описати компоненти, але деякі з компонентів не мають добре розроблених моделей або їх моделі занадто складні та марні для обмеженої в часі процедури діагностики. Такі компоненти можна описати описами на основі графів, які збирають експертні знання про поведінку компонентів. Через два комплементарні підходи, методологія здатна описати системи, які неможливо описати єдиною, спільною для всіх компонентів методологією.

Складність діагностичного опису частково зменшується завдяки ієрархізації, яку іноді називають абстракцією або релаксацією. Надалі будемо називати це ієрархічним підходом. Грубо кажучи, оригінальний опис поділено на кілька рівнів, і кожен рівень складається із «забуття» деяких деталей з нижчого рівня ієрархії. Найвищий рівень є найбільш загальним, нижчий – найбільш детальним і схожим на вихідну задачу.

Тут ієрархічна структура, що включає різні рівні, створюється шляхом включення компонентів в інші компоненти. Якщо компонент включає інші компоненти, то ці підкомпоненти складають наступний більш детальний рівень ієрархії. Кожен рівень ієрархії містить декілька підкомпонентів, межі яких визначаються структурним або функціональним поділом. Діагнози, отримані на вищих рівнях ієрархії, можуть бути краще визначені та підтверджені на нижчих рівнях. Ієрархічний опис зазвичай, але не завжди, менш складний і легший для діагностики, ніж вихідний детальний опис проблеми.

Діагностична процедура для представленого вище ієрархічного діагностичного опису дозволяє генерувати діагнози від найвищого рівня

ієрархії до нижчого, тобто від найбільш абстрактного рівня до найбільш детального рівня (зверху вниз). Вироблені діагнози включають елементарні компоненти, які не мають підкомпонентів, або компоненти, які не можуть бути додатково діагностовані через відсутність діагностичної інформації. Крім того, діагностики можна впорядкувати відповідно до ймовірності несправностей і перевірити між рівнями ієрархії.

Основні визначення та припущення

Нехай дано набір констант *ELEMENTARY COMPONENTS* (*EC*, скорочено), що позначає основні компоненти в сенсі, тобто ті, що позначаються константами, для яких ніколи не розглядається внутрішня структура. *A* (складний, ієрархічно структурований) компонент можна рекурсивно визначити таким чином:

Визначення. Компонент *c* :

- елементарний компонент $c \in EC$, або
- складний компонент $c = (CD, SUBC)$, де *CD* є опис компонента (набір речень першого порядку) і *SUBC* є сукупність його підкомпонентів.

Зауважте, що наведене вище визначення вводить дерево зв'язків моделювання структури компонентів між компонентами та їхніми підкомпонентами. Елементарними компонентами є листя дерева, вони не мають нащадків. Корінь дерева будемо називати основним компонентом. По суті, основний компонент – це складна система ієрархічної структури підкомпонентів.

Якщо $c = (CD, SUBC)$ є компонентом, тоді *c* є прямим суперкомпонентом для будь-якого компоненту $c_i \in SUBC$. І навпаки, будь-який $c_i \in SUBC$ є прямим підкомпонентом *c*. Якщо існує шлях, що йде від кореня через деякі компоненти c_i і c_j до якогось листка тоді c_i є суперкомпонентом для c_j і c_j є підкомпонентом для c_i .

Далі ми призначаємо рівень будь-якому вузлу, що представляє деякі компоненти (підкомпоненти) у дереві. Кореневому вузлу (і головному компоненту) присвоюється рівень з нумерацією 0. Будь-якому іншому

компоненту призначається рівень $m+1$, де m рівень його прямого надкомпонента.

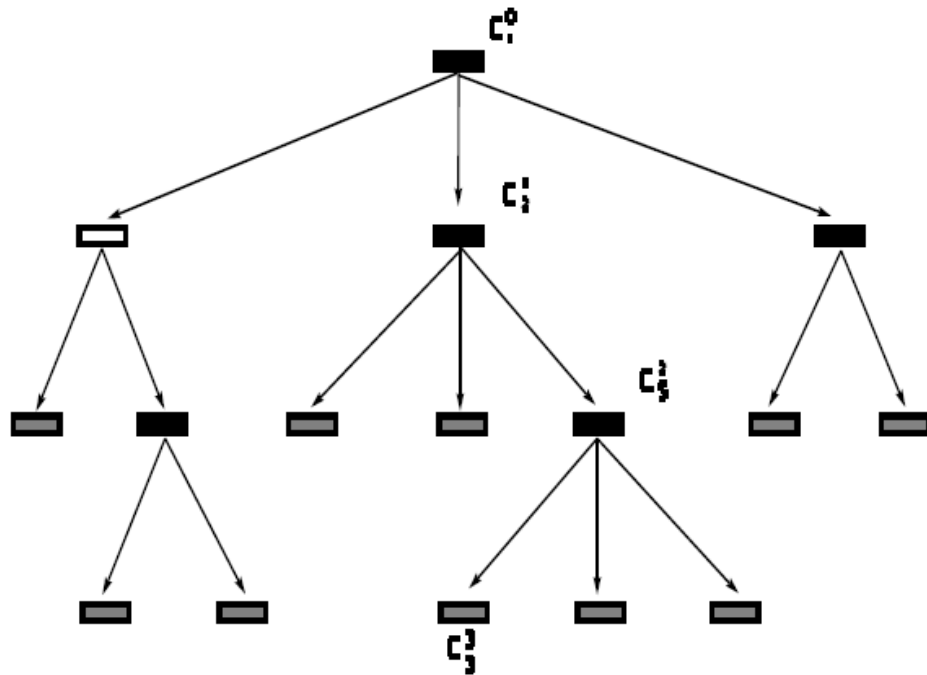


Рисунок – 2.5. Дерево структури компонентів для гіпотетичної системи.

Щоб позначити той факт, що компонент c_i знаходиться на рівні j всередині компонент c ми маємо записати $level(c, c_i) = j$ або $c_i \in_j c$. Якщо компонент c є визначений унікальним чином, нотація може бути спрощена до $level(c_i) = j$ або c_i^j .

Будь-які компоненти c припускається, що ідентифікуються за його (унікальною) назвою або міткою. Допозначають той факт, що компонент c_i є прямим підкомпонентом c контурно-крапкової нотації $c.c_i$ буде використовуватися. Крім того, якщо існує шлях, що йде від кореня через деякі компоненти $c_i, c_{i+1}, \dots, c_{i+k}$ до деякого листка, а наступні вузли є прямими підкомпонентами на попередні буде розширено нотацію шлях-точка до $c_i, c_{i+1}, \dots, c_{i+k}$.

Будь-який компонент (простий або складний) може бути додатково описаний набором змінних, які зазвичай визначають його входи та виходи. Передбачається, що будь-якому компоненту присвоєно певну функціональну поведінку, а його виходи пов'язані з входами певним чином, визначеними моделлю компонента. Вхідні та вихідні змінні разом з деякими іншими

характеристиками компонента (внутрішні змінні стану, параметри тощо) також називаються атрибутами компонента.

Для позначення значення атрибута компонента використовуватиметься стандартна нотація $O-A-V$ (Атрибут (Об'єкт) = Value), наприклад, $Output(a1) = 12$. Набір $ATTR_i^j$ включає усі атрибути зі значеннями компонента c_i^j . Невизначене значення позначається постійним значенням «null».

У запропонованому підході поєднання діагностичних процедур на основі моделі Reiter і засновані на причинно-наслідкових міркуваннях з І/АБО/НІ логічними причинно-наслідковими графіками будь-який компонент можна сприймати з двох різних точок зору: його можна ідентифікувати як елемент системи, що розглядається на певному рівні в модельному підході або в ієрархічному причинно-наслідковому графі в експертному підході.

Ієрархічна модель, як і плоска модель, має включати спостереження реального світу. Визначення спостережень таке:

Визначення. Спостереження комплексу компонент c_i^j є кінцевим набором речень першого порядку. Будемо писати $(CD_i^j, SUBC_i^{j+1}, OBS_i^j)$ для компонент із спостереженням OBS_i^j .

Приклад. Арифметичний блок, представлений на рис. 2.1, моделюється ієрархічно. $Level 0$ описується моделлю OBS_1^0 представлені на рисунку та включає підкомпоненти $SUBC_1^1 = \{m1, m2, m3, a1, a2\}$. Спостереження є $OBS_1^0 = \{A = 3, B = 2, C = 2, D = 3, E = 3, F = 23, G = 12\}$.

Одним з діагнозів системи є $d_1 = \{a1\}$. Опис компонента CD_{a1}^1 суматора $c_1^0 a^1$ показаний на рис. 2.6, його підкомпоненти (однорозрядні суматори) є $SUBC_{a1}^2 = \{\Sigma_0, \Sigma_1, \Sigma_2, \Sigma_3\}$.

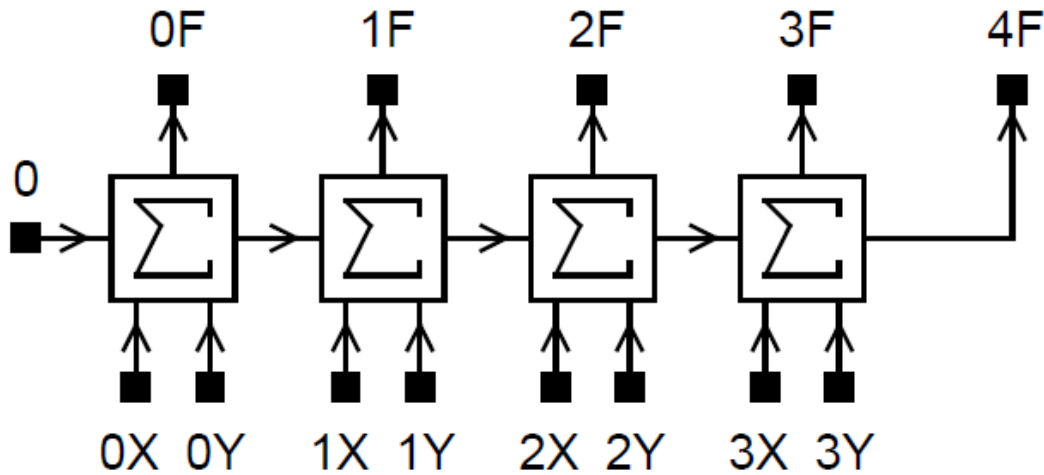


Рисунок – 2.6. Повний суматор

Діагностична проблема

Поточні спостереження можуть узгоджуватися з прогнозованою поведінкою компонента – i в такому випадку вважається, що компонент працює правильно; це позначається як $\neg AB(c_i)$ (виклик це $AB(c_i)$ позначає той факт, що компонент c_i поводить себе ненормально). Якщо поточні спостереження відрізняються від того, що можна очікувати на основі опису компонента (його моделі), компонент вважається несправним – принаймні один його субкомпонент поводить себе ненормально; це позначається як $AB(c_i)$. Діагностичну задачу для ієрархічної моделі можна сформулювати так:

Визначення. Діагностична проблема P_i^j для компонента c_i^j , визначеного на рівні j , визначається як чотири смугова

$$P_i^j = (c_i^j, CD_i^j, SUBC_i^{j+1}, OBS_i^j)$$

де CD_i^j визначення моделі для компонента c_i^j , є його підкомпонентом і OBS_i^j поточні спостереження на рівні для компонента c_i^j .

Приклад. Діагностична задача для арифметичного блоку на рівні задається наступним чином:

$$P_1^0 = (c_1^0, CD_1^0, SUBC_1^1, OBS_1^0)$$

Зауважимо, що в ієрархічному підході визначення діагностичної проблеми завжди формулюється на певному рівні абстракції і відноситься до якогось конкретного компонента. На рівні 0 діагностується вся система, а діагнози стосуються її безпосередніх підкомпонентів; насправді, бо ми маємо класичну однорівневу діагностику проблеми. На якомусь середньому рівні j діагнози компонента c^j посиляться до його прямого підкомпонента.

Міжрівневе відображення спостережень

Для забезпечення справжньої ієрархічної діагностики необхідно визначити спосіб переміщення рівня вниз, щоб інформацію про рівень j можна було ефективно призначити компонентам рівня $j+1$. Визначено дві функції відображення для ієрархічних систем:

Функція фокусування, яка дозволяє зосередити спостереження на атрибутах підкомпонентів.

Ієрархічна функція, яка дозволяє зіставляти атрибути зі спостереженнями підкомпонентів.

Визначення. Функція фокусування F_i^j — це функція, що відображає спостереження, що посиляються на компонент c_i^j на рівні j , на атрибути, що посиляються на його прямі підкомпоненти на рівні $j+1$

$$F_i^j: OBS_i^j \rightarrow ATTRIBUTES_i^{j+1}$$

де $ATTRIBUTES_i^{j+1}$ є колекцією наборів ATR_d^{j+1} . Набір ATR_d^{j+1} містить атрибути підкомпонентів, включених до діагностики d . Діагностика d вирішує діагностичну проблему P_i^j .

Приклад. Знову розглянемо модель арифметичного блоку на рис. 2.1. Одиниця є модельно-описаною, а її атрибутами є входи та виходи її підкомпонентів.

Розраховуються чотири діагнози: $d_1 = \{a1\}$, $d_2 = \{m1\}$, $d_3 = \{a2, m2\}$, $d_4 = \{m2, m3\}$.

Колекція $ATTRIBUTES_1^1$ включає в себе чотири набори атрибутів:

$ATR_{d1}^1, ATR_{d2}^1, ATR_{d3}^1, ATR_{d4}^1$. Наприклад, ATR_{d1}^1 включає в себе такі атрибути: $input1(a1) = 6, input2(a1) = 6, output(a1) = 23$,

Визначення. Ієрархічна функція H_k^{j+1} є функція, що відображає атрибути, що посилаються на підкомпонент c_k^{j+1} на рівні $j+1$ своїх спостережень $H_k^{j+1}: ATR_k^{j+1} \rightarrow OBS_k^{j+1}$.

Ієрархічна функція в основному використовується для перетворення форми діагностичної інформації між ієрархічними рівнями. Перетворення можливе, якщо

$$\exists_{d \in P_i^j \wedge c_k^{j+1} \in d} ATR_k^{j+1} \subseteq ATR_d^{j+1}$$

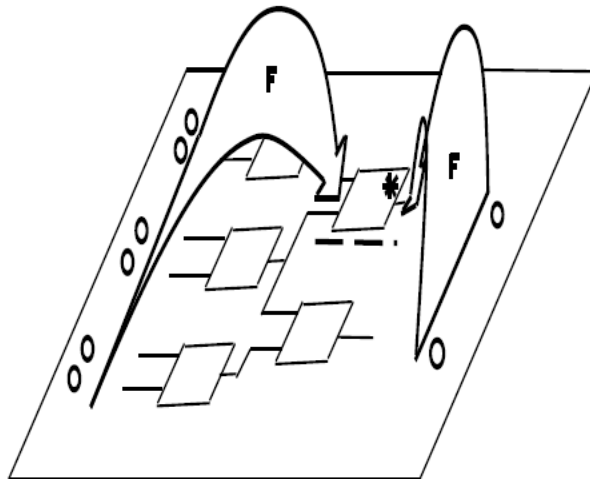


Рисунок – 2.7. Функція фокусування

Приклад. Ієрархічна функція H_{a2}^1 для суматора $a2$ перетворює $\{input1(a2) = 6, input2(a2) = 6, output(a2) = 23\}$, який є підмножиною набору атрибутів $ATR_{d3}^1 \in ATTRIBUTES_1^1$, до спостережень OBS^{j+1} . Спостереження є двійковими значеннями на вхідних даних виходи компонентів нижчого рівня $SUBC_k^2$ (рис. 2.6):

$$0X = 0, 1X = 1, 2X = 1, 3X = 0,$$

$$0Y = 0, 1Y = 1, 2Y = 1, 3Y = 0,$$

$$0F = 1, 1F = 1, 2F = 1, 3F = 0, 4F = 1.$$

Нехай c_i^{j+1} буде деяким прямим підкомпонентом c_k^j . Спостереження з посиланням на c_i^{j+1} буде позначатися як OBS_i^{j+1} , $i \in OBS_i^{j+1} \in HoF(OBS_k^j)$.

Якщо підкомпонент c_i^{j+1} на підставі спостережень OBS_i^j діагностується як несправний на рівні j , то результати спостережень OBS_k^{j+1} від відображення спостережень до рівня $j+1$ можна віднести до однієї з наступних трьох категорій:

Спостереження, які узгоджуються з описом правильної поведінки субкомпонента c_k^{j+1} ; опис некоректної поведінки компонента c_i^j занадто загальний. Діагноз спростовано як помилковий.

Спостереження, які узгоджуються з описом неправильної поведінки субкомпонента c_k^{j+1} . У цьому випадку діагностична процедура може бути продовжена на нижчих рівнях ієрархії.

Спостереження, які не відповідають жодному з описів. Описи неповні, запропоноване рішення - припинити діагностичну процедуру на цьому рівні.

Окремі ситуації показано на рис. 2.8 та 2.9.

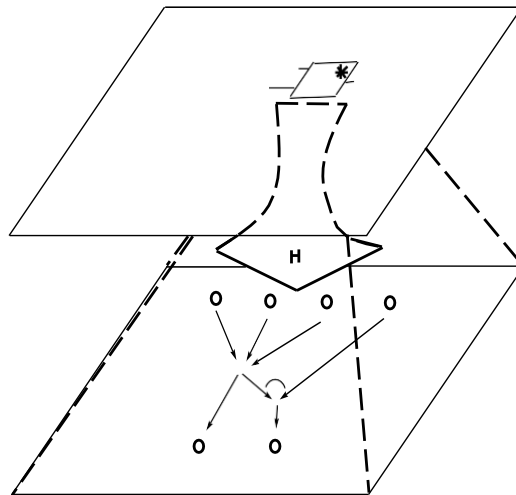


Рисунок – 2.8. Ієрархічна функція.

Резюмуючи, форма функції фокусування залежить від типу опису компонента, форма ієрархічної функції залежить від обох: типу опису компонента і тип опису підкомпонента.

На рис. 2.10 представлено обидві функції відображення в інтуїтивно зрозумілій графічній формі.

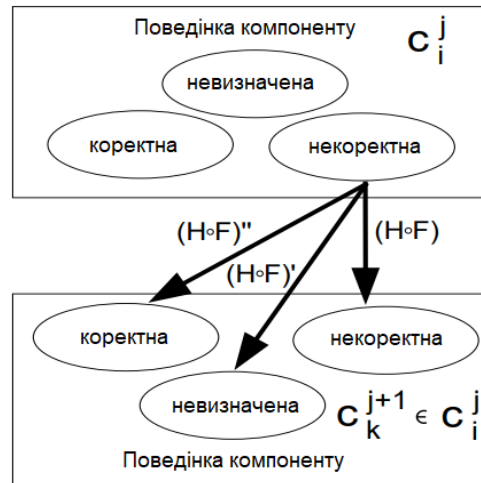


Рисунок – 2.9. Відображення несправностей

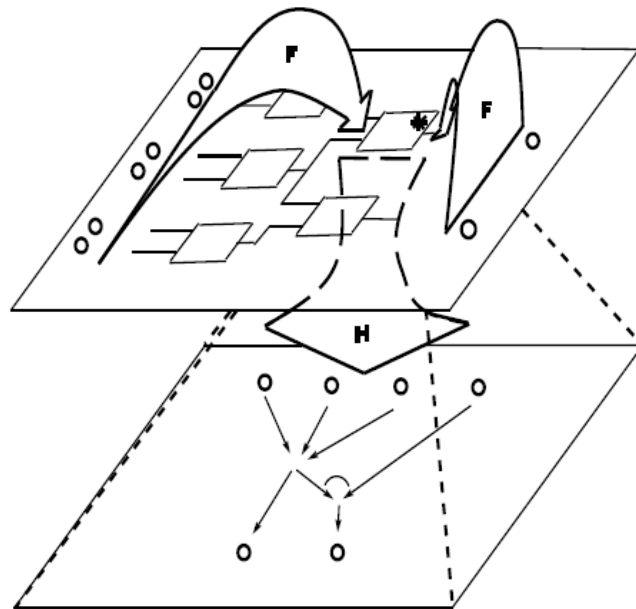


Рисунок – 2.10. Ієрархічна та фокусна функції.

Висновки до розділу 2

В розділі проведено опис методики діагностування DX, яка базується на узгодженості між правильною поведінкою системи та її спостережуваною ненормальною поведінкою.

Як альтернативу розглянуто методику діагностування на основі моделі виявлення та усунення несправностей. Поведінкова модель є основним елементом підходів виявлення та усунення несправностей. Модель, у більшості випадків, походить від структури системи та показує зв'язки між компонентами системи та їхньою моделлю поведінки.

Описано ієрархічну діагностику в гетерогенних середовищах. Наведено основні залежності між компонентами та методи переходів між рівнями діагностування.

3 РОЗРОБКА ДІАГНОСТИЧНИХ АЛГОРИТМІВ ДЛЯ ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ

3.1. Діагностичний алгоритм

Представлений алгоритм є реалізацією методології ієрархічної діагностики. Алгоритм складається з глобальної діагностичної процедури та локальної діагностичної процедури. Процедура глобальної діагностики починається з даного комплексного компонента c_i^j і його спостереження OBS_i^j .

На першому етапі діагностична проблема P_i^j вирішується за допомогою процедури локальної діагностики. Локальні діагнози, отримані на цьому етапі, графічно представлені у вигляді проміжного графіка. На наступному кроці аналізується кожен локальний діагноз d_k і набір атрибутів $ATR_{d_k}^{j+1}$ для діагностики розраховується. Ця частина діагностичної процедури реалізує функцію фокусування. Усі набори атрибутів і проміжний граф повертаються до процедури глобальної діагностики.

Подальша діагностика здійснюється відповідно до стратегії глибини; Також доступна стратегія «спершу вшир», але вона не розглядається в цьому розділі. Перший локальний діагноз d_1 береться з проміжного графіка. Тепер деякий перший підкомпонент c_p^{j+1} вибирається з діагнозу d_1 та його спостережень OBS_p^{j+1} розраховується на основі атрибутів $ATR_{d_1}^{j+1}$. Це Частина діагностичного алгоритму реалізує ієрархічну функцію.

Після завершення діагностики для підкомпонента c_p^{j+1} наступний компонент (у попередньо визначеному порядку) береться з діагнозу d_1 , і процедура повторюється до останнього в діагнозі. Така сама операція виконується з рештою діагнозів для компонента c_i^j .

3.1.1. Алгоритм локальної діагностики

Алгоритм локальної діагностики діагностує компоненти за допомогою відповідної методології для їх діагностичного опису. Результатом локальної діагностики є проміжний граф, який є простим графом зі структурою, специфічною для типу опису компонента. Основною причиною використання графічної форми замість простої колекції діагнозів є те, що графічна форма більш читабельна для користувачів.

За допомогою локального діагностичного алгоритму можна аналізувати два типи діагностичних описів:

Причинно-наслідкові графіки І/АБО/НІ.

Моделі (корисні для аргументації на основі послідовності).

У першому випадку діагностика на основі випадкових графів AND/OR/NOT здійснюється шляхом поширення станів. Результат діагностичної процедури для компонента, описаного на графі, можна записати як диз'юнкцію всіх діагнозів для компонента, де кожен діагноз представлений кон'юнкцією несправних підкомпонентів. Проміжний графік будується так:

```

root := createOrNode()
FOREACH diagnosis  $D_i$  IN diagnosisSet DO
  diagnosisNode := createAndNode()
  addNode(root, diagnosisNode)
  FOR EACH component  $C_{ij}$  IN  $D_i$  DO
    componentLeaf := createLeaf( $C_{ij}$ )
    addNode(diagnosisNode, componentLeaf)
  END FOREACH
END FOREACH

```

Нотація вузлів графа подібна до нотації, яка використовується для причинних графів. АБО вузли є звичайними вузлами графа; вони представляють логічну диз'юнкцію вузлів-нащадків. Вузли AND — вузли з додатковою дугою під вузлом; вони представляють логічне з'єднання вузлів-нащадків. Листи графів позначені назвами компонентів.

Коли необхідно обчислити діагнози з проміжного графа, то стан його кореневого вузла встановлюється на «true». Подальші обчислення виконуються відповідно до правил розповсюдження станів у причинно-наслідкових графах І/АБО/НІ. Якщо стан листка «true», то компонент, назва

якого вказана на етикетці, вважається несправним і стає частиною діагностики.

Приклад. Проміжний граф для гіпотетичного клапана представлений нижче, компонентний опис клапана є причинно-наслідковим графіком. Діагнози для системи: $\{f, a\}$ і $\{e\}$.

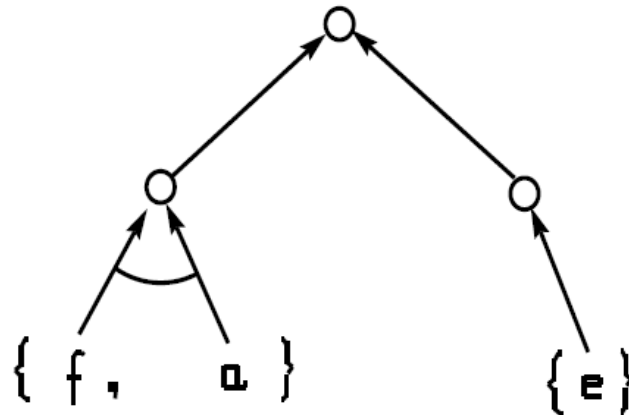


Рисунок – 3.1. Проміжний граф для клапана.

У другому випадку модельно-описані компоненти діагностуються процедурою спростування. Моделі мають відповідати деяким обмеженням; наступні припущення доповнюють ті, що включені в оригінальну теорію:

1. Належна поведінка підкомпонента описується як

$$Y = f(X)$$

де

$$X = (x_1, x_2, x_3, \dots, x_n)$$

$$Y = (y_1, y_2, y_3, \dots, y_n)$$

Y - вектор вихідних змінних, X – вектор вхідних змінних.

2. Функцію f можна обчислити вперед, якщо вказано всі його вхідні дані. Функція f^l може обчислюватися у зворотному порядку для вхідної змінної, якщо відомі інші входи та виходи.

Компоненти з більш ніж одним виходом (концептуально) розділяються перед діагностикою та знову збираються одразу після неї. Кількість нових (віртуальних) компонентів, отриманих у результаті розбиття, дорівнює кількості виходів у вихідному компоненті. Якщо принаймні один із нових

компонентів визнано несправним, то оригінальний компонент є несправним і буде додатково діагностовано на нижчих рівнях ієрархії (якщо застосовано).

Діагнози для компонента, описаного моделлю, генеруються в два етапи. На першому етапі розраховуються конфліктні набори на основі компонентних спостережень і моделі. Якщо існує різниця між значенням спостереження та відповідним значенням, отриманим з моделі, то спостереження не узгоджуються з моделлю. Процедура спростування генерує всі можливі набори компонентів, які впливають на невідповідності. На другому етапі з конфліктних наборів генеруються діагнози; кожен діагноз є набором влучень усіх конфліктних наборів.

Набір конфліктних наборів перетворюється на проміжний граф. Трансформацію можна виконати подібно до трансформації для компонентів, описаних на графі, але це також можна зробити більш компактним способом, який краще пояснює, як генеруються діагнози для компонентів, описаних моделлю. Остання властивість особливо корисна для взаємодії з користувачами.

Проміжний графік будується так:

```

root := createAndNode()
FOREACH conflictSet  $C_i$  IN conflictSets DO
  conflictNode := createOrNode()
  addNode(root, conflictNode)
  FOR EACH component  $C_{ij}$  IN  $C_i$  DO
    componentLeaf := createLeaf( $C_{ij}$ )
    addNode(conflictNode, componentLeaf)
  END FOREACH
END FOREACH

```

Приклад. Проміжний граф для арифметичної системи, описаної в вище представлений на рис. 3.2. Набори конфліктів для системи: $\{a1, m1, m2\}$ і $\{a1, a2, m1, m3\}$.

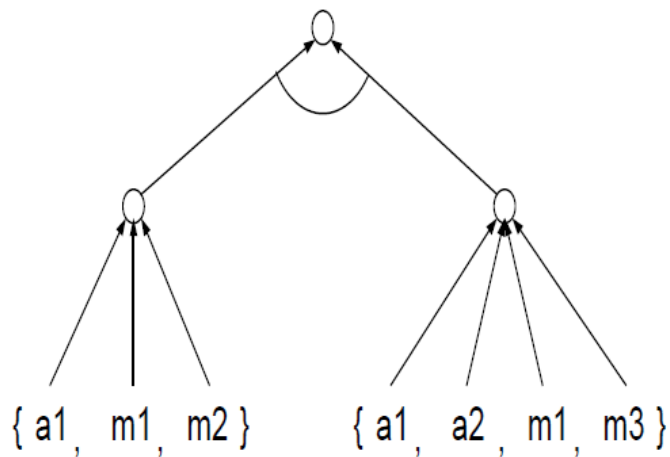


Рисунок – 3.2. Проміжний граф для арифметичної системи, описаної моделлю.

Діагноз — це будь-яка множина конфліктів, що збігаються. Це означає, що для формування діагнозу необхідно взяти хоча б один підкомпонент з кожного конфліктного набору.

Проста графова процедура, яка поширює стани в такому проміжному графі, бере один компонент із кожного конфліктного набору. Зверніть увагу, що деякі діагнози, створені таким чином, можуть бути не мінімальними щодо включення набору. Було показано, що мінімальні множини влучень мають деякі корисні властивості для ієрархічної діагностики, тому ми навмисно наблизили діагнози з проміжного графа до чистих мінімальних множин влучень.

Процедура графа візьме підкомпонент лише з цих наборів збігів, які не включають жодного з попередньо вибраних підкомпонентів (з попередніх наборів збігів).

Правило частково виключає набори збігів, які включають більше компонентів, ніж мінімум, і воно дуже схоже на основне правило побудови HS-дерева.

На жаль, лише деякі немінімальні набори збігів усуваються таким чином через послідовність вибраних конфліктних наборів (послідовний характер процедури графа). Коли необхідно обчислити справді мінімальні набори збігів, у значенні включення набору, тоді кожен згенерований набір збігів слід

порівнювати з уже створеними наборами збігів і або відхилити, якщо це надмножина, або прийняти замість усіх її надмножин.

3.1.2. Глобальний діагностичний алгоритм

Результатом діагностичного процесу для складної складової є сукупність альтернативних діагнозів. Кожен діагноз є поясненням аномальної поведінки складного компонента та включає деякі його підкомпоненти. Усі підкомпоненти в діагнозі повинні бути несправними для збереження узгодженості між спостереженнями та описом складних компонентів DS .

Логічний аспект ієрархічної діагностики вимагає рекурсивного виконання наступної формули:

$$AB(c_i^j) = \bigvee_{d_k \in D_i} \left(\bigvee_{c_n^{j+1}} AB(c_n^{j+1}) \right)$$

Рекурсія формули закінчується, коли або колекція діагнозів для складного компонента порожня, або компонент є елементарним компонентом. Точніше, діагностична процедура зупиняється і повертається у вигляді дерева діагнозів, коли:

1. Діагностований компонент є елементарним компонентом. Алгоритм повертається до свого суперкомпонента, і компонент розпізнається як атомарний елемент діагнозу.

2. Неможливо встановити значення для відповідної кількості атрибутів або спостережень для компонента. Компонент розглядається як елементарний компонент.

3. Діагностична процедура завершується без встановлення діагнозу. Це трапляється, якщо або спостереження породжують конфлікт у причинно-наслідковому графіку, або спостереження не породжують жодного конфлікту в моделі. Алгоритм повертається до суперкомпонента, поточний обстежуваний діагноз пропускається, і аналізується наступний за порядком діагноз.

4. Встановлені значення атрибутів S_1 суперечать або доменам атрибутів, або іншим умовам, накладеним на атрибути. Алгоритм повертається до свого суперкомпонента, поточний діагноз пропускається, і аналізується наступний за порядком діагноз. Усі набори атрибутів S_2 , такі, що буде $S_1 \subseteq S_2$ будуть також конфліктними (як для компонентів, описаних моделлю, так і для описаних графом).

Кінцевий результат діагностичного процесу можна показати у візуальній формі у вигляді простого дерева діагнозів T_R (наприклад, рис. 3.3). Кожен складний компонент c_i^j позначає вузол компонента; вузол є АБО вузлом. Кожна діагностика d_k складного компонента c_i^j представлена вузлом-нащадком вузла компонента. Вузли, позначені діагнозами, є діагностичними вузлами і вони є І вузлами.

Кожен підкомпонент компонента c_i^j , який належить до діагнозу, є нащадком вузла діагнозу. Діагностичні вузли можна розширити до компонентних вузлів, а компонентні вузли можна розширити до додаткових діагностичних вузлів (якщо діагнози існують). Найвищий комплексний компонент в ієрархії представлено коренем T_R . Решта дерева будується рекурсивно вниз за результатами діагностики складних компонентів.

Розглянемо дерево, що моделює ієрархічну діагностичну процедуру, як показано на рис. 3.3. Введемо таке визначення:

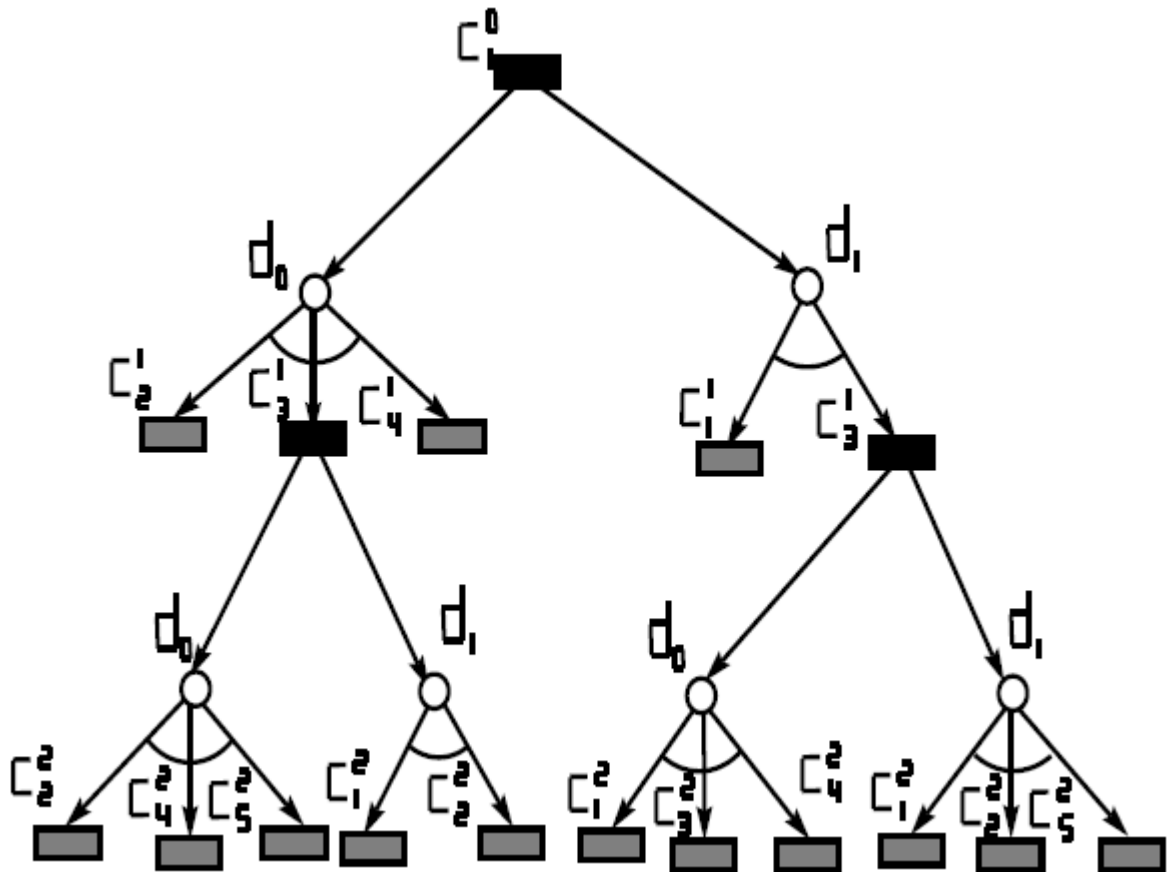


Рисунок – 3.3. Дерево діагнозів T_R для складної системи.

Означення. Ієрархічна діагностика для будь-якого компонента c_i^j на рівні j — це будь-яке піддерево, яке задовольняє такі умови:

Коренем піддерева є будь-який з діагностичних вузлів, розташованих безпосередньо під компонентним вузлом c_i^j .

Для будь-якого компонентного вузла c_n^k в дереві щонайбільше одне піддерево з кореневим вузлом, який є вузлом діагностики для вузла компонента c_n^k , належить до дерева, що визначає ієрархічну діагностику.

З огляду на наведене вище визначення, будь-який компонент ієрархічної діагностики моделювання дерева може бути розширений (униз), і таким чином розробляється ієрархічна діагностика для нього; для листових вузлів, однак, розширення неможливе (це елементарні діагнози). Ієрархічна діагностика для всієї системи побудована як піддерево, що відповідає наведеному вище визначенню та з кореневим вузлом діагностики, розташованим безпосередньо під компонентом c_1^0 .

Зверніть увагу, що для даного компонента можна визначити різні діагнози щодо ступеня розширення. Наприклад, є два різні ієрархічні діагнози, позначені на рис 3.4 та рис. 3.5.

Найбільш розширена ієрархічна діагностика – це діагностика, у якій усі компонентні вузли не мають вузлів-нащадків, тобто така, у якій усі компонентні вузли фактично є елементарними компонентами (вони не мають внутрішньої структури підкомпонентів). Але в більшості випадків такий діагноз важко отримати через проблеми з відображенням під час ієрархічного діагностичного процесу. Діагноз, представлений на рис. 3.5 ($\{c_1^1, c_1^2, c_1^3, c_1^4\}$), є найбільш розширеним ієрархічним діагнозом, якщо всі його компоненти є елементарними компонентами EC .

Глобальна діагностична процедура передбачає діагностику всієї системи, і діагноз ґрунтується на результатах, отриманих під час локальної діагностичної процедури. Алгоритм глобальної діагностики заснований на пошуку зверху вниз в дереві діагнозів. Такий пошук має дві основні версії: пошук вглиб і пошук вшир.

Найпростіший ескіз процедури пошуку в глибину виглядає наступним чином:

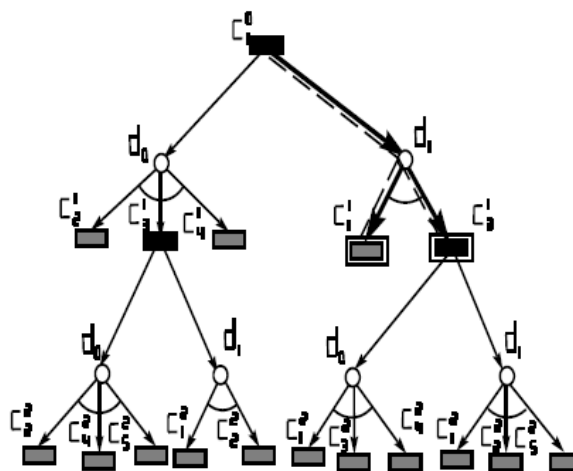


Рисунок – 3.4. Дерево діагнозів T_R з позначеним ієрархічним діагнозом $\{c_1^1, c_3^1\}$

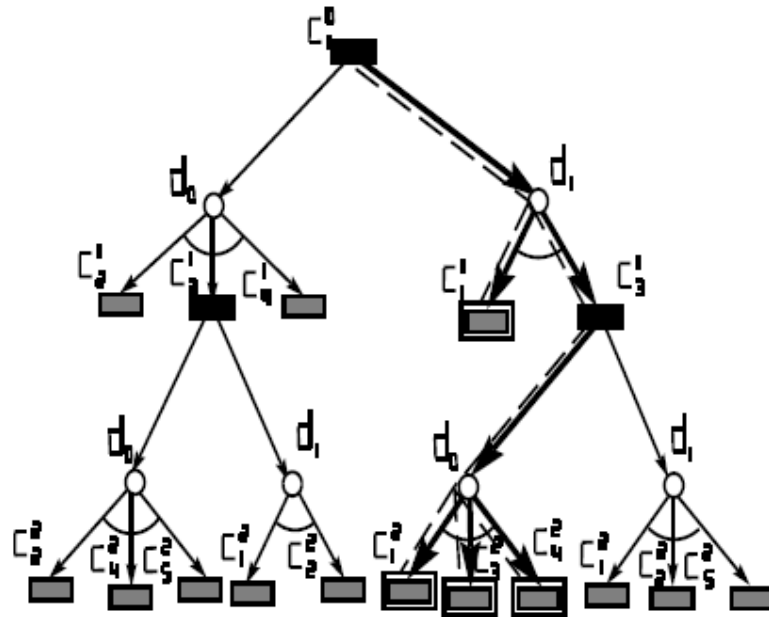


Рисунок – 3.5. Дерево діагнозів T_R з позначеним ієрархічним діагнозом $\{c_1^1, c_1^2, c_3^2, c_4^2\}$

1. $i = 1, j = 0$
2. $P_i^j \rightarrow D_i^j$
3. SELECT p, k THAT $c_k^{j+1} \in d_p$ IS NOT PROCESSED AND $d_p \in D_i^j$
4. IF k EXISTS THEN $j = j + 1, i = k$ AND GOTO 2
5. IF $j = 0$ THEN EXIT
6. IF k NOT EXISTS THEN $j = j - 1$ AND GOTO 3

Детальна форма процедури пошуку в глибину представлена нижче (написана псевдо-код). Він запускається з кореневого компонента на рівні 0 параметра i далі викликається рекурсивно до генерації всіх можливих діагнозів.

```

PROCEDURE globalDiagnosticD(component, observations, nodeOR)
  intermediateGraph, attributesF := localDiagnosticF(
    component, observations)
  IF intermediateGraph IS EMPTY THEN
    RETURN component_refused
  END IF
  diagnosis := getFirstDiagnosis(intermediateGraph)
  WHILE diagnosis IS NOT EMPTY DO
    diagnosisNode := createAndNode()
    addNode(nodeOR, diagnosisNode)
    FOREACH subcomponent IN diagnosis DO
      subcomponentNode := createOrNode()
      addNode(diagnosisNode, subcomponentNode)
      IF subcomponent IS elementary THEN
        closeNode(subcomponentNode)
    
```

```

        CONTINUE
    END IF
    atr := getAttributes(diagnosis, subcomponent,
        attributesF)
    conditions := getConditionsCND(subcomponent)
    IF NOT checkConditions(atr, conditions) THEN
        EXIT FOREACH
    END IF
    subobservations:=mapAttributesH(subcomponent, atr)
    IF NOT correctSetOf(atr, subobservations) THEN
        closeNode(subcomponentNode)
        CONTINUE
    END IF
    IF globalDiagnosticD(subcomponent, subobservations,
        subcomponentNode) = component_refused THEN
        closeNode(diagnosisNode)
        EXIT FOREACH
    END IF
END FOREACH
diagnosis := getNextDiagnosis(intermediateGraph)
END WHILE
END PROCEDURE

```

Діагнози для всієї комплексної системи можуть бути показані після завершення діагностичної процедури.

3.2. Діагностичний процес

Процес діагностики для ієрархічних складних систем подібний до типового процесу діагностики та включає наступні три основні кроки:

- виявлення несправностей,
- генерація діагнозів (ізоляція несправностей),
- верифікація діагнозів (з певною процедурою виключення, додатковими вимірюваннями, тестами тощо).

3.2.1. Виявлення несправностей

Діагностична система працює одночасно з діагностованою комплексною системою. Деякі змінні процесу з діагностованої системи зчитуються датчиками та надсилаються в діагностичну систему. Вони стають спостереженнями *OBS*.

Спостереження порівнюються з розрахованою поведінкою для порівняння на основі моделі компонентів або очікуваної поведінки для компонентів, описаних на графі. Якщо характер спостережень є дискретним, що в основному відбувається, коли діагностуються цифрові одиниці, тоді процес діагностики є легшим і базується на порівнянні між прогнозованими значеннями змінних моделі та їх спостережуваними значеннями.

Гірший випадок для безперервних змінних. Вони більш поширені, але також складніші для виявлення несправностей. Саме шум вимірювання та тимчасові флуктуації процесу змінюють спостереження та ускладнюють їх порівняння. У такому випадку визначаються допустимі діапазони помилок, і несправність виявляється, коли помилки виходять за межі цих діапазонів (рис. 3.6). Часто необхідно виконувати додаткові умови часу або амплітуди.

Несправності можуть бути виявлені на кожному рівні ієрархічної моделі, де для такого виявлення достатньо кількох спостережень. Достатня кількість компонентів, описаних моделлю, повинна дозволяти генерувати надлишкові значення для деяких змінних моделі: розрахункових і спостережуваних. Для компонентів, описаних на графіку, кількість і значення спостережень повинні дозволяти зробити висновок про те, що поведінка компонента є неправильною або правильною.

3.2.2. Локалізація проблем

Якщо виникає несправність, то всі компоненти, які є попередниками несправного компонента c_i^j можуть бути виявлені як несправні. Теоретично всі компоненти c_k^i що $c_k^i \in ANTECEDENTS(c_i^j) \cup c_i^j$ можуть бути помилковими, і їхні спостереження не повинні узгоджуватися з їх правильною поведінкою.

Якщо набір *FAULTS* включає всі несправні компоненти, і ці несправності виявлено для деяких компонентів $c_m^n \in DETECTED$, тоді починається діагностика для кореневого компонента c_i^j , щоб

$$c_i^j \in DETECTED \text{ and } ANTECEDENTS(c_i^j) \text{ not } \in DETECTED$$

що дозволяє знайти найбільш повний набір діагнозів. Виявлення можливе лише на рівнях, де достатньо кількох спостережень. Подальша діагностика компонента c_i^j здійснюється загальною діагностичною процедурою. Якщо діагноз не знайдено, то процедура повторюється для наступного кореневого компонента, який знайдено для нового набору виявлених несправнб[компонент:

$$DETECTED = DETECTED - \{c_i^j\}$$

Приклад. На рис. 3.6 представлена гіпотетична ієрархічна модель, де компонент c_3^3 є несправний і два компоненти виявлені як несправні: c_2^1 і c_2^5 . Діагностика розпочата для вищого компоненту: c_2^1 .

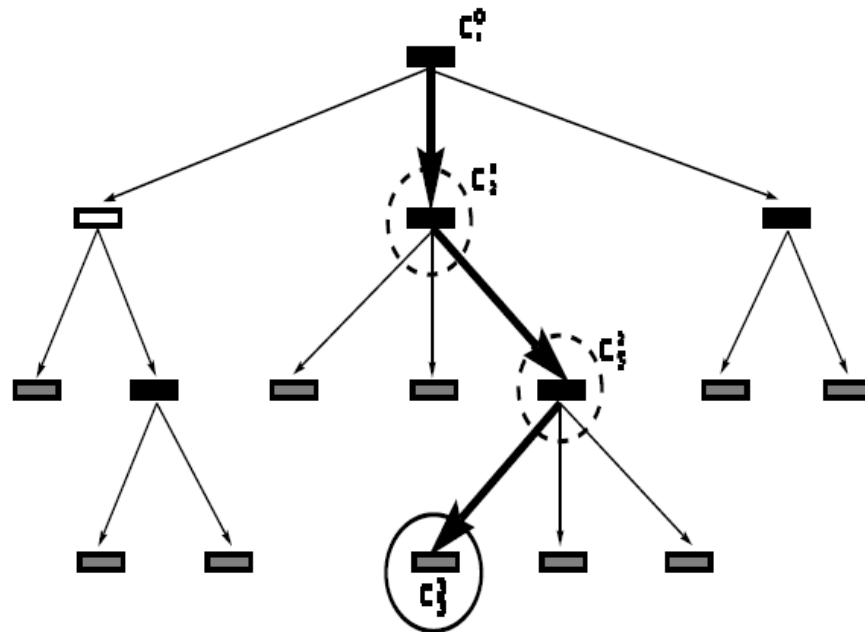


Рисунок – 3.6. Виявлення помилок в ієрархічній моделі

3.2.3. Верифікація діагнозів

Існують два типи верифікації для ієрархічної діагностики:

Теоретична – на основі наявних спостережень і властивостей моделей.

Фізична - заснована на додаткових спостереженнях або тестах, які виконуються після виявлення несправності та дають додаткові спостереження.

Теоретична перевірка

Як зазначалося в попередньому розділі, глобальний діагностичний алгоритм може відхиляти деякі діагнози в кількох випадках. Два з них важливі для діагностичної перевірки:

Значення атрибутів виходять за межі своїх областей (фізична неможливість). Діагностична процедура на ієрархічному рівні завершується без будь-якої діагностики.

Крім того, можна додати сюди ще одну умову:

Спостереження $OBS_i^{j/calc}$ обчислюється функціями відображення F і H є непослідовними зі спостереженнями реального світу OBS_i^j .

Приклад. Наприклад, одним із діагнозів для арифметичної одиниці на є $\{m1\}$. Значення на виході несправного підкомпонента має бути рівним 17 для спостережень, які показані на рис. 2.1. Розраховуємо його на основі компонентів: $m2$, $a1$. Множник $\{m1\}$ на нижчому ієрархічному рівні може виглядати так, як показано на рис. 3.7.

Модель на нижньому рівні показує, що множник не здатний створити вихідне значення, що дорівнює 17 (максимальне значення дорівнює 15), а 17 виходить за межі області визначення виходу множника. Отже, діагноз $\{m1\}$ буде відхилений через фізичну неможливість.

Перший випадок очевидний.

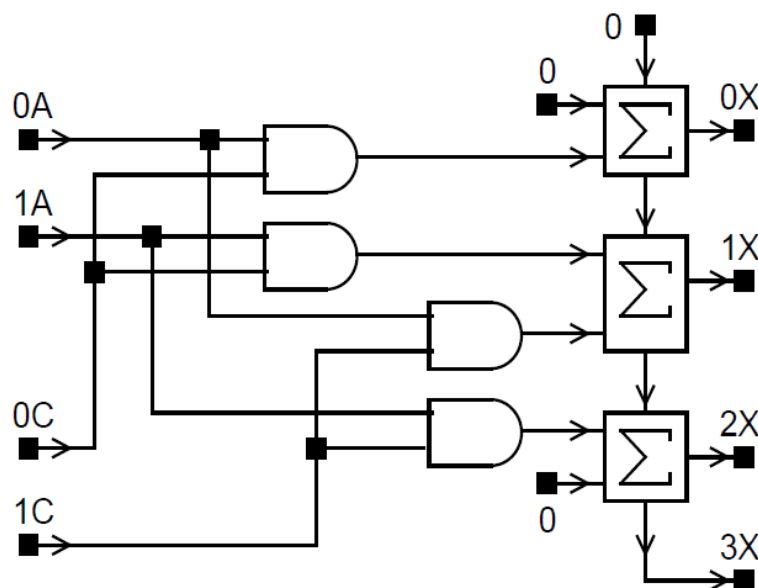


Рисунок – 3.7. Помножувач.

Припустимо, що компонент описується набором атрибутів

$$ATR = ATR_d \cup ATR_u$$

де: ATR_d є атрибут з визначеним значенням,

ATR_u є атрибут з невизначеним значенням.

Тепер, якщо компонент відхилено як частину діагностики, компонент буде відхилено також для набору атрибутів, таких як

$$ATR' = ATR'_d \cup ATR'_u$$

де:

$$ATR'_d \supseteq ATR_d$$

Доказ цього факту є простим для відмови на основі обмежень домену для компонентів на основі моделі. Відмова на основі конфліктів у причинно-наслідкових графах AND/OR/NOT також відповідає наведеній вище властивості. Якщо неможливо пояснити поведінку компонента за спостережуваними симптомами ATR_d та будь-яким значенням невизначених симптомів ATR_u , тоді поведінка також буде непоясненою на основі надмножини спостережуваних симптомів ATR'_d .

Другий випадок виникає, коли компонент визначається як несправний на рівні j , але його поведінка класифікується як нормальна або невизначена на нижчому рівні $j+1$. Причини такої ситуації можуть бути наступними:

слабкість опису компонентів на рівні $j+1$,

навмисне усунення деяких неможливих випадків на рівні $j+1$,

навмисне послаблення опису компонента на рівні j .

Наприклад, логічна функція, яка представлена причинно-наслідковим графіком, не може бути виконана для поточних значень спостережуваних симптомів і будь-яких значень невизначених симптомів.

Приклад. На рис. 3.8 представлено простий експертний граф моделювання вентиля EX-OR, вентиль має два входи $in1(X1) = 0$, $in2(X1) = 0$ і один вихід $m1(X1) = 1$.

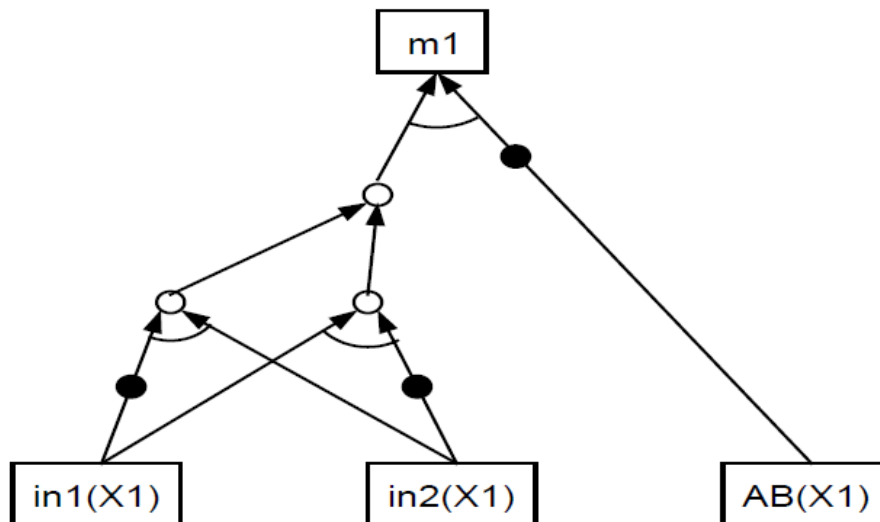


Рисунок – 3.8. Граф AND/OR/NOT для вентилля XOR

Логічна функція, пов'язана з графіком, така:

$$m1(X) = (in1(X)XOR in2(X))AND NOT AB(X)$$

Зазначена вище функція має три спостережувані симптоми: $in1(X1)$, $in2(X1)$, $m1(X1)$; і один елементарний діагноз $AB(X1)$. Зазначені вище значення симптомів породжують конфліктну функцію, яка не може бути пояснена жодним значенням діагнозу $AB(X)$.

Невідповідності між спостереженнями $OBS_i^{j/calc}$, обчисленими за допомогою функцій відображення (F і H) і справжні слова спостереження OBS_i^j також призводять до відмови від розглянутого діагнозу. Відмінності повинні бути суттєвими, як і при виявленні несправності. Якщо обидва спостереження закриті, то для подальших обчислень береться справжнє слово спостереження. В іншому випадку діагностичний алгоритм зупиняється та повертається назад до суперкомпонента, поточний досліджуваний діагноз пропускається.

Є ще один шанс для теоретичної перевірки деяких діагнозів. Якщо один і той же компонент повторно використовується на кількох рівнях (рис. 3.9), тоді можна порівняти його діагнози з рівнів, наборів несправних підкомпонентів.

Під повторним використанням тут розуміється використання одного і того ж компонента кількома суперкомпонентами, але без одночасного впливу на компонент. Якщо вплив є одночасним, тоді вплив слід моделювати на вищому ієрархічному рівні, який є загальним для всіх впливових суперкомпонентів (рис. 3.10).

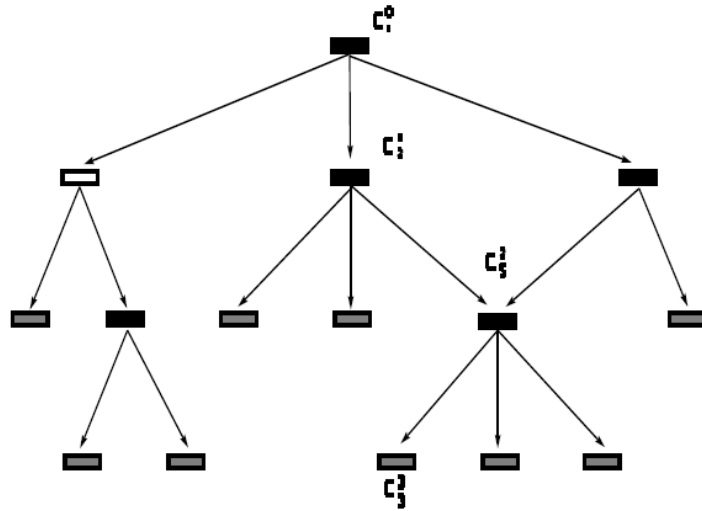


Рисунок – 3.9. Приклад повторного використання компонента.

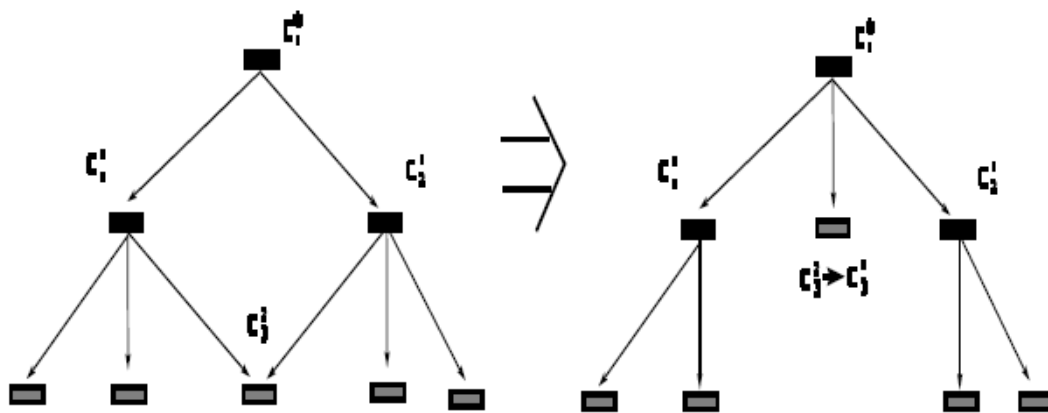


Рисунок – 3.10. Необхідне перетворення для компонентів з більш ніж одним суперкомпонентом.

Хорошим прикладом правильного повторного використання є суматор, який використовується кількома іншими електронними блоками. Найпростіша стратегія для такого порівняння діагнозів субкомпонента полягає в тому, що тільки ті діагнози, які повторюються, будуть далі розглядатися на нижчих рівнях ієрархії.

3.3. Приклади використання ієрархічного діагностування

3.3.1. Графічне зображення діагнозів

Взаємодія між людиною та діагностичною системою вимагає дружньої форми подання діагнозів, яка також повинна бути здатною пояснити, як генеруються діагнози. Пропонована графічна форма базується на проміжних графах і може бути легко реалізована як комп'ютерний додаток.

Графічне представлення результатів, отриманих у результаті процедури глобальної діагностики, є комбінацією проміжних графіків, отриманих у результаті процедури локальної діагностики.

Приклад. На рисунку 3.11 у графічній формі представлені результати процедури глобальної діагностики для ієрархічної моделі арифметичної системи на рис. 2.1.

Елементарні компоненти позначені порожніми квадратами (наприклад, $a1$). Квадратики з хрестиком означають, що компонент має діагнози, а підкомпоненти в діагнозах можна відобразити (наприклад, $m1$). Якщо наразі відображаються підкомпоненти компонента, то всередині квадрата є позначка мінус (наприклад $m2$). Квадрат сірий, якщо компонент описується моделлю (наприклад, $a2$). Квадрат зелений, якщо компонент описується графіком.

На найвищому рівні моделі є три множника і два суматора. Конфліктні набори для спостережень: $\{a1, m1, m2\}$, $\{a1, a2, m1, m3\}$. Мінімальний набори влучень для спостережень: $\{a1\}$, $\{m1\}$, $\{a2, m2\}$, $\{m2, m3\}$. Компоненти $a2, m1, m2, m3$ включають підкомпоненти.

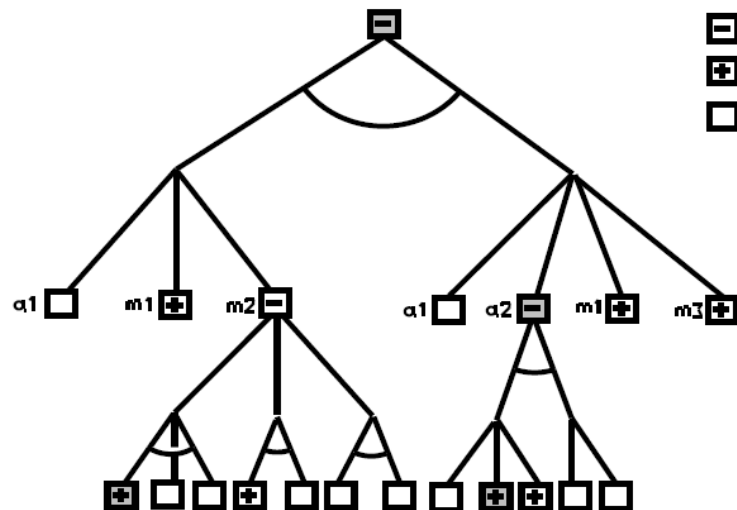


Рисунок – 3.11. Графічне представлення результатів.

Діагнози для компонента, описаного на графі, можуть бути представлені безпосередньо як проміжний граф, з'єднаний з вузлом, який є представленням компонента.

Інша ситуація з компонентами, описаними моделлю. Існує два варіанти вирішення цієї проблеми. Перший тривіальний; графіки для компонентів, описаних моделлю, мають бути подібними до цих для компонентів, описаних на графіках, і включати всі можливі діагнози (наприклад, мінімальні набори попадань). Друге рішення полягає в тому, що з проміжного графа можна вибрати лише один діагноз, і тільки цей діагноз буде графічно «розгорнутий» на нижчих рівнях ієрархії.

3.3.2. Концептуальна система для ієрархічного моделювання

Запропонована концептуальна система є комп'ютерним додатком із графічним інтерфейсом, основною метою якого є підтримка проектування ієрархічних моделей складних систем. Система підтримки діагностики (DSS) також може використовуватися для моніторингу та контролю процесу діагностики.

Діагностовані системи можна змоделювати та представити за допомогою кількох основних видів:

Вид рівня– показ діагностичного опису заданого рівня. Цей перегляд дає можливість вводити поточні значення спостережень. Існує два основних види цього перегляду:

- І/АБО/НЕ переглядати– візуалізація графоописаних рівнів.
- Вид моделі– візуалізація модельно-описаних рівнів.

Ієрархічний вигляд моделі– представлення всіх компонентів разом із позначеними залежностями між ними. Цей режим перегляду забезпечує легкий доступ до режимів перегляду рівня.

Діагностичний вид– відображення розрахованих діагнозів у графічній формі, яка представлена в попередньому розділі. Це подання надає доступ до подання рівнів і може використовуватися для керування діагностичними міркуваннями.

Подання ієрархічної моделі і перегляди рівнів є найважливішими під час розробки діагностичних моделей. Користувач може визначити ієрархічну структуру складної системи у вигляді ієрархічної моделі; тут також можна визначити тип опису компонентів.

Тепер кожен компонент в ієрархічній моделі може бути «відкритим» і показаним в окремому рівні. Усі його підкомпоненти, які були визначені в ієрархічній моделі, будуть доступними. Крім того, користувач може додати деякі елементарні компоненти та елементи, які є специфічними для типу опису, наприклад, зв'язки між компонентами для компонентів, описаних моделлю, або дуг і проміжних вузлів для компонентів, описаних на графі.

Після завершення розробки можна ввести значення деяких системних змінних тобто вхідні спостереження. Тепер користувач може вибрати напрямок діагностики на нижчих рівнях шляхом вибору діагнозу з уже діагностованих рівнів; Форма діагностичного перегляду представлена на рис. 3.13.

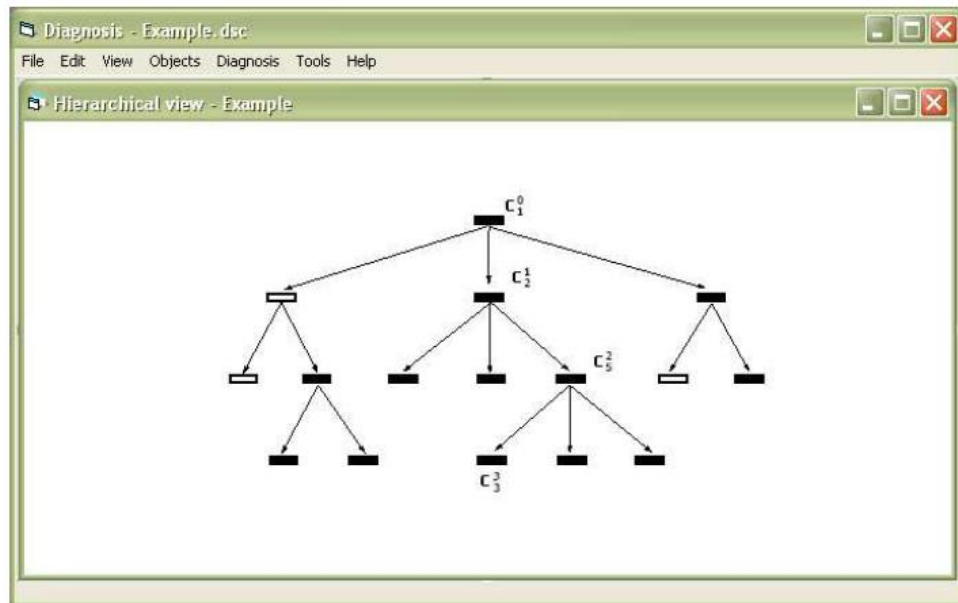


Рисунок – 3.12. Подання ієрархічної моделі.

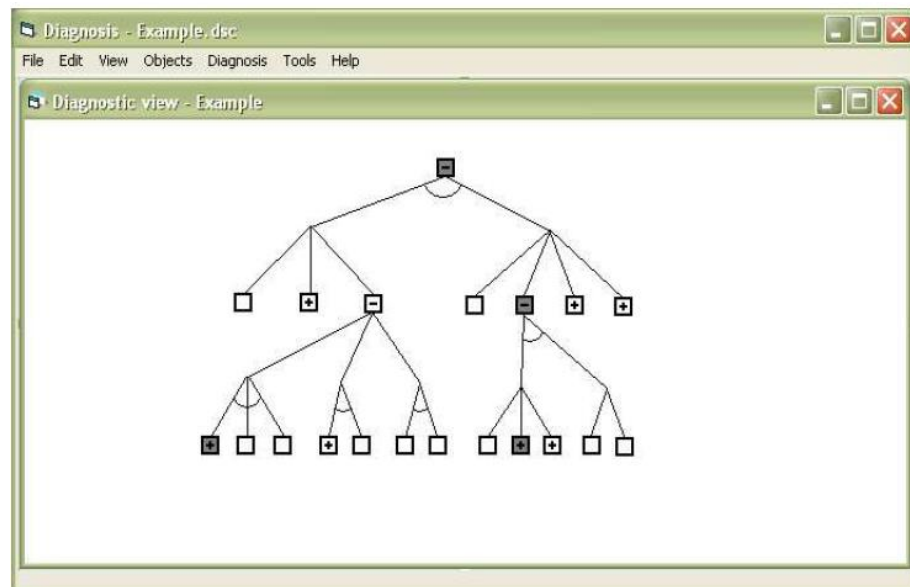


Рис. 3.13. Діагностичне подання

3.3.3. Приклад діагностики системи трьох ємностей

Наступна розглянута система є частиною більшої системи наповнення. Мета системи полягає в підтримці постійного рівня рідини в резервуарах T_I і T_{II} . У комплект ємності (рис. 3.14) входять:

- T_M основний бак,
- T_I вихідний бак I ,
- T_{II} вихідний бак II ,

- V_M головний керований запірний клапан,
- V_I регулюючий клапан для випускного ємності I ,
- V_{II} регулюючий клапан випускного ємності II ,
- V_{MI} контрольована запірна арматура між ємностями M і I ,
- V_{MII} контрольована запірна арматура між баками M і II ,
- S_M давач рівня в основній ємності M ,
- S_I давач рівня в основній ємності I ,
- S_{II} давач рівня в основній ємності II ,
- C цифровий контролер,
- P живлення.

Діагностичний опис

Модель резервуарної системи представлена на рис. 3.15. Резервуари контролюються окремо. Отже, контролер C може бути розділений на три віртуальні контролери C_M , C_I і C_{II} .

Модель керованої арматури («Клапан - I», «Клапан - II») представлена на рис. 3.16. Є наступні підкомпоненти та змінні:

- P_S адаптер живлення,
- C_N контрольний сигнал,
- W_P вхідний тиск води,
- C_M контролер клапанів,
- D_A перетворювач сигналу,
- M кроковий двигун, позиціонер,
- V_M механічний клапан,
- * віртуальний компонент, що перераховує відкриття клапана на потік,
- F_L потік води.

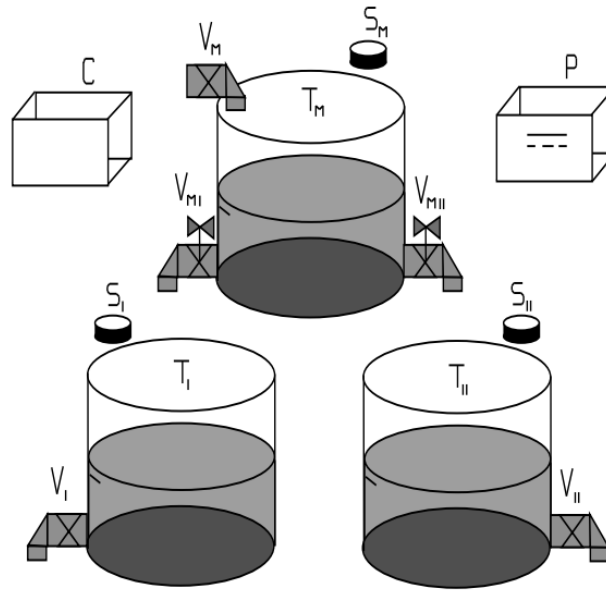


Рисунок – 3.14. Схема системи ємностей.

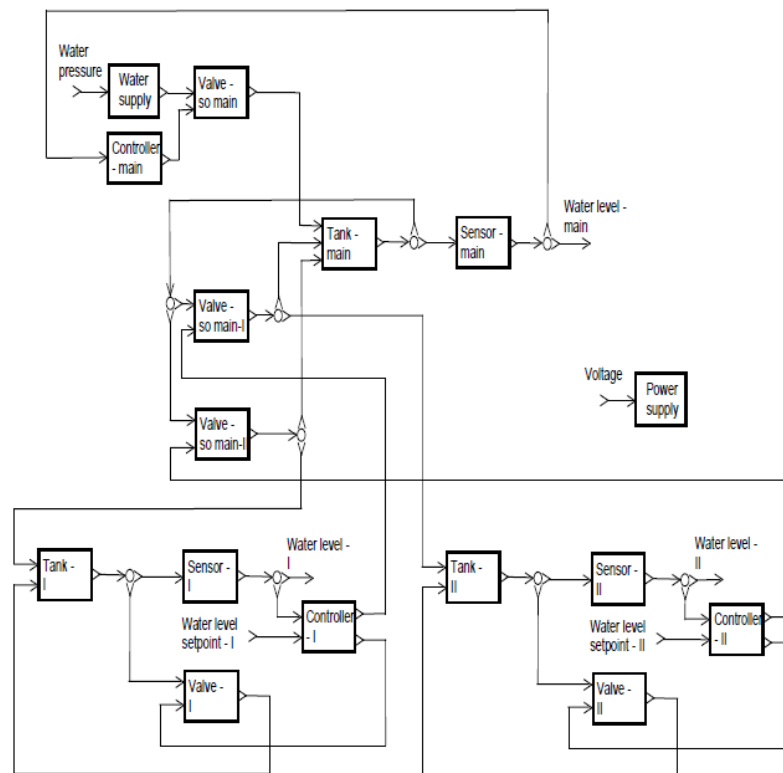


Рисунок – 3.15. Принципова схема резервуарної системи.

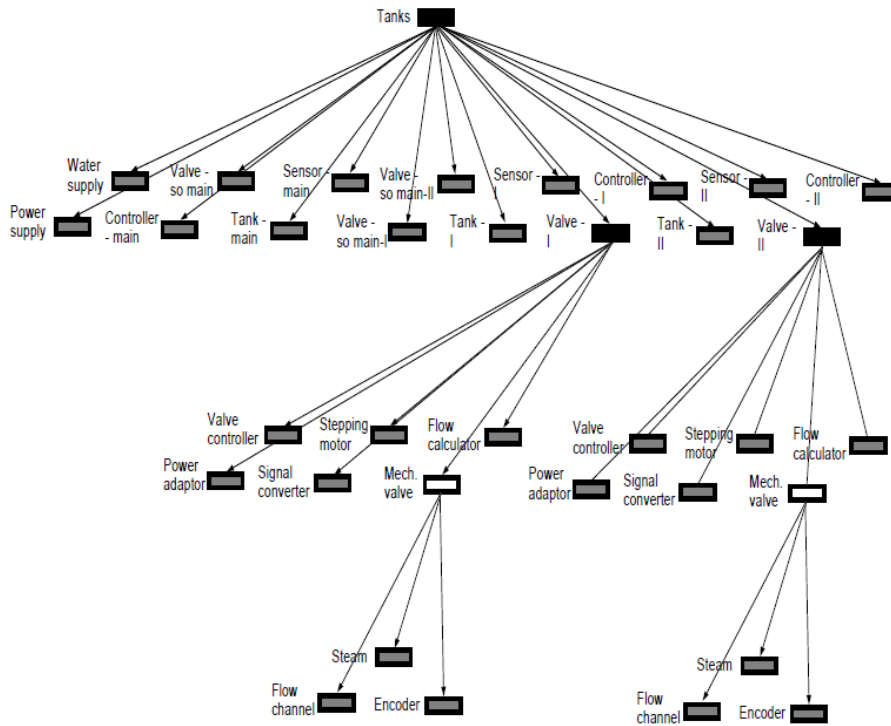


Рисунок – 3.16. Дерево структури компонентів для ємностей

Немає сенсу проводити більш глибоку діагностику клапанного контролера і перетворювача сигналу, тому що вони об'єднані в одну схему. Продіагностуємо механічну частину клапана (рис. 3.18).

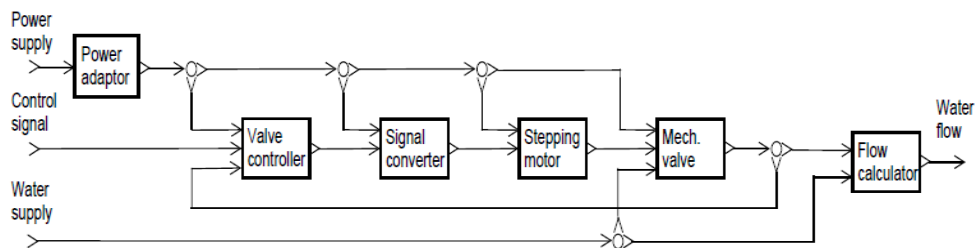


Рисунок – 3.17. Схема клапана

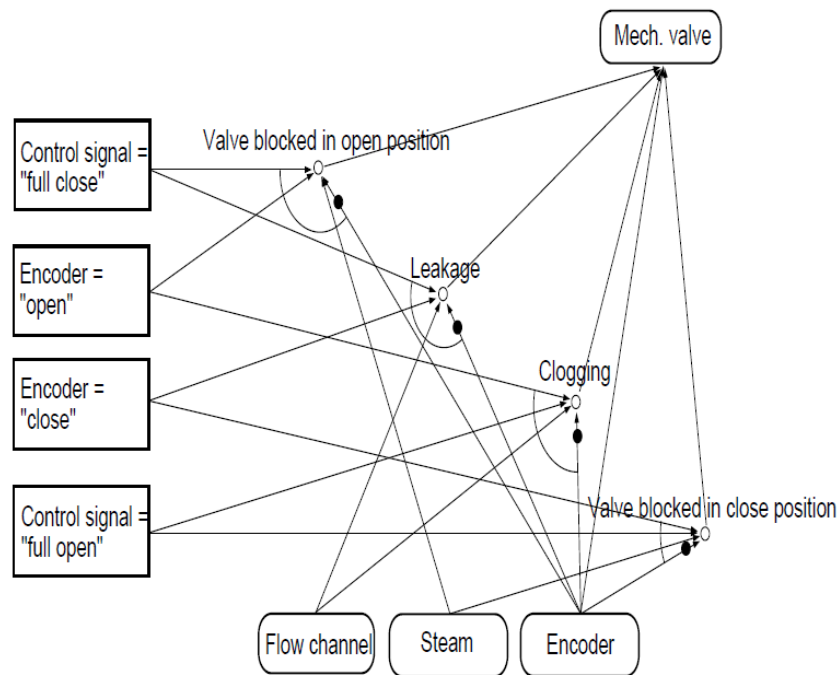


Рисунок – 3.18. Причинно-наслідковий графік для електромагнітного клапана.

На основі представленої моделі можна провести діагностику і правильно розрахувати діагнози. Через петлі зворотного зв'язку на рівні та просту структуру моделей на нижчих рівнях, є кращі результати, якщо доступні деякі додаткові спостереження (крім входів і виходів компонентів) (наприклад, положення клапанів).

3.3.4 Застосування систем діагностування для виявлення несправностей при діагностуванні технологічного процесу енергогенерації

Проведення діагностики онлайн промислових технологічних процесів є ефективною процедурою для підвищення функціональної безпеки самого процесу та зменшення економічних втрат, спричинених збоями в роботі чи несправностями обладнання, що збільшує тривалість ремонтного простою. Раннє розпізнавання несправностей, що виникають, викликає потребу для операторів технологічних процесів вживати відповідних дій, які би могли дозволити відносно швидко та успішно проводити діагностування з метою попередження збоїв. Таким чином, блокування та заходи безпеки, передбачені та реалізовані в системах керування при правильному підході до

діагностування фактично не будуть активовані, а отже, зупинка процесу не відбувається [10].

Таким чином, діагностичні процедури гарантують, серед іншого, значну економію при виконанні технологічного процесу. При цьому точне та якісне діагностування, дозволяє ізолювати несправності та застосовувати відповідні рішення та ремонтні дії. Точність діагностування можна визначити шляхом додавання кількості несправностей, виявлених у кожному елементарному блоці [11]. Чим менша кількість несправностей зібрано в елементарному блоці, тим точніша процедура діагностування. Точність залежить від ступеня розрізнення несправності, досягнутого в даній системі. Під розрізненням дефектів автори розуміють здатність розпізнавати (розрізняти) одиничні дефекти. Як правило, розпізнавання несправностей базується на аналізі відповідно оброблених результатів діагностичних тестів. Діагностичні тести виявляють набір характерних симптомів, пов'язаних із несправністю або набором несправностей. Вони можуть бути визначені або як патерн конкретних значень (симптомів) діагностичних сигналів, а також як певні послідовності (порядки) цих сигналів [11]. Зрозуміло, що недоліки неможливо розрізнити, якщо їх усі симптоми ідентичні. Тому розрізнення несправності залежить від вибору симптомів. Показники розрізнення несправностей є дуже потрібними при побудові систем діагностування. Вимоги до розрізнення несправностей можуть бути визначені, наприклад, шляхом вказування набору пар несправностей, які необхідно розрізняти. Як правило, оператори процесу вимагають розрізнення всіх окремих несправностей. Однак варто зазначити, що це не обов'язково. На практиці достатньо розрізняти ті несправності, наслідки яких є особливо загрозливими. Іншою типовою вимогою є розрізнення несправностей, які можуть бути використані для запуску процедур страхування безпеки процесу. У цьому випадку несправності, до яких відносяться ці процедури, не потрібно розрізняти.

Усунення несправності зазвичай визначається в контексті прийнятого методу діагностики. Його часто аналізують на основі структурних залишків (бінарна діагностична матриця, матриця інцидентності, структурна матриця),

отриманих із лінійних рівнянь залишків у внутрішній формі [12]. Основні визначення були сформульовані термінах структур залишкових множин, а також у контексті залишкового простору [13]. Розпізнавання несправностей є важливим питанням, яке впливає на процес проєктування діагностичних систем. Зрозуміло, що індикація помітних і нерозрізнених несправностей необхідна для перевірки того, чи задовольняються необхідні вимоги ізоляції несправностей. Відповідний вибір змінних процесу є ключовим у питаннях виявлення та розрізнення несправностей. Як правило, чим більше доступних змінних процесу, тим більше моделей процесів потрібно побудувати. Примітно, що розробка моделей, корисних для діагностики, не є тривіальною. Крім того, це складне питання для складних великомасштабних систем. Методика генерації структур усіх моделей за заданим набором вимірювань обговорюється в [14]. На практиці використовуються також евристичні підходи. У цьому випадку як первинні, так і вторинні залишки отримують шляхом об'єднання (зв'язування) сусідніх часткових моделей діагностованої системи. Часткові моделі призначені для невеликих частин системи, яка підлягає діагностуванню. Як правило, збільшення кількості моделей призводить до збільшення показників розрізнення несправностей. Крім того, розрізнення несправностей можна збільшити за допомогою багатозначних залишків. Як правило, використовується тризначна оцінка залишків. При цьому значення діагностичних сигналів належать набору $v = \{0, -1, +1\}$. Значення діагностичного сигналу 0 означає, що значення залишку знаходиться в певній визначеній і прийнятній межі. Таке значення вказує на номінальний поведінковий стан системи. Інші значення називаються симптомами несправності. Прийняття тризначних залишків є виправданим, оскільки деякі несправності можуть спричинити збільшення залишкових значень вище прийнятних порогів (+1), в той час як інші несправності знижують ці значення нижче прийнятних порогових значень (-1). Тому, принаймні, тризначна оцінка залишків гарантує кращу ступінь розрізнення несправностей. Багатозначна оцінка залишків вимагає знання зв'язку несправність—діагностичні сигнали у формі системи інформації про несправності. Під

симптомами несправності розуміються інтерпретовані значення діагностичних сигналів. Оскільки, загалом, діагностована система є динамічною системою, зазвичай виникає затримка між моментом виникнення несправності та моментом, коли з'являється відповідний симптом цієї несправності. Це відставання залежить від динамічних властивостей діагностованої системи. Іншими словами, одна і та ж помилка впливає на різні діагностичні тести в різні моменти часу. Попередження діагностичних сигналів може надати додаткову інформацію, яка може бути корисною в діагностиці [15]. Крім того, це може бути корисним для диференціації несправностей, які неможливо розрізнити на основі оцінки залишкової вартості. Щоб розрізнити будь-яку пару дефектів, достатньо, щоб передування симптомів цих дефектів було різним.

Співтовариство інформатики та штучного інтелекту розробило численні методи виявлення та ізоляції несправностей, що належать до сімейства методів DX [16]. Діагнози отримують на основі конфліктів, де конфлікт – це набір компонентів системи, для яких припущення, що всі ці компоненти є здоровими, суперечить спостереженням. Остаточними діагнозами є набори конфліктних наборів. Суттєва перевага методів DX порівняно з методологіями виявлення та усунення несправностей (Fault Detection and Isolation, FDI) полягає в тому, що не потрібні додаткові заходи для ізоляції множинних недоліків. Одиначні та множинні несправності виділяються таким же чином. Метою розділу моделювання роботи систем діагностування для технологічного процесу керування турбіною.

Виправдання помилок часто передбачається в методологіях FDI, заснованих на аналітичних відношеннях надлишковості (Analytical Redundancy Relation, ARR). Відповідно до цього припущення, якщо деякі ARR задовольняються спостереженнями, то всі компоненти, пов'язані зі спостереженнями, вважаються невідмовними. У підходах DX такого припущення не існує. Загальна структура системи діагностування приведена на рис. 3.19.

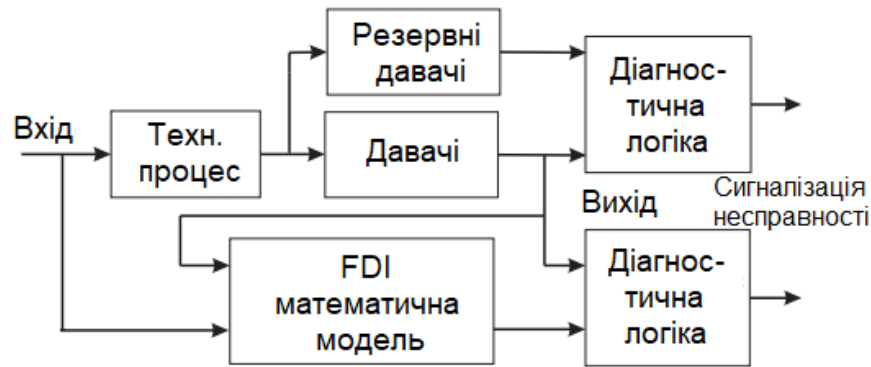


Рисунок – 3.19. Загальна структура системи діагностування технологічного процесу

Усі методи FDI на основі моделі використовують модель моніторингової системи для створення генератора симптомів. Якщо система не є складною і може бути точно описана математичною моделлю, FDI здійснюються безпосередньо за допомогою простого геометричного аналізу залишків. Однак у реальних промислових системах невизначеності моделювання не уникнути. Розробка надійної схеми FDI повинна брати до уваги невизначеність моделювання щодо чутливості недоліків [17]. Метод FDI на основі моделі вимагає високоточного математичного опису системи, що контролюється. Чим краще модель представляє динамічну поведінку системи, тим кращою буде точність FDI. Якщо можна розробити такий метод, який нечутливий до невизначеності моделювання, точна модель не обов'язково потрібна. Усі невизначеності можна підсумувати як збурення, що діють на систему. Хоча вектор збурення невідомий, його матрицю розподілу можна отримати за допомогою процедури ідентифікації. Згідно з цим припущенням, принцип розв'язки завад може бути використаний для розробки схеми виявлення несправності з використанням невідомого вхідного спостерігача (Unknown Input Observer, UIO). Якщо зробити припущення, що система може бути описана як модель рівнянь помилки, то стає можливим отримання матриці розподілу збурень з даних безвідмовної системи, враховуючи член помилки. UIO, який забезпечує розв'язання збурень, може бути розроблений на основі моделі вимірювання помилки. У подальшому, фактично, припускається, що контрольована система, зображена на рис. 2, може бути описана моделлю похибки лінійного рівняння з дискретним часом типу [18]:

$$y_i^*(t) = \sum_{k=1}^n \alpha_{ik} y_i^*(t-k) + \sum_{j=1}^r \sum_{k=1}^n \beta_{ik} u_j^*(t-k) + \varepsilon_i(t) \quad (3.1)$$

де $y_i^*(t)$ ($i = 1, \dots, m$) - i -компонента вихідного вектора системи $y^*(t)$;

u_j^* - j -компонента вхідного вектора керування $u^* \in R^r$

n, α_{ik} та β_{ikj} є параметрами, які мають бути визначені методом ідентифікації.

Термін $\varepsilon_i(t)$ враховує похибку моделювання, що зумовлена шумами процесу, змінами параметрів тощо.

Використовуючи опис передавальної функції, систему (3.1) можна переписати у вигляді:

$$y_i^*(t) = F_i(z)u^*(t) + G_i(z)\varepsilon_i(t)$$

ї її структура зображена на рис. 3, де z є оператором унітарного випередження.

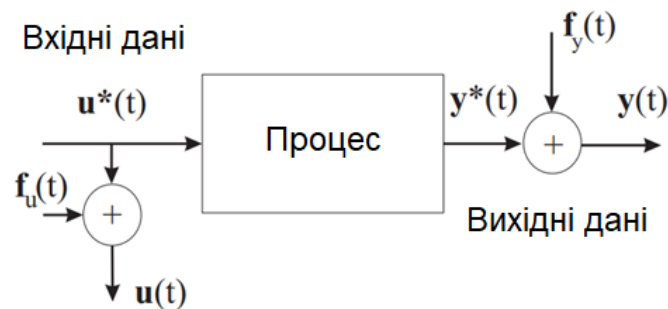


Рисунок – 3.20. Структура системи моніторингу технологічного процесу

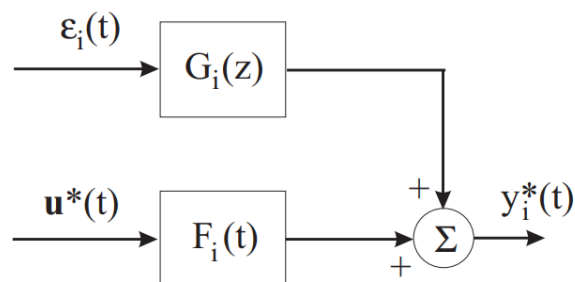


Рисунок – 3.21. Структура моделі похибки

Для моделювання системи ідентифікації несправностей FDI створено модель системи парової турбіни для технологічного процесу генерації електроенергії потужність 120 МВт.

Система складається з двовальної промислової газової турбіни, що працює паралельно з електромережею.

Блок-схема системи показана на рис. 3.22.

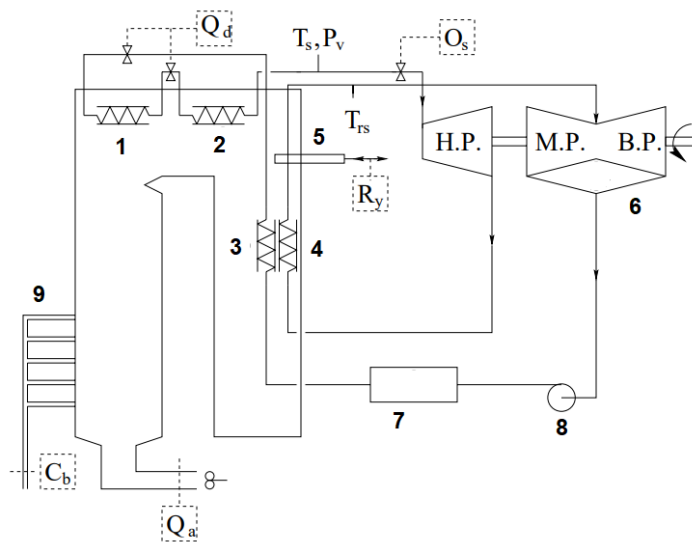


Рисунок – 3.22. Структура системи енергогенерації: 1- нагрівач великої потужності (випромінювання); 2 - нагрівач великої потужності (конвекція); 3, 4 – теплообмінники; 5 – заслонки; 6 – конденсатор; 7 - барабан, 8 - водяний насос; 9 - пальник.

Для симуляції системи застосували метод спостерігачів виходу, для FDI прототипу промислової газової турбіни і технологічному процесі генерації електроенергії.

Прототип Simulink, зображений на рис. 3.23, можна описати схемою замкнутого циклу, представленої на рис. 3.24, в якій змінні f_u , f_s , f_c і f_y , це види несправностей, які можуть виникнути в реальному технологічному процесі.

Вони представляють мультиплікативні несправності виконавчого механізму, системи, компонента контролера та вихідного датчика, відповідно. Зокрема, їх можна змоделювати як функції лінійного висхідного сигналу.

Розглянута проблема стосується виявлення та ізоляції несправностей на основі знання вимірної $y_i(t)$ та оціненої послідовності $\hat{y}_i(t)$ щодо вихідної потужності i -ї турбіни.

Структура пристрою виявлення несправностей зображена на рис. 3.24.

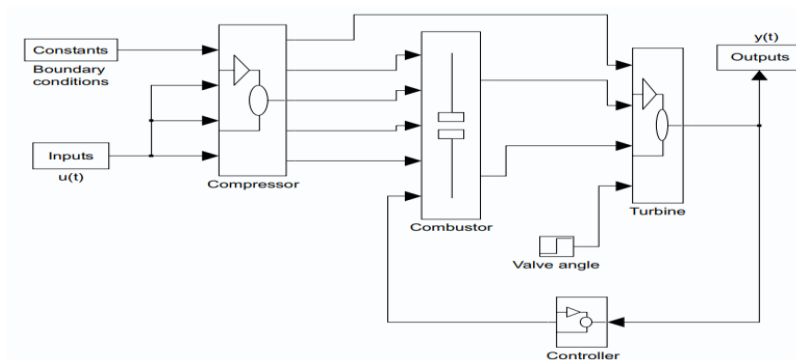


Рисунок – 3.23. Прототип процесу генерації в пакеті MATLAB/Simulink

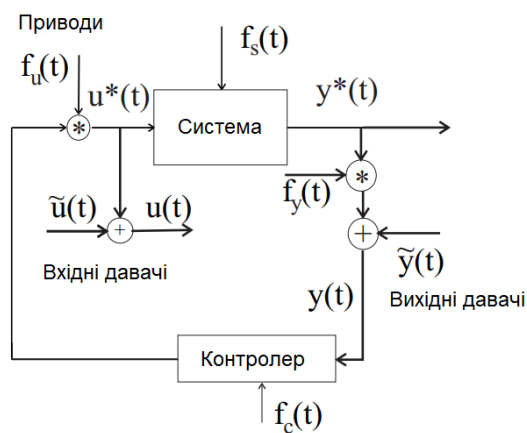


Рисунок – 3.24. Схема замкнутого циклу турбіни.

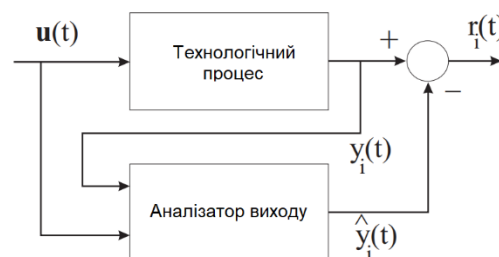


Рисунок – 3.25. Логічна схема генератора залишку.

Динаміка входу показана на рис. 3.26.

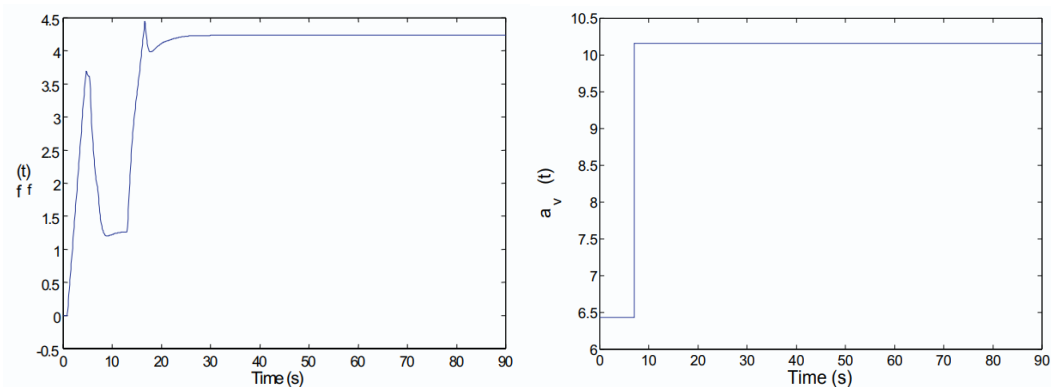


Рисунок – 3.26. Динаміка вхідних параметрів системи.

Було проведено симуляцію для чотирьох несправностей, а саме:

1. Забрудненість компресора, $f_c(t)$;
2. Несправність датчика термопари, $f_y(t)$;
3. Пошкодження ущільнення турбіни, $f_s(t)$;
4. Пошкодження приводу, $f_u(t)$.

Ці мультиплікативні несправності, які мають розвиваються повільний розвиток, були змодельовані за допомогою лінійних функцій.

Випадок несправності 1 являє собою забруднення поверхонь лопаток компресора, що зменшує потік повітря, змінюючи їх аеродинаміку і, отже, змінюючи шорсткість поверхні.

Сигнал, що відповідає витраті повітря компресора, зображено на рис. 3.27а, а несправність сигналу – на рис. 3.27б.

Виявлення несправності $f_s(t)$ щодо компресора було виконано за допомогою конфігурації вихідного спостерігача. Вхід $u(t)$ і вихід $y_i(t)$ подають спостерігачу для оцінки самого сигналу $\hat{y}_i(t)$. Полюси вихідного спостерігача для сигналу були обрані близько 0:4. На рисунку 10 показані безвідмовний (суцільна лінія) і несправний залишок (пунктирна лінія), створений першим вихідним спостерігачем.

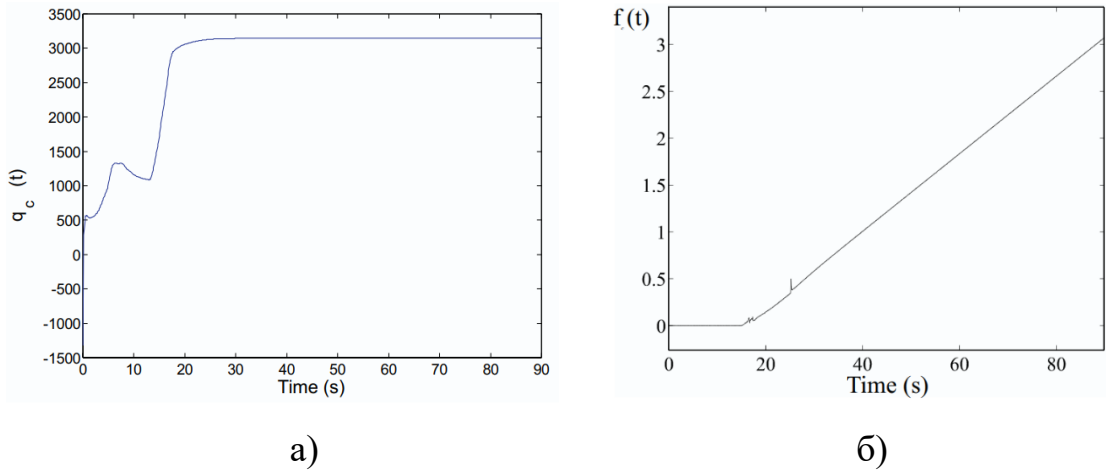


Рисунок – 3.27. Графіки роботи та несправності компресора: а) витрата повітря компресора; б) динаміка несправності компресора.

Графіки вхідних процесів, сигналу несправності та залишків для решти видів несправностей приведені на рис. 3.29.

При перевищенні функції залишку заданого порогу приймається рішення про наявність несправності системи. Виявлення несправності відбувається із затримкою.

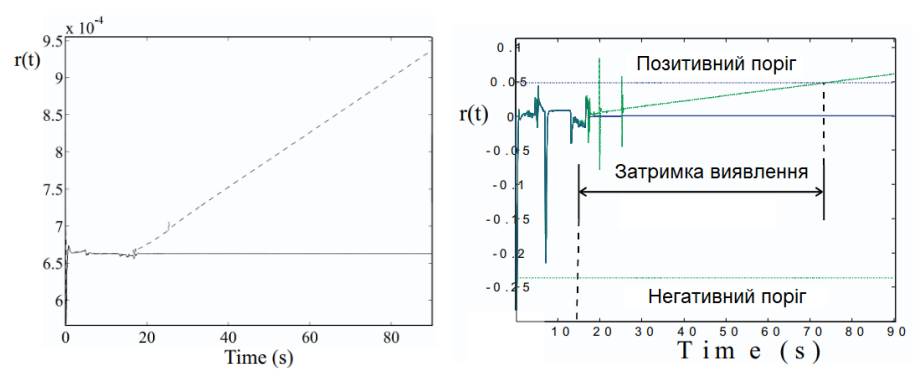
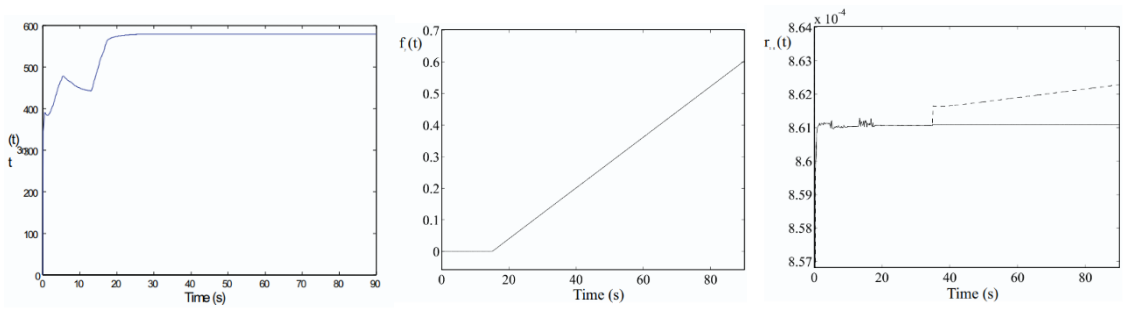
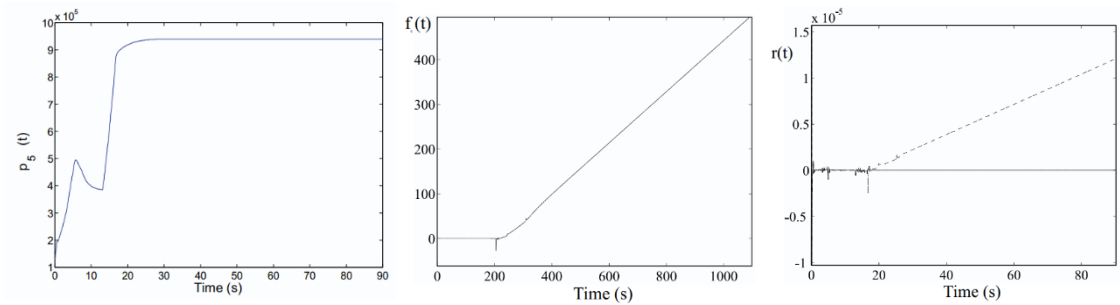


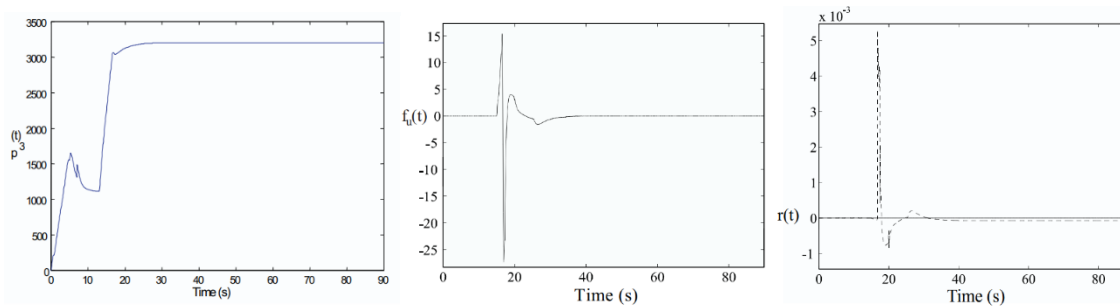
Рисунок – 3.28. Залишок несправності компресора та виявлення затримки при несправності.



а)



б)



в)

Рисунок – 3.29. Графіки нормальної роботи, несправності та залишку отримані при моделюванні роботи компонентів системи: а) несправність давача термопари, $f_y(t)$; б) пошкодження ущільнення турбіни, $f_s(t)$; в) пошкодження приводу, $f_u(t)$.

Всі отримані значення затримок при виявленні несправності приведено в таблиці 3.1.

Таблиця 3.1 підсумовує продуктивність методу FDI як у детерміністичному, так і в стохастичному середовищі. Таблиця містить мінімальні значення часу виявлення несправності на чотирьох виходах вимірювання, якщо залишкові або інноваційні значення контролюються за допомогою геометричного тесту та фіксованих порогів (рис. 3.30).

Мінімальні значення несправності, у таблиці 1 виражені у відсотках від контрольованого сигналу і відносяться до випадку, коли появу несправності необхідно виявити якомога швидше.

Несправності, змодельовані за допомогою лінійних функцій, можуть не негайно виявлені, оскільки затримка відповідного тривоги зазвичай залежить від режиму несправності.

Таблиця 3.1. Значення затримок при визначенні несправності на основі залишків

Вид несправності	Вимірювання без шуму	Вимірювання з наявністю шуму	Затримка виявлення
Забрудненість компресора, $f_c(t)$	1,0%	1,5%	35с
Несправність датчиків термодатчиків, $f_y(t)$	11%	12,5%	35с
Пошкодження ущільнення турбіни, $f_s(t)$	6%	8%	65с
Пошкодження приводу, $f_u(t)$	1,0%	3,5%	20с

Отримані результати виявлення несправностей на різних датчиках є достатніми для промислових діагностичних застосувань, враховуючи також, що мінімальна кількість виявлених несправностей може бути зменшена, якщо застосувати додаткові алгоритми аналізу залишків.

Висновки до розділу 3

Розроблено діагностичний алгоритм для проведення ієрархічної діагностики технологічних процесів. Він складається з двох частин: локального алгоритму та глобального. Перший проводить процедуру діагностики на певному концептуальному рівні, другий – між рівнями.

Також описано процедури виявлення несправностей, локалізації проблем та верифікації діагнозів.

Наведено приклад реалізації ієрархічного моделювання для системи контролю ємностей.

4. РОЗРОБКА СТАРТАП ПРОЄКТУ «РОЗРОБЛЕННЯ МЕТОДІВ ТА АЛГОРИТМІВ АВТОМАТИЗОВАНОГО КОНТРОЛЮ ТА ДІАГНОСТУВАННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ»

4.1 Опис та технологічний аудит ідеї стартап-проекту

На основі розглянутого вище матеріалу в цьому розділі буде проведено аналіз стартап проекту системи діагностування технологічних процесів.

Дана система призначена для аналізу параметрів технологічного процесу. Система дозволить скоротити витрати часу на ресурсів на процес пошуку несправностей, що дозволить скоротити ремонтний простій виробничих ліній.

Зміст ідеї, що пропонується, можливі напрямки її застосування та основні вигоди, що може отримати користувач за кожним напрямком застосування наведені в таблиці 4.1 [21].

Таблиця 4.1. Опис ідеї стартап проекту.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблення методів та алгоритмів автоматизованого контролю та діагностування технологічних процесів	Система діагностування технологічного процесу з ефектом «Фокусу»	Можливість планувати ремонтні періоди заміни обладнання
	Верифікація діагностичних процедур на точність	Зменшення помилок при проведенні діагностування, вища точність діагностування
	Ієрархічні процедури діагностування на різних рівнях ведення технологічного процесу	Економія часу на проведення діагностування всієї системи та окремих елементів

Використовуючи розроблену систему можна досягти економії загальних витрат підприємства, підвищити час роботи обладнання, збільшити міжремонтний час роботи обладнання технологічного процесу.

На наступному етапі проводимо порівняльний аналіз переваг стартапу порівняно з пропозиціями конкурентів. Такий аналіз наведено в таблиці 4.2.

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Siemens Simatic	OpenScada			
1.	Точність діагностичної процедури	Висока	Низька	Середня	-	-	+
2.	Швидкість впровадження	Низька	Висока	Висока	+	-	-
3.	Складність керування	Висока	Висока	Висока	-	+	-
4.	Економічність	Висока	Середня	Середня	-	-	+
5.	Швидкість діагностичної процедури	Висока	Середня	Середня	-	-	+

Спираючись на порівняльний аналіз техніко-економічних характеристик ідеї проекту, дослідженого у таблиці 4.2, можна зробити висновок, що розроблений проект є більш точним, швидким та економічнішим, ніж аналоги конкурентів.

Далі аналізуємо технічні особливості, спектр можливих технологічних рішень. Для аналізу здійсненності проекту використаємо таблицю 4.3.

Використання діагностичних процедур не є інноваційним, але для актуальності на ринку необхідно використовувати максимально ефективні рішення, розробляти нові кращі алгоритми.

Технологічна реалізація проекту можлива, так як всі необхідні технології є наявними на ринку та знаходяться у широкому доступі.

Таблиця 4.3 Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Автоматична діагностика обладнання технологічного процесу	Діагностичні алгоритми	+	+
		Ієрархічні процедури	+	+
		Підтримка інтелектуальної логіки	+	+
		Система фокусування	-	-
		Багатошарова архітектура	+	+
Обрана технологія реалізації ідеї проекту: Система на діагностувальних алгоритмів з використанням багатошарової архітектури та функцій фокусування				

4.2 Аналіз ринкових можливостей запуску стартап проекту

Визначення основних ризиків та можливостей, які можна очікувати від ринку є необхідним, оскільки дозволяє спроектувати напрями розвитку стартапу із врахуванням всіх складових ринкового положення, попиту цільових клієнтів та пропозицій конкурентів. Доцільно використати ці знання під час ринкового впровадження проекту.

Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4) [22].

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	300 000
3	Динаміка ринку (якісна оцінка)	Повільно зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Вузька галузь, репутація продукту
5	Специфічні вимоги до стандартизації та сертифікації	Вимоги щодо безпеки та надійності системи
6	Середня норма рентабельності в галузі (або по ринку), %	20%

Попередній аналіз вказує на гарну рентабельність проекту та позитивну динаміку зростання ринку вцілому, що є привабливим для входу. Серед наявних обмежень можна виділити високу конкурентність.

Визначимо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи. Результати приведені в таблиці 4.5.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до послуги
1	Автоматизація процесу діагностування технологічного процесу	Виробництва та підприємства машинобудівного та приладобудівного сегменту	Швидкість роботи системи, точність роботи системи, рентабельність	Надійна робота системи, можливість якісного планування

Дослідивши цільову аудиторію на ринку визначено, що основними клієнтами є виробництва та підприємства машинобудівного та приладобудівного сегменту.

Відмінностями у поведінці різних потенційних цільових груп клієнтів є загальна якість діагностування, швидкість роботи системи, точність діагностичних процедур та рентабельність.

Інформація про вимоги майбутніх покупців до товару буде враховано в подальшій розробці стартап-проекту.

Після визначення потенційної цільової аудиторії продукту проведемо аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та фактори, що йому перешкоджають. Дані наведені у таблиці 4.6

В заповненій таблиці наведені фактори, які можуть в значній мірі нашкودити стартап проекту. Як протидію цим факторам запропоновані контрдії які мають покращити становище проекту в складних ситуаціях.

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Висока вартість продукції	Високі затрати на розробку та впровадження	Зменшення ціни за рахунок використання масштабованих та ієрархічних алгоритмів
2.	Висока конкуренція	Клієнти звертаються до інших замовників	Виділяти сильні сторони продукту та просувати це при маркетинговій стратегії
3.	Нові технології на ринку	Поява більш інноваційних технологій	Розробка з урахуванням появи нових технологій
4.	Недостатній попит товару на ринку продукції	Відсутність нових замовлень	Вкладати більше коштів в рекламу товару, висвітлювати переваги
5.	Швидкість виконання замовлення	Незадовільняюча швидкість установки системи у клієнтів	Залучення працівників з високою кваліфікацією

Проаналізувавши фактори загроз, які впливають на продукт, можна визначити такі основні негативні фактори: висока конкуренція, висока вартість систем такого класу, нові технології на ринку, недостатній попит товару, швидкість виконання замовлення. Для успішного ведення стартапу необхідно постійно проводити аналіз конкурентів, слідкувати за трендами та технологіями у сфері та залучати кращих спеціалістів галузі.

Слід зазначити, що окрім загроз, що можуть перешкоджати розвитку продукту, існують і фактори можливостей, що відображають переваги, на яких буде проводитись впровадження проекту. Фактори можливостей проекту зображені у таблиці 4.7.

В таблиці наведено фактори, які дозволяють вивести продукт на ринок та підвищити ефективність його впровадження.

При появі на ринку важливим є розуміння своїх конкурентів, їх можливостей та швидкості адаптації. Також важливо враховувати їх можливий вплив на ринок.

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Вдосконалення	Модернізація функціонування системи	Розвиток та залучення додаткових коштів
2.	Вихід на закордонні ринки	Ріст можливих сфер застосування та продажу	Аналіз закордонних ринків
3.	Збільшення виробництв у галузі приладобудування та машинобудування	Різке збільшення попиту на засоби діагностування	Збільшення потужностей виробництва для покриття попиту
4	Тренд на автоматизацію виробництв	Бажання автоматизації виробничих процесів	Пришвидшення темпу розроблення проєкту, впровадження більш адаптивних систем

Ця інформація заноситься в таблицю 4.8

Проаналізувавши дані таблиці 4.8 можна зробити висновок, що на ринку присутня олігополія, так як ринок не перенасичений компаніями зі схожою продукцією. Рівень конкурентної боротьби – національний з внутрішньогалузевою ознакою. Конкуренція за видами є товарно видовою.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції: монополістична	На ринку існує певна кількість фірм, що випускають схожі системи	Створення високо технологічної системи
За рівнем конкурентної боротьби: національний	Конкуренція компаній на міжнародному рівні	Аналіз аудиторії та їх потреб з врахуванням національних особливостей
За галузевою ознакою: міжгалузева	Виготовлення різного роду продукції	Орієнтування на якість та ціну
За характером конкурентних переваг: нецінова	Конкурентні переваги не залежать від вартості продукції	Увага до якості системи
За інтенсивністю: марочна	Бренд має значний вплив	Надання функцій, які не надають конкуренти

Після аналізу конкуренції нижче наведений більш детальний аналіз умов конкуренції в галузі.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	OpenScada Siemens Simatic	Компанії, що займаються виробництвом систем діагностування	Вартість на обладнання, кількість наявного обладнання	Промислові виробництва	Відсутні
Висновки:	Середня інтенсивність конкурентів	Присутні можливості для входу на ринок	Диктують вартість системи діагностування	Прагнуть мати якісну продукцію, надійність та швидкість	Відсутні

Аналіз конкуренції в галузі за М.Портером показав прямих та потенційних конкурентів у галузі, можливості входу на ринок та інтенсивність конкурентів, клієнтів та їх основні вимоги до продукції, що виготовляється, товари замітники є відсутніми.

Після всіх аналізів визначається та обґрунтовується перелік факторів конкурентоспроможності.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Точність діагностування	Наявність процедур верифікації отриманих діагностичних даних
2	Час проведення процедури	Наявність функції фокусування на несправності
3	Технічна підтримка	Допомога спеціалістів на всіх етапах впровадження системи
4	Якість діагностичного аналізу	Чітке визначення несправних елементів та ведення статистики
5.	Якість впровадження	Наведені чіткі інструкції та алгоритми керування системою

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін «системи керування мікрокліматичними параметрами у виробничих приміщеннях»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розробкою						
			-3	-2	-1	0	+1	+2	+3
1	Технологічність	16					+		
2	Ефективність	16							+
3	Висока якість	17							+
4	Технічна підтримка	18							+
5.	Ціна	15		+					

Дослідивши слабкі і сильні сторони проекту, зроблено висновок, що розроблена система є конкурентоспроможною завдяки ефективності, високій якості та технічній підтримці, при цьому програє у вартості.

Проведено SWOT-аналіз, що представлений у таблиці 4.12, на основі якого ми можемо стверджувати про сильні та слабкі сторони продукту та можливості і загрози. На базі даного аналізу виконуються альтернативи ринкової поведінки та приблизний час реалізації з врахуванням конкурентів.

Таблиця 4.12. SWOT- аналіз стартап-проекту

Сильні сторони: Надійність процедур, якість діагностування, швидкість діагностичних алгоритмів.	Слабкі сторони: Висока складність застосування, вартість диктується конкурентами
Можливості: 1. Удосконалення 2. Вихід на міжнародний ринок. 3. Збільшення виробництв у галузі приладобудування та машинобудування. 4. Підтримка держави. 5. Тренд на процедури діагностування.	Загрози: 1. Висока вартість продукції. 2. Нові технології на ринку. 3. Висока конкуренція. 4. Недостатній попит товару на ринку продукції. 5. Якість виконання замовлення.

Визначені альтернативи ринкового впровадження стартап-проекту показано у таблиці 4.13.

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Стратегія орієнтування продукту на нішу державних підприємств великих міст	Середня ймовірність	9-14 місяців
2	Стратегія орієнтування продукту на нішу приватних підприємств великих міст	Висока ймовірність	9-14 місяців
3	Стратегія орієнтування продукту на нішу приватних підприємств невеликих міст	Низька ймовірність	6-8 місяців
4	Стратегія орієнтування продукту на нішу державних підприємств невеликих міст	Низька ймовірність	9-14 місяців

Проаналізувавши таблицю 4.13 можна прийти до висновку, що доцільніше використовувати стратегію орієнтування продукту на нішу державних та приватних підприємств великих міст через високу ймовірність отримання ресурсів не дивлячись на великі строки реалізації в порівнянні з іншими альтернативами.

4.3. Розроблення ринкової та маркетингової програми стратегії проекту

У даному підрозділі проводиться розробка ринкової стратегії, що в першу чергу передбачає визначення стратегії охоплення ринку, а саме опис цільових груп споживачів. Отриманий результат занесено у таблицю 4.14.

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Мережеві підприємства	Так	Високий	Середня	Середня
	Локальні підприємства	Так	Середній	Середня	Складна
Які цільові групи обрано: Під час аналізу потенційних груп споживачів було прийнято рішення що компанія буде працювати із мережевими підприємствами та локальними виробництвами.					

Цільовими групами продукту будуть мережеві підприємства та локальні виробництва, таким чином обрано використовувати диференційну маркетингову стратегію, так як комунікація буде відбуватись одночасно з двома сегментами ринку, але для кожного буде розроблятися окрема програма ринкового впливу.

Для роботи в обраному сегменті ринку необхідно сформувати базову стратегію розвитку. Отриманий результат представлено в таблиці 4.15.

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Орієнтація продукту на ринок приватних та державних промислових компаній великих міст, мережевих підприємств
Стратегія охоплення ринку	Диференційований маркетинг - це стратегія, коли компанія розробляє окремі маркетингові кампанії для різних сегментів ринку, враховуючи особливості та потреби кожного з них.
Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Технологічність, точність, швидкість, якість
Базова стратегія розвитку	Стратегія диференціації - полягає у створенні унікальної ціннісної пропозиції для споживачів, яка відрізняє продукт або послугу компанії від конкурентів.

Базовою стратегією розвитку обрано стратегію диференціації, через наявність на ринку конкурентів, котрі займаються сегментуванням ринку, ключовими конкурентоспроможними позиціями є технологічність, точність та ефективність продукції.

Далі визначимо базову стратегію конкурентної поведінки, що зазначена у таблиці 4.16.

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Компанія буде шукати нових споживачів	Ні	Заняття конкурентної ніші

Обрано стратегію заняття конкурентної ніші, так як продукт не є «першопрохідцем» у галузі систем діагностування технологічних процесів, компанія буде шукати нових споживачів та відмовляється від копіювання характеристик товару конкурентів.

На основі побажань покупців обраної ніші до постачальника і продукту, а також в залежності від стратегії розвитку та стратегії конкурентної поведінки розробляємо стратегію позиціонування яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проект. Результати наведені у таблиці 4.17.

Таблиця 4.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Точність діагностики, швидкодія, аналіз архівних даних	Стратегія диференціації	Якість продукції, широкий функціонал, швидкодія	Швидкість, ефективність, точність

У результаті проведеного аналізу у даному підрозділі визначена система рішень, щодо ринкової поведінки стартапу, вона визначає вектор напрямку роботи компанії на ринку та ключову конкурентоспроможність.

Під час створення маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач. У таблиці 4.18 наведені результати ключових переваг.

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
Точність	Наявність функцій фокусування	Точність забезпечується використанням функцій фокусування
Швидкість роботи	Застосування ієрархічних процедур	Поділ процесу на рівні, швидкий аналіз рівня з подальшим фокусуванням
Якість	Сучасні алгоритми верифікації результатів	Використання сучасних методів та технологій для верифікацій процедур діагностики

У таблиці 4.19 було описано три рівні моделі товару, а саме товар за задумом, товар у реальному виконання, товар із підкріпленням.

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Автоматизована система діагностування технологічного процесу		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Економічна	М	Е
	2. Технологічна	Нм	Тл
	3. Швидкість роботи	Нм	Тх
	4. Призначення	Нм	Ор
	5. Технічна підтримка	Нм	Тх
	Якість: стандарти, нормативи, технічний регламент		
Пакування: Мікроконтролер з набором датчиків, програмне забезпечення, герметична упаковка			
Марка: Назва бренду та назва установки			
III. Товар із підкріпленням	До продажу с: Консультація з можливості інтегрування		
	Після продажу: Підтримка товару		

Отже під час визначення ключових переваг концепції потенційного товару було визначено такі ключові потреби: точність, швидкість роботи та технологічність. Дані потреби задовольняються якістю обладнання, що використовується, невеликим часом перехідного процесу регулювання та завдяки впровадженню сучасних методів та технологій контролю мукроклімату.

Проаналізувавши рівні моделі товару, можна дійти до висновку, що продукт у реальному виконанні в достатній мірі відповідає товару за задумом, який був спроектований ще на стадії ідеї стартап-проекту, однак було внесено певні незначні корективи для того, щоб товар був краще адаптований до умов ринку.

Після аналізу рівнів моделей товару, проведено визначення встановлення меж ціни, які необхідно враховувати для встановлення адекватної цінової політики на товар. Проведено аналіз цін товарів конкурентів та доходів користувачів продукту і його результати занесено до таблиці 4.20

Таблиця 4.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	300 000 грн	400 000 грн	Високий	300 000 – 400 000 грн

Проаналізувавши ціни, визначено такі межі встановлення цін на власний продукт, щоб він міг конкурувати з товарами-аналогам та товарами-замінниками, при цьому бути доступними для групи споживачів.

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (див. табл. 4.21).

Таблиця 4.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Підприємства виконують закупівлю системи діагностування	Своєчасне виконання запланованих робіт та поставок, технічна підтримка системи	Ринок України та міжнародні поставки	Торгівля через спеціалізовані магазини та партнерів

Проведено формування системи збуту де розглянуто специфіку закупівельної поведінки цільових клієнтів, функції збуту постачальника, глибина каналу збуту та оптимальна система збуду, що заключається в торгівлі через спеціалізовані магазини та партнерів.

Далі проводиться аналіз концепції маркетингових комунікацій, що дозволить збільшити загальні продажі товару та прибуток компанії шляхом налаштування каналів збуту продукції.

Таблиця 4.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Пошук якісної продукції з високою точністю	Тематичні виставки присвячені системам діагностування, таргетована реклама в інтернеті	Вигідна ціна на систему, дотримання високої якості та швидкодії	Показати сильні сторони продукту	Демонстрація можливостей

Провівши необхідний аналіз та дослідження, створено концепцію маркетингових комунікацій, яка виражається в активній рекламі на тематичних виставках та таргетованій рекламі в інтернет ресурсах, завданням якої є демонстрація сильних сторін продукту компанії.

Бізнес модель дозволить надати цінність нашим клієнтам, залучити їх та забезпечити дохідні потоки через продаж представленої системи для діагностування технологічних процесів.

Наш стартап пропонує систему діагностування технологічних процесів з використанням інтелектуальних діагностичних ієрархічних процедур з функцією фокусування. Передбачається співпраця з науковцями та постачальниками датчиків, щоб постійно вдосконалювати точність та швидкодію роботи системи, яка аналізує параметри роботи обладнання технологічних процесів. Основні активності включають розробку програмного забезпечення. Перевагами цієї системи є висока точність, швидкодія та можливість аналізу тривалості роботи обладнання.

Цільова аудиторія - виробництва та підприємства приладобудівного та машинобудівного сегменту. Прибуток буде отримано за рахунок продажу розробленої технології, яка забезпечує точне, швидке та ефективне діагностування виробничого обладнання технологічних процесів. Просування продукту планується через активну рекламу на спеціалізованих виставках та цілеспрямовану рекламу через партнерів, з фокусом на демонстрації переваг продукту.

4.4 Економічна ефективність стартап проєкту

При аналізі ефективності витрачених коштів на реалізацію стартап проєкту необхідно оцінити показники його ефективності. Для інноваційних проєктів необхідно провести аналіз доходів та витрат при реалізації проєкту. Вони мають як постійний так і змінний характер.

Отримані прибутки при реалізації ідеї ми отримуємо від продажу мікроконтролерних систем керування, продажу програмного забезпечення; підтримки систем.

До постійних витрат відносяться ті витрати, загальна сума яких за певний час не залежить від кількості виготовленої продукції.

- Оренда приміщень.
- Реклама
- Зарплата постійних працівників компанії (інженери, маркетологи, менеджери)

Змінні витрати представляють собою витрати, загальна сума яких за певний час залежить від обсягу виготовленої продукції.

- Закупка обладнання (мікроконтролери та давачі) необхідних для виготовлення комплексного обладнання;
- Витрати на електроенергію;
- Зарплата робочих;

При отриманні необхідного фінансування ми будемо мати наступну фінансову ситуацію, при розробці системи:

Стаття постійних витрат складає:

- Для оренди приміщень - 10 000 грн/міс.
- Для рекламної кампанії - 10 000 грн/міс.
- зарплата постійних працівників - 70 000 грн/міс.

Змінні витрати на виробництво одного комплексного обладнання:

- зарплата робочих - 25 000 грн.;
- додаткове обладнання, датчики - 25 000 грн.

Можливий отриманий прибуток складає:

- відпускна вартість системи - 300 000 грн. - + пдв - 60 000 грн.

Обсяги виробництва продукції на перші 5 місяців наведено в таблиці 4.23.

Таблиця 4.23 – Обсяги виробництва продукції

Показник	Значення по місяцях, тис. грн.				
	1	2	3	4	5
Загальна потреба в продукції, шт.	2	4	6	8	10
Ціна одного набору тис. грн.	300	300	300	300	300
Річні обсяги випуску в вартісних показниках (тис. грн.)	600	1200	1800	2400	3000

Визначимо витрати, необхідні для реалізації поточної діяльності за проектом (таблиця 4.24).

Таблиця 4.24 – Виробничі витрати

№ з/п	Стаття витрат	Сукупні витрати за період місяців, тис. грн.				
		1	2	3	4	5
1	Загальногосподарські витрати, оплата праці постійного персоналу	10	20	30	40	50
1.1	Витрати на оренду та утримання приміщень, обладнання	10	20	30	40	50
1.2	Витрати на збут, просування та рекламу	10	20	30	40	50
2	Витрати на матеріальні ресурси (комплектуючі, сировина)	25	50	75	100	125
3	Витрати на оплату праці	25	50	75	100	125
Разом:		80	160	240	320	400

Точка беззбитковості відображає обсяг виробництва інноваційної продукції, при досягненні якого виручка від реалізації покриває сумарні витрати на її виробництво. Розрахунок точки беззбитковості року проводиться за формулою:

$$T_6 = \frac{C}{P-V} \quad (4.1)$$

де C – постійні витрати на весь обсяг продукції (ті, які не залежать від обсягу виробництва продукції – загальногосподарські витрати та витрати на оплату праці); P – ціна одиниці продукції; V – змінні витрати на одиницю продукції.

$$T_6 = \frac{C}{P-V} = \frac{90000}{300000-240000} = 1,5 \quad (4.2)$$

Визначимо формування грошового потоку від реалізації проекту.

Чистий дисконтований дохід (NPV, Net Present Value) – це різниця між надходженнями за весь період інноваційного проекту та інвестиціями в проект. Розрахунок чистого дисконтованого доходу наведений в таблиці 4.25.

Таблиця 4.25 – Формування грошового потоку від реалізації проекту

№	Показник	Значення по місяцях					Разом
		1	2	3	4	5	
1	Надходження від проекту (виручка від реалізації продукції, послуг)	600	1200	1800	2400	3000	9000
2	Загальні витрати	80	160	240	320	400	1200
3	Грошовий потік	520	1040	1560	2080	2600	7800
4	Акумуляований грошовий потік	520	1040	1560	2080	2600	7800

Оскільки $NPV > 0$ протягом розрахункового періоду, то даний інноваційний проект доцільно прийняти.

Індекс рентабельності інвестицій в проект (ROI, Return On Investment) характеризує рівень грошового потоку, що припадає на одиницю інноваційних витрат і обчислюється за формулою (4.3), коли інвестиція здійснюється багато разів:

$$ROI = \frac{\sum_T D_T}{\sum_T I_T} \quad (4.3)$$

де D_T – надходження у відповідному періоді;

I_T – інвестиція у відповідному періоді.

Індекс рентабельності за 5 місяців:

$$ROI = \frac{9000}{1200} = 7.5 \quad (4.4)$$

Показник $ROI > 1$, отже даний інноваційний проект доцільно прийняти. Даний критерій ROI використовують при виборі певного проекту із декількох альтернативних, у яких NPV приблизно однакові.

Період окупності інвестицій (Ток) – це розрахунковий термін від початку реалізації проекту, починаючи з якої акумуляований грошовий потік (ACF) приймає стійке позитивне значення. Іншими словами, це – період (вимірюваний в місяцях, кварталах або роках), починаючи з якого первинні вкладення та інші витрати, пов'язані з інвестиційним проектом, покриваються сумарними результатами його здійснення.

Термін окупності розраховується за формулою :

$$T = t + \frac{|ACF_{t-}|}{(|ACF_{t-}| + |ACF_{t+}|)} = 1 + \frac{0}{0+1} = 1 \quad (4.5)$$

де t – останній період реалізації проекту, при якому акумульований грошовий потік (різниця накопиченого доходу і витрат) приймає від’ємне значення;

ACF_{t-} – це остання від’ємна різниця накопиченого доходу та витрат (грн.);

ACF_{t+} – це перша позитивна різниця накопиченого доходу та витрат (грн.).

Отже, на 1-му місяці після початку реалізації проект повністю окупить себе.

Отже, враховуючи розрахунки можна стверджувати, що проект є ефективним для реалізації і може принести прибуток.

Висновки до розділу 4

Даний розділ магістерської дисертації присвячений розробці стартап-проекту «Автоматизована система контролю та діагностування технологічних процесів». Висвітлено загальну ідею проекту, система призначена контролю та діагностування технологічних процесів з метою оптимізації ремонтних робіт та заміни обладнання. Система дозволить виконати економію витрат часу на діагностування та покращити стабільність ведення технологічного процесу.

Проведено аналіз потенційних клієнтів стартапу, дослідження показало що основними клієнтами являються підприємства приладобудівного та машинобудівного сегменту. Після визначення цільової аудиторії виконано аналіз ринкового середовища, що включає фактори загроз та можливостей.

Аналіз конкуренції в галузі за М.Портером показав, що потенційними конкурентами є компанії, що займаються виробництвом систем діагностування, автоматизованих систем, можливості для входу на ринок присутні. Дослідивши слабкі і сильні сторони проекту, зроблено висновок, що розроблена система є конкурентоспроможною завдяки ефективності, високій якості, швидкості та технічній підтримці, при цьому програє у вартості.

У підсумку можна зазначити, що ринкова комерціалізація проекту можлива виходячи з наведеного вище аналізу. Стримка динаміка росту ринку створює нові можливості для впровадження та розвитку спроектованої системи. Не дивлячись на рівень конкуренції існує перспектива входу та закріплення на ринку, оскільки є можливість запропонувати якісні переваги продукту групі потенційних клієнтів.

ВИСНОВКИ

Проведено аналіз методів та факторів, які використовуються в системах діагностування та засновані на знаннях. Оглянуто діагностичні методики, їх переваги та недоліки. Розглянуто системи міркування на основі випадків, ієрархічні діагностичні матриці, причинно-наслідкові графи. Описана причинно-наслідкова методика для систем діагностики.

Розроблено діагностичний алгоритм для проведення ієрархічної діагностики технологічних процесів. Він складається з двох частин: локального алгоритму та глобального. Перший проводить процедуру діагностики на певному концептуальному рівні, другий – між рівнями. Також описано процедури виявлення несправностей, локалізації проблем та верифікації діагнозів. Наведено приклад реалізації ієрархічного моделювання для системи контролю ємностей.

Розроблено діагностичний алгоритм для проведення ієрархічної діагностики технологічних процесів. Він складається з двох частин: локального алгоритму та глобального. Перший проводить процедуру діагностики на певному концептуальному рівні, другий – між рівнями. Також описано процедури виявлення несправностей, локалізації проблем та верифікації діагнозів. Наведено приклад реалізації ієрархічного моделювання для системи контролю ємностей.

Створено стартап проект атоматизованої системи діагностування технологічних процесів. Виствітлено загальну ідею проекту, визначено слабкі та сильні сторони. Проведено порівняльний аналіз ринку та визначено цільову ідею продукту, фактори загроз та можливостей, створено бізнес-модель.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Junjian Hou, Xikang Lu, Yudong Zhong, Wenbin He, Dengfeng Zhao, Fang Zhou. A comprehensive review of mechanical fault diagnosis methods based on convolutional neural network. *Journal of Vibroengineering*. 2023. Vol. 26, No. 1, p. 44–65. DOI <https://doi.org/10.21595/jve.2023.23391>.

[2] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri, Kewen Yin. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering*. Volume 27. Issue 3. 2003. p. 327-346.

[3] Hang Gao, Jiangmeng Li, Wenwen Qiang and other. Introducing Expertise Logic into Graph Representation Learning from A Causal Perspective. 2023. DOI: 10.48550/arXiv.2301.08496.

[4] Olav M. Underdal, Harry M. Gilbert. Hierarchy of diagnosis for advanced diagnostics equipment. Application US12/038,351. Patent Application Publication. 2008. URL : <https://patentimages.storage.googleapis.com/ff/30/4a/f467ae1d663343/US20090216493A1.pdf>. (дата звернення 12.08.2024).

[5] Ko'ścielny J. M. Diagnostics of industrial processes in decentralised structures. A Chapter in Korbicz et al. 2004. pp. 763–779.

[6] Antoni Ligeża, Jan Maciej Kościelny. A New Approach to Multiple Fault Diagnosis: A Combination of Diagnostic Matrices, Graphs, Algebraic and Rule-Based Models. The Case of Two-Layer Models. *International Journal of Applied Mathematics and Computer Science*. 2008. 18(4). P.465-476. DOI: 10.2478/v10006-008-0041-8.

[7] Koorosh Aslansefat, Sohag Kabir, Youcef Gheraibia, Yiannis Papadopoulos. Dynamic Fault Tree Analysis: State-of-the-Art in Modelling, Analysis and Tools. In book: *Reliability Management and Engineering: Challenges and Future Trends* Edition: 1 Chapter: 4 Publisher: CRC Press. 2020. DOI: 10.1201/9780429268922-4.

[8] Vincent Cabeli. Learning causal graphs from continuous or mixed datasets of biological or clinical interest. URL : https://theses.hal.science/tel-03693627/file/CABELI_Vincent_2021.pdf. (дата звернення 14.08.2024).

[9] Cordier M-O., Dague P., Dumas M., Lévy F., Montmain J., Staroswiecki M., Trave-Massuyes L. A comparative analysis of AI and control theory approaches to model-based diagnosis. In: ECAI 2000, 14th European Conference on Artificial Intelligence (W. Horn, Ed.). IOS Press. Berlin. 2000. p. 136–140.

[10] Kościelny J.M., Bartyś M., Rostek K. The comparison of fault distinguishability approaches – case study // Technical sciences. – 2019. – Vol. 67, No. 6. – P. 1059-1068. DOI: 10.24425/bpasts.2019.131566.

[11] Kościelny J.M., Syfert M., Rostek K., Szyber A. Fault isolability with different forms of faults-symptoms relation // Journal of Applied Mathematics and Computer Science. – 2016. – Vol. 26, No. 4. – P. 815–826.

[12] Blanke M., Staroswiecki M. Structural design of systems with safe behaviour under single and multiple faults // IFAC Symposium Safeprocess. – 2006. – P. 511–515.

[13] Isermann R. Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. – Springer Science & Business Media, 2006. – 475 p.

[14] Szyber A. Graph of a Process: A New Tool for Finding Model Structures in a Model-Based Diagnosis // IEEE Transactions on Systems, Man and Cybernetics: Systems. – 2015. – Vol. 45, No. 7. DOI: 10.1109/TSMC.2014.2384000.

[15] Syfert M., Kościelny J.M. Diagnostic reasoning based on symptom forming sequence // IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes. – 2009. – P. 89–94. <https://doi.org/10.3182/20090630-4-ES-2003.00015>.

[16] Trapani N., Longo L. Fault Detection and Diagnosis Methods for Sensors Systems: Scientific Literature Review // IFAC-PapersOnLine. – 2023. – Vol. 56, Issue 2. – P. 1253-1263. <https://doi.org/10.1016/j.ifacol.2023.10.1749>.

[17] Simani S., Fantuzzi C., Patton R.J. Model-based fault diagnosis in dynamic systems using identification techniques. – Springer London, 2013. – 282 p.

[18] Нестеренко А. Алгоритми автоматизованого контролю та діагностування технологічних процесів / Нестеренко А. // VII МСНТК, 25-26 квітня 2024 року. — Т. : ТНТУ, 2024. — С. 267–268.

[19] Нестеренко А. Автоматизоване діагностування технологічних процесів методами ієрархічного моделювання та діагностичних процедур // Контроль і управління в складних системах (КУСС-2024). ВНТУ. Вінниця. – 2024.

[20] Автоматизована система діагностування технологічних процесів / Нестеренко А.В., Філіппова М.В. // Збірник праць XX Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні». 04-05 грудня 2024 р. – К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2024.

[21] Гавриш, О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів. Конспект лекцій : навчальний посібник для студентів спеціальностей 151 – «Автоматизація та комп'ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірвальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,88 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 188 с.

[22] Гавриш, О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів: практикум: навчальний посібник для студентів спеціальностей 151 – «Автоматизація та комп'ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірвальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,11 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 116 с.