

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

**Басюк Наталія Василівна**

**Нечітка система діагностики комп'ютерної системи / Fuzzy system of the computer system diagnosing**

напрямок підготовки: 6.050102 - Комп'ютерна інженерія  
фахове спрямування - Комп'ютерні системи та мережі  
Бакалаврська робота

Виконав студент групи КСМ-41/1  
Н.В. Басюк

Науковий керівник:  
к.т.н., Л.О. Дубчак

Тернопіль –2018

## РЕЗЮМЕ

Дипломний проект містить 54 сторінки пояснюючої записки, 17 рисунки, 7 таблиць, 3 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Мета проекту полягає в аналізі роботи комп'ютерних систем, їх основних характеристик та розробці нечіткої системи аналізу їх стану.

Для досягнення поставленої мети необхідно розв'язати наступні задачі: проаналізувати всі можливі стани комп'ютерних систем; дослідити нечітку логіку та обрати підходящий алгоритм нечіткого виводу по базі знань; побудувати функції належності; сформуванати базу правил; дослідити роботу реалізованої нечіткої системи; зробити висновки щодо правильності роботи розробленої системи.

Побудовано функції належності вхідних та вихідного значень. Сформовано базу правил нечіткої системи. Проаналізовано роботу нечіткої системи за допомогою поверхонь значень нечіткої системи.

Розроблено структурну схему нечіткої системи, що дозволяє краще зрозуміти принципи роботи. Зроблено висновки щодо правильності роботи нечіткої системи. Використання розробленої нечіткої системи діагностики роботи комп'ютерних систем, дозволить зменшити шанси виникнення проблем шляхом їх попередження та вчасної діагностики, що дозволить значно зекономити час та ресурси.

Ключові слова: НЕЧІТКА ЛОГІКА, ФУНКЦІЯ НАЛЕЖНОСТІ, НЕЧІТКА СИСТЕМА ДІАГНОСТИКИ.

## RESUME

The diploma project contains 54 pages of explanatory note, 17 figures, 7 tables, 3 appendices. Volume of graphic material 2 sheets of A3 format.

The aim of the project is to analyze the operation of computer systems, their main characteristics and to develop a fuzzy system of analysis of their condition.

To achieve this goal it is necessary to solve the following tasks: to analyze all possible states of computer systems; explore fuzzy logic and choose an appropriate fuzzy inference algorithm based on knowledge; build membership functions; to form a base of rules; to investigate the work of the implemented fuzzy system; draw conclusions about the correctness of the developed system.

The membership functions of input and output values are constructed. The base of rules of fuzzy system is formed. The operation of the fuzzy system is analyzed using the surfaces of the fuzzy system values.

A block diagram of a fuzzy system has been developed, which allows to better understand the principles of operation. Conclusions are made on the correctness of the fuzzy system. Using the developed fuzzy system for diagnosing the operation of computer systems will reduce the chances of problems by preventing them and timely diagnosis, which will save a lot of time and resources.

Keywords: FUZZY LOGIC, BELONGING FUNCTION, FUZZY DIAGNOSTIC SYSTEM.

## ЗМІСТ

Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Поняття комп'ютерної системи.....	11
1.2 Основні характеристики комп'ютерних систем.....	15
1.3 Постановка задачі.....	18
2 Нечітка система аналізу стану комп'ютерної системи.....	21
2.1 Застосування нечіткої логіки в інженерних задачах.....	21
2.2 Аналіз стану комп'ютерної системи.....	24
2.3 Загальна схема нечіткої системи аналізу стану комп'ютерної системи.....	28
3 Дослідження роботи пристрою.....	31
3.1 Вибір середовища моделювання.....	31
3.2 Розробка моделі розробленого засобу.....	34
3.3 Верифікація роботи нечіткої системи.....	39
4 Техніко-економічний розділ.....	42
4.1 Розрахунок витрат на розробку програмного забезпечення.....	42
4.2 Визначення експлуатаційних витрат.....	47
4.3 Розрахунок ціни споживання проектного рішення.....	50
4.4 Визначення показників економічної ефективності.....	51
Висновки.....	53
Список використаних джерел.....	54

					ДП.КСМ. 07097/14.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Басюк Н.В.			НЕЧІТКА СИСТЕМА ДІАГНОСТИКИ КОМП'ЮТЕРНОЇ СИСТЕМИ	Літ.	Арк.	Акрушів
Перевір.		Дубчак Л.О.				8	94	
		Паздрій І.Р.				ТНЕУ. ФКІТ. КСМ-41/1		
Н. Контр.		Гураль І.В.						
Затверд.		Березький О.М.						

## ВСТУП

Комп'ютерні системи мають дуже велике значення у сучасному світі . Вони використовуються, практично, в усіх сферах життя суспільства, стали незамінні для сучасних професій практично усіх галузей. Завдяки комп'ютерній системі можна вирішувати прикладні завдання в предметних галузях діяльності такі як технологічна підготовка, керування, облік, автоматизація процесів. Практичне застосування комп'ютерні системи знайшли при дистанційному навчанні. Раніше, дистанційне навчання означало заочне навчання. Зараз це засіб навчання, що використовує кейс-, ТБ-і мережеві технології навчання. Користувачам комп'ютера доступне програмне забезпечення, а саме: системні та прикладні програми (наприклад, компілятори, текстові редактори, системи управління базами даних тощо). Ці програми взаємодіють з операційною системою, яка, в свою чергу, управляє роботою комп'ютера.

Неможливо уявити сучасну людину без комп'ютера, або ж сучасне підприємство без застосування великих комп'ютерних систем та мереж.

Тому є актуально захистити комп'ютерну систему шляхом вчасної та регулярної діагностики, яка зможе попередити 90% усіх існуючих проблем, що можуть виникнути.

Нечітка система дозволить підтримувати задану функціональність та стійкість будь-якої комп'ютерної системи.

Мета проекту полягає в аналізі роботи комп'ютерних систем, їх основних характеристик та розробці нечіткої системи аналізу їх стану.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- проаналізувати всі можливі стани комп'ютерних систем;
- дослідити нечітку логіку та обрати підходящий алгоритм нечіткого виводу по базі знань;
- побудувати функції належності;

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		9

- сформувати базу правил;
- дослідити роботу реалізованої нечіткої системи;
- зробити висновки щодо правильності роботи розробленої системи.

Об'єкт дослідження - комп'ютерні системи та їх основні характеристики.

Предмет дослідження - методи та засоби підвищення стійкості комп'ютерних систем.

Методи дослідження - методи нечіткої логіки, а саме - алгоритм Мамдані.

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		10

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Поняття комп'ютерної системи

Комп'ютерна система - це сукупність різних компонентів, що використовуються для спільної обробки даних. Мета комп'ютерної системи - зробити процес вирішення завдання на комп'ютері найбільш простим. Функціонує комп'ютерна система об'єднує елементи програмного і апаратного забезпечення. Апаратні елементи - це механічні пристрої комп'ютера, які виконують всі фізичні функції. Програмні елементи - це додатки, написані під систему; саме вони виконують логічні і математичні операції і надають користувачеві можливість управління комп'ютером. Документація включає в себе керівництво і списки допустимих операцій, завдяки яким можна повноцінно використовувати програмні і апаратні складові комп'ютера [2].

Разом ці компоненти утворюють комп'ютерну систему: системне апаратне забезпечення плюс системні програми плюс документація до них дорівнює комп'ютерна система. Зазвичай, до складу комп'ютерної системи входить три базові апаратних складових: сам комп'ютер, який обробляє всі дані; термінальний пристрій, що використовується як друкарська машинка для двостороннього контакту між користувачем і системою; і медіа накопичувач - для зберігання програм і даних. Три цих пристрої - комп'ютер, термінал і медіа-сховище - необхідні складові будь-якої комп'ютерної системи.

Найпоширенішою класифікацією комп'ютерних систем є класифікація (або ж таксономія) за Флінном. Це загальна класифікація архітектур ЕОМ за ознаками наявності паралелізму в потоках команд і даних. Була запропонована Майклом Флінном в 1966 році і розширена в 1972 році. Все розмаїття архітектур ЕОМ в цій таксономії Флінна зводиться до чотирьох класів [1]:

- SISD (Single Instruction stream over a Single Data stream) - обчислювальна система з одиночним потоком команд і одиночним потоком даних;

					ДП.КСМ. 07097/14.00.00.000 ПЗ	11
Змн.	Арк.	№ докум.	Підпис	Дата		

- SIMD (Single Instruction, Multiple Data) - обчислювальна система з одиночним потоком команд і множинним потоком даних;
- MISD (Multiple Instruction Single Data) - обчислювальна система з множинним потоком команд і одиночним потоком даних;
- MIMD (Multiple Instruction Multiple Data) - обчислювальна система з множинним потоком команд і множинним потоком даних.

Архітектура SISD - це традиційний комп'ютер фон-Нейманівської архітектури з одним процесором, який виконує послідовно одну інструкцію за одною, працюючи з одним потоком даних. В даному класі не використовується паралелізм ні даних, ні інструкцій, тому SISD-машина не є паралельною. До цього класу також прийнято відносити конвеєрні, суперскалярні і VLIW-процесори. На рисунку 1.1 зображено тип архітектури SISD.

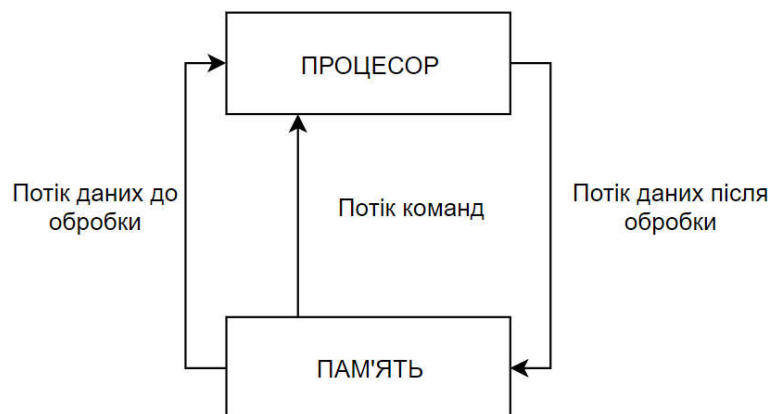


Рисунок 1.1 – Тип архітектури SISD

Типовими представниками SIMD є векторні процесори, звичайні сучасні процесори, коли працюють в режимі виконання команд векторних розширень, а також особливий підвид з великою кількістю процесорів - матричні процесори [7]. У SIMD-машинах один процесор завантажує одну інструкцію, набір даних до них і виконує операцію, описану в цій інструкції, над усім набором даних одночасно. Тип архітектури SIMD зображений на рисунку 1.2.



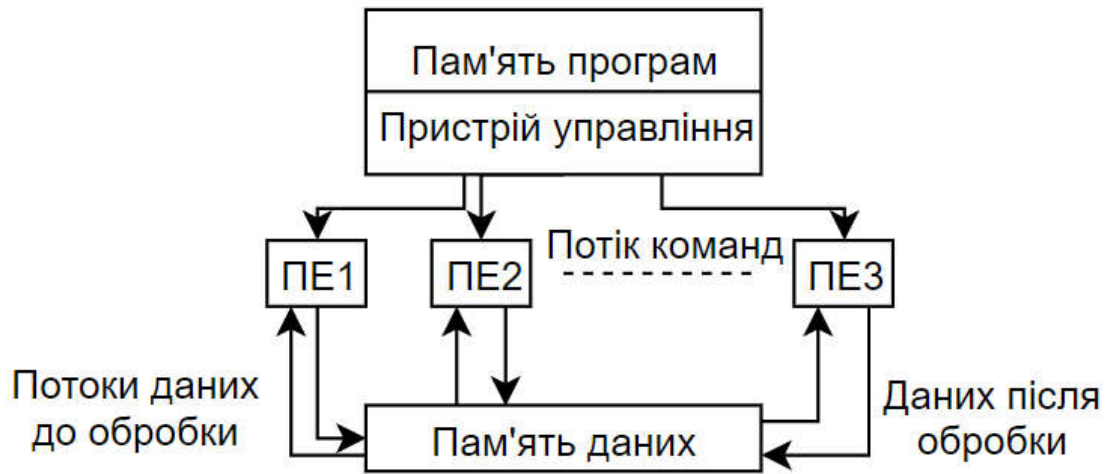


Рисунок 1.2 – Тип архітектури SIMD

До класу MISD ряд дослідників відносить конвеєрні EOM, однак це не знайшло остаточного визнання. Також, можливо вважати MISD системами, системи з гарячим резервуванням. Крім цього, до архітектури MISD деякі відносять систолічні масиви процесорів. Архітектура типу MISD показана на рисунку 1.3.

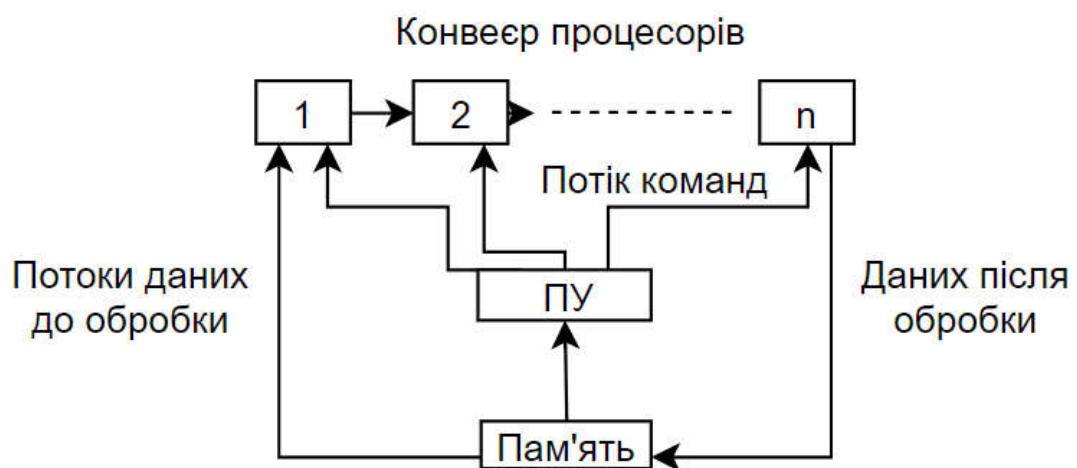


Рисунок 1.3 – Архітектура MISD

Клас MIMD включає в себе багатопроцесорні системи, де процесори обробляють множинні потоки даних. Сюди відносять традиційні мультипроцесорні машини, багатоядерні і багатопотокові процесори, а також комп'ютерні кластери [10]. Архітектура даного класу зображена на

рисунку 1.4.

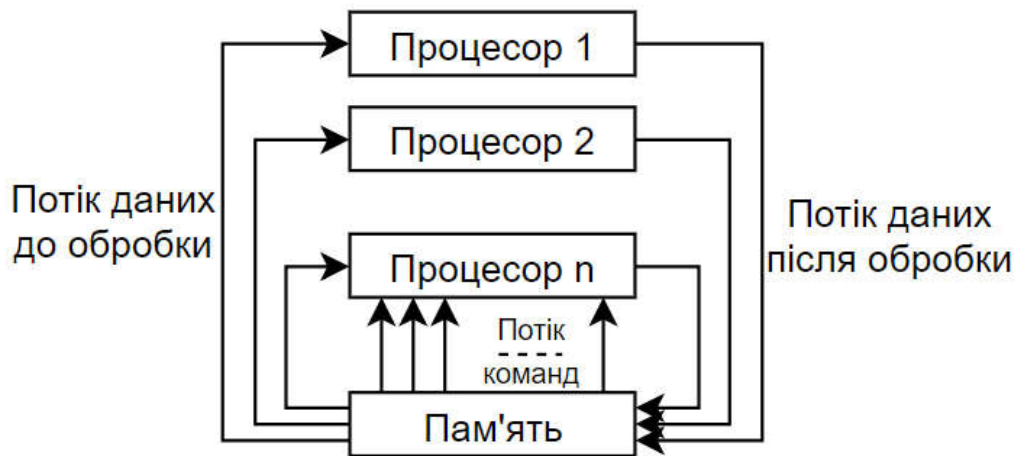


Рисунок 1.4 – Архітектура типу MIMD

У спеціалізованій літературі можна зустріти ще такі підкласи MIMD-класу:

- SPMD (Single Program Multiple Data);
- MPMD (Multiple Programs Multiple Data).

SPMD (Single Program Multiple Data) - описує систему, де на всіх процесорах MIMD-машини виконується тільки одна єдина програма і на кожному процесорі вона обробляє різні блоки даних.

MPMD (Multiple Programs Multiple Data) - описує систему в двох випадках:

- на одному процесорі MIMD-машини працює майстер-програма, а на інших підпорядкована програма, роботою якої керує майстер-програма (принцип master / slave або master / worker);

- на різних вузлах MIMD-машини працюють різні програми, які по-різному обробляють один і той же масив даних (принцип coupled analysis), здебільшого вони працюють незалежно один від одного, але час від часу надсилати та отримувати дані для переходу до наступного кроку [5].

Відношення конкретних машин до конкретного класу сильно залежить від точки зору дослідника. Так, конвеєрні машини можуть бути віднесені і до класу SISD (конвеєр - єдиний процесор), і до класу SIMD (векторний потік даних з конвеєрним процесором), і до класу MISD (безліч процесорів конвеєра

обробляють один потік даних послідовно), і до класу MIMD - як виконання послідовності різних команд (операцій шаблів конвеєра) з множинним скалярним потоком даних (вектором).

## 1.2 Основні характеристики комп'ютерних систем

Основні показники КС - характеристики продуктивності, енергетичні характеристики, характеристики надійності та ефективності систем, економічні показники - взаємопов'язані і взаємозалежні. Поліпшення однієї групи показників якості, наприклад, збільшення продуктивності, веде до погіршення інших - ускладнення структури, збільшення вартості, зниження надійності і т. д.

Однією з найголовніших характеристик є швидкість. Швидкість - це час, який комп'ютер витрачає на виконання завдання операції. Час, який витрачає комп'ютер для виконання певного завдання, набагато менший, ніж той, який витрачається людиною. Різні комп'ютери класифікуються на основі їхньої швидкості, порівнюючи їх MIPS (Million Instructions per Second). Всі складові повинні бути приблизно однакового рівня, тому що якщо якась із деталей буде набагато старше інших, то вона буде гальмувати і все інше, якщо звичайно вони працюють синхронно. Синхронно – це означає, що інше обладнання буде чекати сигналу-відповіді від повільного обладнання і поки не дочекається теж не почне роботу. Асинхронне (багатопотокове), тобто незалежне – це коли воно не чекає відповіді, а займається чимось іншим поки повільне устаткування не закінчило свої розрахунки. Тому повинні бути сучасними: оперативна пам'ять, мікропроцесор, відеокарта, жорсткий диск і материнська плата (вона з'єднує їх між собою).

Ще однією важливою характеристикою є точність. Точність - це ступінь правильності та точності операцій, що виконуються комп'ютером. За відсутності поганого програмування комп'ютери не роблять помилок і здатні

					ДП.КСМ. 07097/14.00.00.000 ПЗ	15
Змн.	Арк.	№ докум.	Підпис	Дата		

точно виконувати складні інструкції. Якщо дані, що надходять на комп'ютер, не мають помилок, це може призвести до неточних результатів. При розраховуванні складних задач чи багатопроцесних програм точність відіграє провідну роль. Адже дані з входу на вихід повинні потрапляти у точно визначеному форматі, з певним розміром та інтервалом часу. Якщо щось піде не так, тоді у всій системі може статись збій, що призведе до непоправних результатів. Таким чином буде втрачений дорогоцінний час, а також в окремих випадках пошкоджено деякі елементи комп'ютерної системи [8].

Не менш важливою характеристикою в роботі комп'ютерних систем є надійність. Комп'ютерні системи не відповідають на людські фактори, такі як стомлюваність або нудьга. Тому вони частіше працюють неодноразово і ефективно. У разі виникнення будь-яких неполадків в комп'ютерній системі існують положення про негайне резервне копіювання інформації та програм. Резервне копіювання є важливим компонентом комп'ютерних систем, бо в разі серйозних збоїв масштабний проект може бути аварійно закритий, і тоді всі дані будуть втрачені, а клопітка робота впродовж довгого часу виявиться марною. Отож зберігання резервних копій результатів роботи комп'ютерної системи є обов'язковим для подальшого продуктивного використання.

Не менш важливим є і те, що будь яка комп'ютерна система, яка зосереджена на широке коло використання і є багатообіцяючою в плані виконання різносторонніх задач має бути універсальною. Комп'ютери здатні виконувати всі рівні завдань - прості або складні. Тому вони можуть бути використані в будь-якій галузі науки, технології, бізнесу, фінансів, рахунків, комунікацій тощо. В сучасному світі багатозадачність і різнобічність комп'ютерних систем є чи не найважливішою характеристикою для активного користувача персонального комп'ютера [9]. Оскільки навіть навчання в університеті вимагає роботи з різними сферами наукових розробок, то і комп'ютер має відповідати запитам користувачів, щоб залишатися актуальним та затребуваним. Комп'ютерна система, для того щоб мати успіх на ринку продажів має задовільняти потреби як звичайного студента, так і, наприклад, професора чи знаменитого академіка. Успіх гарантований лише при адаптації

					ДП.КСМ. 07097/14.00.00.000 ПЗ	16
Змн.	Арк.	№ докум.	Підпис	Дата		

до потреб цільової аудиторії користувачів.

Не можна уявити повноцінну комп'ютерну систему, що не здатна зберігати певні файли чи програми. Це стосується здатності комп'ютера зберігати дані та програми. Зберігання проводиться на носіях таких як компакт-диски, дискети, DVD-диски, оперативна пам'ять (Random Access Memory), постійний запам'ятовуючий пристрій (Read Only Memory). Працюючи за комп'ютером, людина оперує тисячами різноманітних файлів, програм, застосунків і логічно те, що все це потрібно зберігати. Відповідно, чим більший об'єм опрацьовуваних програм чи файлів, тим більше пам'яті потрібно для ефективної роботи [3].

Проте не менш важливою є і оперативна пам'ять, яка відіграє провідну роль в швидкодії комп'ютера. Від кількості оперативної пам'яті також залежить з якою швидкістю будуть виконуватись операції в комп'ютерній системі. Для сучасних комп'ютерів кількість оперативної пам'яті менше 4 гігабайт - це дуже мало, і неозброєним оком помітно, що час реакції значно зменшується. Основним накопичувачем даних є жорсткий диск (HDD), заснований на принципах магнітного запису. Проте більш швидким його аналогом є твердотілий накопичувач (SSD). Його перевагами є те, що в нього низьке енергоспоживання, відсутність механічних частин, малі розміри і вага, широкий спектр робочих температур і найголовніше - це висока швидкість читання та запису, що значно перевищує пропускну здатність жорсткого диску.

Важливу роль в створенні будь якої комп'ютерної системи відіграють економічні характеристики її складових. Для успішної реалізації ціна повинна відповідати якості, щоб бути доступною для широкого кола користувачів. Всі компоненти, що є вагомими при створенні, мають бути сучасними, відповідати один одному і водночас не надто ускладнювати архітектуру побудови.

Також значний вплив на роботу комп'ютерних систем має підібрана операційна система, а також відповідне системне програмне забезпечення. Вони також мають відповідати фізичним компонентам, для продуктивної та

					ДП.КСМ. 07097/14.00.00.000 ПЗ	17
Змн.	Арк.	№ докум.	Підпис	Дата		

високоєфективної роботи.

Комп'ютерні системи є так званим фундаментом для сучасного виробництва та проектування у будь якій галузі. Таке широке використання комп'ютерів не даремне, адже значно економить час роботи і спрощує виконання, здавалося б громіздких задач. Все, починаючи від заводу хлібобулочних виробів і закінчуючи надсекретною лабораторією, базується на використанні комп'ютерних технологій. Тому розробка ефективної та швидкої комп'ютерної системи є головним завданням, бо в майбутньому все буде базуватись саме на ній [1].

Важливим є і те, що сучасне життя неможливо уявити без використання комп'ютерів. І саме розробки та наукові винаходи у цій сфері мають високу рентабельність та перспективу на майбутнє. Працюючи над розробкою, потрібно враховувати що є необхідним, враховувати основні компоненти, а також думати над тим, що технічний прогрес не стоїть на місці і для забезпечення реалізації проекту треба створювати інновації, які обов'язково стануть корисними і затребуваними для майбутніх поколінь.

### 1.3 Постановка задачі

Оскільки при діагностиці комп'ютерної системи необхідно врахувати поточні параметри системи, такі як продуктивність, допустимі затрати пам'яті та необхідний рівень стійкості до часового аналізу, то для вирішення цієї задачі варто застосувати апарат нечіткої логіки.

Математична теорія нечітких множин (fuzzy sets) і нечітка логіка (fuzzy logic) є узагальненнями класичної теорії множин і класичної формальної логіки. Дані поняття були вперше запропоновані американським ученим Лотфі Заде (Lotfi Zadeh) у 1965 р. Основною причиною появи цієї теорії стала наявність нечітких і наближених міркувань при описі людиною процесів, систем, об'єктів.

					ДП.КСМ. 07097/14.00.00.000 ПЗ	18
Змн.	Арк.	№ докум.	Підпис	Дата		

Основними перевагами нечітких систем у порівнянні з іншими є:

- можливість оперувати вхідними даними, заданими нечітко, наприклад, значеннями, що невинно змінюються в часі (динамічні задачі);
- можливість нечіткої формалізації критеріїв оцінки і порівняння;
- можливість проведення якісних оцінок як вхідних даних, так і виведених результатів, оскільки система оперує не тільки власне значеннями даних, а й їх ступенем вірогідності та її розподілом;
- можливість проведення швидкого моделювання складних динамічних систем та їх порівняльний аналіз із заданим ступенем точності.

В інженерних задачах застосовується, як правило, механізм нечіткого висновку Мамдані. В ньому використовується мінімаксна композиція нечітких множин. Даний механізм включає наступну послідовність дій:

- процедура фазифікації: визначаються ступені істинності, тобто значення функцій належності  $MF_i(x)$  для лівих частин кожного  $i$ -го правила (передумов);

- нечіткий висновок. Спочатку визначаються мінімальний рівень "відсічення" для лівої частини кожного з правил  $A_i = \min(MF_i(x))$ , а потім знаходяться "усічені" функції належності висновку  $B_i = \min(A_i, B_i)$ ;

- композиція або об'єднання отриманих "усічених" функцій, для чого використовується максимальна композиція нечітких множин  $MF(y) = \max(B_i(y))$ ;

- дефазифікація або приведення до чіткості.

Існує декілька методів дефазифікації. Наприклад, метод середнього центру або центроїдний метод. Геометричний зміст такого значення – центр ваги для кривої функції належності отриманого виходу.

Застосування апарату нечіткої логіки при створенні апаратно-програмного засобу для здійснення діагностики комп'ютерної системи шляхом вибору оптимального методу модулярного експоненціювання для кожної окремої характеристики та врахування поточних параметрів самої КС дозволить забезпечити стійкість системи до часового аналізу в режимі

реального часу [4].

Для вирішення цього завдання необхідно (рисунок 1.5):

- дослідити основні параметри методів модулярного експоненціювання, що найчастіше застосовуються в сучасних комп'ютерних системах та визначити стійкість кожного з них до часового аналізу;
- розробити метод обробки нечіткої інформації під час діагностики комп'ютерної системи, враховуючи значення всіх характеристик;
- розробити структуру засобу реалізації розробленого методу та дослідити його основні характеристики.



Рисунок 1.5 – Дерево рішень дипломного проекту

Спроектвавши схему, за якою буде проводитись подальша робота, можна досягнути найменшої кількості неточностей, а значить досягнути бажаного результату в найкоротші терміни. Це стане основою майбутньої розробки, за допомогою точно поставленого завдання, структурної схеми та проаналізованому матеріалу розробка нечіткої системи буде точною та з найменшою кількістю помилок, що дозволить заощадити значну кількість часу та ресурсів.



## 2 НЕЧІТКА СИСТЕМА АНАЛІЗУ СТАНУ КОМП'ЮТЕРНОЇ СИСТЕМИ

### 2.1 Застосування нечіткої логіки в інженерних задачах

Нечітка логіка, звана "fuzzy control", добре відома інженерам програмістам систем управління як зручний засіб програмування і моніторингу додатків управління технологічними процесами. За аналогією з традиційними засобами управління технологічними процесами, системи на основі нечіткої логіки можуть використовуватися для опису петель регулювання і брати участь в обчисленні керуючого впливу відповідно до однієї або великої кількості точок завдання для одного або більшої кількості вимірів [6].

Правила нечіткої логіки дозволяють забезпечити:

- застосування існуючого досвіду управління;
- використовувати гнучкі правила в разі неможливості точно моделювати систему за допомогою традиційних засобів.

Покращення якості управління при цьому здійснюється за допомогою:

- саморегулювання системи управління;
- випереджаюча зміна вихідного впливу (функція попередження), базуючись на подіях, які не можуть бути враховані у разі застосування традиційних способів управління.

Кількість додатків заснованих на даних методах управління безперервно збільшується для безперервних процесів, для додатків пакетної обробки, а також для автоматизованих систем. Нечітка логіка, завдяки використанню її в цій галузі, отримала опис і формулювання в якості методу програмування. Вона дозволяє систематизувати емпіричні знання і застосувати їх для управління процесами в разі труднощів із застосуванням класичних методів управління [4]. Теорія нечіткої логіки дозволяє описати набори методів управління, які нескладно застосувати для реальної системи і дозволяє врахувати досвід операторів і технологів для динамічного управління процесом. Це дозволяє описувати на нечіткій логіці окремі частини

					ДП.КСМ. 07097/14.00.00.000 ПЗ	21
Змн.	Арк.	№ докум.	Підпис	Дата		

виробничого процесу, такі як ініціалізація, завдання параметрів.

В інженерних задачах застосовується, як правило, механізм нечіткого виводу Мамдані, який можна реалізувати у середовищі Matlab. Метод Мамдані був одним з перших, побудованих за допомогою теорії нечітких множин. Він був запропонований в 1975 році Ібрагімом Мамдані.

Метод Мамдані є нечіткою системою виведення (НСВ). НСВ являє собою систему, що використовує теорію нечітких множин для відображення входів (функцій в разі нечіткої класифікації) до виходів (класів в разі нечіткої класифікації) [15].

Метод складається з наступних етапів:

- формування бази правил;
- фазифікація;
- агрегування підумов;
- активізація підвисновків;
- акумулювання висновків;
- дефазифікація.

Кожен етап виконується послідовно, до того ж кожен наступний етап отримує на вхід значення, що були отримані в результаті роботи попереднього. Для вхідних значень метод Мамдані використовує певну базу правил.

Для прикладу, правила можуть бути представлені у такому вигляді: якщо  $x \in Y_1$ , тоді  $z \in M_1$ . В такому випадку  $x$  – вхідне значення,  $z$  – вихідне значення,  $Y$  і  $N$  – нечіткі множини.

Фазифікацію вхідних змінних ще називають приведенням їх до нечіткості. На вхід надходить вже сформована база правил і певний масив вхідних даних  $A = \{a_1, \dots, a_n\}$ . В такому масиві містяться значення всіх вхідних змінних. Метою проведення даного етапу є отримання значень істинності для всіх умов з бази правил. Це відбувається наступним чином: для кожної з умов знаходиться значення  $b_i = \mu(a_i)$ . Таким чином існує безліч значень  $b_i$  ( $i=1..k$ ).

На етапі агрегування умов визначається ступінь істинності умов для кожного з правил системи нечіткого виведення [20]. Тобто, для кожної умови

знаходиться мінімальне значення істинності всіх її підумов. Формально, це виглядає так (формула 2.1):

$$c_j = \min \{b_i\}, \quad (2.1)$$

де  $j = 1..q$  ;

$i$  – число з безліччю номерів підумов, в яких бере участь  $j$ -а вхідна змінна.

На етапі активізації виводів відбувається перехід від умов до підвыводів. Для кожного підвывода знаходиться ступінь істинності  $d_i = c_i * F_i$ , де ( $i = 1..q$ ). Потім, кожному  $i$ -му підвыводу зіставляється безліч  $D_i$  з новою функцією належності. Її значення визначається як мінімум з  $d_i$  і значення функції належності терму з підвыводу. Цей метод називається *min-активізацією*, який формально записується в такий спосіб (формула 2.2):

$$\mu_i' = \min \{d_i, \mu_i(x)\}, \quad (2.2)$$

де  $\mu_i'(x)$  – «активована» функція належності;

$\mu_i(x)$  – функція належності терму;

$d_i$  – ступінь істинності  $i$ -го підвыводу.

Отже, мета цього етапу – отримання суми «активованих» нечітких множин  $D_i$  для кожного з підвыводів в базі правил ( $i = 1..q$ ).

На етапі акумуляції виводи отримують нечітку множину (або їх об'єднання) для кожної з вихідних змінних. Виконується цей етап наступним чином:  $i$ -тій вихідній змінній зіставляється об'єднання множин  $E_i = \cup D_j$ . Де  $j$  – номер підвыводів, в яких бере участь  $i$ -та вихідна змінна ( $i = 1..s$ ).

Об'єднанням двох нечітких множин є третя нечітка множина з наступною функцією приналежності (формула 2.3):

$$\mu_i'(x) = \max \{\mu_1(x), \mu_2(x)\}, \quad (2.3)$$

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		23

де  $\mu_1(x), \mu_2(x)$  – функції приналежності поєднаних множин [11].

Метою дефазифікації є отримання кількісного значення для кожної з вихідних лінгвістичних змінних. На практиці це відбувається так: розглядається  $i$ -та вихідна змінна і її відповідна множина  $E_i(1..s)$ . Далі за допомогою методу дефазифікації знаходиться кінцеве кількісне значення вихідної змінної. У такій реалізації алгоритму використовується метод центру тяжіння, в якому значення  $i$ -ої вихідної змінної розраховується за формулою 2.4 :

$$y_i = \frac{\int_{\min}^{\max} x * \mu_i(x) dx}{\int_{\min}^{\max} \mu_i(x) dx}, \quad (2.4)$$

де  $\mu_i(x)$  – функція приналежності відповідної нечіткої множини  $E_i$ ;

$\min$  і  $\max$  – кордони універсуму нечітких змінних;

$y_i$  – результат дефазифікації.

## 2.2 Аналіз стану комп'ютерної системи

Переважає більшість користувачів регулярно працюють за комп'ютером і не замислюються про те, що в певний момент він може вимкнутися і більше не ввімкнутись зовсім. Так, досить часто виникає проблема коли щойно зібраний або оновлений комп'ютер не включається. Або ж, якщо комп'ютер раптово перестає працювати. У такому випадку головне - правильно визначити несправність. Тому, що ремонт в певних випадках може бути не обов'язковим. Спочатку треба розібратися з причинами, що можуть викликати певну несправність [12]. Як відомо, і пил і несприятливі кліматичні умови погіршують стан комплектуючих персонального комп'ютера. Тому вихід апаратного забезпечення з ладу може бути викликаний окисленням контактів,

					ДП.КСМ. 07097/14.00.00.000 ПЗ	24
Змн.	Арк.	№ докум.	Підпис	Дата		

потраплянням пилу (відповідно, статичної електрики) на мікросхеми і роз'єми, їх температурний збій.

Також всі несправності можуть бути наслідком стрибків напруги, неправильної роботи блоку живлення, або ж неправильного заземлення. Найперше, що можна порекомендувати в такому випадку - використання мережевих фільтрів та заземлення комп'ютера. В першу чергу при несправності персонального комп'ютера слід провести візуальний огляд, зняти захисну кришку та добре оглянути зовнішній вигляд всіх компонентів чи нема відхилень від норми, а також варто звернути увагу чи не присутній запах диму, що часто притаманно проблемам, що пов'язані з стрибками напруги. Якщо явних ознак несправностей на елементах комп'ютера не виявлено, то варто перевірити надійність підключення до живлення. Якщо перевірка не дала результатів, то можна включити комп'ютер і перевірити чи функціонують вентилятори на блоку живлення (БЖ) і на кулері процесора, також можна перевірити кріплення кулера. Якщо вентилятор не функціонує і жорсткий диск не відтворює характерного звуку розкручування, то можна зробити висновок, що несправність стосується блоку живлення [13].

Наявність напруги на виході блоку живлення можна перевірити тестером, помірявши величину напруги на контактах системної плати в тому місці, де вони з'єднані з блоком живлення. Для впевненості можна підключити інший блок живлення і перевірити справність інших компонентів комп'ютера. Незважаючи на те, що монітор ламається нечасто, потрібно перевірити чи подаються йому сигнали з відеоадаптера. Для цього осцилографом необхідно перевірити наявність діючих сигналів.

Система автоматичної діагностики комп'ютера являє собою комплекс програмних, мікропрограмних і апаратних засобів. Розрізняють системи тестової і функціональної діагностики. У системах тестової діагностики результати на діагностуючій пристрій надходять від засобів діагностування. У середніх і великих електронно-обчислювальних машинах зазвичай використовуються спеціалізовані засоби діагностування [14]. У мікроелектронно-обчислювальних машинах частіше використовуються вбудовані

					ДП.КСМ. 07097/14.00.00.000 ПЗ	25
Змн.	Арк.	№ докум.	Підпис	Дата		

засоби подачі тестових впливів в зовнішні засоби для зняття і обробки результатів. Процес діагностики включає в себе декілька етапів (простих перевірок), кожна з яких характеризується подачею на пристрій або ж робочим тиском, що знімається з відповідного пристрою. Через те, що існує велика кількість проблем, які можуть виникнути, не існує єдиного підходу до діагностики комп'ютера. Тому до пошуку проблем потрібно підходити з наполегливістю. Існує перелік проблем з апаратною частиною комп'ютера, які зустрічаються найчастіше:

- комп'ютер не вмикається;
- комп'ютер вмикається, але на екрані немає інформації;
- комп'ютер вмикається і видає специфічні звуки ("пищить");
- різноманітні помилки пов'язані зі стартом BIOS;
- комплектуючі не функціонують;
- комп'ютер перегрівається;
- апаратна несумісність обладнання;
- система працює дуже повільно.

Отож, очевидно, що діагностику комп'ютерної системи можна проводити за допомогою фізичних ресурсів людини, проте аналогом служать програмні засоби, які зараз є досить поширеними серед активних користувачів комп'ютера [15]. Ці програмні засоби - це так званий набір команд, які надходять до кожної складової комп'ютера і надають певний результат, що відображається в діалоговому вікні. Програмна система діагностики опирається на вже внесені дані щодо нормальної роботи складових комп'ютера, порівнює їх, аналізує та відображає отримані результати. Дізнатись найпростіші характеристики персонального комп'ютера можна і стандартними засобами Windows.

					ДП.КСМ. 07097/14.00.00.000 ПЗ	26
Змн.	Арк.	№ докум.	Підпис	Дата		

Один із способів - це перегляд за допомогою диспетчера завдань (рисунок 2.1).

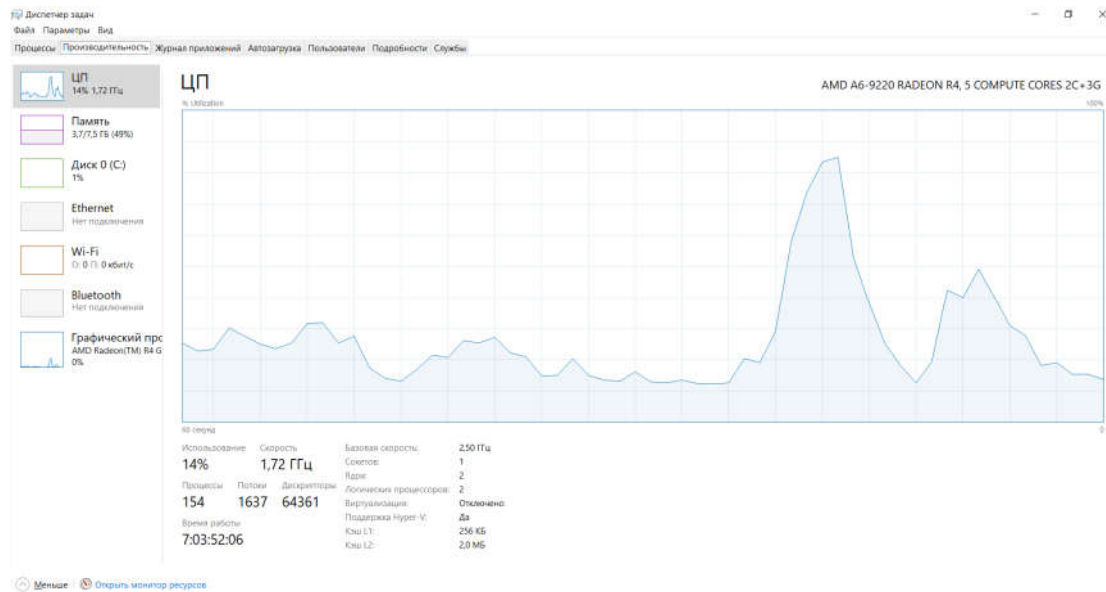


Рисунок 2.1 – Поточні характеристики комп'ютера

Для діагностики комп'ютерних систем існує безліч способів, програмних, апаратних та навіть спеціальних веб-ресурсів. Кожен обирає той, який буде зручніший та функціональніший для своїх же потреб. Але загальні методи аналізу та діагностики є для всіх засобів однаковими. Це обов'язкова перевірка складових системного блоку як на наявність фізичних дефектів, так і перевірка за допомогою програмних засобів, що допомагають ідентифікувати неточності та неполадки, що можуть стосуватись програмного коду системних додатків або характеристик тих же апаратних засобів [17].

Аналіз комп'ютерної системи – поширена та затребувана тематика для всіх користувачів комп'ютера. Вчасна діагностика дозволяє запобігти серйозним проблемам і відповідно уникнути ремонту, який може вимагати великої кількості коштів. Тому в діагностиці важливо враховувати кожен елемент, незалежно від його значущості. Це особливо важливо при перевірці системних додатків на наявність різноманітних вірусів, адже вони можуть ховатися в найменших файлах і згодом нанести безповоротну шкоду програмному та апаратному забезпеченню [25].

## 2.3 Загальна схема нечіткої системи аналізу стану комп'ютерної системи

Схема нечіткої системи аналізу комп'ютерної системи буде розроблена за допомогою вже відомого методу нечіткого виводу Мамдані. Для створення такої системи потрібен нечіткий контролер, що описаний за допомогою програмного середовища Matlab. Його загальна схема зображена на рисунку 2.2.

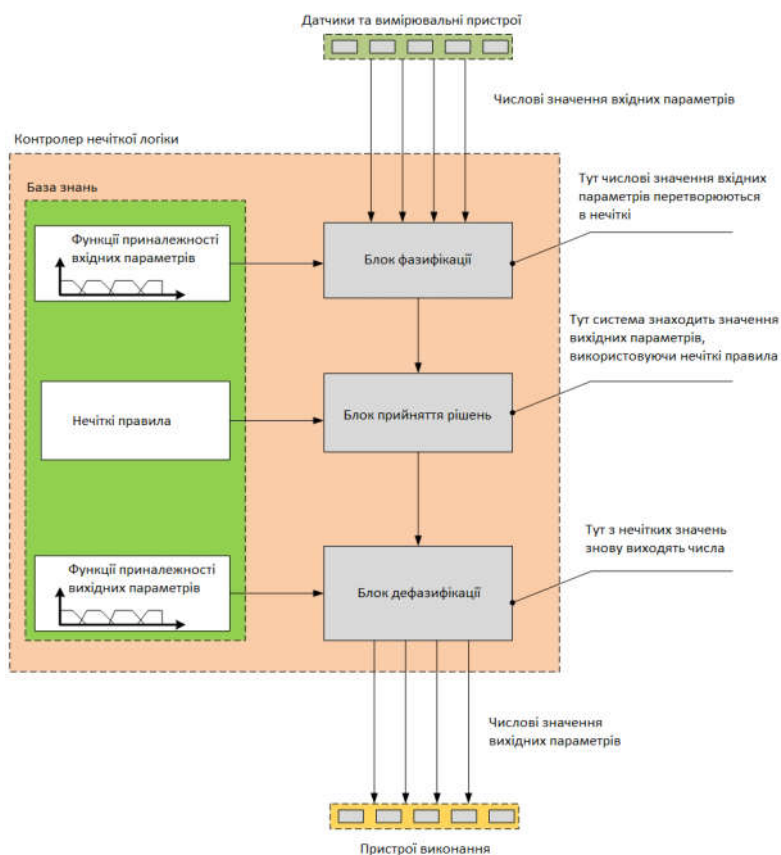


Рисунок 2.2 – Загальна схема нечіткого контролера

Отже, для розробки нечіткої системи, що діагностує комп'ютерну систему потрібні певні вхідні дані, а також нечіткі правила, за допомогою яких і буде проводитись діагностика [27].

База правил системи нечіткого виводу призначена для формального подання емпіричних знань або знань експертів в тій чи іншій проблемній області. У системах нечіткого виведення використовуються правила нечітких продукцій, в яких умови і укладення сформульовані в термінах нечітких

Змн.	Арк.	№ докум.	Підпис	Дата



лінгвістичних висловлювань. База правил нечітких продукцій - це кінцева множина правил нечітких продукцій, узгоджених щодо використовуваних в них лінгвістичних змінних. Найчастіше база правил представляється в формі певного структурованого тексту:

- правило 1: якщо «умова\_1, то висновок\_1», або ж, до прикладу, якщо температура центрального процесора (ЦП) наближена до 100 градусів за Цельсієм, то стан ЦП критичний і потребує втручання з боку користувача.

Для того, щоб перевести вхідний параметр в нечітку область використовується фазифікація, яка в свою чергу керується функціями приналежності  $M_x$ . Вона показує ступінь приналежності параметра до одного з нечітких значень. Чим більша ступінь приналежності, тим більша ймовірність, що обчислювальна машина присвоїть змінній відповідне нечітке значення [21]. Однак не варто плутати функцію приналежності з функцією імовірнісного розподілу. Тому, зокрема, сума ступенів належності одного вхідного параметра до різних нечітких значень не обов'язково дорівнює 1. Графік функції належності показано на рисунку 2.3.

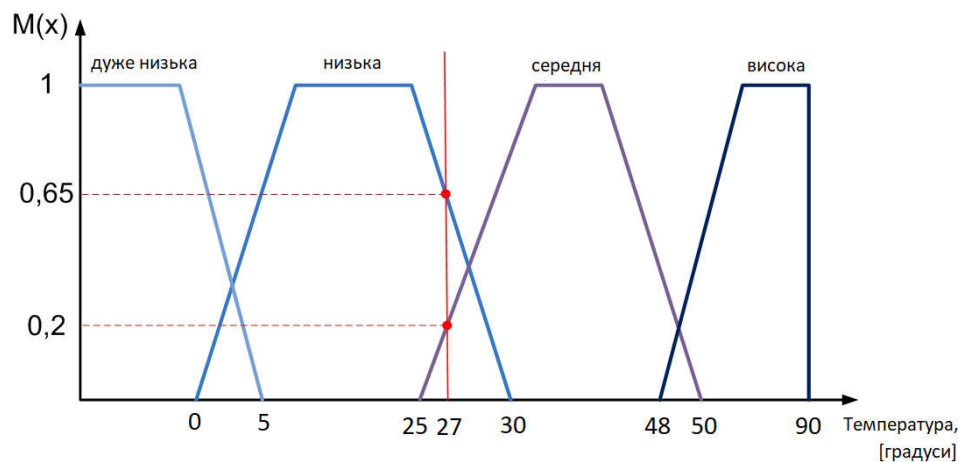


Рисунок 2.3 – Графік функції належності

Точно такі ж функції приналежності потрібно визначити і для інших вхідних і вихідних параметрів системи. Як тільки система управління фазифікує всі вхідні параметри по заданих функціях приналежності, блок прийняття рішення знайде відповідні значення вихідних параметрів,

користуючись нечіткими правилами, що були складені попередньо [19].

На етапі дефазифікації система управління буде робити зворотне перетворення з нечітких значень вихідних параметрів (знайдених по таблиці) до точних цифр. Математичні алгоритми цих перетворень різноманітні і залежать від конкретного завдання. В нашому випадку алгоритм роботи вже визначений і формули, які потрібні для роботи також.

Як висновок, можна сказати, що нечіткі системи справді є дуже зручними для моделювання певних складних систем пов'язаних з ЕОМ. Вони мають велику кількість переваг перед іншими, серед яких:

- можливість оперувати вхідними даними, заданими нечітко: наприклад, такі що безупинно змінюються в часі (динамічні задачі), значення, що неможливо задати однозначно (результати статистичних опитувань, рекламні компанії);

- можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями "більшість", "можливо", переважно";

- можливість проведення якісних оцінок як вхідних даних, так і виведених результатів;

- можливість проведення швидкого моделювання складних динамічних систем і їх порівняльний аналіз із заданим ступенем точності: оперуючи принципами поведінки системи, описаними fuzzy-методами, по-перше, не витрачається багато часу на з'ясування точних значень змінних і складання рівнянь, що їх описують, по-друге, можна оцінити різні варіанти вихідних значень [30].

## 3 ДОСЛІДЖЕННЯ РОБОТИ ПРИСТРОЮ

### 3.1 Вибір середовища моделювання

Для створення будь-якої нечіткої системи логічними є використання програмного середовища Matlab. Matlab - це високопродуктивна мова для технічних обчислень. Вона об'єднує обчислення, візуалізацію та програмування в простому середовищі, де проблеми та рішення виражаються у знайомому математичному позначенні [22]. Типовими варіантами використання Matlab є:

- математика та обчислення;
- розробка алгоритмів;
- моделювання та прототипування;
- аналіз даних, дослідження та візуалізація;
- наукова та інженерна графіка;
- розробка додатків, включаючи створення графічного інтерфейсу користувача.

Matlab - це інтерактивна система, основним елементом якої є масив, який не вимагає розмірів. Це дозволяє вирішити багато технічних обчислювальних завдань, особливо тих, що мають матричні та векторні формулювання, на частку часу, необхідного для написання програми в скалярній неінтерактивній мові, такій як C або Fortran [23].

Назва Matlab означає матричну лабораторію. Спочатку Matlab був написаний, щоб забезпечити легкий доступ до матричного програмного забезпечення, розробленого проектами Linpack та Eispack, які разом представляють найсучасніші рішення у програмному забезпеченні для матричних обчислень.

Matlab розвивався протягом декількох років за допомогою численних користувачів. У середовищі університетів це стандартний навчальний інструмент для просунутих курсів з математики, техніки та інших дисциплін. У промисловості Matlab є основним інструментом для розробки рішень для

					ДП.КСМ. 07097/14.00.00.000 ПЗ	31
Змн.	Арк.	№ докум.	Підпис	Дата		

високопродуктивних досліджень та аналізу [26].

Matlab - це сімейство рішень для конкретних програм, які називаються наборами інструментів. Дуже важливим для більшості користувачів Matlab, є те що панелі інструментів дозволяють вивчати та застосовувати спеціалізовані технології. Панелі інструментів - це всеосяжні набори функцій Matlab (М-файли), які розширюють середовище MATLAB для вирішення конкретних класів проблем. Області, в яких доступні набори інструментів, включають обробку сигналів, системи управління, нейронні мережі, нечітку логіку, симуляцію та багато інших. На рисунку 3.1 представлений типовий інтерфейс програмного середовища Matlab.

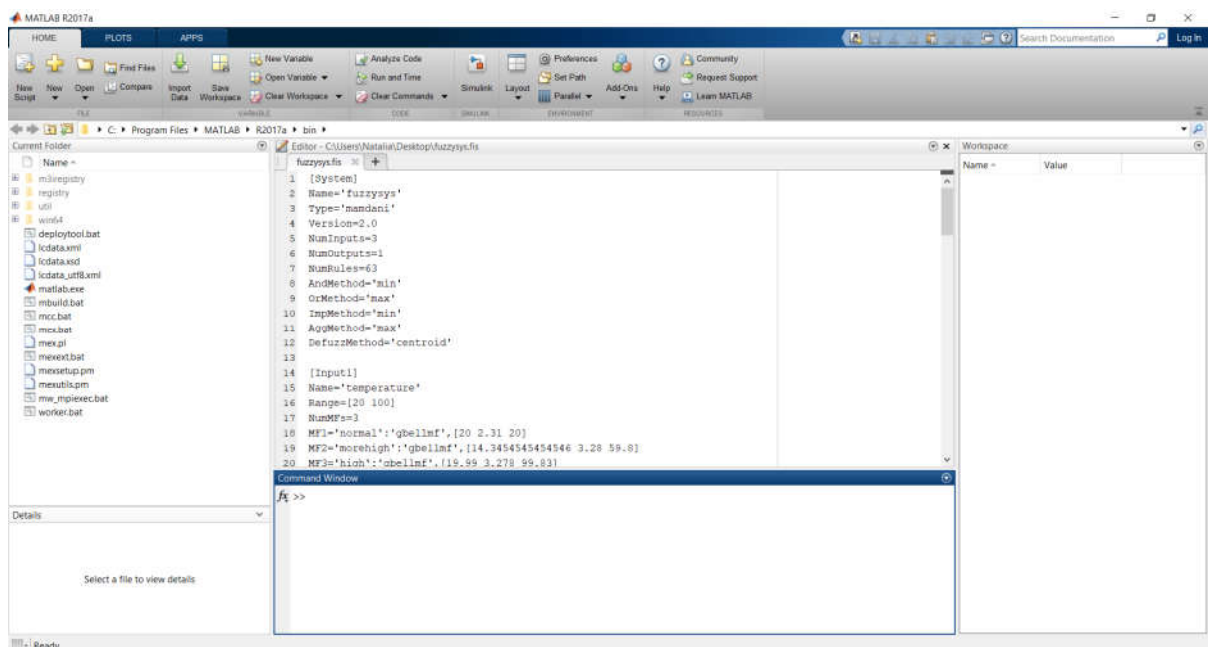


Рисунок 3.1 – Інтерфейс Matlab

Для Matlab є можливість створювати спеціальні набори інструментів (англ. Toolbox), що розширюють його функціональність. Набори інструментів - це набори функцій, написаних на мові Matlab для вирішення певного класу задач [18]. Компанія Mathworks поставляє набори інструментів, які використовуються в багатьох областях, включаючи наступні:

- цифрова обробка сигналів , зображень та даних (DSP Toolbox , Image Processing Toolbox , Wavelet Toolbox , Communication Toolbox , Filter Design Toolbox - набори функцій, що дозволяють вирішувати широкий спектр завдань

обробки сигналів, зображень, проектування цифрових фільтрів і систем зв'язку);

- системи управління (Control Systems Toolbox ,  $\mu$ -Analysis and Synthesis Toolbox , Robust Control Toolbox , System Identification Toolbox , LMI Control Toolbox , Model Predictive Control Toolbox , Model-Based Calibration Toolbox - набори функцій, що полегшують аналіз і синтез динамічних систем, проектування, моделювання та ідентифікацію систем управління, включаючи сучасні алгоритми управління, такі як  $H^\infty$ -управління , ЛМН-синтез ,  $\mu$ -синтез та інші);

- фінансовий аналіз (GARCH Toolbox , Fixed-Income Toolbox , Financial Time Series Toolbox , Financial Derivatives Toolbox , Financial Toolbox , Datafeed Toolbox - набори функцій, що дозволяють швидко і ефективно збирати, обробляти і передавати різну фінансову інформацію);

- аналіз і синтез географічних карт, включаючи тривимірні (Mapping Toolbox);

- збір і аналіз експериментальних даних (Data Acquisition Toolbox , Image Acquisition Toolbox , Instrument Control Toolbox , Link for Code Composer Studio - набори функцій, що дозволяють зберігати та обробляти дані, отримані в ході експериментів, в тому числі в реальному часі. Підтримується широкий спектр наукового та інженерного вимірювального обладнання);

- візуалізація та уявлення даних (Virtual Reality Toolbox - дозволяє створювати інтерактивні світи і візуалізувати наукову інформацію за допомогою технологій віртуальної реальності і мови VRML);

- засоби розробки (Matlab Builder for COM, Matlab Builder for Excel, Matlab Builder for NET, Matlab Compiler, Filter Design HDL Coder - набори функцій, що дозволяють створювати незалежні програми з середовища Matlab);

- взаємодія з зовнішніми програмними продуктами (Matlab Report Generator , Excel Link , Database Toolbox , Matlab Web Server , Link for ModelSim - набори функцій, що дозволяють зберігати дані різних видів таким чином, щоб інші програми могли з ними працювати);

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		33

- бази даних (Database Toolbox - інструменти роботи з базами даних);
- наукові та математичні пакети (Bioinformatics Toolbox , Curve Fitting Toolbox , Fixed-Point Toolbox , Fuzzy Logic Toolbox , Genetic Algorithm and Direct Search Toolbox , OPC Toolbox , Optimization Toolbox , Partial Differential Equation Toolbox, Spline Toolbox, Statistic Toolbox , RF Toolbox - набори спеціалізованих математичних функцій, що дозволяють вирішувати широкий спектр наукових та інженерних задач, включаючи розробку генетичних алгоритмів, вирішення завдань в приватних похідних, цілочисельні проблеми, оптимізацію систем та інші);
- нейронні мережі (Neural Network Toolbox - інструменти для синтезу та аналізу нейронних мереж);
- нечітка логіка (Fuzzy Logic Toolbox - інструменти для побудови та аналізу нечітких множин);
- символічні обчислення (Symbolic Math Toolbox - інструменти для символічних обчислень з можливістю взаємодії з символічним процесором програми Maple).

Можна зробити висновок, що Matlab є програмним середовищем, яке вирішує величезний спектр проблем та допомагає реалізувати найрізноманітніші наукові рішення та ідеї. Є можливість підібрати будь-який набір функції, що буде краще підходити для проектного рішення, та водночас дасть можливість реалізувати його найбільш простим та зрозумілим способом, не витрачаючи багато часу та ресурсів [13]. За допомогою програмного середовища Matlab можна вирішити всі сучасні задачі, адже з такою великою кількістю наборів функціональних пакетів практично не залишається сфери чи наукової області, де не був би потрібен Matlab.

### 3.2 Розробка моделі розробленого засобу

Розробка нечіткої системи в програмному середовищі Matlab

					ДП.КСМ. 07097/14.00.00.000 ПЗ	34
Змн.	Арк.	№ докум.	Підпис	Дата		

виконується за допомогою вище описаного набору функції, що має назву Fuzzy Logic Toolbox. Fuzzy Logic Toolbox надає функції, додатки і Simulink - блоки для аналізу, проектування і моделювання систем на основі нечіткої логіки. Даний набір функцій в основному працює методами розробки виводів для нечітких систем. Функції надаються для багатьох загальних методів, включаючи нечіткі кластеризації та адаптивне нейрофузійне навчання. До ключових особливостей Fuzzy Logic Toolbox можна віднести наступні:

- програма Fuzzy Logic Design для побудови нечітких систем виведення та перегляду та аналізу результатів;
- функції належності для створення систем нечітких виводів;
- підтримка AND, OR, та NOT логіки в правилах, визначених користувачем;
- стандартні системи нечітких виводів Mamdani та Sugeno;
- формування автоматизованої функції належності за допомогою нейроадаптивних та нечітких технологій кластеризації;
- можливість вставляти нечіткі системи виводів в модель Simulink;
- можливість генерування вбудованого C-коду або автономного виконуваного двигуна нечіткого виведення.

Для побудови нечіткої системи потрібно для початку задати вхідні дані, а також кінцевий результат. На вході було задано наступні значення:

- температура;
- швидкодія;
- навантаженість.

На виході отримано загальний стан системи, який напряму залежить від значень вхідних даних. На рисунку 3.2 показано загальний вигляд розробленої нечіткої системи.

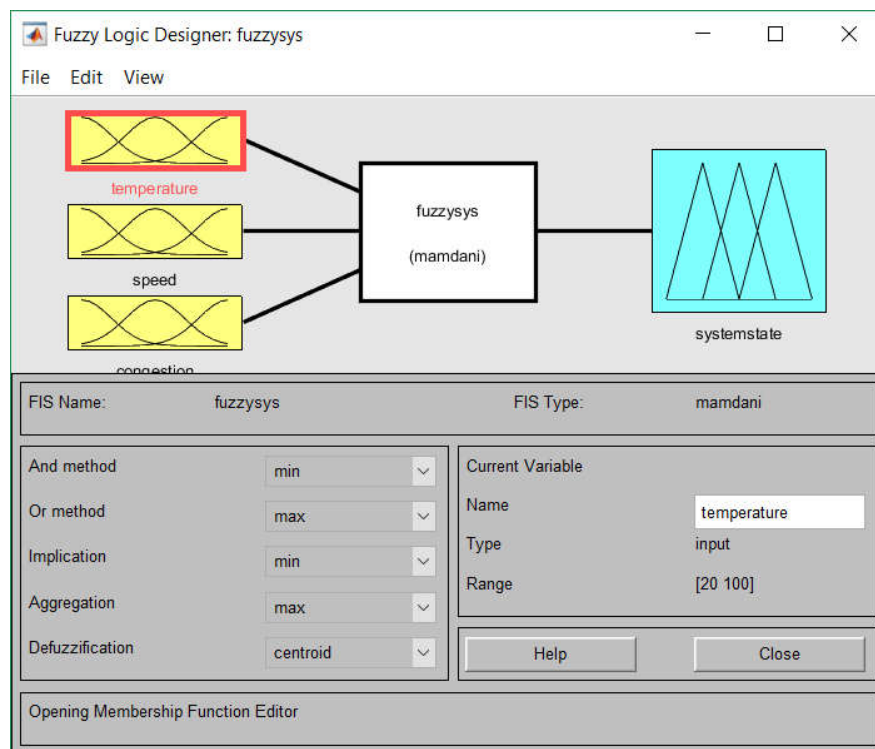


Рисунок 3.2 – Входи розробленої системи

Далі для кожного з входів, а також для виходу необхідно створити функції належності. Для входів було обрано функції типу “gbellmf”, адже вони дозволяють краще візуально показати зміну значень. Для виходу застосовуються функції типу “trimf”. Для подальшої роботи потрібно визначитись в яких межах значень будуть коливатись вхідні та вихідні дані. Проаналізувавши роботу комп’ютерних систем для входів було призначено межі, а саме для температури від 20°C до 100°C:

- до 35°C - температура в нормі;
- до 70°C - температура підвищена;
- більше 70°C – висока температура.

Для швидкодії від 0 до 1:

- від 0 до 0,5 - висока швидкодія;
- від 0,4 до 0,7 - нормальна швидкодія;
- від 0,6 до 1 - низька швидкодія.

Для навантаженості від 0 до 200:

- від 0 до 80 - низька навантаженість;
- від 70 до 140 - середня навантаженість;



- від 130 до 200 - висока навантаженість.

Для загального стану системи було умовно встановлено шкалу від 0 до 10:

- від 0 до 3 - відмінний стан;
- від 3 до 7 - нормальний стан;
- від 7 до 10 - поганий стан.

Функції приналежності температури, швидкодії, навантаженості, а також кінцевого стану системи зображені відповідно на рисунках 3.3 - 3.6.

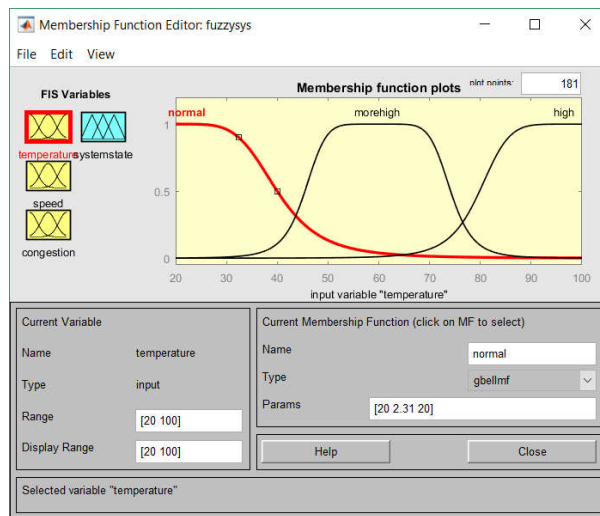


Рисунок 3.3 – Функції приналежності температури

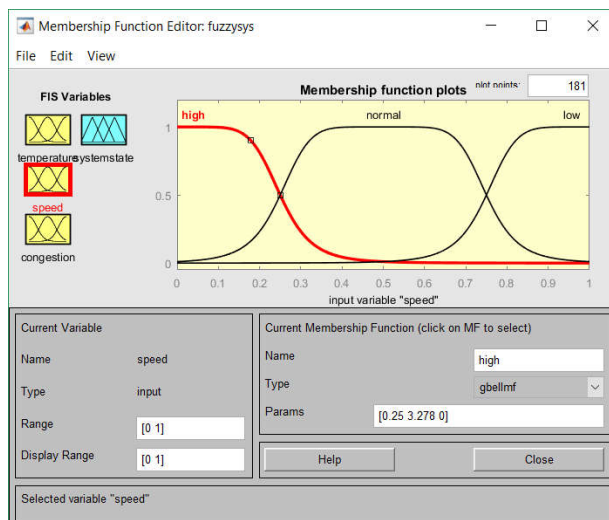


Рисунок 3.4 – Функції приналежності швидкодії



значення вхідної змінної не задане системою. Випадок, коли значення усіх вхідних змінних не задані, на практиці неможливий.

### 3.3 Верифікація роботи нечіткої системи

Для підтвердження працездатності нечіткої системи на рисунках 3.7 та 3.8 подані поверхні значень.

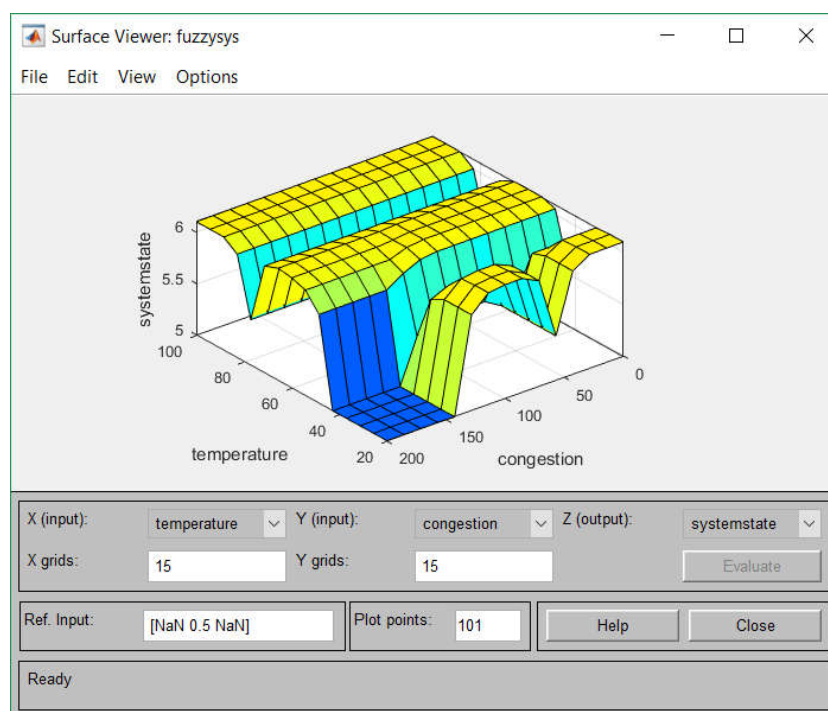


Рисунок 3.7 – Поверхня значень стану системи (systemstate) залежно від відношення температури(temperature) до навантаженості (congestion)

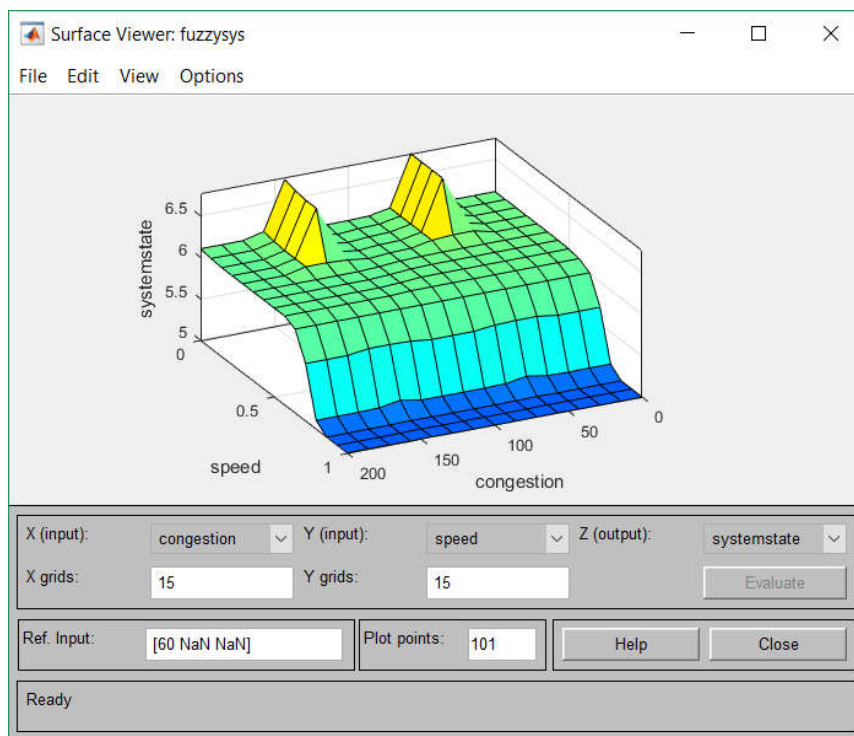


Рисунок 3.8 - Поверхня значень стану системи (systemstate) залежно від відношення швидкодії(speed) до навантаженості (congestion)

Нечіткий вивід моделі вибору стану комп'ютерної системи, побудованого на основі заданих 63 правил з поточними значеннями змінних temperature, speed, congestion та systemstate має вигляд, представлений на рисунку 3.9.

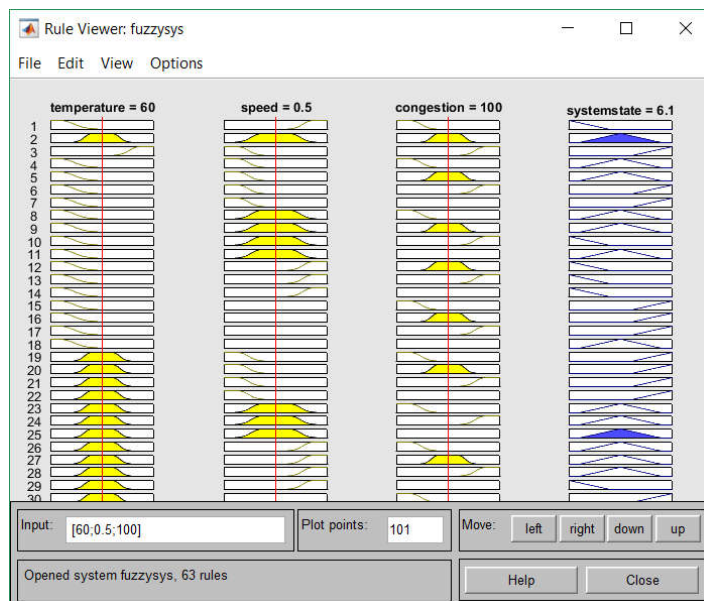


Рисунок 3.9 – Графічне зображення правил нечіткої системи

Також для підтвердження правильності роботи створеної нечіткої

системи, представлено таблицю даних з результатами моделювання (таблиця 3.1).

Таблиця 3.1 – Результати моделювання нечіткої системи

Вхідні значення			Вихідне значення systemstate
temperature	speed	congestion	
31.6	0.28	41.8	5.5
76.5	0.83	157	7
0	0.4	40	3.2
85	0	140	5.3
75	0.2	47	8
15	0.75	47	2.2
30.4	0.48	96	4.6
27.9	0.61	115.3	6.31
12	0.51	100.9	5.2
35	0.5	134.4	4.7

Отже, в результаті верифікації розробленої нечіткої системи, було переглянуто всі вхідні та вихідні дані і зроблено висновок, що система працює коректно, відповідно до встановлених вимог, а також до розробленої бази нечітких правил.

## 4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

У даному розділі дипломної роботи проводиться економічне обґрунтування доцільності розробки програмного засобу. Зокрема, здійснюється розрахунок витрат на розробку даного програмного забезпечення, експлуатаційних витрат, ціни на споживання проектного рішення, визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблений програмний засіб призначений для діагностики комп'ютерних систем.

### 4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів ( $K$ ) включають:

$$K = K_1 + K_2, \quad (4.1)$$

де  $K_1$  — витрати на розробку програмних засобів, грн;

$K_2$  — витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ( $B_{оп}$ );
- витрати на відрахування у спеціальні державні фонди ( $Bф$ );
- витрати на покупні вироби ( $Пв$ );
- витрати на придбання спецобладнання для проведення експериментальних робіт ( $Об$ );
- накладні витрати ( $H$ );
- інші витрати ( $Iв$ ).

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		42

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоемності відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломант; консультант техніко-економічного розділу.

Таблиця 4.1 — Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Місячний оклад, грн.	Коефіцієнт додаткової з/п
1	Керівник ДП, доцент	5286	0,94
2	Консультант техніко-економічного розділу, доцент	6026	1,47
3	Студент	1287	0

Витрати на оплату праці розробників проекту визначаються за формулою 4.2 :

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.2)$$

де  $n_{ij}$  — чисельність розробників  $i$ -ої спеціальності  $j$ -го тарифного розряду, осіб;

$t_{ij}$  —затрачений час на розробку проекту співробітником  $i$ -ої спеціальності  $j$ -го тарифного розряду, год;

$C_{ij}$  — годинна ставка працівника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою 4.3 :

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

де  $C_{ij}$  — основна місячна заробітна плата розробника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн.;

$h$  — коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$  - місячний фонд робочого часу працівника  $i$ -ої спеціальності  $j$ -го тарифного розряду, год. (приймаємо 168 год.).

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 — Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, доцент	20,5	61	1250,5
2	Консультант техніко-економічного розділу, доцент	2	88,6	177,2
3	Студент	240	7,7	1848
Разом				3275,7

Величну єдиного соціального внеску визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати, тобто

$$B_{\phi} = \frac{20,5}{100} \cdot 3275,7 = 671,5 \text{ грн.}$$

Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		44



забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 5,3 грн. Середній щоденний час роботи на комп'ютері — 2 години.

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 — Розрахунок витрат на матеріали та комплектуючі

Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн.	Транспортні витрати (10% від суми)	Загальна сума, грн
Папір (формат А4)	уп	80,0	2	160,00	16,0	176,0
Ручка кулькова	шт	4,0	2	8,00	0,8	8,80
Диски CD-R	шт	2,0	2	4,00	0,4	4,40
Зошит, 96 арк	шт	9,50	1	9,50	0,95	10,45
Тонер для принтера	уп	49	1	49	4,9	53,9
Ліцензія середовища розробки	шт	6000	1	6000	0	6000
Разом						6252,55

Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4 — Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
1	Проведення досліджень та оформлення їх результатів	60	318
2	Оформлення техніко-економічного розділу	8	42,4
3	Оформлення ДП	12	63,6
Разом		80	424

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати, тому  $H = 1,5 \cdot 3275,7 = 4913,5$  (грн.)

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:  $I = 3275,7 \cdot 0,1 = 327,6$  (грн.)

Витрати на розробку програмного забезпечення складають:

$$K_1 = V_{оп} + V_{ф} + V_{пв} + H + I$$

$$K_1 = 3275,7 + 671,5 + 6252,55 + 4913,5 + 327,6 = 15440,85 \text{ (грн.)}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою 4.4 :

$$K_2 = S_{м.з.} \cdot t_{від} \quad (4.4)$$

де  $S_{м.г.}$  — вартість однієї машино-години роботи ПК, грн./год.

$t_{від}$  — комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері — 2 години. Вартість години роботи комп'ютера дорівнює 5,3 грн. Тому  $K_2 = 5,3 \cdot 60 = 318$  грн.

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 4.5).

Таблиця 4.5 — Кошторис витрат на розробку програмного забезпечення

Найменування витрат	Сума витрат, грн.
Витрати на оплату праці	3275,7
Відрахування у спеціальні державні фонди	671,5
Витрати на куповані вироби	6252,55
Накладні витрати	4913,5
Інші витрати	327,6
Витрати на відлагодження і дослідну експлуатацію програмного продукту	318
Разом	15758,85

#### 4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.5)$$

де  $E_{\Pi}$  — одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$  — вартість підготовки даних для експлуатації ПЗ (аналогу), грн.;

$E_{2\Pi}$  — вартість роботи комп'ютера для виконання проектного рішення (аналогу), грн.

Річні експлуатаційні витрати  $B_{en}$  визначаються за формулою:

$$B_{en} = E_{\Pi} * N_{\Pi}, \quad (4.6)$$

де  $N_{\Pi}$  — періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою 4.7 :

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i c_i, \quad (4.7)$$

де  $i$  — категорії працівників, які приймають участь у підготовці даних;

$n_i$  — кількість працівників  $i$ -ої категорії, осіб;

$t_i$  — трудомісткість роботи співробітників  $i$ -ої категорії по підготовці даних, год.;

$c_i$  — середнього годинна ставка працівника  $i$ -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m}, \quad (4.8)$$

де  $b$  — коефіцієнт, який враховує додаткову заробітну плату (прийmemo 0,57);

$m$  — кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен

					ДП.КСМ. 07097/14.00.00.000 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		48

один працівник, основна місячна заробітна плата якого складає:  $c = 1287$  грн.  
Тоді:  $c_1 = \frac{1287(1+0,57)}{22*8} = 11,5$  грн/год. Трудомісткість підготовки даних для проектного рішення складає 3 год., для аналога 3,5 год. Результати представлені у таблиці 4.6.

Таблиця 4.6 — Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Проектне рішення		
3	11,5	34,5
Аналог		
3,5	11,5	40,25

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t * S_{MG}, \quad (4.9)$$

де  $t$  — витрати машинного часу для реалізації проектного рішення (аналогу), год.;

$S_{MG}$  — вартість однієї години роботи комп'ютера, грн./год.

$$E_{2\Pi} = 3 \cdot 5,3 = 15,9 \text{ грн.}; \quad E_{2a} = 3,5 \cdot 5,3 = 18,55 \text{ грн.};$$

$$E_{\Pi} = 34,5 + 15,9 = 50,4 \text{ грн.}; \quad E_a = 40,25 + 18,55 = 58,8 \text{ грн.};$$

$$B_{en} = 50,4 \cdot 252 = 12700,8 \text{ грн.}; \quad B_{ea} = 58,8 \cdot 252 = 14817,6 \text{ грн.}$$

### 4.3 Розрахунок ціни споживання проектного рішення

Ціна споживання — це витрати на придбання і експлуатацію проектного рішення за весь строк його служби:

$$Ц_{C(П)} = Ц_{П} + B_{(E)NPV}, \quad (4.10)$$

де  $Ц_{П} = K(1 + \frac{П_P}{100}) + K_0 + K_k$  — ціна придбання проектного рішення, грн.:

$K$  — кошторисна вартість;

$П_P$  — рентабельність;

$K_0$  — витрати на прив'язку та освоєння проектного рішення на конкретному об'єкті, грн.;

$K_k$  — витрати на доукомплектування технічних засобів на об'єкті, грн.

Отже,  $Ц_{П} = 15758,85 \cdot (1 + 0,3) = 20486,5$  грн.

Вартість витрат на експлуатацію проектного рішення (за весь час його експлуатації), грн.:

$$B_{епrv} = \sum_{t=1}^T \frac{B_{еп}}{(1+R)^t}, \quad (4.11)$$

де  $B_{еп}$  — річні експлуатаційні витрати, грн.;

$T$  — строк служби проектного рішення, років;

$R$  — річна ставка проценту банку.

Отже,  $B_{епrv} = \sum_{t=1}^3 \frac{12700,8}{(1+0,08)^t} = 35280$  грн;  $B_{епrv} = \sum_{t=1}^3 \frac{14817,6}{(1+0,08)^t} = 41160$  грн.

Тоді ціна споживання проектного рішення дорівнюватиме:

$$Ц_{сн} = 20486,5 + 35280 = 55766,5 \text{ грн.}$$

Аналогічно визначається ціна споживання для аналогу:

					ДП.КСМ. 07097/14.00.00.000 ПЗ	50
Змн.	Арк.	№ докум.	Підпис	Дата		

$$Ц_{ca} = 17352,6 + 41160 = 58512,6 \text{ грн.}$$

#### 4.4 Визначення показників економічної ефективності

Економічний ефект в сфері проектування рішення:

$$E_{пп} = Ц_{п} - Ц_{а} = 58512,6 - 55766,5 = 2746,1 \text{ грн.}$$

Річний економічний ефект в сфері експлуатації:

$$E_{кc} = B_{EA} - B_{EP} = 14817,6 - 12700,8 = 2116,8 \text{ грн.}$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{екc} = \sum_{t=1}^T E_{екc} (1 + R)^{T-t} \text{ грн.}$$

$$\Delta E_{екc} = \sum_{t=1}^3 2116,8 (1 + 0,08)^{3-t} = 7407,1 \text{ грн.}$$

Сумарний ефект складає:

$$E = E_{пп} + \Delta E_{екc} = 10153,2 \text{ грн.}$$

Результати усіх здійснених вище розрахунків представлені в таблиці 4.7.

					ДП.КСМ. 07097/14.00.00.000 ПЗ	51
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.7 — Показники економічної ефективності проектного рішення

Найменування	Одиниці вимірювання	Значення показників	
		Базовий варіант	Новий варіант
Капітальні вкладення	грн.	-	15758,85
Ціна придбання	грн.	17352,7	20486,5
Річні експлуатаційні витрати	грн.	14817,6	12700,8
Ціна споживання	грн.	58512,6	55766,5
Економічний ефект в сфері проектування	грн.	-	2746,1
Економічний ефект в сфері експлуатації	грн.	-	2116,8
Додатковий ефект в сфері експлуатації	грн.	-	7407,1
Сумарний ефект	грн.	10153,2	

Отже, у даному розділі проведено розрахунок витрат на розробку програмного засобу, оплати праці, розрахунок витрат на використання комп'ютерної техніки, а також кошторис витрат на розробку даного програмного забезпечення. Отже, згідно проведеного економічного обґрунтування дане проектно рішення є конкурентноздатним. Отримано економічний ефект у розмірі 10153,2 грн. і тому розробка і впровадження цього проектного рішення є економічно доцільними.



## ВИСНОВКИ

У процесі виконання дипломного проекту, було розроблено нечітку систему діагностики комп'ютерних систем.

Протягом розробки вирішено такі основні завдання:

- 1) проаналізовано проблеми, які виникають при користуванні комп'ютером та в загальному в комп'ютерних системах;
- 2) досліджено можливі алгоритми нечіткого виводу по базі знань і обрано для використання алгоритм Мамдані;
- 3) побудовано функції належності вхідних та вихідного значень;
- 4) сформовано базу правил нечіткої системи;
- 5) проаналізовано роботу нечіткої системи за допомогою поверхонь значень нечіткої системи;
- 6) розроблено структурну схему нечіткої системи, що дозволяє краще зрозуміти принципи роботи;
- 7) зроблено висновки щодо правильності роботи нечіткої системи.

Використання розробленої нечіткої системи діагностики роботи комп'ютерних систем, дозволить зменшити шанси виникнення проблем шляхом їх попередження та вчасної діагностики, що дозволить значно зекономити час та ресурси.

					ДП.КСМ. 07097/14.00.00.000 ПЗ	53
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Функції обробки нечітких даних. Щоденник: веб-сайт. URL: <http://company.shodennik.ua/functions>(дата звернення: 11.11.2018).
2. Novak V., Perfilieva I., Mockor J. Mathematical principles of fuzzy logic: Kluwer Academic Publishers,1999. P. 15.
3. Stanford academic informations. Plato: web-site. URL: <https://plato.stanford.edu/entries/logic-fuzzy>(дата звернення: 22.03.2019).
4. Cintula P. Fuzzy Logics as the Logics of Chains: Fuzzy Sets and Systems. Libor, 2006. P. 606.
5. Godo L. Fuzzy Sets and Systems. Monoidal T-Norm Based Logic: Towards a Logic for Left-Continuous T-Norms. Waweland, 2001. P. 25.
6. What Is Fuzzy Logic? Mathworks: web-site. URL: <https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html>(дата звернення: 05.06.2019).
7. Zadeh L. Real-Life Applications of Fuzzy Logic. Fuzzy logic now and then. Hindawi, 2013. P. 125.
8. Yager R. Fuzzy Sets and Applications. Introduction. Wiley, 1987. P. 8.
9. Encoder the Newsletter of the Seattle Robotics Society (Fuzzy logic – an introduction). Seatlerobotic: web-site. URL: <http://www.seattlerobotics.org/encoder/mar98/part2.html>(дата звернення: 23.01.2019).
10. Zimmerman H. Fuzzy set theory and its applications. Fuzzy logic intoduction. Kluwer, 1991. P. 315.
11. Блюмин С., Шуйкова И., Сараев П. Нечеткая логика: алгебраические основы и приложения: монография. Киев, ЛЭГИ, 2002. 113 с.
12. Cordon O., Herrera F. A General study on genetic fuzzy systems. Genetic Algorithms in computer science. Tante, 1995. P. 33.

13. Леоненков А. Нечеткое моделирование в MATLAB и fuzzyTECH. Санкт–Петербург: БХВ–Петербург, 2005. 736 с.
14. Abadeh M., Habibi J., Lucas C. Intrusion Detection Using a Fuzzy Genetics–Based Learning Algorithm. Journal of Network and Computer Applications. 2007. No.30. P. 414–418.
15. Mamdani E. Application of fuzzy algorithms for the control of a simple dynamic plant. Proc. IEEE 121, 1974. P. 1585–1588.
16. Штовба С. Обеспечение точности и прозрачности нечеткой модели Мамдани при обучении по экспериментальным данным. Харьков, 2007. С. 102–104.
17. Дубчак Л. Метод обробки нечітких даних на основі механізму Мамдані. Системи обробки інформації. 2012. Вип. 7(105). 131с.
18. Ротштейн А., Штовба С. Идентификация нелинейной зависимости нечеткой базой знаний с нечеткой обучающей выборкой. Кибернетика и системный анализ. 2006. Вып.2. С. 17–24.
19. Мирончук Ю., Куріненко О. Побудова функцій належності нечітких множин, які відповідають кількісним експертним оцінкам фізичних величин. Системи обробки інформації. 2017. Вип.1. С. 93–97.
20. Лозинський А., Демків Л. Дослідження впливу вигляду функції належності на динамічні показники системи при багатокритеріальній оптимізації зі змінними ваговими коефіцієнтами. Електротехнічні та комп'ютерні системи. 2012. Вип.5. С. 137–144.
21. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва: Горячая линия – Телеком, 2004. 452 с.
22. Лазарев Ю. Моделювання динамічних систем у Matlab. Київ: НТУУ «КПІ», 2011. 421 с.
23. Passino K., Yurkovich S. Fuzzy Control. California: Addison–Wesley, 2001. 53 P.
24. Koo T. Analysis of a Class of Fuzzy Controllers, in Proc. 1st Asian Fuzzy Systems Sump. Singapore: Way, 1998. P. 35–38.

25. Iancu I. Extended Mamdani Fuzzy Logic Controller. California: ACTA Press, 2001. P. 143–149.

26. Simulation and Model-Based Design. Mathworks: web-site. URL: <https://www.mathworks.com/products/simulink.html>(дата звернення: 17.05.2019).

27. Cintula P. Fuzzy Logics as the Logics of Chains. Fuzzy Sets and Systems. Melburn: Libor, 2006. P. 606.

28. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напрямку підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О. М. Березький, Л. О. Дубчак, Р. Б. Трембач, Г. М. Мельник, Ю.М. Батько, С. В. Івасьєв / Під ред. О. М. Березького. Тернопіль: ТНЕУ, 2014.–65 с.

29. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» напрямку підготовки 6.050102 комп’ютерна інженерія/ І. Р. Паздрій – Тернопіль: ТНЕУ, 2014. – 37 с.

					ДП.КСМ. 07097/14.00.00.000 ПЗ	56
Змн.	Арк.	№ докум.	Підпис	Дата		