

АНОТАЦІЯ

Тема даної дипломної роботи – «Нечітка система управління клімат - контролем фітнес - центру». Дана система призначена для автоматизованого збору й контролю значень температури й вологості повітря в приміщеннях фітнес -центру. Система повинна мати властивість інтелектуальності тобто видавати поради оператором при дистанційному керуванні обладнанням клімат-контролю у залах центру. Створення й впровадження такої системи є актуальним тому що інтелектуальний контроль і регулювання підвищать надійність й економічність роботи електричного обладнання та забезпечить комфортні тренування.

Дипломна робота складається з декількох розділів, які відповідають етапам розробки системи управління. Першим є вступна частина, у якій розглядається предметна область роботи. На підставі вивчених матеріалів були визначені основні функції й характеристики системи, обрані науковий напрямок і середовище програмування (FuzzyTECH). Розробка нечіткої системи клімату-контролю починається з етапу визначення основних вимог до системи. Послідовно була побудована нечітка модель, що імітує дії досвідченого оператора або експерта на основі дистанційного керування. На основі методів нечіткої логіки створені експертна система та програма для контролера у автоматичної системи управління. Розроблена система діагностики обладнання потужного кондиціонера. Система діагностики реалізує методи нечітких продукційних експертних систем.

Застосування системи управління температурою повітря допомагає керівникам фітнес-центрів знизити витрати на управління та технічне обслуговування обладнання систем кондиціонування, покращити роботу з клієнтами, та забезпечити безперебійну роботу центру.

ABSTRACT

The topic of this diploma thesis is "Unclear system of climate control - control of the fitness center". This system is intended for automated collection and control of temperature and humidity values in the premises of the fitness center. The system should have the property of intelligence, that is, give advice to the operator during remote control of the climate control equipment in the halls of the center. The creation and implementation of such a system is relevant because intelligent control and regulation will increase the reliability and economy of electrical equipment and provide comfortable training.

The thesis consists of several sections that correspond to the stages of development of the management system. The first is the introductory part, in which the subject area of the work is considered. Based on the studied materials, the main functions and characteristics of the system were determined, the scientific direction and the programming environment (FuzzyTECH) were chosen. The development of a fuzzy climate control system begins with the stage of determining the basic requirements for the system. A fuzzy model simulating the actions of an experienced operator or expert based on remote control was built sequentially. An expert system and a program for a controller in an automatic control system were created on the basis of fuzzy logic methods. A system for diagnosing the equipment of a powerful air conditioner has been developed. The diagnostic system implements methods of fuzzy production expert systems.

The application of the air temperature management system helps the managers of fitness centers to reduce the costs of management and maintenance of the equipment of air conditioning systems, to improve the work with customers, and to ensure the uninterrupted operation of the center.

ЗМІСТ

| | |
|---|-----------|
| ВСТУП..... | 5 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 7 |
| 1.1 . Аналіз ключових факторів та процесів, що забезпечують стабільне протікання клімат – контролю у фітнес-центрі..... | 7 |
| 1.2 Опис системи клімат-контролю температурних зон фітнес-центру | 9 |
| 1.3 Вибір наукового напрямку для вирішення поставлених завдань | 13 |
| 2 НЕЧІТКАЯ СИСТЕМА УПРАВЛІННЯ КЛІМАТОМ..... | 19 |
| 2.1 Розробка нечіткої системи регулювання температурою та вологістю..... | 19 |
| 2.2. Розробка програми для контролера системі управління..... | 26 |
| 2.3. Розробка експертної системи для підтримки прийняття рішення оператора під час управління системою припливної вентиляції | 36 |
| 3. СИСТЕМА ДІАГНОСТИКИ ОБЛАДНАННЯ КОНДИЦІОНЕРІВ..... | 48 |
| 3.1 Аналіз можливих причин несправності обладнання спліт - системи..... | 48 |
| 3.2 Створення експертної системи для пошуку причин несправності ... | 50 |
| ВИСНОВКИ..... | 57 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 58 |
| Додатки..... | 6 |

ВСТУП

Інтелектуальна система - це система діючій основі теорії штучного інтелекту яка реалізує комплекс програмних, лінгвістичних і логіко-математичних засобів, для реалізації завдань збору, аналіз і керування інформаційними потоками з метою здійснення підтримки діяльності людини в різні науково - технічних галузях [1]. Історія штучного інтелекту як нового наукового напрямку починається в середині ХХ століття. В 1950 році один з піонерів в області обчислювальної техніки, англійський вчений Алан Тьюрінг, пише статтю за назвою «Чи може машина мислити?», у якій приводить свої відповіді на подібні питання та описує процедуру, за допомогою якої можна буде визначити момент, коли машина зрівняється в плані розумності з людиною, що одержала назву тесту Тьюрінга [2].

У цей час в області теорії штучного інтелекту спостерігається розвиток практично всіх аспектів управління складними соціально-технічними об'єктами. Ця тенденція стосується алгоритмів управління, структури та складу технічних засобів, обсягу й розподілу функцій контролю, засобів подання інформації, математичних моделей об'єктів. Як відомо, сучасна кібернетика пройшла шлях від дослідження простих об'єктів і локальних систем до вивчення питань оптимального управління, адаптації та самоорганізації складних технічних систем. Розглянуті системи складаються з багатоланкової сукупності об'єктів змінної структури та характеризуються багатомірністю й невизначеністю фізичних явищ, що протікають у них. Важливим елементом, при вирішенні складних проблемних завдань, є новий напрямок - інтелектуальне управління на основі комп'ютерних систем [2].

Сучасне виробництво характеризується стійкою тенденцією до впровадження спеціальних керуючих обчислювальних комплексів, на базі промислових комп'ютерів, що здатні управляти складними технологічними процесами без участі людини.

Система кліматичного контролю призначена для створення комфортних умов проживання. Інтелектуальний клімат – контроль забезпечує єдине управління та узгоджену роботу систем опалення, вентиляції та кондиціонування. В результаті людина отримує необхідні значення параметрів температури, вологості та припливу свіжого повітря, що максимально сприятливо впливають на організм, що дозволяє продуктивно працювати і повноцінно відпочивати.

З огляду на те, що фітнес - центри є важливими громадськими установами для підвищення рівня здоров'я людини, впровадження інтелектуальної системи клімат-контролю є актуальним завданням.

Впровадження сучасних програмно – технічних засобів та засобів автоматизації дозволяє знизити витрати на опалення та енергозбереження у межах 30 – 40 % [3, 5]. За бажанням адміністрації медичного центру інтелектуальна система автоматично здатна регулювати температуру повітря у приміщеннях та будинках залежно від дня тижня та часу доби. Знижуючи температуру в адміністративних будинках у нічний час та вихідні дні. Такий режим є енергозберігаючим. Також інтелектуальна система здатна робити: видачу повідомлень оперативному персоналу про позаштатні чи аварійні ситуації. Здійснювати збереження та архівація даних про хід технологічного процесу в СУБД, а також здійснювати діагностику роботи обладнання.

Застосування системи управління кліматом допомагає керівникам фітнес-центрів знизити витрати на управління, технічне обслуговування та забезпечити безперебійну роботу центру.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1. 1. Аналіз ключових факторів та процесів, що забезпечують стабільне протікання клімат – контролю у фітнес-центрі

Основою дипломного проектування є етап постановки завдання та опису вимог до об'єкта дослідження, визначення технічних та програмних засобів, які будуть використовуватись при розробці. Предметна область вивчається з допомогою спеціальної літератури, у процесі проходження переддипломної практики, а як і, з допомогою консультацій з експертами з'ясування основних аспектів та характеристик об'єкта проектування, тобто. з людьми, які працюють у цій сфері та знайомими з її основними аспектами.

У процесі вивчення предметної галузі розробки інтелектуальної системи клімат – контролю фітнес – центру проводилися такі роботи:

- а) визначення основних санітарно-технічних параметрів, що впливають на роботу фітнес-центру;
- б) дослідження існуючих систем клімат – контролю діючих на основі технології «розумний будінок»;
- в) визначення основних вимог, що пред'являються до інтелектуальної системи збору та аналізу інформації;
- г) вивчення режимів роботи технологічного обладнання, обов'язків та інструкцій операторів системи управління та технічного обслуговування (осіб, що приймають рішення).

При вивченні спеціалізованої літератури та роботи систем кліматичного контролю було виділено такі основні параметри [1]:

Температура та вологість, швидкість повітря у приміщеннях фітнес – центру.

Температура та вологість повітря у приміщенні є найважливішими параметрами, що визначають стан комфорту всередині приміщення. Організм людини постійно виділяє теплоту залежно від фізичної активності,

так спокійна доросла людина виділяє в середньому близько 80 Вт, а при великих фізичних зусиллях вже 300 Вт. Ця теплота повинна відводитися від людини, щоб запобігти перегріву. Відводиться це тепло головним чином шляхом теплообміну з навколишнім повітрям, тому крім одягу важливим показником теплового комфорту для людини є температура навколишнього повітря. Рекомендовані значення температури повітря в приміщенні за різними стандартами знаходяться в межах 20-22 С і 22-26 С. Ще один фізичний параметр внутрішньої атмосфери, що безпосередньо впливає на теплообмін організму людини - це вологість повітря, що характеризує насиченість водяними парами. Так недолік вологості, менше 20% відносної вологості, призводить до пересихання слизових оболонок, що викликає кашель. А перевищення рівня вологості, понад 65%, призводить до погіршення тепловіддачі при випаровуванні поту, виникає почуття ядухи. Тому температура повинна співвідноситися з рівнем вологості. Ще одним фактором, що впливає на тепловий обмін людського організму з навколишнім повітрям, є швидкість руху повітря.

Швидкість повітря визначається робочої зоні приміщення, тобто. там, де перебувають люди, саме у просторі від 0,15м. від підлоги до 1,8 м за висотою та на відстані не менше 0,15 м від стін. Швидкість повітря у робочій зоні рекомендується в межах 0,13-0,25 м/с. При меншій швидкості - душно або навіть спекотно, при більшій - наявність протягів, допускати який має сенс лише за підвищення температури нормативних значень.

1.2. Опис системи клімат-контролю температурних зон фітнес-центру

У фітнес-центрі працює та оздоровлюється багато людей, яким потрібен комфортний мікроклімат. До того ж, якщо обладнання виділяє велику кількість тепла, перебувати в приміщенні без охолодження кондиціонером стає небезпечно для здоров'я. Для того щоб система кондиціонування справлялася зі своїм завданням, використовується різні типи кондиціонерів, які підходять для конкретного приміщення (рис.1.1.).

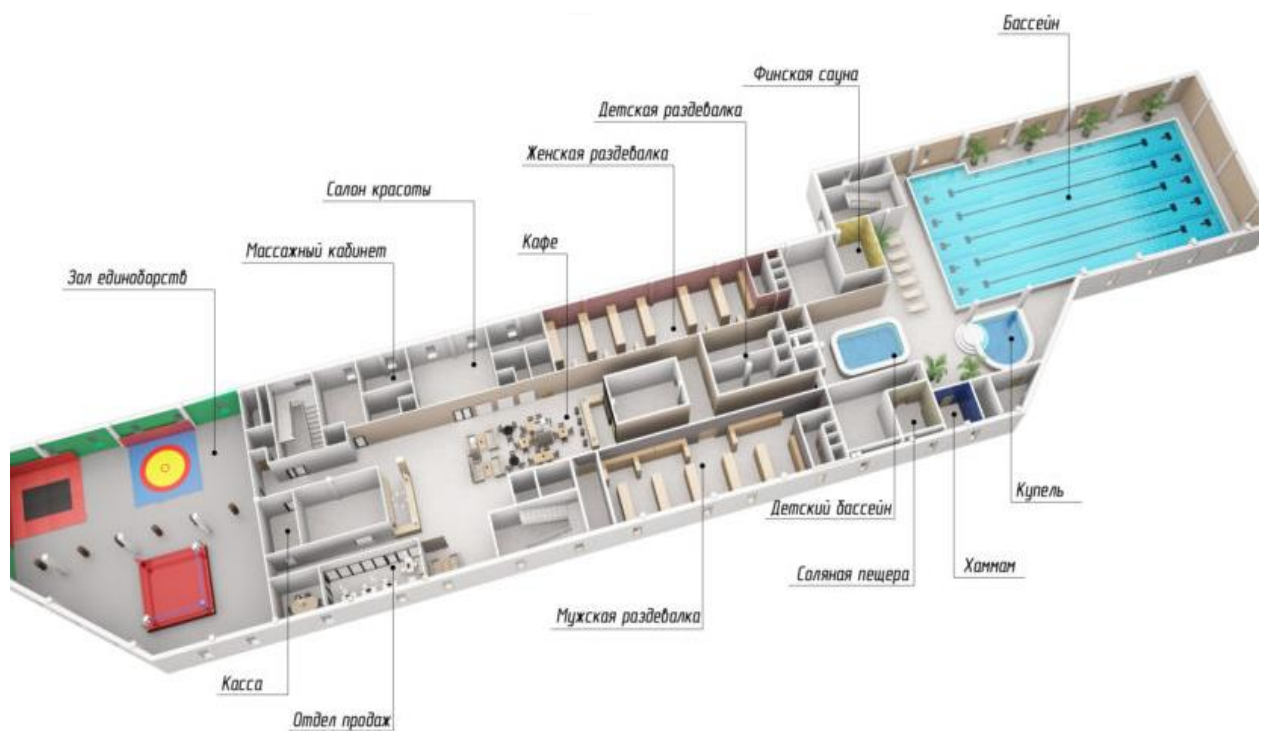


Рисунок 1.1 – План температурних зон фітнес-центру

У більшості приміщень застосовуються мультizonні спліт – системи [2]. Це великі мультizonальні системи кондиціонування з необхідною витратою повітря для виробництва, які складаються з одного зовнішнього блоку та великої кількості внутрішніх – це ідеально для будівель з кількома приміщеннями. Для кожного внутрішнього блоку можуть бути задані свої

параметри, наприклад, для офісів, магазинів, ресторанів, спортивних залів тощо. які знаходяться в одному торговельно-розважальному комплексі (рис.1.2).

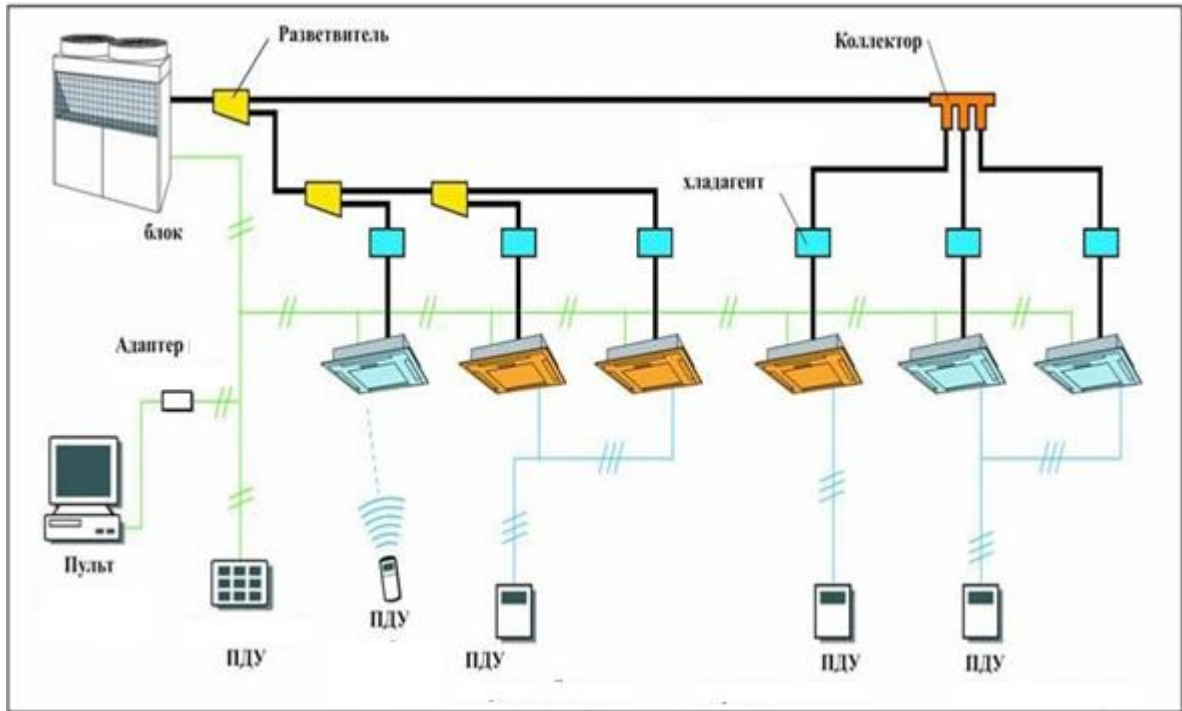


Рисунок 1.2 –Схема мультizonної спліт-системи

Така система також заощаджує на ремонті та обслуговуванні кондиціонерів – за потреби можна замінити тільки той блок, у якому є несправність або якщо приміщення використовується з іншою метою та вимоги до кондиціонування змінилися. Суть такого промислового охолодження у тому, що завдання зовнішнього блоку – чиллера полягає у охолодженні рідини – хладагента. Потім рідина прямує до другого елементу – фанкойлу. Прилад втягує тепле повітря, очищає його, охолоджує і знову поширює. Чилери здатні замінити 2 інженерні системи відразу: опалювальну та кліматичну. Також у мережу можна інтегрувати вентиляцію та якісно покращити параметри мікроклімату. «Чіллер-фанкойли» мають необмежені можливості масштабування та працюють із будь-якою конфігурацією внутрішніх блоків. Завдання чиллера - виробляти холодоагент (охолоджену рідину) оптимальної температури та доставляти її до інших агрегатів.

Приймаючим пристроєм є фанкойл. Прилад монтується у приміщенні та зовні нічим не відрізняється від внутрішнього блоку кондиціонера. Цей процес протікає плавно і синхронно: окремі компоненти працюють одночасно. Агрегати оснащені термостатом та електронними елементами, які дозволяють контролювати як режими роботи пристрою, так і всю установку загалом. Для ефективного контролю значень параметрів температури та вологості необхідно впровадження комп'ютерної системи збору інформації (рис.1.3.). Виміряні дані від датчиків через контролери надходять на персональний комп'ютер оператора. Оператор спостерігає значення контрольованих параметрів у приміщеннях та палатах установи та може також проводити дистанційне керування обладнанням систем вентиляції та кондиціонування. Також оператор виконує функції із супроводу СУБД обладнання установи та адміністрування Web – сайту організації.

На рис. 1.3 система збору даних RNet по однойменному протоколу здійснює опитування стану регуляторів і робить архівування даних, також програма RNet робить опитування не тільки регуляторів, але й підключених у цю ж мережу модулів вводу-виводу. Зв'язок здійснюється також по протоколу RNet. Модулі введення-виводу в цьому випадку можуть використовуватися тільки як пристрої видаленого доступу. Контролери одержують дискретні й аналогові сигнали від датчиків, попередньо обробляють і передають їх у комп'ютер по шині RS-485 у систему RNet для реєстрації і відображення на автоматизированих робочих местах (АРМ) операторів. Зв'язок промислових комп'ютерів один з одним здійснюється за допомогою технології Ethernet і протоколу TCP/IP. У цей час на вітчизняних підприємствах функціонує велика кількість контролерів як імпортованих, так і українського виробництва, що дозволяє будувати розподілені системи комп'ютерного керування.

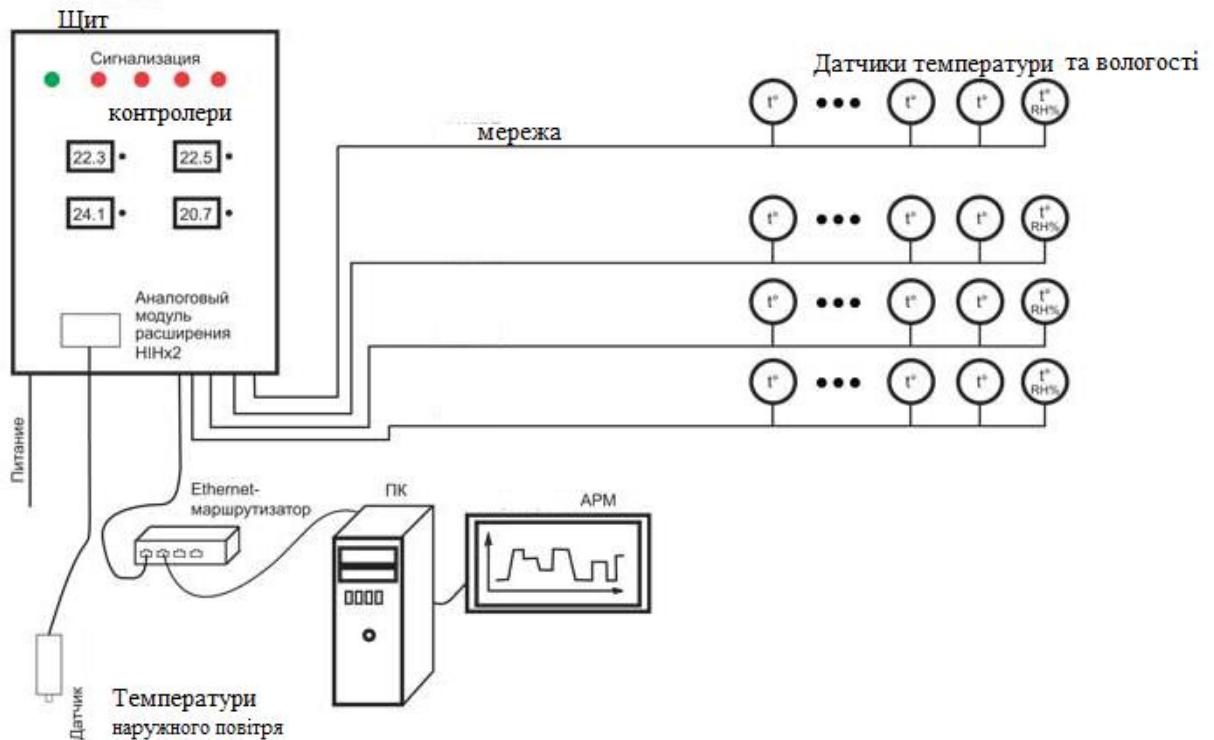


Рисунок 1.3. - Розподілена система моніторингу та управління параметрами клімат-контролю у зонах фітнес-центру

При побудові архітектури системи мають бути закладені такі характеристики системи управління [3]:

- тестування (можливість встановлення факту вірного функціонування);
- діагностування (можливість знаходження несправної частини системи);
- ремонтпридатність (можливість відновлення працездатності за мінімальний час за економічно виправданої вартості ремонту);
- надійність (наприклад, шляхом резервування);

Економічна вигода: персонал своєчасно реагує на аварійні ситуації та усуває їх. У свою чергу, це дозволяє зберегти працездатність обладнання та заощадити на його ремонті.

- Деякі ситуації усуваються системою без залучення персоналу. Це дозволяє працівникам повністю зосередитись на виконанні своїх обов'язків.
- Рациональне споживання електроенергії. Це досягається за рахунок оптимальної роботи чилерів, фанкойлів, припливно-витяжних установок та іншого обладнання.

- Точний контроль споживаних енергоресурсів. Завдяки цьому підприємство ретельніше відстежує свої витрати.
- Облік подій, що відбуваються з вентиляційною системою та виробничим обладнанням.
- Ретельне планування профілактичних робіт інженерних комунікацій.
- Оперативне співробітництво різних інженерних служб.

У представленій системі використовуються ранні кліматичні системи (фанклойли - спліт системи в приміщеннях спортивних залів, а також припливно-витяжна вентиляція в зонах басейнів і кафе-барів, для окремих офісів адміністрації та масажного салону передбачені локальні побутові кондиціонери) (рис. 1.4.).

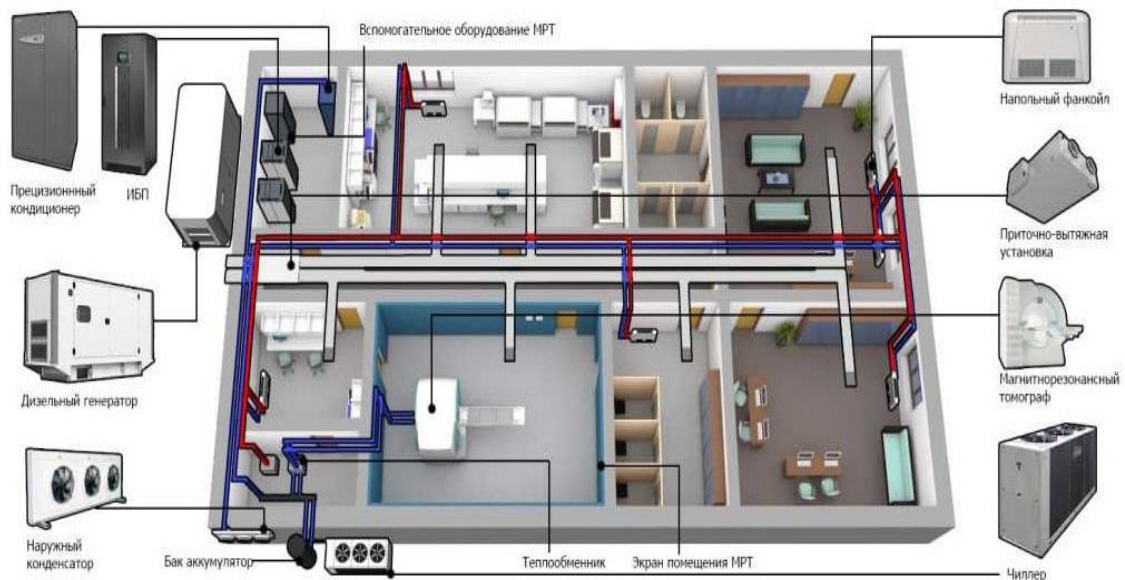


Рисунок 1.4 – План встановлення обладнання

Таким чином, для успішного впровадження системи клімат-контролю фітнес-центру необхідно вирішити такі завдання:

1. Розробити систему управління кондиціонування повітря з написанням програми для контролера у програмі FuzzyTECH;
2. створити експертну систему підтримки прийняття рішень оператора під час дистанційного керування системою вентиляції

3. Розробити систему діагностики обладнання кондиціонерів

1.3. Вибір наукового напрямку для вирішення поставлених завдань

Останнім часом на хвилі інтересу до технологій штучного інтелекту в системах комп'ютерного керування всі частіше стали знаходити застосування так називані нечіткі алгоритми регулювання. Регулятори, побудовані на основі цієї інноваційної концепції, у ряді випадків здатні забезпечити більше високі показники якості перехідних процесів у порівнянні із класичними регуляторами. Крім того, використовуючи технологію нечітких алгоритмів керування, можливо, провести оптимізацію процесів без проведення складних математичних досліджень.

Відомо, що експертів - операторів зручніше описувати властивості об'єкта і його динаміку в термінах, застосовуваних у контексті звичайної мови. Звідси треба, що для складання "словесної" моделі доцільно застосовувати апарат нечітких безлічей, що оперує лінгвістичними змінними.

Засновником теорії нечітких безлічей є американський учений Л.А. Заді. У середині 60 р.г. минулого століття у своїх працях [4, 5,6] він увів термін "Fuzzy sets" який переводиться як "нечіткі, або розмиті, безлічі". Ці безлічі призначені для опису й дослідження складних, погано певних, або "нечітких", систем. Новизна ідеї полягала в тому, що "...складність систем вимагає підходу, у корені відмінного від загальноприйнятих кількісних методів аналізу систем і для моделювання поведінки таких систем може виявитися необхідним трохи скоротити використання кількісних методів і застосовувати замість цього лінгвістичний підхід, відповідно до якого як значення змінних допускаються не тільки числа, але й слова або пропозиції природної або штучної мови" [5].

Даний підхід дозволяє розширити сферу додатка методів кібернетизації за межі застосовності класичної теорії. Точка зору Л.Заді така: " Я вважаю, що зайве прагнення до точності стало робити дію, що зводить на немає теорію керування й теорію систем, тому що воно приводить до того, що дослідження

в цій області зосереджують на тих і тільки тих проблем, які піддаються точному рішенню. У результаті, багато класів важливих проблем, у яких дані, мети й обмеження є занадто складними або погано певними для того, щоб допустити точний математичний аналіз, залишалися й залишаються осторонь із тієї причини, що вони не піддаються математичному трактуванню. Для того, щоб сказати що - або істотне для проблем подібного роду, ми повинні відмовитися від наших вимог точності й допустити результати, які є трохи розмитими або невизначеними" [9].

Таким чином, науковий напрямок, заснований на теорії нечітких безлічей, можна вважати перспективним для рішення поставлених завдань по розробці систем керування й експертної системи підтримки ухвалення рішення. Поняття нечіткої безлічі, уведене Л. Заді [10], ґрунтується на припущенні про те, що будь-який елемент лише до деякої міри належить даній безлічі, тому одним з основних способів математичного опису нечіткої безлічі є визначення ступеня такої приналежності деяким числом з інтервалу $[0, 1]$. При цьому границі інтервалу 1 й 0 означають, відповідно, «належить» й «не належить». Нехай E – універсальна безліч, x – елемент множини E , а P – деяка властивість. Звичайне (чітке) підмножина A універсальна безліч E , елементи якого задовольняють властивості P , визначається як безліч упорядкованих пар $A = \{\mu_A(x) / x\}$, де $\mu_A(x)$ – характеристична функція приналежності, що приймає значення 1, якщо x задовольняє властивості P , і 0 – у протилежному випадку. Нечітка підмножина відрізняється від звичайного тем, що для елементів x з E немає однозначної відповіді «так/ні» щодо властивості P .

Лінгвістичною називають змінну, значеннями якої є слова або пропозиції природної або штучної мови. Як базовий приклад розглядається фізична величина – «температура теплоносія», що описує оператор при спостереженні за температурою води, по термометрі, що надходить від котельні до споживача. У цьому випадку описувана величина є лінгвістичною змінною (ЛП). Сукупність значень ЛП яке описує експерт становить терм – безліч (ТМ). Так, наприклад, якщо температура лежить у діапазоні «від те 10 до 30 °С»

оператор говорить, що „температура теплоносія низька”. У випадку, коли температура перебуває в діапазоні «30 – 90 °С» експерт затверджує, що „температура теплоносія середня”. Якщо температура теплоносія розташована в інтервалі «90 – 120 °С», то лінгвістична змінна приймає позначення „” (рис.1.5). На рис.1.5. числова змінна температура, приймає значення від 10 до 120 С°.

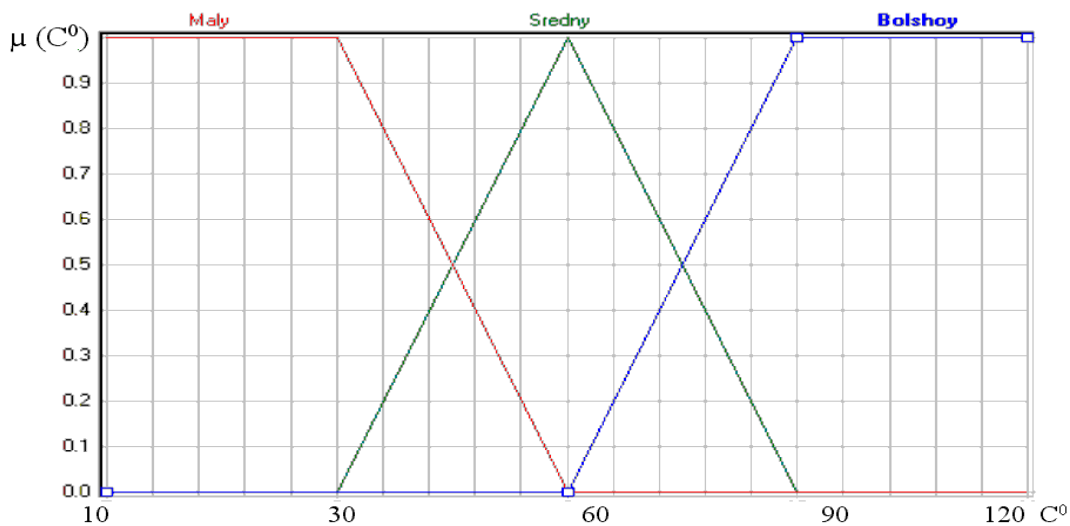


Рисунок - 1.5. Графіки лінійної Z образної трикутної й S образної функцій належностей лінгвістичної змінної “температура теплоносія ”

На рисунку 1.5. №1 - Z – образна, №2 – трикутна, №3 - S – образна функції приналежності. Перша функція описує терм “низька температура”, друга “середня”, третя - “висока”.

Перша із цих функцій у загальному випадку може бути задана аналітично наступним вираженням:

$$\mu_1(x, a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & b \leq x \end{cases}, \quad (1.1)$$

де a, b – числові параметри, що приймають дійсні значення й упорядковані відношенням: $a < b$. Для терм – безлічі «низька» й універсума $X = [0, 120]$ значення параметрів відповідно рівні $a = 30, b = 60$.

Аналітичне вираження «трикутної» функції приналежності записується в наступному виді:

$$\mu_2(x, a, b, c) = \left\{ \begin{array}{ll} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c < x \end{array} \right\}. \quad (1.2)$$

Для трикутної функції приналежності терм – «середня», значення з універсума, що перебуває в інтервалі від 80, до 100, для $a=30$; $b=60$; $c=90$.

Третя S – образна функція приналежності задається наступним вираженням:

$$\mu_3(x, a) = \left\{ \begin{array}{ll} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \end{array} \right\}. \quad (1.3)$$

Значення параметрів, відповідно, також рівні, $a=30$, $b=60$.

У випадку, представленим на рис. 1.5, використався прямий метод визначення функції приналежності, коли експерт задає значення $\mu(x)$ для кожної підмножини. Як правило, прямі методи завдання $\mu(x)$ використовуються при вимірі фізичних величин. У випадку утруднення оцінки застосовується груповий метод, або статистичний підхід, при якому вид функції приналежності і її значень визначаються усередненням оцінок $\mu(x)$, що задають різними експертами. Існують й інші методи оцінок [5].

Терм - безліч або нечітка безліч може представлятися й за допомогою типових форм (трикутної, гауссовської, і т.д.) [6].

Нечіткий вивід займає центральне місце в нечіткій логіці й системах нечіткого керування. Процес нечіткого виводу являє собою деяку процедуру або алгоритм одержання нечітких висновків на основі нечітких умов або передумов.

Системи нечіткого виведення призначені для перетворення значень вхідних змінних процесу керування у вихідні змінні на основі використання нечітких правил продукцій. Для цього системи нечіткого виводу повинні містити базу правил нечітких продукцій і реалізовувати нечіткий вивід

висновків на основі посилок або умов, представлених у формі нечітких лінгвістичних висловлень.

Основними етапами нечіткого виведення є [3] (рис.1.6):

- формування бази правил (бази знань БЗ) систем нечіткого виводу,
- фазифікація вхідних змінних,
- агрегування подумовий у правилах нечітких продукцій,
- активізація або композиція подзаключений у нечітких правилах продукцій,
- акумулювання висновків нечітких правил продукцій;
- дефазифікація



Рисунок 1.6 – Етапи системи нечіткого виведення

2. НЕЧІТКАЯ СИСТЕМА УПРАВЛІННЯ КЛІМАТОМ

2.1. Розробка нечіткої системи регулювання температурою та вологістю

Останнім часом для керування системами кондиціонування повітря активно розвиваються принципово нові закони регулювання, що одержали назву «нечітка логіка» (Fuzzy logic). Це нова технологія керування, у якій як модель використовується нечітка система. Система вимірює температуру в приміщенні й автоматично вибирає режим роботи. Вибір ґрунтується на практичному аналізі — за еталон беруться стандартні переваги людей, що користуються системою. Величини відбивають рівні різних факторів, від значення яких залежить комфорт людини: температура, вологість, інтенсивність повітряних потоків, тип одягу (літня/зимова) і ін. Приведемо приклад обліку впливу вологості на стан людини. Відчуття теплоти або прохолоді є наслідком не тільки температури повітря, але його вологості. Температура повітря 26 °С и вологість 50-60% вважаються комфортними влітку, тоді як температура 22 °С буде комфортної взимку. Однак навіть температура 25°С буде перебувати в зоні комфортності, якщо вологість становить 50 %, тоді як ця ж температура при вологості 70 % буде здаватися високою .

Такий підхід добре погодиться з логічною системою обробки інформації «нечітка логіка» (fuzzy logic), що застосовується в нечітких логічних регуляторах (НЛР). Нечітка логіка має переваги в порівнянні з використанням ПД - регуляторів при обробці дуже складних процесів, нелінійних процесів високих порядків, обробці експертних (лінгвістично сформульованих) даних. Нечітка логіка оперує не цифровими, а лінгвістичними поняттями. Ключовими поняттями нечіткої логіки є: фазифікація - перетворення множини значень аргументу (x) у деяку функцію приналежності $M(x)$, тобто переклад значень (x) у нечіткий формат; дефазифікація - процес зворотний фазифікації.

Побудова нечіткої моделі ґрунтується на формальному поданні характеристик досліджуваної системи в термінах лінгвістичних змінних. Оскільки, крім нечіткого алгоритму, основними засобами опису керування є вхідні й вихідні змінні, те саме вони представляються як лінгвістичні змінні при формуванні бази правил у системах нечіткого виводу.

У цьому випадку нечітка система керування, представлена на рис.21, будується з урахуванням реалізації всіх правил нечіткої логіки, а сам процес знаходження керуючого впливу реалізується на основі алгоритмів нечіткого виводу [7].

Нехай у системі, зображеної на рис.2.1, об'єкт керування має більші фізичні розміри (фітнес-центр 1500 м²). У цьому випадку m датчиків зворотного зв'язку, що формують вектор вимірів U , і n виконавчих органів (ВО), що генерують вектор керування Y , розташовані на значних відстанях друг від друга. Припустимо також, що кожен компонент вектора вимірів та керувань передається по окремій парі проводів.

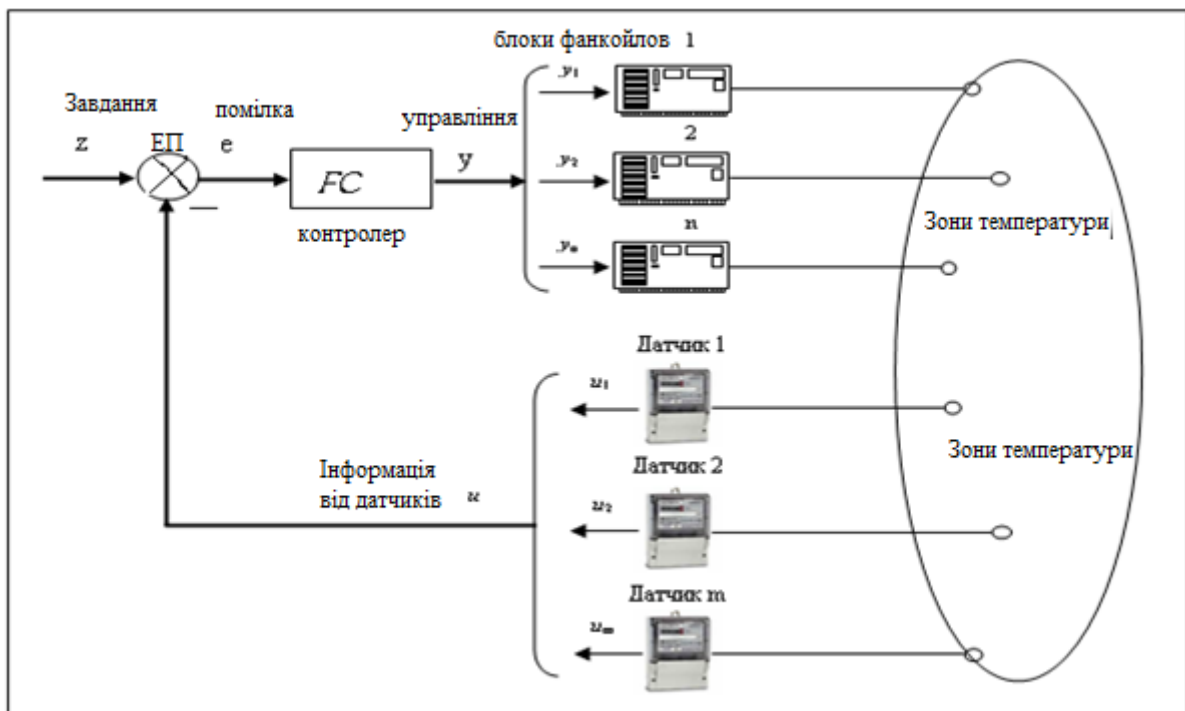


Рисунок 2.1. - Нечітка багатомірна система керування з дистанційними датчиками й виконавчими органами (FC – нечіткий контролер)

Досвід використання кондиціонерів показує, що охолодження або нагрівання повітря в приміщеннях має інерційністю, тобто. після включення режиму «холод» відбувається нагнітання холодного повітря, у зв'язку з чим температура повітря у приміщенні поступово падає. При цьому в момент відключення режиму температура продовжує знижуватися протягом деякого проміжку часу. Аналогічна ситуація спостерігається під час включення режиму «тепло». Щоб врахувати цю особливість процесу управління та виключити додаткові витрати, пов'язані з частим включенням та відключенням зазначених режимів, передбачається врахувати як контрольований параметр не тільки температуру повітря в приміщенні, але і швидкість її зміни.

Розглянемо принцип керування холодопродуктивністю кондиціонера з використанням нечіткої логіки. Холодопродуктивність, що повинен забезпечити кондиціонер, визначається різницею між температурою в приміщенні й температурою, що ми хотіли б одержати (температура уставки). Ця змінна лінгвістично може бути сформульована як «різниця температур» і приймати значення «мала», «середня» й «велика». Природно, чим більше різниця температур у цей момент, тим більше повинна бути холодопродуктивність. Другий лінгвістичної змінної визначимо «швидкість зміни температури» у приміщенні, який також дамо лінгвістичні значення «мала», «середня» й «більша». Якщо швидкість зміни температури більша, то потрібно більша холодопродуктивність. У міру наближення температури в приміщенні до температури уставки швидкість зміни температури в приміщенні буде зменшуватися, а холодопродуктивність кондиціонера знижуватися. Холодопродуктивність є вихідний змінної, котрої привласнюються наступні терми: «дуже мала», «мала», «середня», «більша» й «дуже більша». Зв'язок між входом і виходом занесемо в таблицю нечітких правил. Кожен запис відповідає своєму нечіткому правилу.

Наприклад, якщо різниця температур середня, а швидкість зміни більша, те холодопродуктивність повинна бути більша. Кондиціонер з

нечіткою логікою працює по наступному принципі: сигнали від датчиків будуть фазифіцировані, оброблені, дефазифіцировані й отримані дані у вигляді сигналів надійдуть на частотний регулятор двигуна компресора, швидкість обертання якого (а отже, і продуктивність) будуть мінятися у відповідності зі значенням функції приналежності. Результат спільного впливу двох функцій приналежності на значення вихідного параметра «холодопродуктивність» визначається відповідною програмою, закладеної в логічний пристрій. З огляду що холодопродуктивність пропорційно частоті обертання компресора, можна від частоти побудувати залежність результуючої функції приналежності обертання компресора, додавши лінгвістичним термам швидкість обертання компресора з рангом 1,0 наступні значення (рис. 8.3): мала - 20, середня — 50 Гц; більша — 75 Гц; дуже більша — 100 Гц.

Після дефазифікації можна перейти до чіткого значення вихідного параметра - частоті обертання компресора або холодопродуктивність. У ПЧ (перетворювачі частоти) вхідна напруга 220/380В, з частотою 50 Гц перетворюється на вихідну імпульсну напругу за допомогою широтно-імпульсної модуляції (ШІМ), яке формує в обмотках двигуна синусоїдальний струм частотою від 0 до 400 Гц. Збільшенням частоти та амплітуди напруги, що подається на обмотки асинхронного двигуна, забезпечується плавне регулювання швидкості обертання валу електродвигуна.

Системи з нечіткою логікою функціонують по наступному принципі: показання вимірювальних приладів фазифіцируються (переводяться в нечіткий формат), обробляються, дефазифіцируються й потім у вигляді звичайних сигналів подаються на виконавчі пристрої. Ключовим елементом в нечіткій системі управління являється - нечіткий контроллер или регулятор (рис.2.2)

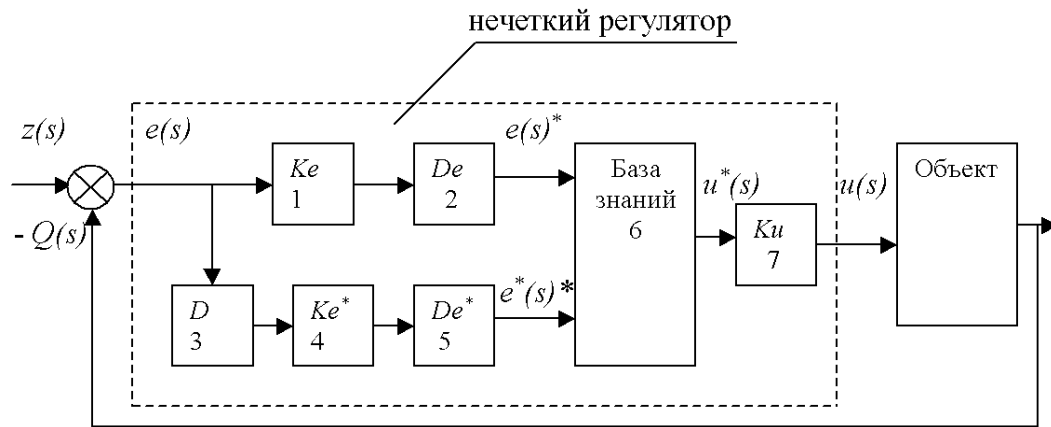


Рисунок 2.2. - Структура системи з нечітким регулятором

На рис. 2.2. вихідна змінна об'єкта регулювання технологічного процесу у $Q(s)$ рівняється із заданим значенням $z(s)$ і помилка неузгодженості $e(s) = Z(s) - Q(s)$ надходить у масштабний елемент 1 з коефіцієнтом K_e й у диференціатор 3, вхід якого множиться на K_e^* у масштабному елементі 4. Елементи 2, 5 призначені для перетворення поточних значень неузгодженості й похідній від неузгодженості (швидкості зміни неузгодженості) у їхні лінгвістичні значення (фазифікації).

Нечіткі терми $e(s)^*$, $e^*(s)^*$ надходять у головний елемент нечіткого регулятора 6 - базу знань (БЗ). База знань (БЗ) будується на основі продукційної моделі знань, що має конструкцію виду ЯКЩО...ТО...

Знайдене лінгвістичне значення керування після множення на масштабний коефіцієнт K_u в елементі 7 і перетворення його в чітке значення $u(s)$ (етап дефазифікації) надходить на виконавчі пристрої об'єкта керування. Масштабні коефіцієнти K_e , K_e^* , K_u є параметрами універсальних безлічей E , E^* й U , на яких, виходячи з умов конкретного керованого об'єкта, визначаються нечіткі безлічі $e(s)_1^*$, $e^*(s)^*$, $u(s)^*$. Наприклад, якщо універсальною безліччю E є діапазон $(-10, -9, \dots, +9, +10)$ і потрібно, щоб неузгодженість у системі перебувало в діапазоні $(-1; +1)$, тоді K_e береться рівним $10,0$, для того, щоб нечіткий регулятор міг використати всю універсальну безліч, на якому визначаються нечіткі безлічі.

Для нечіткого регулятора будь-яке правило в БЗ може бути представлено у вигляді:

$$\text{ЯКЩО } (e(s) \in e(s)_1^*) \text{ І } (e(s)^* \in e^*(s)_1^*) \text{ ТЕ } (u(s) \in u(s)_1^*),$$

де $e(s)$, $e(s)^*$, $u(s)$ - змінні; $e(s)_1^*$, $e^*(s)_1^*$, $u(s)_1^*$ - лінгвістичні значення змінних.

Згідно з методикою Алієва [5], база знань (НР) має табличний вигляд і в БЗ визначені нечіткі множини помилки $e(t)$, швидкості зміни $e'(t)$ і керуючого впливу $u(t)$, вищезазначені нечіткі множини описані за допомогою лінгвістичної мови, де лінгвістичні змінні (ЛП) такі: (NB – негативно велике, NS – негативно середнє, NM – негативно мале, Z – нульове, PS – позитивно середнє, PM – позитивно мале, PB – позитивно велике) [4]. Продукційна база правил виду ЯКЩО $e(t) = NM$ І $e'(t) = Z$ ТО $u(t) = Z$ у вигляді таблиці 2.1.

Таблиця 2.1. - База знань нечіткого регулятора (контролера)

| Помилка e | Похідна помилки e' | | | | |
|-------------|----------------------|----|----|----|----|
| | NB | NM | Z | PM | PB |
| NB | NB | NB | NM | PM | PM |
| NM | NM | NM | Z | Z | PM |
| Z | NM | NM | Z | PM | PM |
| PM | NM | Z | Z | PM | PM |
| PB | NM | Z | PM | PB | PB |

Графіки функцій наєжності вхідних та вихідних параметрів контролера на рис. 2.3 – 2.5.

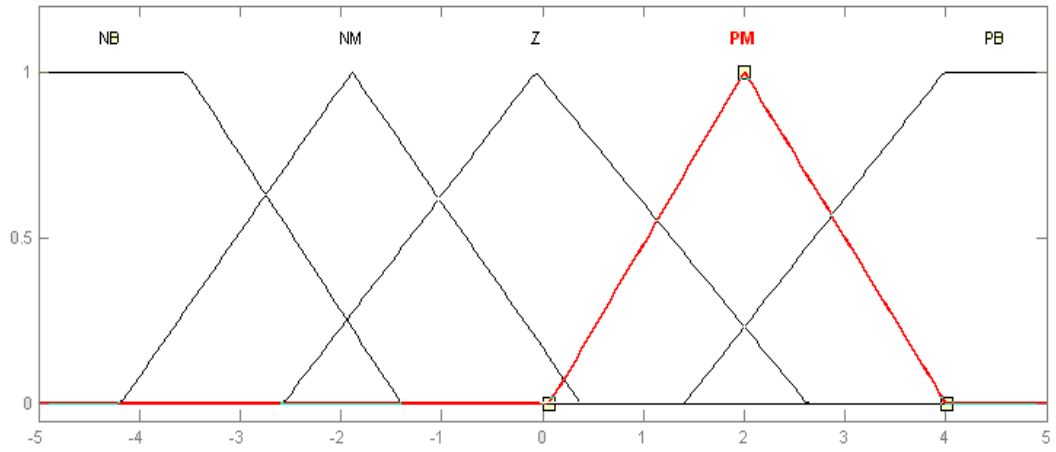


Рисунок 2.3 - Функції належності « помилка»

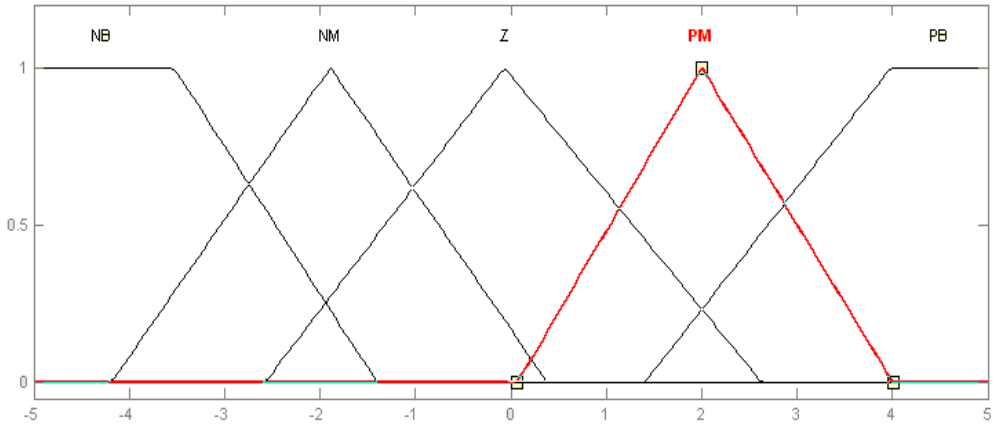
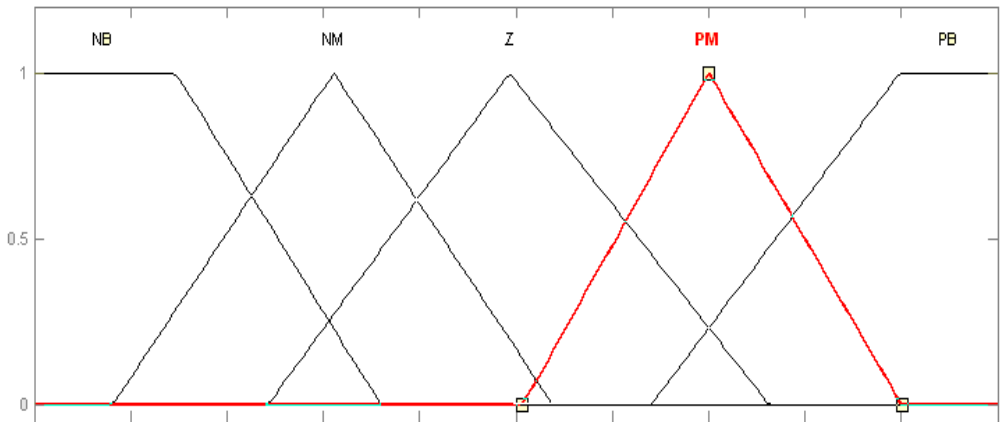


Рисунок 2.4 - Функції належності «Похідна помилки»



0

125 Гц.

Рисунок 2.5 - Функції належності «керуючий вплив»

Етап дефазифікації по першому каналу регулювання проводиться методом центру тяжкості:

$$H_c = \frac{\sum_{c=1}^7 H_c \cdot \mu(H_c)}{\sum_{c=1}^7 \mu(H_c)}$$

Перевірка ефективності роботи нечіткого контролера з відпрацювання завдання проводилося за допомогою моделювання у програмі Simulink (MatLab) (рис. 2.5). Передавальні функції відображають модель об'єкта.

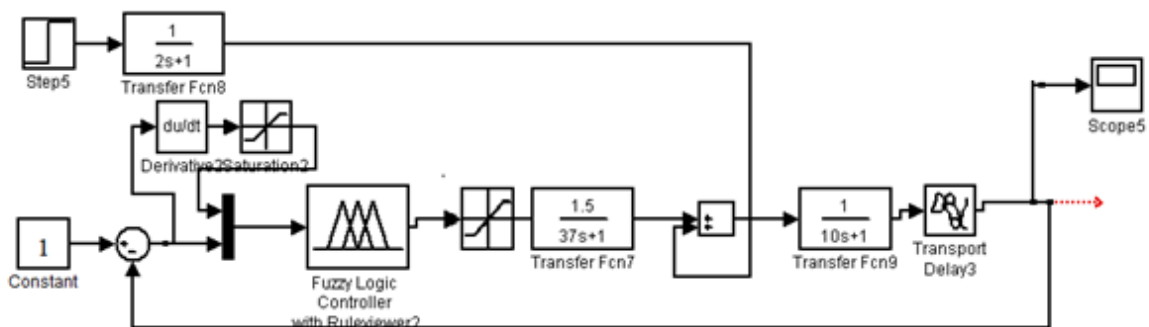


Рисунок 2.5 – Моделювання нечіткого контролера

2.2. Розробка програми для контролера системи управління

Платформа fuzzyTech призначена для рішення різних завдань нечіткого моделювання. На відміну від системи MatLab, fuzzyTECH є спеціалізованим засобом, що дозволяє розробляти й досліджувати різні нечіткі моделі в графічному режимі, а так само перетворювати їх у програмний код на одній з мов програмування з можливістю наступного виконання. Дана програма надає можливість створення клієнтського або серверного додатка для нечіткого керування вилученими об'єктами.

Загальна структура контролера, що використовує нечітку логіку, містить у своєму складі наступні складові частини :

- блок фазифікації;
- базу знань;
- блок рішень;
- блок дефазифікації.

Реалізувати фаззи - контролер можна практично на будь-якому мікроконтролері, наприклад, Motorola 68HC11 або Intel MCS 96.

Існують мікроконтролери, у яких фаззи-команди інтегровані безпосередньо в асемблер. Наприклад, це кристали Motorola 68HC12 або STMicroelectronics ST52x301. У ядро цих мікросхем убудована апаратна підтримка базових фаззи-операцій. Зокрема, мікроконтролер 68HC12 базується на ядрі з малим споживанням, що має розширений набір інструкцій 68HC11 на 16-розрядному рівні. Стік переривань і програмна модель повністю ідентичні 68HC11. Ядро контролера 68HC12 включає 64 нові інструкції, 20-розрядний арифметико-логічний пристрій, черга інструкцій і розширену індексну адресацію. Через нові інструкції й сторінкову адресацію пам'яті розроблювачі мають доступ до більш ніж чотирьох мегабайтів програмного коду й одного мегабайтові даних. Архітектура 68HC12 оптимізована для мов високого рівня, швидких математичних операцій і фаззи-логіки.

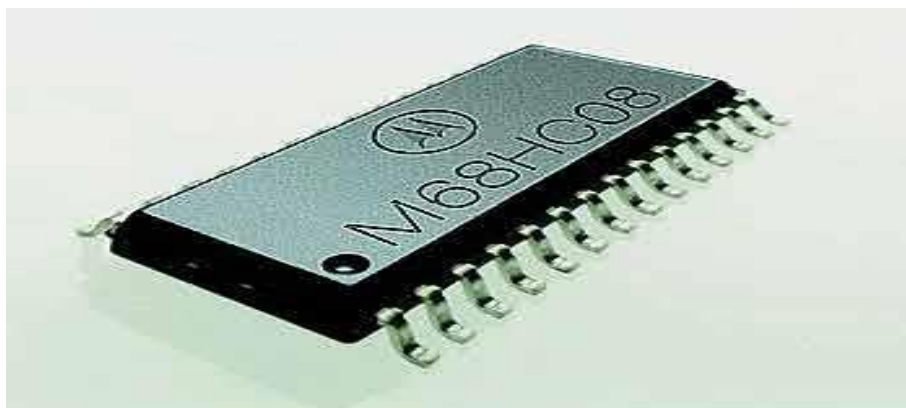


Рисунок 2.6 - Мікроконтролер Motorola 68HC12

Розробку системи керування на базі нечіткої логіки умовно можна розбити на чотири етапи [13]:

1. Опис системи.

На цьому етапі за допомогою програмних засобів, пакета FuzzyTech, завдання фазифіцирується. Тут описуються лінгвістичні змінні, використовувані в моделі, їхньої функції приналежності, розробляється стратегія керування за допомогою нечітких правил, що поєднується в єдину базу правил.

2. Off - line - оптимізація.

У цьому пункті перевіряється працездатність створеної моделі за допомогою засобів fuzzyTech. Використається створений програмний симулятор об'єкта керування. Виробляється аналіз його адекватності, працездатності й підтримки заданих параметрів.

3. On - line - оптимізація

На цьому етапі розроблювальна система керування й реальний об'єкт з'єднуються фізичною лінією зв'язку. Для зв'язку системи керування з моделлю використовується протокол зв'язку fTlink, в основу якого покладена концепція обміну повідомленнями Windows. Всі необхідні засоби для встановлення зв'язку об'єкта з моделлю перебувають у вихідних текстах програм поставляють із пакетом.

Даний вид налагодження дозволяє спостерігати поведінку системи в реальних умовах й, при необхідності, вносити зміни в систему керування.

4. Реалізація.

На цьому етапі необхідно одержати остаточний варіант коду для конкретного мікроконтролера. Основу програмного коду, генерируемого пакетом fuzzyTech 5.5, становить апаратно – орієнтоване на конкретний тип процесора ядро. Поставляє з пакетом fuzzyTECH програмне ядро сумісне з такими контролерами, як, 8096 ВН, 8019МС, 80196 NT/NQ, 68НС12 й ін.

Пакет fuzzyTECH дозволяє створювати програму шляхом вибору необхідних блоків у меню. Останнім етапом є запуск програми й перегляд результатів керуючого впливу.

На першому етапі розробки програми вказуються кількість вхідних та вихідних змінних, а також кількість терм (нечітких множин) у кожній змінній (рис.2.7). Вікно редактора проекту представлено на рис. 2.8.

Виробляється фазифікація помилки (E) (рис.2.9) та її похідної (E') (рис.2.10) та керуючого впливу – частоти обертання двигуна компресора (рис. 2.12). Всі основи етапи створення програмного продукту для програмування нечіткого контролера в системі управління кліматом представлені на рис. (2.13 – 2.18).

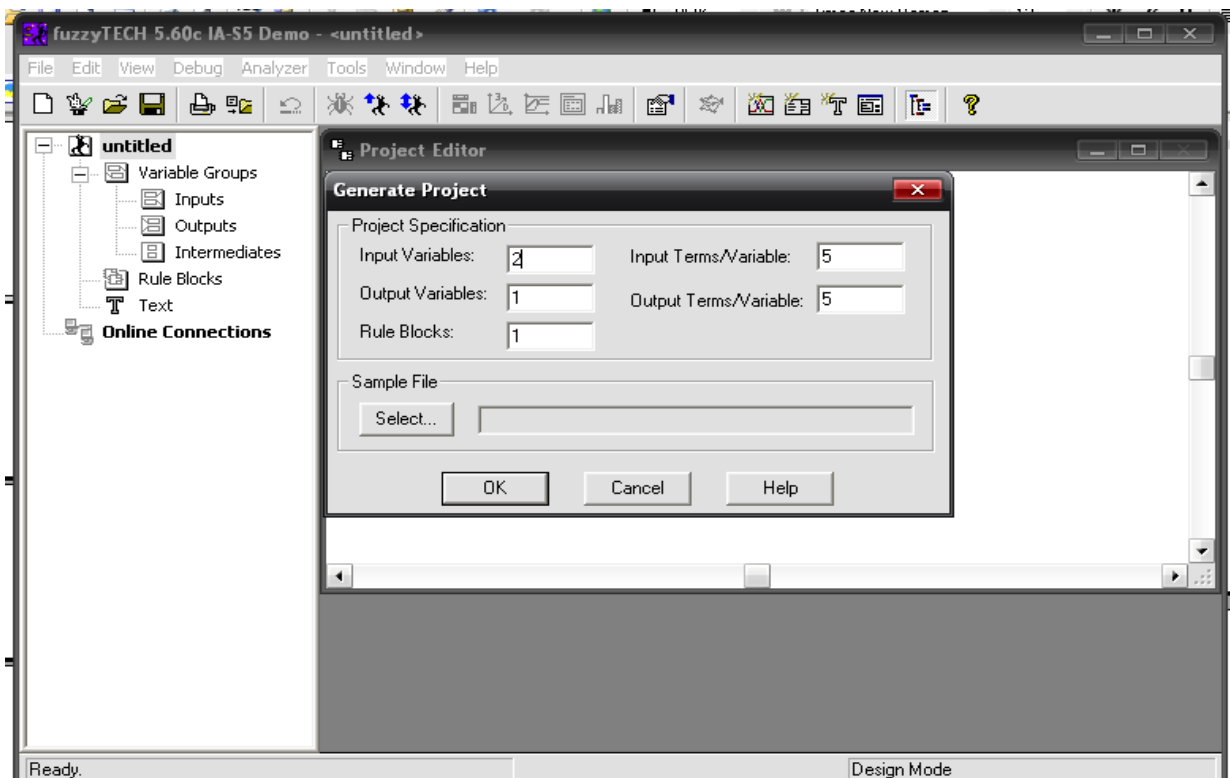


Рисунок 2.7 - Етап вибору кількості вхідних та вихідних змінних

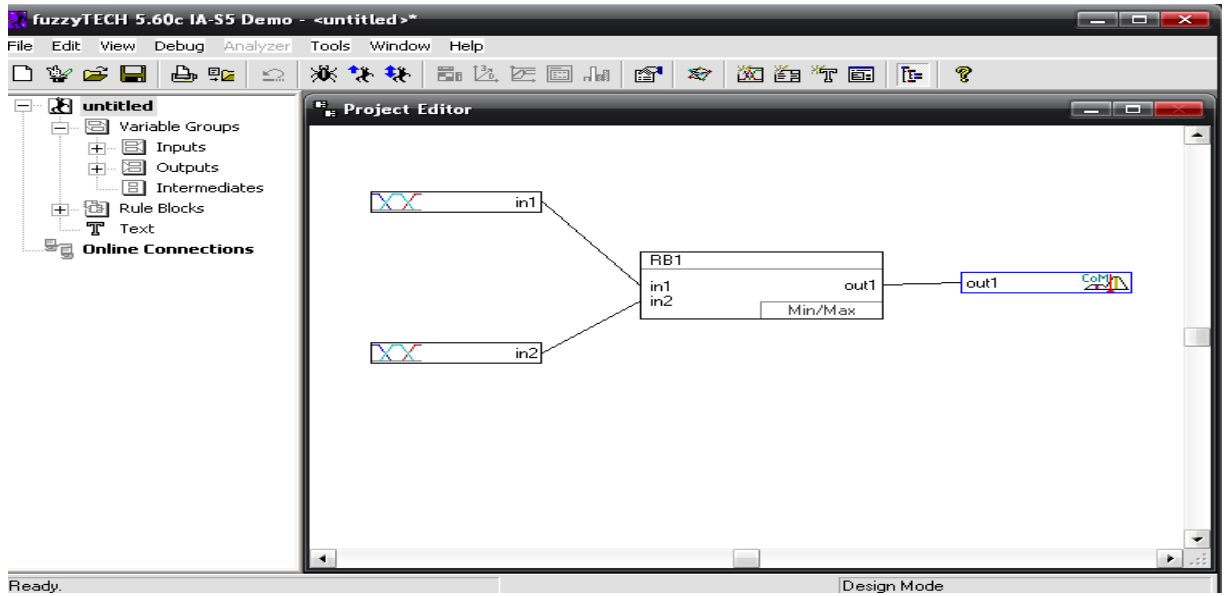


Рисунок 2.8 - Редактор проекту програми

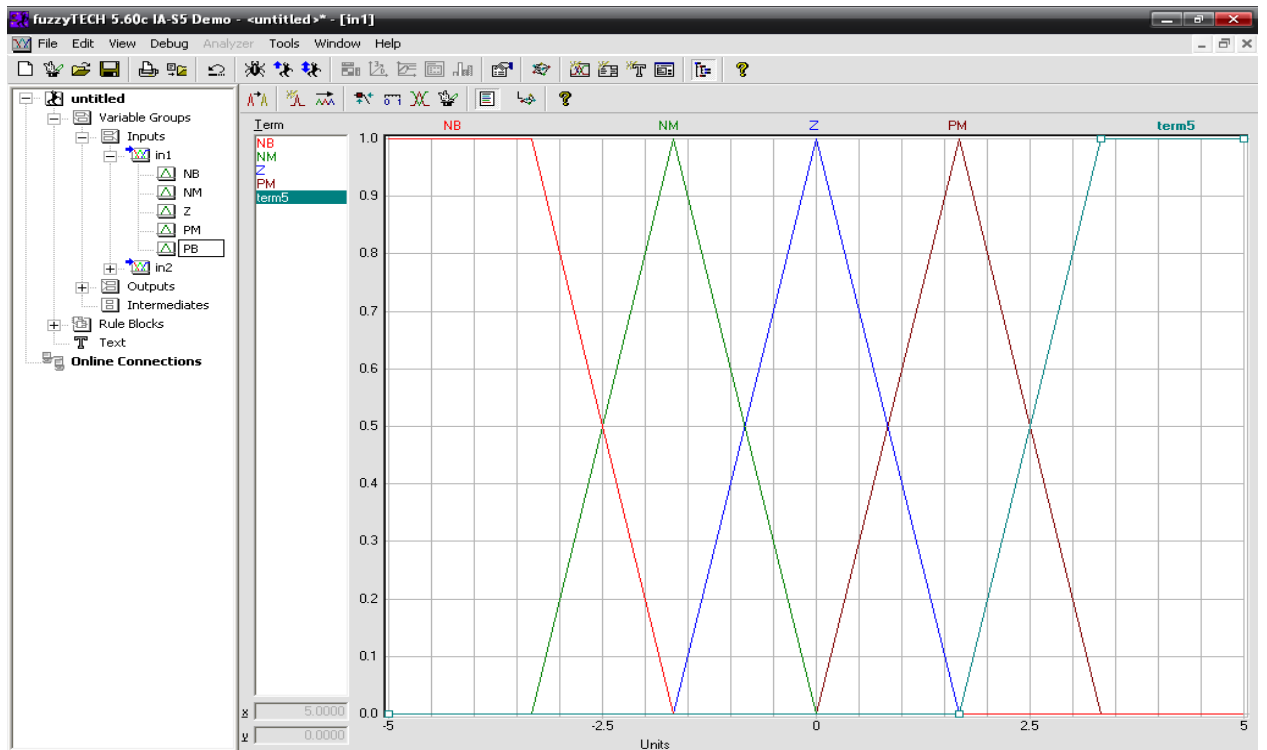


Рисунок 2.9 - Фаззифікація вхідної змінної «помилка» у вікні проекту.

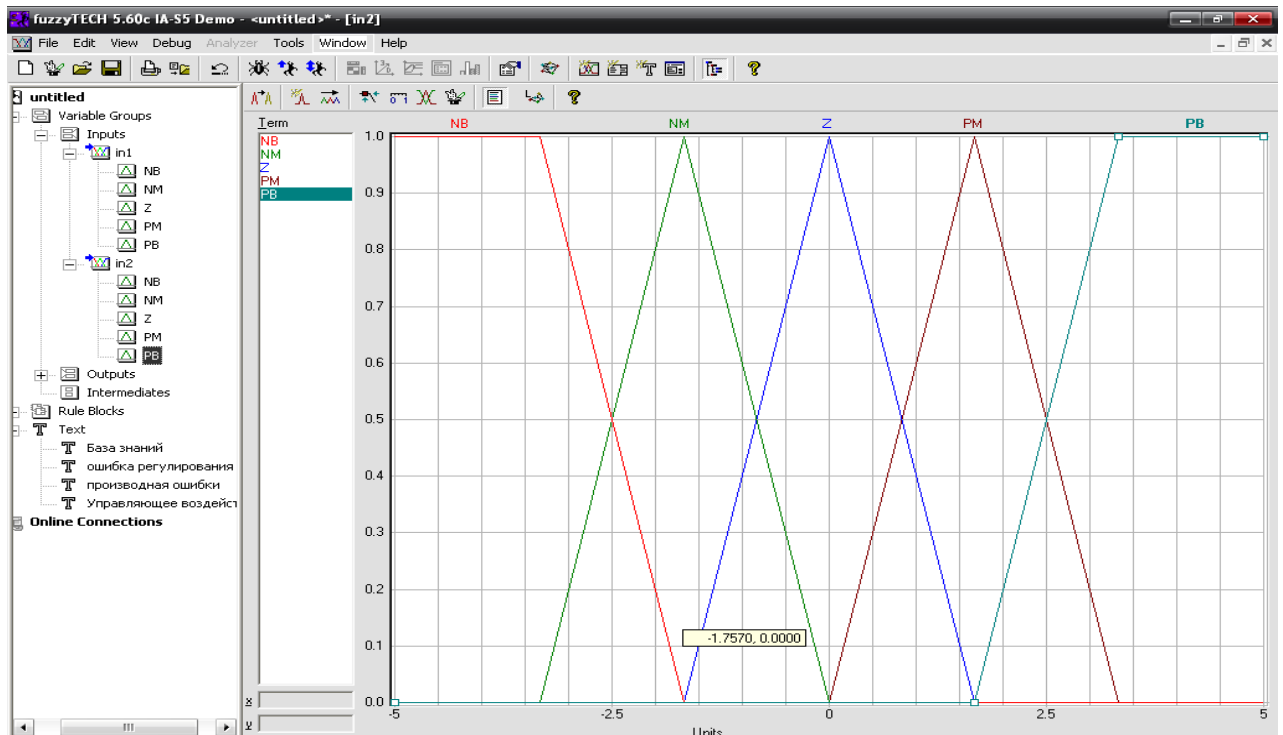


Рисунок 2.10 - Фаззифікація вхідної змінної «похідна помилка» у вікні проекту

The screenshot shows the 'Variable Properties: out 1' dialog box. It has two tabs: 'Base Variable' and 'Comments'. The 'Base Variable' tab is active and contains the following settings:

- Base Variable Range:**
 - Min: 0 (Shell Values) as 0 (Code Values) [Minimum]
 - Max: 125 (Shell Values) as 32767 (Code Values) [Maximum]
 - Default: 0.5
 - Keep Definition Points
 - Unit: Units
- Grid Resolution:**
 - Base: 5e-005 = 20000 Interval(s)
 - Membership: 1 = 1 Interval(s)
 - Snap to Grid

Buttons at the bottom: OK, Отмена, Справка.

Рисунок 2.11 - Вікно програми з функцією універсаму вихідної змінної

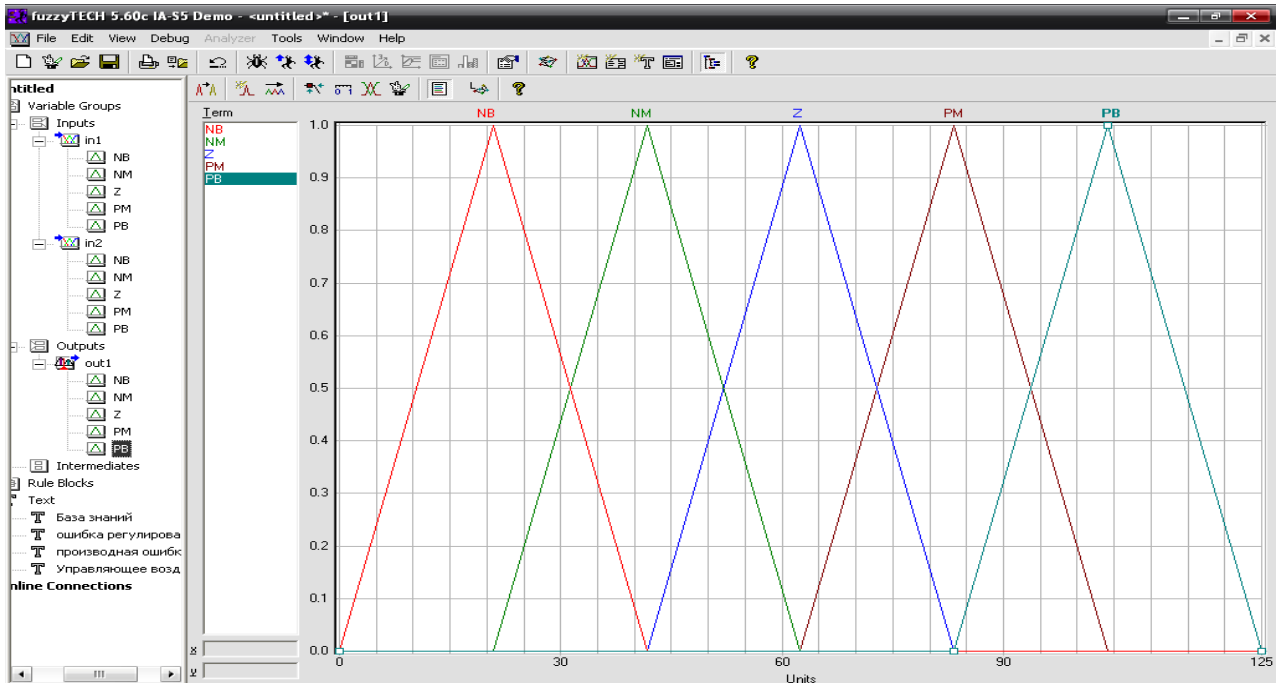


Рисунок 2.12 - Вікно фазифікації вихідної змінної (керуюча дія)

| # | IF | THEN | DoS | U |
|----|----|------|------|----|
| 1 | NB | NB | 1.00 | NB |
| 2 | NB | NM | 1.00 | NM |
| 3 | NB | Z | 1.00 | |
| 4 | NB | PM | 1.00 | |
| 5 | NB | PB | 1.00 | |
| 6 | NM | NB | 1.00 | |
| 7 | NM | NM | 1.00 | |
| 8 | NM | Z | 1.00 | |
| 9 | NM | PM | 1.00 | |
| 10 | NM | PB | 1.00 | |
| 11 | Z | NB | 1.00 | |
| 12 | Z | NM | 1.00 | |
| 13 | Z | Z | 1.00 | |
| 14 | Z | PM | 1.00 | |
| 15 | Z | PB | 1.00 | |
| 16 | PM | NB | 1.00 | |
| 17 | PM | NM | 1.00 | |
| 18 | PM | Z | 1.00 | |
| 19 | PM | PM | 1.00 | |
| 20 | PM | PB | 1.00 | |
| 21 | PB | NB | 1.00 | |
| 22 | PB | NM | 1.00 | |
| 23 | PB | Z | 1.00 | |
| 24 | PB | PM | 1.00 | |
| 25 | PB | PB | 1.00 | |
| 26 | | | | |

Рисунок 2.13 - Граничне вікно табличного редактора бази правил нечітких продукцій

База правил містить 25 правил нечітких продукцій виду: «ЯКЩО ... Умова (Помилка = великий ... І похідна помилки велика) ... ТО Управління = Велике»

| # | IF | E | THEN |
|----|------|----|---------|
| | dEdt | | DoS U |
| 1 | NB | NB | 1.00 NB |
| 2 | NB | NM | 1.00 NM |
| 3 | NB | Z | 1.00 NM |
| 4 | NB | PM | 1.00 NM |
| 5 | NB | PB | 1.00 NM |
| 6 | NM | NB | 1.00 NB |
| 7 | NM | NM | 1.00 NM |
| 8 | NM | Z | 1.00 NM |
| 9 | NM | PM | 1.00 Z |
| 10 | NM | PB | 1.00 Z |
| 11 | Z | NB | 1.00 NM |
| 12 | Z | NM | 1.00 Z |
| 13 | Z | Z | 1.00 Z |
| 14 | Z | PM | 1.00 Z |
| 15 | Z | PB | 1.00 PM |
| 16 | PM | NB | 1.00 PM |
| 17 | PM | NM | 1.00 Z |
| 18 | PM | Z | 1.00 PM |
| 19 | PM | PM | 1.00 PM |
| 20 | PM | PB | 1.00 PB |
| 21 | PB | NB | 1.00 PM |
| 22 | PB | NM | 1.00 PM |
| 23 | PB | Z | 1.00 PM |
| 24 | PB | PM | 1.00 PM |
| 25 | PB | PB | 1.00 PB |
| 26 | | | |

Рисунок 2.14 - База правил нечітких продукцій для контролера (алгоритм пошуку сигналу керування)

| Inputs: | Outputs: |
|-------------|-----------|
| dEdt 0.9810 | U 95.6000 |
| E 3.6230 | |

Рисунок 2.15 - Вікно перегляду результатів вибору керуючого значення

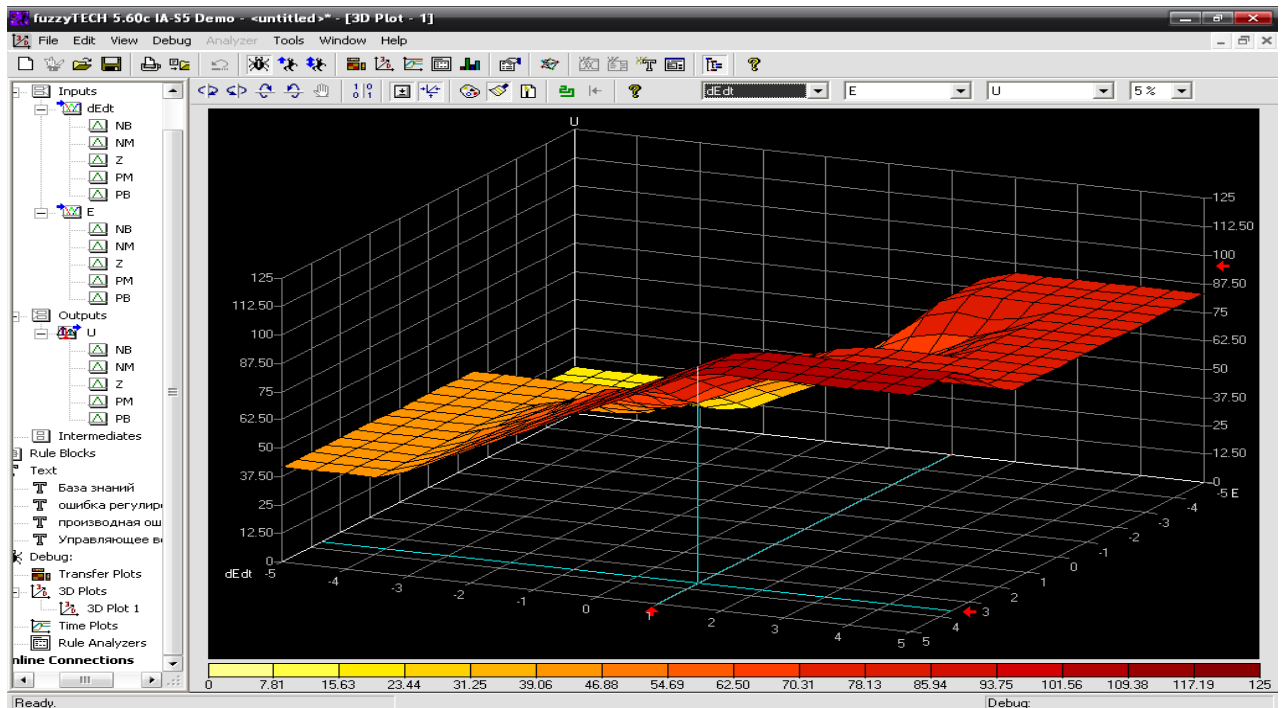


Рисунок 2.16 - Вікно перегляду тривимірної поверхні системи нечіткого виводу

Для генерації коду програми для мікропроцесора існує меню Tools (рис. 2.2.12)

Останнім етапом є генерація програмного коду й програмування ППЗУ (постійний запам'ятовувальний пристрій). Даний пункт виконується за рахунок меню вибору протоколу зв'язку із зовнішнім контролером або типом процесора у випадку підключення контролера до ЕОМ через послідовний СОМ порт і передачу програми по локальній мережі (рис. 2.18).

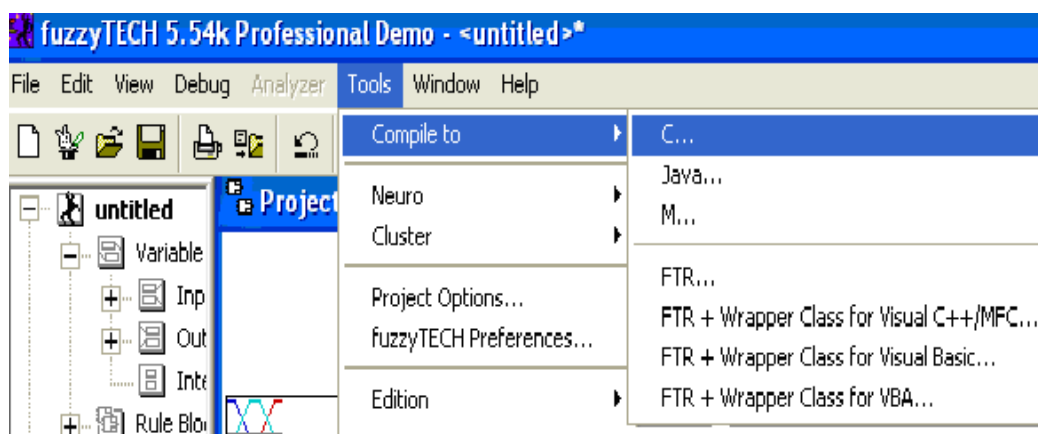


Рисунок 2.17 - Генерація програмного коду

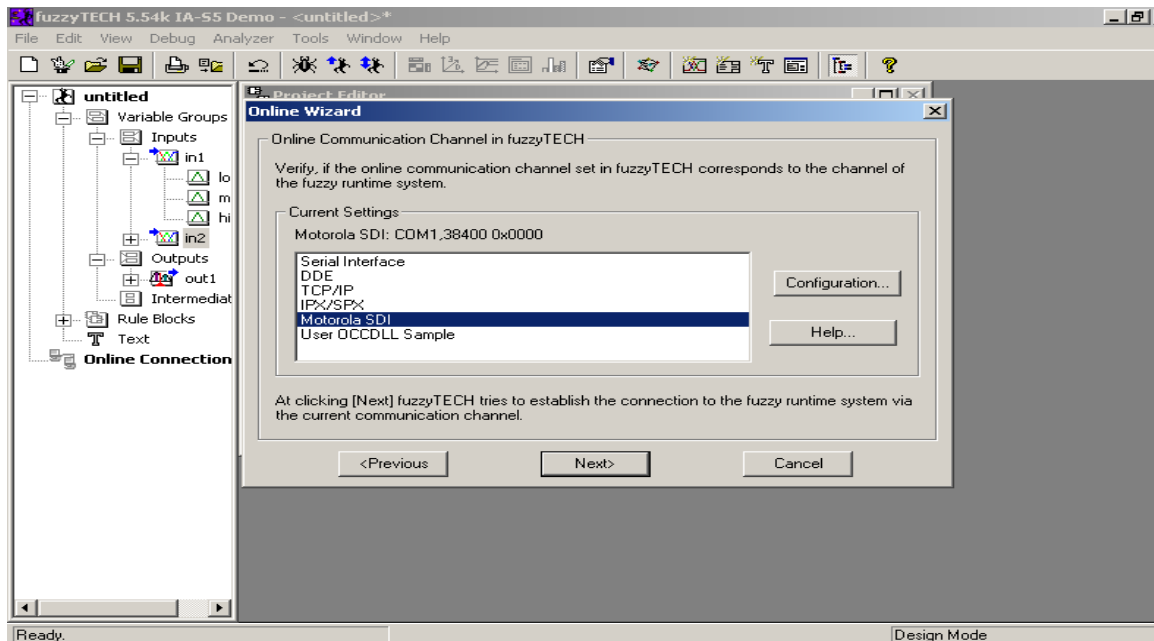


Рисунок 2.18 - Передача програми по локальній мережі

Існуючі програмувальні контролери на платформі Motorola і д.р., а також спеціалізовані програмні пакети успішно дозволяють розробити нечітку модель керування технологічним процесом, апробувати її у віртуальному середовищі пакета й згенерувати програмний код для контролера. Підтримувана температура в приміщенні перебуває на рівні допуску, завдяки чому знижується енергоспоживання. Таким чином, керування кондиціонером із застосуванням нечітких логічних регуляторів (рис.2.19) забезпечує: зміна температури відповідно до санітарних норм (відсутність різкого перепаду температур у приміщенні, підтримка припустимої швидкості потоку повітря й ін.); установку необхідної холодопродуктивності; вибір режиму роботи й уставку температури, виходячи з температури й вологості в приміщенні; вибір комфортного розподілу й інтенсивності потоку повітря; мінімальний час виходу на заданий режим; зменшення витрати електроенергії на 20-40 %.



Рисунок 2.19 – Контроллер ADAM с функцией нечеткой логики

2.3. Розробка експертної системи для підтримки прийняття рішення оператора під час управління системою припливної вентиляції

Функціональна схема системи керування групою вентиляційних камер для басейна представлена на рис. 2. 20. У цій схемі група повітрепідігрівальних установок припливних камер ПК1-ПКп, з'єднаних паралельно по теплоносію, пов'язана з вузлом підготовки теплоносія, що складається з насосів Н1 та Н2 (один резервний), зворотного клапану К1, регулюючого клапану К2 та регулятора тиску Р. На зворотному трубопроводі перед вузлом підготовки встановлено реле протоки теплоносія РПТ.

Виконавчий механізм клапану К2 електрично пов'язаний з регулятором ТС, на входи якого приєднані датчики температури D_1 (ТЕ) у приміщеннях ТРЦ та датчик температури зовнішнього повітря D_n (ТЕ). Розроблена система забезпечує керування групою припливних камер у ручному та автоматичному режимах. У ручному режимі управління система дозволяє запуснути та зупинити двигун вентилятора будь-якої припливної камери ПК1-ПКп, запуснути у відповідному напрямку та зупинити виконавчий механізм

регулюючого клапану K2; запустити у відповідному напрямку та зупинити виконавчі механізми будь-якого повітряного клапану. У режимі автоматичного керування система дозволяє здійснити програмний запуск та вимкнення припливних камер ПК1-ПК, автоматична підтримка заданої температури повітря на виході з припливних камер; контроль температури теплоносія на виході з калорифера, температури та швидкості повітря на виході із припливних камер з сигналізацією аварійного режиму.

Увімкнення системи та вибір режиму «Ручний-Автомат» здійснюється з дистанційного щита або з комп'ютера зі SCADA - системою. Після 5-хвилинного прогрівання калориферів автоматично вмикаються електродвигуни вентиляторів та відкриваються повітряні приймальні клапани. Після повного відкриття спрацьовують кінцеві мікроперемикачі, підключаючи до роботи ланцюги сигналізації та контролю припливних камер. За відсутності або зниженні витрати теплоносія спрацьовує реле РПТ і знеструмлює проміжне реле, яке, у свою чергу, розмикає контакти для живлення магнітних пускачів вентиляторів.

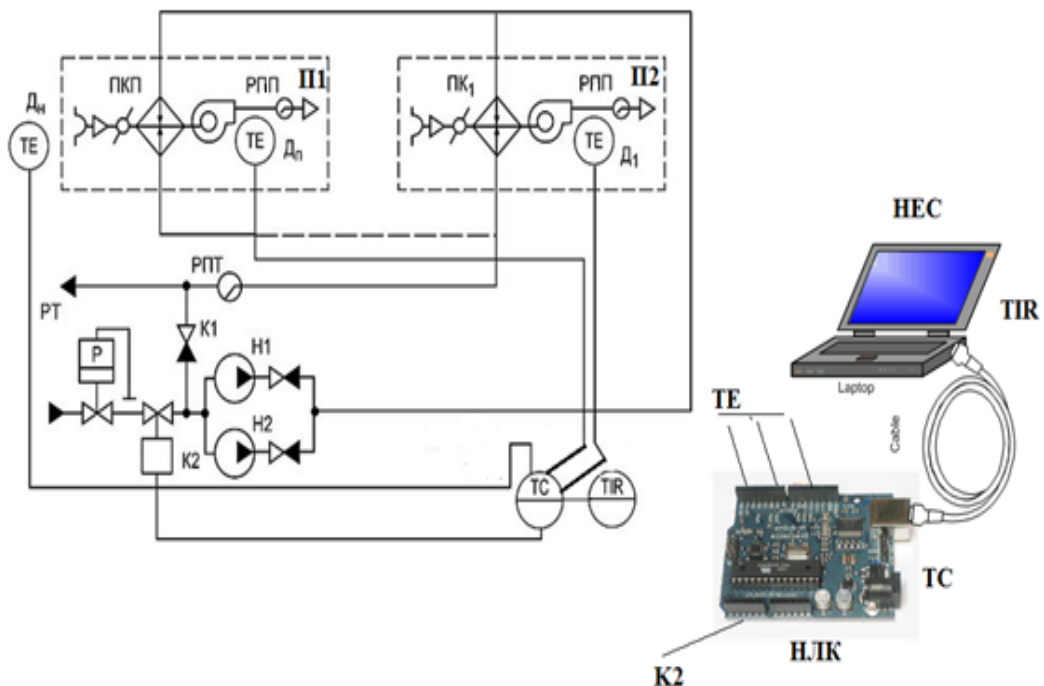


Рисунок 2.20 – Функціональна схема керування групою припливних камер приточної вентиляції басейна:

TE – датчики температури; ЛЛК (ТС) – логічний контролер; НЕС – експертна система; ТІР – комп'ютер зі SCADA

Вимкнення системи автоматичного керування здійснюється також із дистанційного щита або комп'ютера. При цьому знеструмлюються магнітні пускачі насоса та електродвигунів вентиляторів, закриваються повітряні приймальні клапани та клапан К2 на теплоносії.

Розробка функцій належності зон комфорту фітнес - центру .
Змодельємо систему керування повітряним опаленням приміщення з погляду оператора при роботі у режимі дистанційного керування тепловим пунктом . Як вхідні дані системи виступають значення: температура повітря зовнішнього (навколишнього) середовища та температура повітря приміщення (зал ТРЦ). Як вихідні параметри системи - буде відсоток відкриття вентиля (клапана К2) приладу обігрівача для регулювання витрати теплоносія (див. рис.1).

Для відповідності температурним параметрам відповідно до санітарних норм та приміщень різного призначення в фітнес – залах використовувався апарат нечітких множин [10]. Рівень прийнятого комфорту (діапазон 20-25.5°C), представлений у вигляді функції належності (ФП), за вимогами нормативних документів та побажань відвідувачів центру (ді. рис.2 Додатка). Для рівня підвищеного комфорту (ресторани, спа і т.д.) та прийнятого (спортивний та ігровий зали) над значеннями функції належності проведено логічні операції концентрації (зведення у квадрат значень ступенів істинності ФП звичайного комфорту (див. рис.2)) та розмивання (вилучення квадратного кореня значень ступенем істинності ФП).

Для отримання ФП написано програмний продукт мовою програмування C# в середовищі програмування Visual Studio 2022 використовуючи шаблон Додаток Windows Forms (.NET Framework).

Можна бачити (див.рис.2. Додаток), що перший графік жовтого кольору «рівень підвищеного комфорту» визначає найкращу температуру для зон

кофе та ресторанів. Другий графік червоного кольору - «рівень звичайного комфорту»» виділяє більш виразно прийнятну температуру для зон магазинів. Третій графік фіолетового кольору «більш-менш комфортна температура» виділяє температуру для зон заняття спортом. Це функції можливо використовувати для подальший розробки нечітких експертних

Розробка нечіткої експертної системи (НЕС). Розробимо програму для нечіткої системи порад оператору системою повітряного опалення у приміщеннях круїзного судна. Розробка НЕС здійснена у програмному середовищі Fuzzy TECH [25].

Хід етапів створення НЕС у програмі Fuzzy TECH (рис.2.21- 2.35):

1) Спочатку розробляється база знань або правил по ручному куруванню обладнанням вентиляції. На погляд експерта – оператора база знань має наступний вид:

Опишемо всі експертно-створені правила у вигляді наступних виразів:

Правило 1. ЯКЩО "температура повітря в приміщенні низька" І "температура повітря зовні низька", ТО "становище вентиля відкрито високе".

Правило 2. ЯКЩО "температура повітря в приміщенні низька" І "температура повітря зовні середня", ТО "становище вентиля відкрито високе".

Правило 3. ЯКЩО "температура повітря в приміщенні низька" І "температура повітря зовні висока", ТО "становище вентиля середнє".

Правило 4. Якщо «температура повітря в приміщенні середня» та «температура повітря зовні низька», то «становище вентиля середнє».

Правило 5. ЯКЩО «температура повітря в приміщенні середня» та «температура повітря зовні середня», ТО «становище вентиля середнє».

Правило 6. ЯКЩО «температура повітря в приміщенні середня» І «температура повітря зовні висока», ТО «положення вентиля низьке».

Правило 7. Якщо «температура повітря в приміщенні висока» та «температура повітря зовні низька», то «становище вентиля низьке».

Правило 8. Якщо «температура повітря в приміщенні висока» та «температура повітря зовні середня», то «становище вентиля низьке».

Правило 9. ЯКЩО «температура повітря у приміщенні висока» І «температура повітря зовні висока», ТО «положення вентиля низьке».

Другий етап.

2) Створимо вищеописані параметри (змінні) системи та задамо діапазони вимірювань даних для вхідних та вихідних змінних. А також скоригуємо функції належності власності цих змінних.

Для вхідних лінгвістичних змінних (рис.2.26):

1) Лінгвістична змінна "Air_Temperature_Inside":

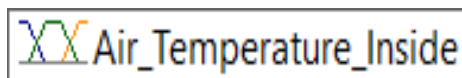


Рисунок 2.21 – Змінна температура повітря в середині приміщення у програмі Fuzzy TECH

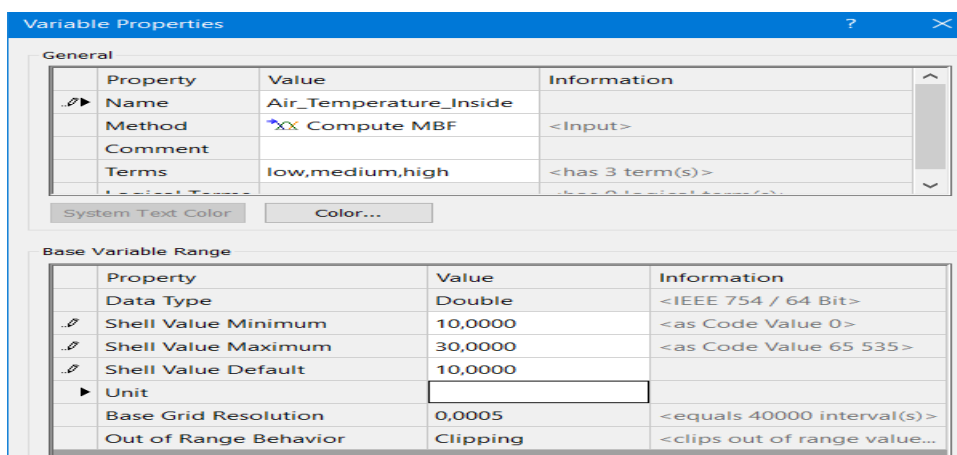


Рисунок 2.22 – Налаштування параметрів змінної температури повітря в середині приміщення ТРЦ

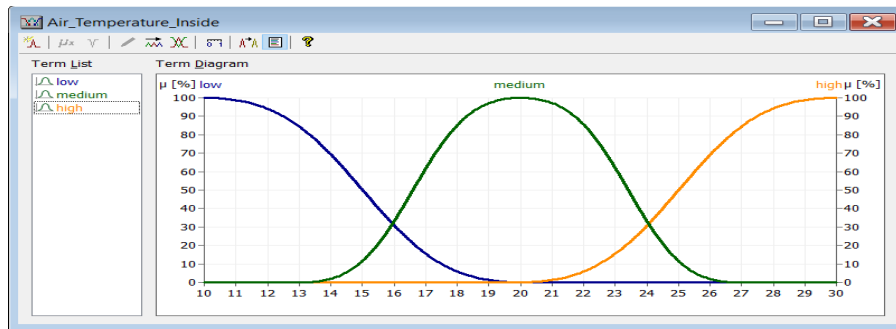


Рисунок 2.23 – Функції належності змінної температури повітря усередині (Мала, Середня, Висока) приміщення та змінна "Air_Temperature_Outside":

`XX Air_Temperature_Outsi...`

Рисунок 2.24 – Змінна температура зовнішнього повітря програмі Fuzzy TECH

| Variable Properties | | | |
|----------------------------|-------------------------|-------------------------------|--|
| General | | | |
| Property | Value | Information | |
| Name | Air_Temperature_Outside | <Input> | |
| Method | XX Compute MBF | | |
| Comment | | | |
| Terms | low,medium,high | <has 3 term(s)> | |
| Linked Terms | | <has 0 linked term(s)> | |
| System Text Color Color... | | | |
| Base Variable Range | | | |
| Property | Value | Information | |
| Data Type | Double | <IEEE 754 / 64 Bit> | |
| Shell Value Minimum | -15,0000 | <as Code Value 0> | |
| Shell Value Maximum | 15,0000 | <as Code Value 65 535> | |
| Shell Value Default | 0,0000 | | |
| Unit | | | |
| Base Grid Resolution | 0,0005 | <equals 60000 interval(s)> | |
| Out of Range Behavior | Clipping | <clips out of range value...> | |

Рисунок 2.25 – Налаштування параметрів змінної температури зовнішнього повітря

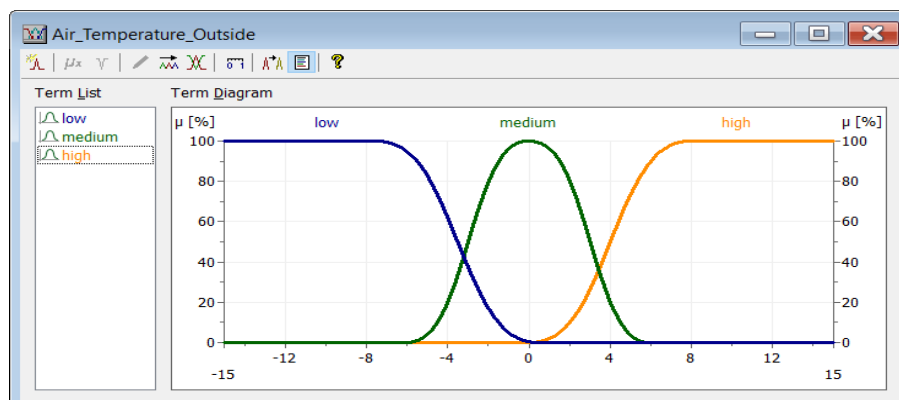


Рисунок 2.26 – Функції належності змінної температури зовнішнього повітря
(Мала, Середня, Висока)

Для вихідних змінних:

Змінна "Heater_Valve_Position":


Heater_Valve_Position 

Рисунок 2.27 – Змінна положення вентиля обігрівача у програми Fuzzy
TECH

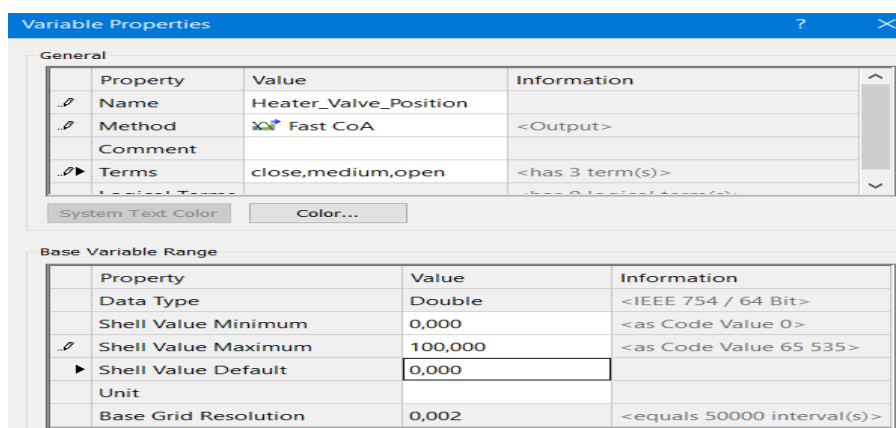


Рисунок 2.28 – Налаштування параметрів змінної положення вентиля
обігрівача

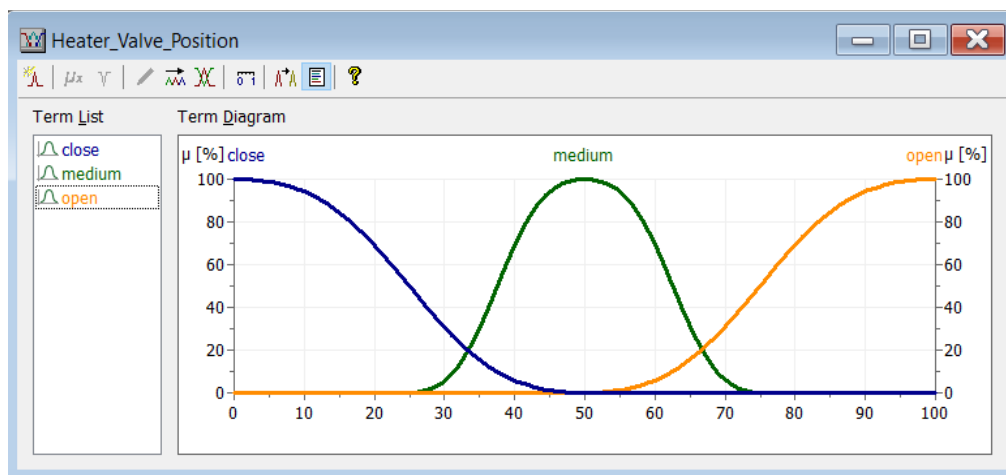


Рисунок 2.29– Функції належності змінної положення вентиля обігрівача
(Мале, Середнє, Високе)

Наступним кроком, коли всі змінні визначені та налаштовані, створимо базу правил, на основі якої буде працювати наша нечітка система:

| L1 | RB1 | | | B1 |
|--------------------------|-----------------------|---|---|-----|
| Air_Temperature_Inside | Heater_Valve_Position | | | |
| Air_Temperature_Outsi... | | | | |
| Min | 9 | - | - | Max |

Рисунок 2.30 – Графічний елемент основи правил на принциповій схемі

| Rule Blocks | | | | | | | | | | |
|-------------|----------|-------------------------------|--------------------------------|-----------|------------------------------|-------|-------|---------------------|------|--------|
| RB1 | | | | | | | | | | |
| | Na... | If | And | Operators | Then | Wi... | Co... | Audit | GUID | |
| ▶ | B1 | RB1 | | Min / Max | | | | 2023-05-15 13:04:35 | Vlad | 161CD3 |
| ▶ | B1.G1 | Air_Temperature_Inside | Air_Temperature_Outside | | Heater_Valve_Position | DoS | | 2023-05-15 13:04:35 | Vlad | D5E911 |
| ▶ | B1.G1.R1 | Air_Temperature_Inside.Low | Air_Temperature_Outside.Low | => | Heater_Valve_Position.open | 100 | | 2023-05-15 12:47:55 | Vlad | 878104 |
| ▶ | B1.G1.R2 | Air_Temperature_Inside.Low | Air_Temperature_Outside.medium | => | Heater_Valve_Position.open | 100 | | 2023-05-15 13:03:14 | Vlad | EC05D1 |
| ▶ | B1.G1.R3 | Air_Temperature_Inside.Low | Air_Temperature_Outside.high | => | Heater_Valve_Position.medium | 100 | | 2023-05-15 13:03:25 | Vlad | 09DBA1 |
| ▶ | B1.G1.R4 | Air_Temperature_Inside.medium | Air_Temperature_Outside.Low | => | Heater_Valve_Position.medium | 100 | | 2023-05-15 13:03:41 | Vlad | 7306F5 |
| ▶ | B1.G1.R5 | Air_Temperature_Inside.medium | Air_Temperature_Outside.medium | => | Heater_Valve_Position.medium | 100 | | 2023-05-15 13:03:59 | Vlad | FF2212 |
| ▶ | B1.G1.R6 | Air_Temperature_Inside.medium | Air_Temperature_Outside.high | => | Heater_Valve_Position.close | 100 | | 2023-05-15 13:04:07 | Vlad | 186E50 |
| ▶ | B1.G1.R7 | Air_Temperature_Inside.high | Air_Temperature_Outside.Low | => | Heater_Valve_Position.close | 100 | | 2023-05-15 13:04:16 | Vlad | 8E8E50 |
| ▶ | B1.G1.R8 | Air_Temperature_Inside.high | Air_Temperature_Outside.medium | => | Heater_Valve_Position.close | 100 | | 2023-05-15 13:04:27 | Vlad | 39C5BE |
| ▶ | B1.G1.R9 | Air_Temperature_Inside.high | Air_Temperature_Outside.high | => | Heater_Valve_Position.close | 100 | | 2023-05-15 13:04:35 | Vlad | 6B14D1 |

Рисунок 2.31 – Безпосередньо сама база правил системи у вигляді логічних виразів

4) Наступним етапом побудови НЕС є загальний вигляд розробленої системи та структурні зв'язки її елементів:

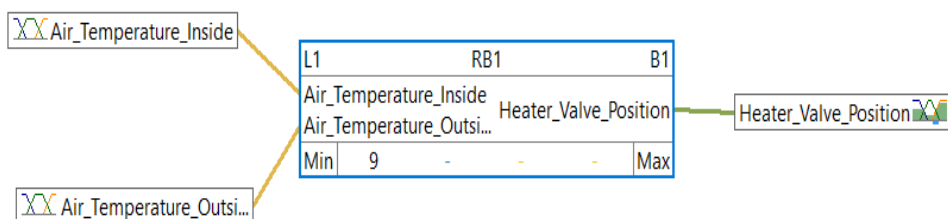


Рисунок 2.32 – Структурна схема нечіткої експертної системи (НЕС) у програмі Fuzzy TECH

Видно три змінні і база правил, які пов'язані між собою.

5) Запустимо програму і проаналізуємо отримані результати системи при деяких довільних заданих значеннях.

Нехай поставимо змінній температури повітря зовнішнього середовища та змінній температури повітря всередині приміщення залів фітнес – клуба значення: 7°C і 25°C відповідно. Тоді система порекомендує оператору встановити вентиль обігрівального пристрою на зразкове значення ~ 13 відсотків від максимального робочого ходу:

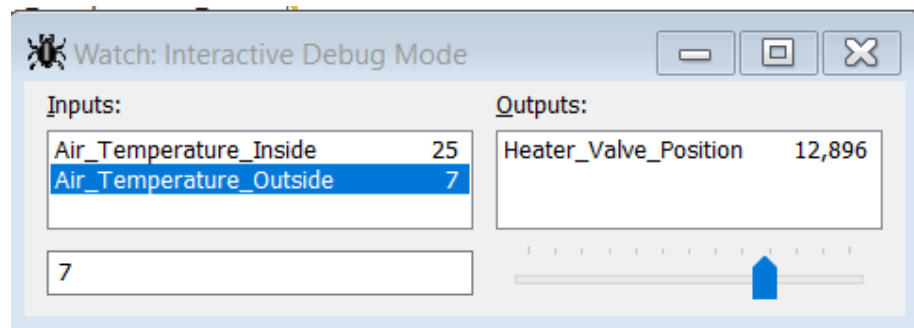


Рисунок 2.33 – Результати рекомендації НЕС на основі перших вхідних даних

Змінимо значення вхідних даних на значення 15°C та -11°C (відповідно температура повітря всередині приміщення та температура повітря зовні) та подивимося результати нечіткої системи:

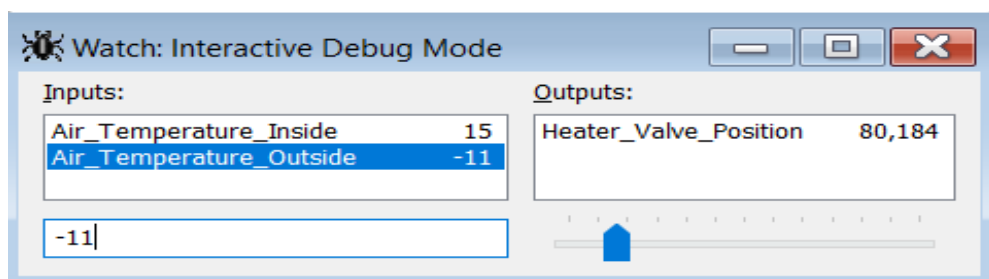


Рисунок 2.34 – Результати рекомендації системи з урахуванням других вхідних даних

Додатково, для наочності, продемонструємо графічну інтерпретацію залежності у вигляді тривимірного графіка:

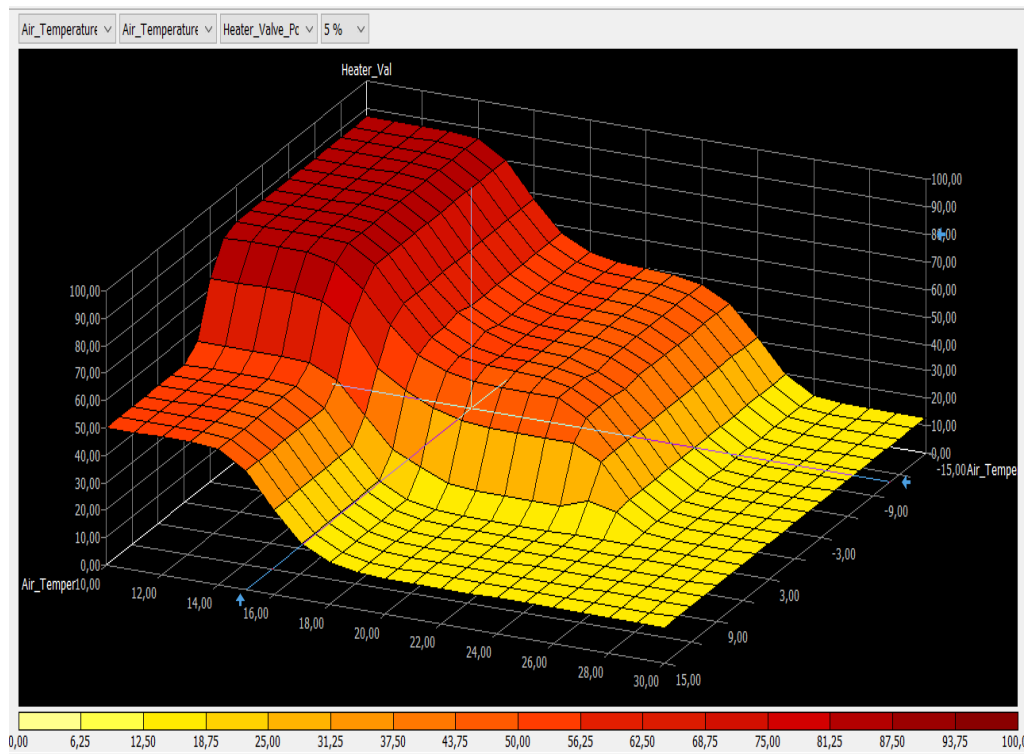


Рисунок 2.35 – Графічна тривимірна інтерпретація

До функцій комп'ютерно-інтегрованої САУ температурою повітря ТРЦ круїзного судна також можна віднести:

- безперервний моніторинг стану системи вентиляції;
- контроль та управління системою вентиляції;
- безперервне відображення даних;
- оповіщення про аварії, нештатні ситуації;
- архівування даних у СУБД;
- надання даних користувачам через Інтернет.

- робота разом з системою пожежогасіння.

Після отримання поради від НЕС, оператор КІСУ за допомогою мнемосхеми SCADA – системи (рис.10. 18) змінює положення клапана гарячої води до 82,5 %, з метою збільшення температури повітря в спортивному залі до 20 °С.

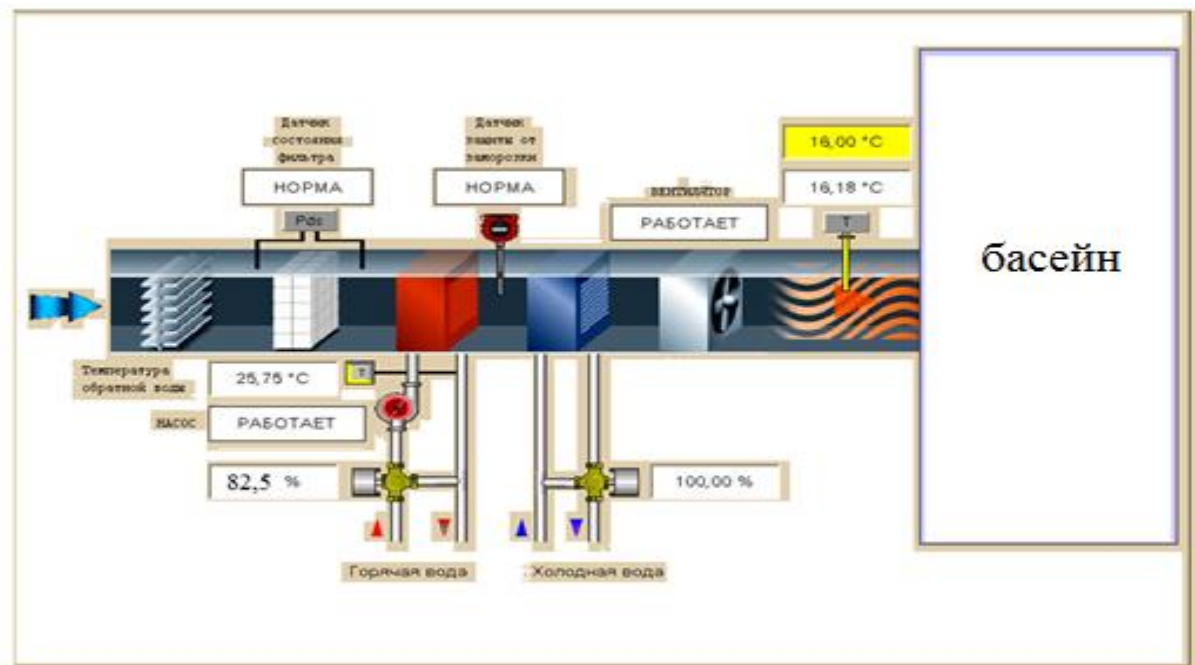


Рисунок 2.36 – Мнемосхема процесу повітряного опалення для басейну фітнес-клуба

Було побудовано нечітку експертну систему клімат- контролю для порад оператору фітнес -центру. У НЕС використовувалось два вхідних параметра: температура зовнішнього середовища та температура внутрішнього середовища, та один вихідний – положення клапана.

Як можна бачити за результатами імітаційного моделювання, НЕС працює коректно та виконує поставлені завдання по зміні положення регулюючого клапана. Вирішення задачі полягає в підвищенні ефективності засобів управління системою припливної вентиляції та підтримці

оптимальної температури повітря в різних зонах фітнес –центру згідно рекомендації санітарних стандартів та вподобань відвідувачів.

В процесі розробки НЕС встановлено, що для отримання правильних рішень по вибору керуючого впливу рекомендується:

- використовувати алгоритм І. Мамдані;
- у процесі фазифікації вхідних і вихідних змінних - використовувати гаусівський тип функцій власності;
- мінімальна кількість терм – множин – три функції належності;
- вагові коефіцієнти кожного правила в базі знань однакові та дорівнюють 100;
- метод дефазифікації – метод центру тяжкості.

За даними компаній розробників САУ систем клімат – контролю [28-31], впровадження SCADA – систем з функцію експертних порад операторам також може привести до зниження часу реакції на позаштатні ситуації до 90%; зниження збитків від позаштатних ситуацій на 50 %; зниження зносу обладнання на 15% та підвищення рівня комфорту у приміщеннях.

В случае оснащения рабочего места оператора компьютерно – интегрированной системой, ЛПР, наблюдая текущую информацию в окне просмотра, может с помощью интерфейса осуществлять дистанционное управление оборудованием (супервизорный режим).

3. СИСТЕМА ДІАГНОСТИКИ ОБЛАДНАННЯ КОНДИЦІОНЕРІВ

3.1 Аналіз можливих причин несправності обладнання спліт-системи

Спліт-системи промислового типу можуть кондиціонувати повітря в кількох кімнатах (офісний будинок) або у просторому приміщенні (конференц-зал, закритий стадіон, виробничий цех тощо). Таке обладнання спроектовано для обробки та кондиціювання значних обсягів повітря та навантаження на промислові кондиціонери більше, ніж на побутові, особливо у випадках, коли спліт-система встановлена на виробничому підприємстві, де повітря насичене пилом та газами. Тривала та ефективна робота цього обладнання потребує своєчасного обслуговування промислових кондиціонерів, щоб усунути можливі несправності та не погіршувати мікроклімат у приміщеннях, де працює спліт-система (рис.4.1).



Рисунок 3.1 – Перевірка параметрів спліт-системи

Головним чином проблеми та несправності кліматичного обладнання виникають через відсутність регулярного огляду та діагностики, а ігнорування початкових несправностей може призвести до того, що користувачам потрібно

звертатися вже не за діагностикою, а за повноцінним ремонтом промислових кондиціонерів.

Щоб запобігти несправності в роботі кліматичного обладнання пропонується створення системи діагностики (експертної системи), що діє на основі операцій нечіткої логіки та враховує досвід та знання експертів з обслуговування та ремонту.

Ознаки несправності промислових кондиціонерів

Якщо промислова спліт-система довго не отримує обслуговування, то потім пил, грязь та жиrowі відкладення накопичуються в блоках системи та шкодять роботі кондиціонера. Основні ознаки, що час замовити ремонт промислових кондиціонерів, виглядають так:

- Нестача холодоагенту - пристрій працює, але недостатньо охолоджує приміщення або не працює в режимі охолодження, помітні маслянисті патьоки на корпусі кондиціонера, обмерзає випарник внутрішнього блоку;
- несправність компресора – при включенні не чути шуму двигуна, відсутня подача холоду;
- поломка вентиляторів - пристрій перезапускається самостійно через короткі проміжки часу, обмерзають теплообмінні батареї спліт-системи, повітря не надходить до приміщення;
- вихід з ладу електронних компонентів – кондиціонер не вмикається, самостійно перезапускається або працює безперервно, без циклу перемикань.

Причин того, що спліт-система вийшла з ладу, кілька. Це і не професійний монтаж, і відсутність належного догляду та обслуговування техніки, підвищена забрудненість повітря, нестача холодоагенту та механічні несправності. Якщо в роботі спліт-системи почалися збої, то їх не варто ігнорувати і звернутися до фахівців ТРІАТЕК, які обслуговують промислові кондиціонери на професійному рівні. Своєчасна діагностика проблем у роботі кліматичної системи допоможе запобігти серйозним несправностям, адже тоді знадобиться вже не тільки сервіс, а й ремонт, який може бути дорожчим за очищення та налаштування техніки.

Обслуговування кліматичного обладнання починається з перевірки стану всіх деталей та запчастин кондиціонера (з допомогою запропонованої експертної системи) та їх заміни, якщо компоненти остаточно вийшли з ладу. Після цього проводиться чищення кондиціонера очищають деталі та блоки кондиціонера, щоб усунути пил, бруд, плісняву та бактерії. Це дозволяє покращити якість повітря у приміщенні та продовжити термін служби обладнання. Далі здійснюють перевірку рівня холодоагенту та заправку ним, у разі потреби. Це не тільки гарантує ефективнішу роботу кондиціонера, а й дозволяє знизити витрати на електроенергію. Після цього відбувається перевірка дренажної системи та електронних компонентів. Промислове обладнання, зокрема, кондиціонери, потребує не лише професійного монтажу, а й своєчасного обслуговування та ремонту. Своєчасна діагностика та ремонт промислового кондиціонера у виконанні наших фахівців дозволить продовжити час його ефективної роботи та знизити витрати на усунення несправностей, адже профілактика завжди дешевша, ніж усунення проблем з технікою.

3.2. Створення експертної системи для пошуку причин несправності

Постановка задачі: розглядаємо системи нечітких продукцій. І як приклад, буде вивчена ситуація пошуку несправностей в системі клімат – котролю (обладнання системи кондиціонування) за допомогою обліку знань експерта - електромеханіка.

З цією метою як причини або причини несправності розглядаємо безліч $X = \{x_1, x_2, x_3, x_4\}$, в якому x_1 - "несправність контролера", x_2 - "несправність компресора", x_3 - "несправність вентилятора", x_4 - "несправність системи холодоагенту". Як другий універсум розглянемо безліч висновків або проявів несправності, $Y = \{y_1, y_2, y_3\}$ де y_1 - "кондиціонер не запускається", y_2 -

"кондиціонер працює нестійко", y_3 - "кондиціонер не розвиває повної потужності".

Причинний взаємозв'язок між безліччю передумов і безліччю наслідків представлений у вигляді бінарного нечіткого відношення $P = \{ \langle x_i, y_j \rangle, \mu_P(\langle x_i, y_j \rangle) \}$, заданого у формі матриці M_P цього нечіткого

$$\text{відношення : } M_P = \begin{pmatrix} 1 & 0.1 & 0.2 \\ 0.8 & 0.9 & 1 \\ 0.7 & 0.8 & 0.5 \\ 1 & 0.5 & 0.2 \end{pmatrix} \quad (3.2.1)$$

Перепишемо для наочності матрицю нечітких відносин як правил нечітких продукцій с значеннями вагових коефіцієнтів з точки зору експерта F :

- 1) ЯКЩО « y_1 », ТО « x_1 » ($F_1 = 1.0$).
- 2) ЯКЩО « y_1 », ТО « x_2 » ($F_2 = 0.8$).
- 3) ЯКЩО « y_1 », ТО « x_3 » ($F_3 = 0.7$).
- 4) ЯКЩО « y_1 », ТО « x_4 » ($F_4 = 1.0$).
- 5) ЯКЩО « y_2 », ТО « x_1 » ($F_5 = 0.1$).
- 6) ЯКЩО « y_2 », ТО « x_2 » ($F_6 = 0.9$).
- 7) ЯКЩО « y_2 », ТО « x_3 » ($F_7 = 0.8$).
- 8) ЯКЩО « y_2 », ТО « x_4 » ($F_8 = 0.5$).
- 9) ЯКЩО « y_3 », ТО « x_1 » ($F_9 = 0.2$).
- 10) ЯКЩО « y_3 », ТО « x_2 » ($F_{10} = 1.0$).
- 11) ЯКЩО « y_3 », ТО « x_3 » ($F_{11} = 0.5$).
- 12) ЯКЩО « y_3 », ТО « x_4 » ($F_{12} = 0.2$).

Тепер припустимо, що результати огляду конкретного обладнання показують, що двигун не запускається, хоча раніше працював стійко та розвивав повну потужність. У дужках вказані міри істинності відповідних нечітких висловлювань для перевірки ситуації (T).

Кондиціонер не запускається ($T = 0.1$).

Кондиционер працює нестійко ($T = 0.3$).

Кондиционер не розвиває повну потужність ($T = 0.8$).

Розглянемо один із можливих способів вирішення даної проблеми з використанням прямого методу виведення, правила для оцінки однакових висновків. З цією метою послідовно розглянемо усі правила отримання відповідних висновків. Метод реалізує алгоритм максиміна.

Напишемо мовою програмування «Python» програму, яка реалізує вирішення поставленого завдання у середовищі розробки «Google COLAB» (рис. 3.3-3.4.) . Код програми з коментарями наведено нижче:

```
import math
def calculate_Fi(last_digit):
    F_values = [1.0, 0.8, 0.7, 1.0, 0.1, 0.9, 0.8, 0.5, 0.2, 1.0, 0.5, 0.2]
    modified_F_values = []
    for F in F_values:
        modified_F = F - (last_digit / 20)
        modified_F_values.append(modified_F)
    return modified_F_values
def calculation(rules, situations):
    modified_F_values = calculate_Fi(2)
    on_rule_1 = (min(situations[0][0], rules[0][0]), "Кондиционер не запускається
через проблеми з контролером.")
    on_rule_2 = (min(situations[0][0], rules[1][0]), " Кондиционер не
запускається через проблеми з копресором.")
    on_rule_3 = (min(situations[0][0], rules[2][0]), "Кондиционер не запускається
через проблеми з вентилятором.")
    on_rule_4 = (min(situations[0][0], rules[3][0]), "Кондиционер не запускається
через проблеми із системою хлорогента.")
    on_rule_5 = (min(situations[1][0], rules[4][0]), "Кондиционер працює
нестійко через проблеми з контролером.")
```

```
on_rule_6 = (min(situations[1][0], rules[5][0]), "Кондиционер працює  
нестійко через проблеми з компресором.")
```

```
on_rule_7 = (min(situations[1][0], rules[6][0]), "Кондиционер працює  
нестійко через проблеми з вентилятором.")
```

```
on_rule_8 = (min(situations[1][0], rules[7][0]), "Кондиционер працює  
нестійко через проблеми із системою хладагенту.")
```

```
on_rule_9 = (min(situations[2][0], rules[8][0]), "Кондиционер не розвиває  
повної потужності через проблеми з контролером.")
```

```
on_rule_10 = (min(situations[2][0], rules[9][0]), "Кондиционер не розвиває  
повної потужності через проблеми з компресором.")
```

```
on_rule_11 = (min(situations[2][0], rules[10][0]), "Кондиционер не розвиває  
повної потужності через проблеми з вентилятором.")
```

```
on_rule_12 = (min(situations[2][0], rules[11][0]), "Кондиционер не розвиває  
повної потужності через проблеми із системою хладагенту.")
```

```
# Порівнюємо змінені значення Fi
```

```
result_value = (max(max(on_rule_1[0], on_rule_2[0]), max(on_rule_3[0],  
on_rule_4[0])), "Проблема полягає або в несправності невідомих елементів -  
звертайтеся в сервісний центр!")
```

```
if result_value[0] >= 0.9:
```

```
    result_value = (result_value[0], " Проблема полягає у несправності  
системи хладагенту.")
```

```
elif result_value[0] >= 0.7:
```

```
    result_value = (result_value[0], " проблема полягає у несправності системи  
компресора.")
```

```
elif result_value[0] >= 0.5:
```

```
    result_value = (result_value[0], " проблема полягає у несправності системи  
вентилятора.")
```

```
else:
```

```
    result_value = (result_value[0], " проблема полягає у несправності  
контролера.")
```

```

return result_value
def print_result(result):
    print(f"Значення істинності: {result[0]}\nРезультат: {result[1]}")
def main():
    y_1 = "кондиционер не запускається"
    y_2 = "кондиционер працює нестійко"
    y_3 = "кондиционер не розвиває повної потужності"
    x_1 = "несправність контролера"
    x_2 = "несправність компресора"
    x_3 = "низька якість хлорогенту"
    x_4 = "несправність вентилятора"
    rule_1 = (calculate_Fi(5)[0], f"ЯКІ {y_1}, ТО {x_1}.")
    rule_2 = (calculate_Fi(5)[1], f"ЯКЩО {y_1}, ТО {x_2}.")
    rule_3 = (calculate_Fi(5)[2], f"ЯКІ {y_1}, ТО {x_3}.")
    rule_4 = (calculate_Fi(5)[3], f"ЯКЩО {y_1} ТО {x_4}.")
    rule_5 = (calculate_Fi(5)[4], f"ЯКІ {y_2}, ТО {x_1}.")
    rule_6 = (calculate_Fi(5)[5], f"ЯКІ {y_2}, ТО {x_2}.")
    rule_7 = (calculate_Fi(5)[6], f"ЯКІ {y_2}, ТО {x_3}.")
    rule_8 = (calculate_Fi(5)[7], f"ЯКІ {y_2}, ТО {x_4}.")
    rule_9 = (calculate_Fi(5)[8], f"ЯКІ {y_3}, ТО {x_1}.")
    rule_10 = (calculate_Fi(5)[9], f"ЯКІ {y_3}, ТО {x_2}.")
    rule_11 = (calculate_Fi(5)[10], f"ЯКІ {y_3}, ТО {x_3}.")
    rule_12 = (calculate_Fi(5)[11], f"ЯКІ {y_3}, ТО {x_4}.")
    situation_1 = (0.9, f"{y_1}.")
    situation_2 = (0.1, f"{y_2}.")
    situation_3 = (0.2, f"{y_3}.")
    rules = [rule_1, rule_2, rule_3, rule_4, rule_5, rule_6, rule_7, rule_8, rule_9,
rule_10, rule_11, rule_12]
    situations = [situation_1, situation_2, situation_3]
    result = calculation(rules, situations)

```

```

print_result(result)

if __name__ == "__main__":
    main()

```

Результати роботи програми продемонструємо у вигляді скриншота консолі (рис.3.2):

```

➔ Значення істинності: 0.75
Результат: проблема полягає у несправності системи компресора.

```

Рисунок 3.2 - Висновки нечіткої експертної системи (діагностики)

```

import math

def calculate_Fi(last_digit):
    F_values = [1.0, 0.8, 0.7, 1.0, 0.1, 0.9, 0.8, 0.5, 0.2, 1.0, 0.5, 0.2]
    modified_F_values = []
    for F in F_values:
        modified_F = F - (last_digit / 20)
        modified_F_values.append(modified_F)
    return modified_F_values

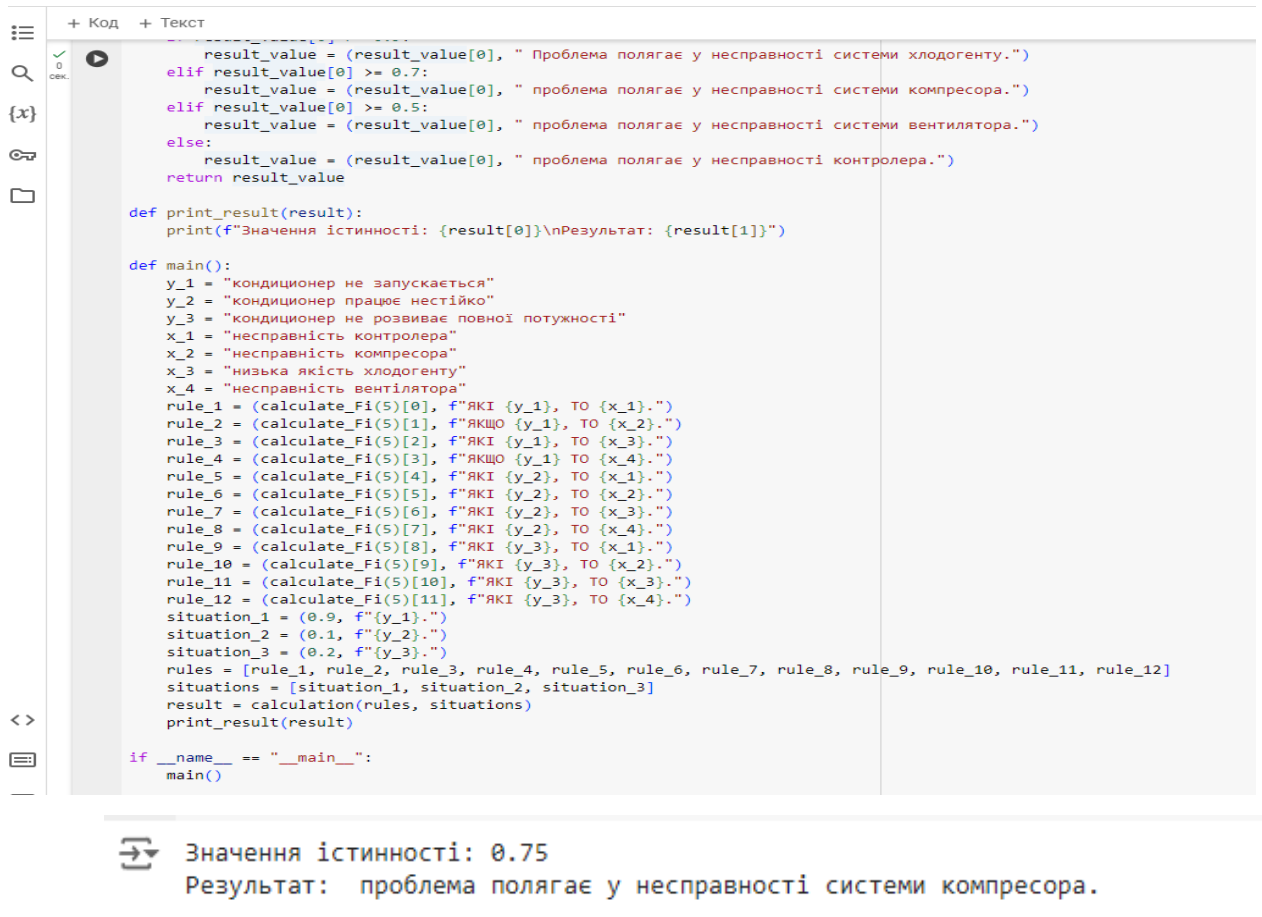
def calculation(rules, situations):
    modified_F_values = calculate_Fi(2)
    on_rule_1 = (min(situations[0][0], rules[0][0]), "Кондиционер не запускається через проблеми з контролером.")
    on_rule_2 = (min(situations[0][0], rules[1][0]), "Кондиционер не запускається через проблеми з компресором.")
    on_rule_3 = (min(situations[0][0], rules[2][0]), "Кондиционер не запускається через проблеми з вентилятором.")
    on_rule_4 = (min(situations[0][0], rules[3][0]), "Кондиционер не запускається через проблеми із системою хладогенту.")
    on_rule_5 = (min(situations[1][0], rules[4][0]), "Кондиционер працює нестійко через проблеми з контролером.")
    on_rule_6 = (min(situations[1][0], rules[5][0]), "Кондиционер працює нестійко через проблеми з компресором.")
    on_rule_7 = (min(situations[1][0], rules[6][0]), "Кондиционер працює нестійко через проблеми з вентилятором.")
    on_rule_8 = (min(situations[1][0], rules[7][0]), "Кондиционер працює нестійко через проблеми із системою хладогенту.")
    on_rule_9 = (min(situations[2][0], rules[8][0]), "Кондиционер не розвиває повної потужності через проблеми з контролером.")
    on_rule_10 = (min(situations[2][0], rules[9][0]), "Кондиционер не розвиває повної потужності через проблеми з компресором.")
    on_rule_11 = (min(situations[2][0], rules[10][0]), "Кондиционер не розвиває повної потужності через проблеми з вентилятором.")
    on_rule_12 = (min(situations[2][0], rules[11][0]), "Кондиционер не розвиває повної потужності через проблеми із системою хладогенту.")
    # Порівнюємо змінені значення Fi
    result_value = (max(max(on_rule_1[0], on_rule_2[0]), max(on_rule_3[0], on_rule_4[0])), "Проблема полягає або в несправності невідомих елементів - звертайтеся в сервісний центр!")
    if result_value[0] >= 0.9:
        result_value = (result_value[0], "Проблема полягає у несправності системи хладогенту.")
    elif result_value[0] >= 0.7:
        result_value = (result_value[0], "проблема полягає у несправності системи компресора.")
    elif result_value[0] >= 0.5:
        result_value = (result_value[0], "проблема полягає у несправності системи вентилятора.")
    else:
        result_value = (result_value[0], "проблема полягає у несправності контролера.")
    return result_value

def print_result(result):
    print(f"Значення істинності: {result[0]}\nРезультат: {result[1]}")

def main():
    y_1 = "кондиционер не запускається"
    y_2 = "кондиционер працює нестійко"
    y_3 = "кондиционер не розвиває повної потужності"
    x_1 = "несправність контролера"

```

Рисунок 3.2 – Тестування програми у Colab



```

+ Код + Текст
0 сек.
result_value = (result_value[0], " Проблема полягає у несправності системи хлодогенту.")
elif result_value[0] >= 0.7:
    result_value = (result_value[0], " проблема полягає у несправності системи компресора.")
elif result_value[0] >= 0.5:
    result_value = (result_value[0], " проблема полягає у несправності системи вентилятора.")
else:
    result_value = (result_value[0], " проблема полягає у несправності контролера.")
return result_value

def print_result(result):
    print(f"Значення істинності: {result[0]}\nРезультат: {result[1]}")

def main():
    y_1 = "кондиционер не запускається"
    y_2 = "кондиционер працює нестійко"
    y_3 = "кондиционер не розвиває повної потужності"
    x_1 = "несправність контролера"
    x_2 = "несправність компресора"
    x_3 = "низька якість хлодогенту"
    x_4 = "несправність вентилятора"
    rule_1 = (calculate_Fi(5)[0], f"ЯКІ {y_1}, ТО {x_1}.")
    rule_2 = (calculate_Fi(5)[1], f"ЯКЩО {y_1}, ТО {x_2}.")
    rule_3 = (calculate_Fi(5)[2], f"ЯКІ {y_1}, ТО {x_3}.")
    rule_4 = (calculate_Fi(5)[3], f"ЯКЩО {y_1}, ТО {x_4}.")
    rule_5 = (calculate_Fi(5)[4], f"ЯКІ {y_2}, ТО {x_1}.")
    rule_6 = (calculate_Fi(5)[5], f"ЯКІ {y_2}, ТО {x_2}.")
    rule_7 = (calculate_Fi(5)[6], f"ЯКІ {y_2}, ТО {x_3}.")
    rule_8 = (calculate_Fi(5)[7], f"ЯКІ {y_2}, ТО {x_4}.")
    rule_9 = (calculate_Fi(5)[8], f"ЯКІ {y_3}, ТО {x_1}.")
    rule_10 = (calculate_Fi(5)[9], f"ЯКІ {y_3}, ТО {x_2}.")
    rule_11 = (calculate_Fi(5)[10], f"ЯКІ {y_3}, ТО {x_3}.")
    rule_12 = (calculate_Fi(5)[11], f"ЯКІ {y_3}, ТО {x_4}.")
    situation_1 = (0.9, f"{y_1}.")
    situation_2 = (0.1, f"{y_2}.")
    situation_3 = (0.2, f"{y_3}.")
    rules = [rule_1, rule_2, rule_3, rule_4, rule_5, rule_6, rule_7, rule_8, rule_9, rule_10, rule_11, rule_12]
    situations = [situation_1, situation_2, situation_3]
    result = calculation(rules, situations)
    print_result(result)

if __name__ == "__main__":
    main()

```


 Значення істинності: 0.75
 Результат: проблема полягає у несправності системи компресора.

Рисунок 3.3 – Висновки програми

У роботі застосували систему нечітких продукцій для діагностики несправностей в системі кондиціонування. Використовуючи знання експерта - електромеханіка, ми встановили зв'язок між причинами (несправності контролера, компресора, вентилятора та системи хлодогенту) і наслідками (не запуск кондиционера, нестійка робота і недостатня потужність). Ми розробили програму, яка обчислює істинність кожного правила та визначає можливість того, що проблема полягає у несправності системи компресора. Цей підхід дозволяє ефективно вирішувати складні ситуації за умови обмежених даних та використовувати досвід експерта.

ВИСНОВКИ

Представлена дипломна робота спрямована на розробку і реалізацію системи управління параметрами температури та вологості у приміщеннях фітнес-центру. Було проаналізовано основні фактори, що впливають на роботу системи кондиціонування та встановлено нормативні показники для різних температурних зон. Вибір методів теорії штучного інтелекту (нечіткої логіки) дозволило розробити програмне забезпечення для контролера системи управління з урахуванням знань експертів з експлуатації та думок клієнтів. Створено експертну систему, яка дозволяє допомагати в управлінні процесом вентиляції оператору. Було проаналізовано основні причино-слідчі фактори, що впливають на роботу обладнання спліт-системи. З урахуванням цих чинників створено експертну систему діагностики причин несправності.

На нашу думку застосування системи управління температурою повітря допомагає керівникам фітнес-центрів знизити витрати на управління та технічне обслуговування обладнання систем кондиціонування, покращити роботу з клієнтами, та забезпечити безперебійну роботу центру.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://olvent.ua/stati/stati-o-ventiltjatsii/ventiljatsija-sportzala/>
2. <http://intercon.com.ua/ua/o-proizvodstve/>
3. Розподілені мікропроцесорні системи: конспект лекцій [Електронний ресурс]: для підготовки докторів філософії в галузі знань 17 Електроніка та телекомунікація за спеціальністю 171 Електроніка за спеціалізацією «Електронні системи» / КПІ ім. Ігоря Сікорського ; уклад.: Т. О. Терещенко– Київ : КПІ ім. Ігоря Сікорського, 2018. – 192 с
4. Пупена О.М. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI: підручник / За ред. О.М. Пупени. – К.: «Ліра-К», 2020. – 594 с.
5. Mamdani, E. H. Application of fuzzy algorithms for control of simple dynamic plant / Proc. Inst. Elect. Eng. Contr. Sci. 1974. vol. 121. pp. 1585 – 1588.
6. Zadeh, L. A. Fuzzy sets / Information and Control. 1965. №8. pp. 338 – 353.
7. Михайленко В.С. Аналіз методів створення бази правил нечіткого контролера / Михайленко В.С. // Холодильна техніка і технології. – 2009. – № 2 (118). – С. 77 – 82.
8. Сухіна О.О., Гунченко Ю.О. Нечітка система управління клімат - контролем фітнес - центру / Інформатика, інформаційні системи та технології: тези доповідей двадцять першої всеукраїнської конференції студентів і молодих науковців. Одеса, 26 квітня 2024 р. – Одеса, 2024.
9. fuzzyTECH [Електронний ресурс]. 2023. Режим доступу до ресурсу: <https://www.fuzzytech.com>.
10. Швидкість реакції оператора і підвищення ефективності з iFIX [Електронний ресурс] DIGITAP, 2023. Режим доступу до ресурсу: <https://digitap.com.ua/company-products/hmi-scada-ifix/>.
11. Кацадзе Т. Л. Експертні системи прийняття рішень в енергетиці: навч. посіб. / Т. Л. Кацадзе. К.: ЛОГОС, 2014. - 173 с.

12. Bolloju, N. et al. A knowledge-based system for improving the consistency between object models and use case narratives / Expert Systems with Applications. 2019. vol. 39, pp. 9398-9410.

ДОДАТОК А

Програма для отримання значення своєї зони комфорту.

Напишемо потрібну програму мовою програмування С# в середовищі програмування Visual Studio 2019 використовуючи шаблон Додаток Windows Forms (.NET Framework).

Наведемо код програми з коментарями нижче:

```
public partial class Form1 : Form
{
    Public Form1()
    {
        InitializeComponent();
        // Викликаємо обробник.
        this.Load += Form1_Load;
    }
    #region Поля: змінні та константи.
    /// <summary>
    /// Ліва межа графіка функції.
    /// </summary>
    private double XMin = 0;
    /// <summary>
    /// Права межа графіка функції.
    /// </summary>
    private double XMax = 45;
    /// <summary>
    /// Крок графіка функції.
    /// </summary>
    private double Step = 1;
    /// <summary>
    /// Масив значень X.
    /// </summary>
    private double[] x;
    /// <summary>
    /// Масиви значень Y.
    /// </summary>
    private double[] y1;
    private double[] y2;
    /// <summary>
    /// Глобальну змінну типу Chart.
    /// </summary>
```

```

static Chart chart;
#endregion
#Region Методи.
/// <summary>
/// Розрахунок значень графіка функції.
/// </summary>
private void Calculate()
{
    // Створюємо масиви відповідних розмірів.
    x = new double[10];
    y1 = new double[10];
    y2 = new double[10];
    // Розраховуємо крапки для графіків функції.
    x[0] = 0;
    y1[0] = 0;
    x[1] = 5;
    y1[1] = 0.25;
    x[2] = 10;
    y1[2] = 0.5;
    x[3] = 15;
    y1[3] = 0.75;
    x[4] = 20;
    y1[4] = 1;
    x[5] = 25;
    y1[5] = 1;
    x[6] = 30;
    y1[6] = 0.75;
    x[7] = 35;
    y1[7] = 0.5;
    x[8] = 40;
    y1[8] = 0.25;
    x[9] = 45;
    y1[9] = 0;
    for(int i = 0; i < 10; i++)
    {
        y2[i] = Math.Pow(y1[i], 2);
    }
}
/// <summary>
/// Створюємо елемент управління Chart та налаштовуємо
його.
/// </summary>
private void CreateChart()

```

```

{
    // Створюємо новий елемент керування Chart.
    chart = New Chart ();
    // Поміщаємо його форму.
    chart.Parent = this;
    // Задаємо розміри елемента.
    chart.SetBounds(10, 10, ClientSize.Width - 20,
ClientSize.Height - 20);
    DataPointCustomProperties points = New
DataPointCustomProperties();
    // Створюємо нову область для побудови графіка.
    ChartArea area = New ChartArea();
    // Даємо їй ім'я (щоб потім додавати графіки).
    area.Name = "myGraph";
    // Задаємо ліву та праву межі осі X.
    area.AxisX.Minimum = XMin;
    area.AxisX.Maximum = XMax;
    // Визначаємо крок сітки.
    area.AxisX.MajorGrid.Interval = Math.Abs(Step);
    // Підписи до осей.
    area.AxisX.Title = "T, °C";
    area.AxisY.Title = "μ(T)";
    // Задаємо шрифт та стиль підписів осей графіка.
    area.AxisX.TitleFont = новий Font("Times New Roman",
12, FontStyle.Bold);
    area.AxisY.TitleFont = новий Font("Times New Roman",
12, FontStyle.Bold);
    // Додаємо область у діаграму.
    chart.ChartAreas.Add(area);
    // Створюємо об'єкт для графіка.
    Series series1 = new Series();
    // Посилаємося область для побудови графіка.
    series1.ChartArea = "myGraph";
    // Задаємо тип графіка - лінії.
    series1.ChartType = SeriesChartType.Line;
    // Вказуємо ширину лінії графіка.
    series1.BorderWidth = 4;
    // Назва графіка для відображення у легенді.
    series1.LegendText = "Рівень звичайного комфорту";
    // Додаємо колекцію крапок.
    series1.Points.Add(y1);
    // Додаємо до списку графіків діаграми.
    chart.Series.Add(series1);

```

```

        // Завдання іміджу точок.
        chart.Series[0].MarkerStyle = MarkerStyle.Circle;
        // Завдання розміру точок.
        chart.Series[0].MarkerSize = 10;
        // Завдання кроку між точками.
        chart.Series[0
].MarkerStep = 1;
        // Завдання кольору крапок.
        chart.Series[0].MarkerColor = Color.Red;
        // Аналогічні дії другого графіка.
        Series series2 = new Series();
        series2.ChartArea = "myGraph";
        series2.ChartType = SeriesChartType.Line;
        series2.BorderWidth = 4;
        series2.LegendText = "Підвищений комфорт.";
series2.Points.Add(y2);
        chart.Series.Add(series2);
        chart.Series[1].MarkerStyle = MarkerStyle.Square;
        chart.Series[1].MarkerSize = 10;
        chart.Series[1].MarkerStep = 1;
        chart.Series[1].MarkerColor = Color.Green;
// Створюємо легенду, яка показуватиме назви.
        Legend legend = New Legend ();
        chart.Legends.Add(legend);
        // Задаємо шрифт та стиль легенди.
        chart.Legends[0].Font = новий Font("Times New
Roman", 12, FontStyle.Bold);
    }
    /// <summary>
    /// Обробник подій форми.
    /// </summary>
    private void Form1_Load(object sender, EventArgs e)
    {
        // Створюємо елемент управління.
        CreateChart();
        // Розраховуємо значення точок графіка функції.
        Calculate();
        // Додаємо обчислені значення графіки функцій.
        chart.Series[0].Points.DataBindXY(x, y1);
        chart.Series[1].Points.DataBindXY(x, y2);
    }
#endregion
}

```

Продемонструємо скріншот виконаної програми:

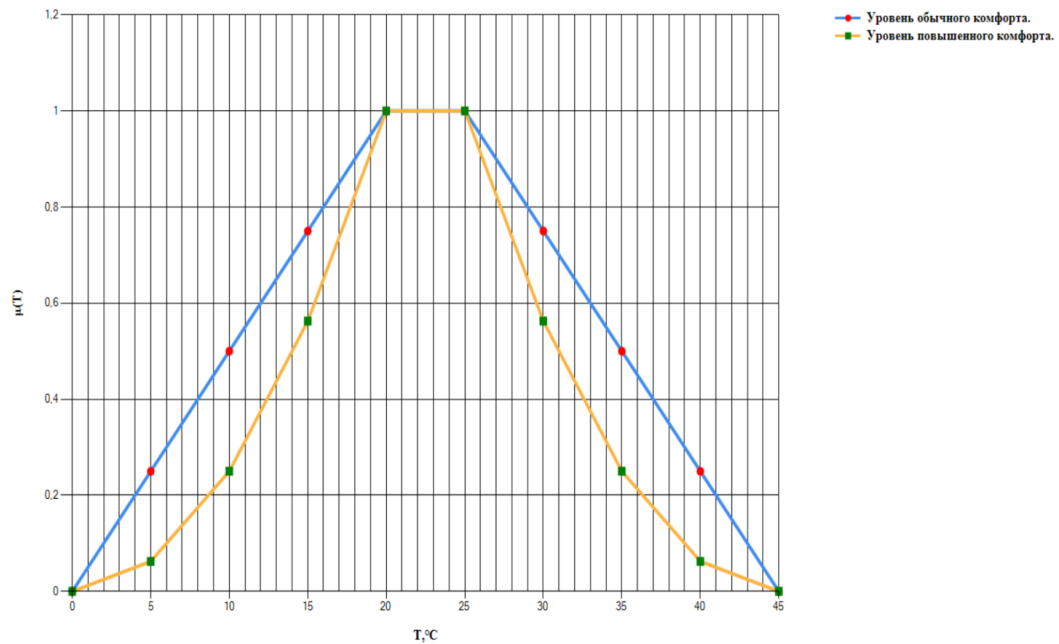


Рисунок 1. - Зони комфорту (для програмування контролера)

Параметр $\mu(T)$ визначає рівень комфорту споживача, де $\mu(T) \in [0;1]$. "0" - відповідає рівню мінімального комфорту (абсолютно неприйнятний), а "1" - відповідає рівню максимального комфорту (абсолютно прийнятно). Другий параметр $T, ^\circ\text{C}$ – температура, яка задана в межах від 0 градусів за Цельсієм до 45 градусів за Цельсієм.

Ми можемо бачити, що перший графік синього кольору «Рівень звичайного комфорту» визначає найкращу температуру 20-25 градусів за Цельсієм і найбільш некомфортну температуру в крайніх точках 0 і 45 градусів за Цельсієм. Інші значення приймають проміжні значення від 0 до 1, які можна легко встановити за графіком.

Другий графік жовтого кольору «Рівень підвищеного комфорту» виділяє більш виразно прийнятну температуру.