

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

Гайкова Ілона Віталіївна

**Навчально-методичний комплекс з дисципліни
«Програмні засоби моделювання для комп'ютерної
інженерії» / Educational and methodical complex of the
discipline "Modeling software for Computer Engineering"**

напрямок підготовки: 123 Комп'ютерна інженерія
фахове спрямування - Комп'ютерна інженерія
Бакалаврська робота

Виконав студент групи КСМ 43/2
Гайкова Ілона Віталіївна

Науковий керівник:
Савка Н.Я.

Тернопіль - 2018

РЕЗЮМЕ

Дипломний проект містить 54 сторінки пояснюючої записки, 13 рисунків, 7 таблиць, 3 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Мета проекту полягає в розробленні навчально-методичного комплексу із дисципліни «Програмні засоби моделювання для комп'ютерної інженерії». При цьому одержано такі практичні результати:

1. Розроблено структуру та зміст комплексу лабораторних робіт із дисципліни, що розкриває основні практичні та теоретичні знання та вміння студентів.

2. Розроблено структуру та зміст лабораторної роботи й зазначено основні пункти, які має містити лабораторна робота для розуміння та практичного виконання.

3. Проаналізовано основні програмні середовища для моделювання комп'ютерних систем й показано, що в основному вони налаштованні на розв'язання вузького кола задач.

4. Охарактеризовано основні переваги та можливості пакету прикладних програм Matlab для розв'язування інженерних задач.

5. Реалізовано комплекс лабораторних робіт із дисципліни у середовищі Matlab, проведено аналіз результатів виконання робіт.

Ключові слова: МОДЕЛЮВАННЯ, ПРОГРАМНІ ЗАСОБИ, КОМПЛЕКС ЛАБОРАТОРНИХ РОБІТ.

RESUME

The diploma project contains 54 pages of explanatory note, 13 figures, 7 tables, 3 appendices. Volume of graphic material 2 sheets of A3 format.

The aim of the project is to develop an educational and methodological complex in the discipline "Software modeling tools for computer engineering". The following practical results were obtained:

1. The structure and content of a set of laboratory works in the discipline, which reveals the basic practical and theoretical knowledge and skills of students.

2. The structure and content of laboratory work are developed and the basic points which laboratory work should contain for understanding and practical performance are specified.

3. The main software environments for modeling computer systems are analyzed and it is shown that they are mainly set up to solve a narrow range of problems.

4. The main advantages and capabilities of the Matlab application package for solving engineering problems are described.

5. The complex of laboratory works on discipline in the Matlab environment is realized, the analysis of results of performance of works is carried out.

Keywords: SIMULATION, SOFTWARE, COMPLEX OF LABORATORY WORKS.

ЗМІСТ

Вступ.....	3
1 Аналіз структури, змісту та програмних засобів реалізації лабораторних робіт..	5
1.1 Роль дисципліни “Програмні засоби моделювання для комп’ютерної інженерії” в структурі навчальних планів по підготовці бакалаврів.....	5
1.2 Аналіз типових структур лабораторних робіт	7
1.3 Аналіз програмних засобів для реалізації лабораторних робіт.....	11
1.4 Постановка задачі дипломного проектування	14
реалізувати комплекс лабораторних робіт в програмному середовищі Matlab;	2
Структура та зміст комплексу лабораторних робіт	14
2.1 Структура комплексу лабораторних робіт	15
2.2 Особливості пакету прикладних програм Matlab	17
2.3 Структура та зміст лабораторної роботи	20
3 Реалізація комплексу лабораторних робіт у середовищі Matlab.....	30
3.1 Приклад виконання лабораторної роботи № 2.....	30
3.2 Приклад виконання лабораторної роботи № 3.....	32
3.3 Приклад виконання лабораторної роботи № 4.....	36
3.3 Приклад виконання лабораторної роботи № 5.....	40
Висновки	47
Список використаних джерел	48

					ДП.КСМ. 07097/14.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Гайкова І.В..			НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС З ДИСЦИПЛІНИ "ПРОГРАМНІ ЗАСОБИ МОДЕЛЮВАННЯ ДЛЯ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ"	Літ.	Арк.	Акрушів
Перевір.		Савка Н.Я.				8	94	
		Паздрій І.Р.				ТНЕУ. ФКІТ. КСМ-43/26		
Н. Контр.		Гураль І.В.						
Затверд.		Березький О.М.						

ВСТУП

Швидкий розвиток комп'ютерної техніки та різного програмного забезпечення - характерні риси сучасного періоду розвитку суспільства. Комп'ютерні технології присутні майже в усіх сферах діяльності. Сьогодні фактично неможливо уявити будь-який кабінет без щоденного і широкого впровадження та використання комп'ютерних технологій.

Зважаючи на це, найбільш перспективною сьогодні є спеціальність "Комп'ютерна інженерія". Інформатизація суспільства стрімко розвивається і потреба в спеціалістах зростає з кожним днем. Сьогодні цінуються ті фахівці, які не тільки працюють за комп'ютером, а й мають навички його обслуговування і ремонту.

Комп'ютерна інженерія це - технічні засоби та системне програмне забезпечення комп'ютерних систем і мереж універсального і спеціального призначення, а також їх компоненти та прикладне програмне забезпечення для їх функціонування. Зміст діяльності фахівців з комп'ютерної інженерії полягає в розробці апаратно-програмних засобів сучасних інформаційних технологій, розробці та застосуванні комп'ютерних систем і мереж загального і спеціального призначення, їх системного програмного забезпечення, спеціалізованих комп'ютерних систем і мереж з оптимізованими параметрами, інтегрованих комп'ютерних систем, технічних засобів захисту інформації в комп'ютерних системах і мережах.

В той же ж час комп'ютерна інженерія тісно пов'язана із програмною інженерією. Адже для розробки комп'ютерних систем потрібне програмне забезпечення, яке уможливить реалізувати функції системи та спростить роботу користувача. Розробка комп'ютерних систем включає етап моделювання, при якому створюється модель реального об'єкта, яка описує ті властивості, які необхідні для розв'язку поставленої задачі. Правильно побудована модель сприяє успішній та швидкій реалізації комп'ютерної системи.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Для моделювання комп'ютерних систем існує чимало програмних засобів. В останній час широкого застосування набув пакет прикладних програм Matlab через велику кількість вбудованих тулбоксів, що об'єднують функції із певної галузі.

Зважаючи на вищеписане, метою дипломної роботи є розробка навчально-методичного комплексу із дисципліни “Програмні засоби моделювання для комп'ютерної інженерії”, яка викладається на бакалавраті спеціальності “Комп'ютерна інженерія”.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ СТРУКТУРИ, ЗМІСТУ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ЛАБОРАТОРНИХ РОБІТ

1.1 Роль дисципліни “Програмні засоби моделювання для комп’ютерної інженерії” в структурі навчальних планів по підготовці бакалаврів

Програма та тематичний план дисципліни “Програмні засоби моделювання для комп’ютерної інженерії” орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок використання теорії та методів моделювання для комп’ютерної інженерії, використання сучасних програмних засобів для проектування.

Метою дисципліни є вивчення науково-практичного інструментарію моделювання комп’ютерних систем та його застосування при розв’язуванні актуальних інженерних задач [12].

Вивчення курсу “Програмні засоби моделювання для комп’ютерної інженерії” передбачає наявність систематичних та ґрунтовних знань із суміжних курсів («Дискретна математика», «Математичний аналіз», «Теорія імовірності», «Програмування», «Алгоритми та методи обчислень», «Комп’ютерна графіка»), а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи студентів. Структурно-логічну схему із дисципліни наведено у додатку А.

Вивчення дисципліни “Програмні засоби моделювання для комп’ютерної інженерії” забезпечує такі компетентності [12]:

- знання сучасних методів і засобів моделювання для комп’ютерних систем;
- знання сучасних методів та засобів створення програм для комп’ютерних систем;
- вміти аналізувати і застосовувати сучасні методи моделювання у комп’ютерній інженерії;
- вміти використовувати методи та програмні засоби моделювання для

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

комп'ютерних систем;

- вміти вирішувати основні проблеми моделювання у комп'ютерних системах;
- вміти моделювати комп'ютерні системи із застосуванням програмних засобів.

В результаті вивчення курсу „Програмні засоби моделювання для комп'ютерної інженерії” студенти повинні [12]:

- володіти методам та засобами моделювання для комп'ютерних інженерії;
- володіти програмним середовищем пакету прикладних програм Matlab;
- володіти Neural Network Toolbox;
- володіти Fuzzy Logic Toolbox;
- моделювати штучні нейронні мережі;
- моделювати нечіткі системи;
- вирішувати поставлену задачу моделювання за допомогою сучасних програмних середовищ.

Завданням вивчення дисципліни є освоєння науково-практичного інструментарію моделювання для комп'ютерної інженерії сучасними програмними засобами. В результаті вивчення дисципліни студенти мають вміти моделювати комп'ютерні системи із застосуванням програмних засобів моделювання.

Мета проведення лекцій із дисципліни полягає у тому, щоб ознайомити студентів із основним сучасним програмним інструментарієм для моделювання комп'ютерних систем та їх застосування у комп'ютерній інженерії.

Основна мета проведення лекцій із дисципліни полягає у:

- викладенні студентам у відповідності з робочою програмою та робочим планом основних питань по особливостях підходів до вирішення задач моделювання комп'ютерних систем, звертаючи увагу на можливість застосування для вирішення реальних прикладних задач;
- сформуванню у студентів цілісну систему теоретичних знань з курсу “Програмні засоби моделювання для комп'ютерної інженерії”.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Лабораторні роботи проводяться для того, щоб виробити у студентів практичні навички використання теоретичного матеріалу в галузі моделювання комп'ютерних систем, вивчення застосування та використання сучасних програмних засобів моделювання при розробці моделей для комп'ютерної інженерії.

Завдання проведення лабораторних занять:

- ознайомитися та навчитися користуватися сучасними програмними засобами моделювання комп'ютерних систем;
- практично розробляти моделі для комп'ютерної інженерії;
- глибше засвоїти та закріпити теоретичні знання, одержані на лекціях.

1.2 Аналіз типових структур лабораторних робіт

Лабораторні роботи займають важливе місце серед різноманітних форм організації навчальної роботи у вищих навчальних закладах України. Саме через них здійснюється дотримання принципу зв'язку теорії з практикою.

Піл лабораторною роботою розуміють форму навчального заняття, при якій студент під керівництвом викладача особисто проводить відповідні експерименти або досліди з метою практичного підтвердження окремих теоретичних положень навчальної дисципліни, набуває практичних навичок роботи з лабораторним устаткуванням, обладнанням, обчислювальною технікою, відповідною апаратурою, програмним забезпеченням. При цьому, студент вчиться проволити експериментальні дослідження і на цій основі проводити відповідний аналіз та приймати рішення.

Лабораторні заняття значно корисніші за практичні, оскільки на них студент має змогу не тільки засвоїти теоретичний матеріал, а й ознайомитися із наявними програмними засобами, порівняти їх із аналогами, дослідити їх обчислювальну складність, специфіку, інтерфейс та особливості роботи.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Згідно навчальних планів та робочих програм, лабораторні заняття уможливлують [13]:

- поєднати лекційну форму занять із самостійною роботою, яка включає освоєння додаткового матеріалу, яке не охоплено на лекційному занятті;
- опрацювання студентами рекомендованих джерел літератури для достатнього володіння теоретичним матеріалом;
- ознайомлення студентів із методами пізнання, переконання в тому, що для ефективного проведення дослідження необхідні ґрунтовні теоретичні знання;
- зародження у студентів інтересу до науково-дослідної діяльності, залучення їх до дослідницьких проектів;
- систематично контролювати знання, уміння і навички студентів з окремих тем чи розділів дисципліни.

При проведенні лабораторних робіт необхідно дотримуватися чіткого розкладу із такими вимогами:

- лабораторне заняття планується для кожної навчальної групи окремо з виділенням одного, якщо група не ділиться на підгрупи чи двох, при умові поділу на підгрупи, викладачів;
- лабораторне заняття з певної тематики слід планувати після лекційного заняття;
- одне лабораторне заняття варто розраховувати не більше, ніж на 4 родини (2 пари), що сприяє кращому засвоєнню матеріалу;
- усі лабораторні роботи повинні бути більш-менш рівні за складністю виконання;
- варто застосовувати метод переходу результатів виконання однієї лабораторної роботи у іншу, що сприяє систематичному їх виконанні і розумінні, чому спочатку необхідно виконати попередню роботу, щоб мати можливість приступити до наступної.

Виконання лабораторної роботи – це реалізація таких принципів:

- оволодіння системою засобів і методів експериментально-практичного дослідження;

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

- оволодіння системою засобів і методів експериментально-практичного дослідження;

- розширення можливостей використання теоретичних знань для вирішення практичних завдань;

Щоб виконання лабораторної роботи було досягло відповідного рівня, потрібна присутність таких елементів:

- обговорення завдань для виконання на лабораторній роботі із групою, з'ясування всіх неясностей;

- самостійне виконання індивідуального завдання із можливістю консультування із групою;

- обговорення й оцінка отриманих результатів;

- письмовий та усний звіт студентів про виконання індивідуального завдання, що стосується певної лабораторної роботи.

Застосування лабораторних робіт при закріпленні теоретичного матеріалу є виправданим тоді, коли:

- нові знання є складними для словесного пояснення, але вони добре засвоюються при самостійних виконаннях на лабораторних роботах, при спостереженнях студентів за алгоритмом появи результатів;

- знання носять більш практичний характер, ніж теоретичний.

Лабораторні роботи класифікують за ступенем самостійності студентів:

- усі студенти виконують одні і ті ж етапи робіт за вказівкою викладача із методичними вказівками щодо виконання лабораторних робіт. У цьому випадку самостійність студентів невелика, проте відбувається колективне закріплення теоретичного матеріалу;

- студенти одержують індивідуальне завдання для виконання лабораторної роботи. У цьому випадку лабораторна робота містить дослідницький та експериментально-порівняльний характер.

Залежно від суті виконуваних робіт на лабораторних заняттях розрізняють роботи, спрямовані на:

- дослідження й аналіз різних явищ, процесів, станів системи та властивостей;

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- дослідження й аналіз роботи устаткування, та програмного забезпечення;
- дослідження якісних і кількісних залежностей.

Лабораторне заняття включає проведення поточного контролю підготовленості студентів до виконання конкретної лабораторної роботи, виконання завдань згідно ходу лабораторної роботи, оформлення звіту з виконаної роботи та його захист перед групою. Виконання лабораторної роботи оцінює викладач із винесенням двох оцінок – за володіння теоретичним виконанням та за практичні навички.

Кожна лабораторна із дисципліни розрахована на розгляд окремого тулбоксу матлабу. Це сприяє кращому засвоєнню теоретичного матеріалу та уможлиблює студентам поступово освоювати всі основні можливості пакету прикладних програм Matlab. У теоретичному матеріалі представлено приклад виконання лабораторної роботи із ілюстраціями всіх ключових моментів, що сприяє швидшому та кращому розумінню того, який має бути кінцевий результат виконання індивідуального завдання лабораторної роботи.

Зважаючи на вищевикладене, можна запропонувати загальний порядок виконання лабораторних робіт з дисципліни.

1. Уважно і детально вивчити загальну частину методичних вказівок до лабораторної роботи, попередньо ознайомившись з метою роботи.
2. Завантажити програмне середовище на робочому комп'ютері.
3. Реалізувати прикладу виконання лабораторної роботи, проаналізувати результати, виявити помилки, якщо вони присутні. Вивчити на основі цього нові функції програмного середовища, необхідні для виконання лабораторної роботи.
4. Згідно свого варіанту розробити оригінальний звіт із виконання лабораторної роботи.
5. Продемонструвати основні результати виконання роботи на комп'ютері із внесенням змін, запропонованих викладачем та аналізу зміни стаєну системи при цьому.
6. Файл із результатами виконання зберегти у програмному середовищі надавши йому ім'я, з якого є зрозуміло, чия це робота і яка вона є по порядку.

Звіт про виконання лабораторної роботи повинен містити:

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

- тему лабораторної роботи та прізвище і групу виконанця (титульний аркуш);
- мету лабораторної роботи та короткі теоретичні відомості (тезисно);
- індивідуальне завдання (згідно варіанту);
- опис основних етапів виконання завдання;
- екранні форми реалізації основних етапів виконання роботи;
- висновки.

1.3 Аналіз програмних засобів для реалізації лабораторних робіт

Сучасний процес математичного моделювання комп'ютерних систем важко реалізувати без комп'ютерів або вбудованих контролерів. Для цього використовуються різноманітні інструментальні програмні засоби та середовища (MathCad, MatLab, Mathematica, Maple, Derive, VisSim, Genius й інші), що суттєво спрощують моделювання [1, 2, 4]. Сучасні математичні пакети можна використовувати і як звичайний калькулятор, і як засоби для спрощення виразів при розв'язанні будь-яких математичних задач, і як генератори графіки або навіть звуку. Стандартними стали також засоби взаємодії з Internet шляхом генерації HTML-сторінок прямо в процесі обчислень. Звичайно, кваліфікований користувач, який в достатній мірі володіє однією з мов програмування (C++, Java, Prolog та ін.), може самостійно створити окрему програму або комплекс програм, що дозволить реалізувати на комп'ютері алгоритм його задачі. Проте такий підхід потребує, як правило, великих працездат на програмування, відлагодження та тестування кожної програми.

Тому, для скорочення часу програмування, було створено вищезгадані прикладні програмні пакети, сфери використання яких у значній мірі перекриваються. Для найбільш ефективного використання обчислювальної техніки необхідно правильно вибрати найкращий пакет програм на ранній стадії

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

розв'язання прикладної задачі. Адже реальна мета полягає в вирішенні певної проблеми, а обчислення – всього лише проміжний етап на шляху до цього.

Пакет прикладних програм Mathcad має зручний формат блокнота проектування у поєднанні з активним математичним представленням, аналізом одиниць вимірювання і потужними обчислювальними функціями. Це програмне забезпечення призначене для інженерних розрахунків й уможливорює представляти свої розрахунки за допомогою діаграм, графіків, тексту і зображень в єдиному документі. Користувачу не потрібні спеціалізовані навички, щоб розуміти основні функції пакету прикладних програм Mathcad.

Mathcad характеризується простотою освоєння - виконання базових операцій в стандартній нотації із інтуїтивно зрозумілим інтерфейсом. Студенти можуть легко створювати і читати документи після досить обмеженого часу навчання, оскільки середовище Mathcad використовує відомі стандарти і доповнює інші засоби математичних позначень. Mathcad поєднує числові і символні інструменти і розширені засоби форматування тексту в одному інтерфейсі для роботи з документами. Зазначене середовище пропонує засоби ведення документації та взаємодії, які стимулюють творчий підхід, гнучкість, різні уявлення і точність та рахунок їх розміщення в одному інтерфейсі з використанням стандартного синтаксису і зручних панелей інструментів.

Mathcad підтримує - реальні навички, необхідні для користувачів, що виходять за рамки вміння програмувати. Студенти мають можливість демонструвати проведений аналіз за допомогою засобів візуалізації.

Пакет прикладних програм Mathematica – це система комп'ютерної алгебри, що використовується при розв'язуванні наукових, інженерних, математичних задач. Mathematica реалізує різноманітні математичні розрахунки від знаходження простих значень функцій до різноманітних спрощень та візуалізації результатів. Система забезпечує автоматичне генерування програмного коду на мові Сі і його компонування. При цьому згенеровані програми можуть бути використані автономно. Для створення, обробки і оптимізації сі-коду підтримується використання SymbolicC. Програми можуть використовувати зовнішні динамічні

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

бібліотеки.

Крім того, Mathematica - це інтерпретована мова функціонального програмування, яка написана на мові Mathematica, хоч деякі функції, особливо що відносяться до лінійної алгебри, написані на мові Сі. Mathematica підтримує і процедурне програмування із застосуванням стандартних операторів управління виконанням програми і об'єктно-орієнтований підхід.

Пакет прикладних програм Maple - система комп'ютерної алгебри (система комп'ютерної математики). Система Maple призначена для символічних обчислень, хоч має ряд засобів і для чисельного вирішення диференціальних рівнянь і знаходження інтегралів. Володіє розвиненими графічними засобами. Має власну мову програмування, що нагадує Паскаль.

Пакет прикладних програм VisSim - візуальна мова програмування, призначена для моделювання динамічних систем, а також проектування, що базується на моделях для вбудованих мікропроцесорів. VisSim поєднує в собі характерний для Windows інтуїтивний інтерфейс для створення блокових діаграм. VisSim дозволяє створювати ієрархічні діаграми. Як правило, будується модель деякого процесу, що складається з декількох рівнів. При необхідності, частина блоків може бути розроблена на мові Сі або Фортран самим користувачем. Потім система доповнюється віртуальним контролером і налаштовується до отримання бажаного відгуку системи.

Із проведеного аналізу програмних засобів моделювання для комп'ютерної інженерії видно, що вони налаштовані на обмежене коло інженерних задач й здебільшого спрямовані на проведення різноманітних математичних розрахунків та побудови графічних об'єктів із можливістю їх візуалізації. Це в основному середовища для обчислення виразів та знаходження значень функцій із виведенням їх графіків.

Суттєвими перевагами серед пакетів прикладних програм для математичного моделювання вирізняється Matlab, в якому наявні вбудовані інструменти, що уможливають розв'язати будь-яку інженерну задачу.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4 Постановка задачі дипломного проектування

В першому розділі проведено аналіз структури, змісту та програмних засобів реалізації лабораторних робіт з дисципліни «Програмні засоби моделювання для комп'ютерної інженерії». Цей аналіз в розрізі навчальних планів по підготовці фахівців галузі знань «Інформатика та обчислювальна техніка», напряму підготовки «Комп'ютерна інженерія», показав необхідність розробки програмного комплексу лабораторних робіт із вищевказаної дисципліни у зв'язку із розвитком програмних засобів, які можна використати для моделювання комп'ютерних систем.

Огляд типових структур лабораторних робіт і аналіз програмного забезпечення для їх реалізації уможливив виділити основні вимоги, що ставляться до розробників задач із комп'ютерної інженерії. Аналіз існуючих пакетів прикладних програм для математичного моделювання показав доцільність використання для реалізації лабораторних робіт пакету прикладних програм Matlab.

Зважаючи на вищезазначене, метою дипломного проектування є розробка навчально-методичного комплексу із дисципліни «Програмні засоби моделювання для комп'ютерної інженерії». Для досягнення мети дипломного проектування необхідно виконати такі завдання:

- розробити структуру та зміст комплексу лабораторних робіт з дисципліни «Програмні засоби моделювання для комп'ютерної інженерії»;
- розробити структуру та зміст лабораторної роботи;
- охарактеризувати програмні засоби моделювання комп'ютерних систем;
- проаналізувати основні можливості пакету прикладних програм Matlab;

**РЕАЛІЗУВАТИ КОМПЛЕКС ЛАБОРАТОРНИХ РОБІТ В ПРОГРАМНОМУ
СЕРЕДОВИЩІ MATLAB;**

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

2 СТРУКТУРА ТА ЗМІСТ КОМПЛЕКСУ ЛАБОРАТОРНИХ РОБІТ

2.1 Структура комплексу лабораторних робіт

Комплекс лабораторних робіт, що розроблений для студентів 2-го курсу напряму підготовки «Комп'ютерна інженерія» охоплює всі ключові теми дисципліни «Програмні засоби моделювання для комп'ютерної інженерії». Перша лабораторна робота призначена для ознайомлення студентів із середовищем пакету прикладних програм Matlab. Метою цієї лабораторної роботи є дослідження основних тулбоксів програми, вивчення основних її функцій.

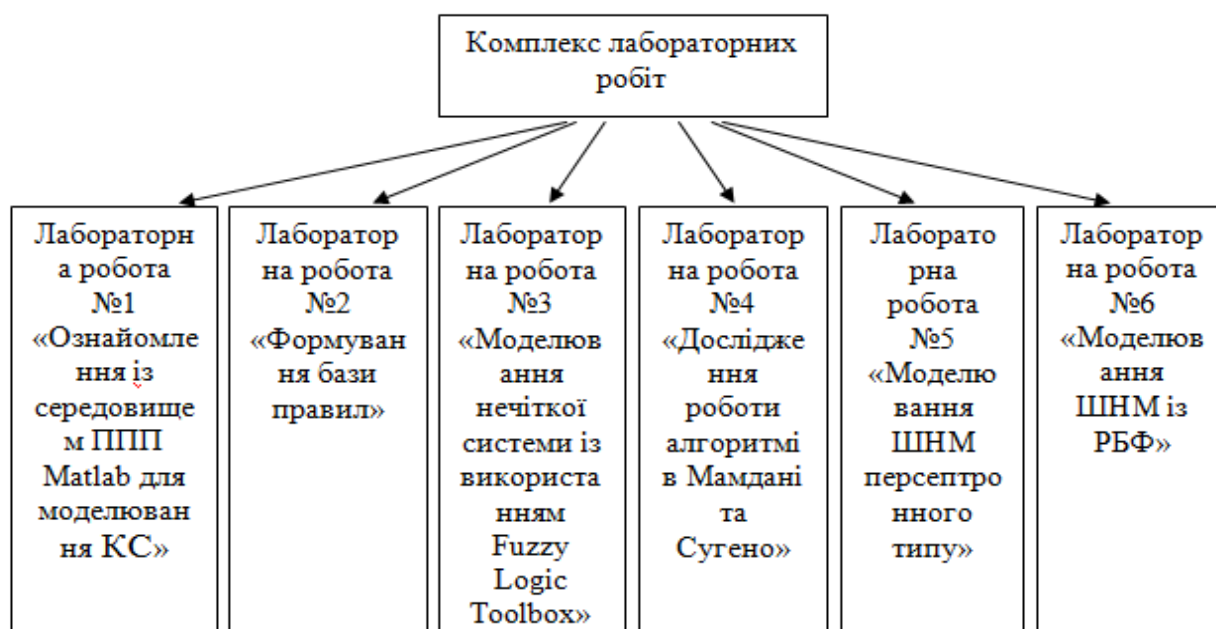


Рисунок 2.1 – Структура комплексу лабораторних робіт

Друга лабораторна робота пов'язана з ознайомленням із теорією нечітких множин, нечітких систем, функцій належності. Студенти мають навчитися будувати нечіткі правила і на їх основі базу знань для побудови комп'ютерних систем. Третя лабораторна робота полягає у розробці нечіткої системи із застосуванням тулбоксу нечіткої логіки. У четвертій лабораторній роботі

студенти досліджують алгоритми нечіткого виведення Мамдані та Сугено на основі розробленої нечіткої системи у попередній роботі. Зміст п'ятої лабораторної роботи полягає у побудові штучної нейронної мережі перцептронного типу із застосуванням Neural Networks Toolbox пакету прикладних програм Matlab. Шосту лабораторну роботу виконують студенти, як комплексне індивідуальне науково-дослідне завдання. Метою її виконання є дослідження спеціалізованої архітектури штучних нейронних мереж, в основі якої радіально-базисні функції, а також функцій Matlab, які відповідають за побудову таких моделей штучних нейронних мереж.

Таблиця 2.1 - Структура залікового кредиту дисципліни “Програмні засоби моделювання для комп'ютерної інженерії” [12]

	Кількість годин			
	Лекції	Лабораторні заняття	Самостійна робота	Індивідуальна робота
Тема 1. Сучасні програмні засоби моделювання для комп'ютерної інженерії	2	3	10	
Тема 2. Вбудовані тулбокси пакету прикладних програм для комп'ютерного моделювання Matlab	2	5	14	1
Тема 3. Нечіткі системи	2	4	14	
Тема 4. Fuzzy Logic Toolbox пакету прикладних програм MatLab	2	4	18	1
Тема 5. Моделювання нечітких систем	3	5	14	
Тема 6. Штучні нейронні мережі	2	3	12	
Тема 7. Neural Networks Toolbox пакету прикладних програм MatLab	2	6	18	1

Із результатів у таблиці 2.1 бачимо, що для того, щоб студенти могли виконувати лабораторні роботи, передбачено 15 годин лекційного матеріалу. Решту теоретичного матеріалу дається на самостійне вивчення. Перше лекційне заняття стосується розгляду сучасних програмних засобів для комп'ютерної інженерії. На другому лекційному занятті студенти знайомляться із вбудованими тулбоксами пакету прикладних програм Matlab. З третього по п'яте лекційні заняття присвячені нечітким системам від розгляду поняття нечітких систем, до засобів їх моделювання. Дві останні лекції стосуються штучних нейронних мереж.

На лабораторні заняття, згідно робочої програми, виділено 30 годин. На індивідуальну роботу студентів – 3 години.

2.2 Особливості пакету прикладних програм Matlab

При дослідженні комп'ютерних систем для розв'язування різних задач найбільш ефективним є використання програмної системи Matlab з широким класом предметно-орієнтованих бібліотек (Toolbox) та інструментом візуального моделювання Simulink.

Matlab - це високорівнева мова і інтерактивне середовище для програмування, чисельних розрахунків і візуалізації результатів. За допомогою Matlab можна аналізувати дані, розробляти алгоритми, створювати моделі і додатки [4, 18].

Мова, інструментарій та вбудовані математичні функції дозволяють досліджувати різні підходи і отримувати рішення швидше, ніж з використанням електронних таблиць або традиційних мов програмування, таких як C / C ++, Java. Matlab широко використовується в таких областях, як [6]:

- обробка сигналів;
- обробка зображень і відео;

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- системах управління;
- обробка нечітких даних;
- автоматизація тестування і вимірювання;
- штучний інтелект;
- фінансовий інжиніринг.

У системі Matlab існують широкі можливості для програмування. Її бібліотека C Math (компілятор Matlab) є об'єктною і містить понад 300 процедур обробки даних мовою C. Бібліотека у Matlab – це збірник підпрограм чи об'єктів, що використовуються для розробки програмного забезпечення. У середині пакета можна використовувати як процедури самої Matlab, так і стандартні процедури мови C, що робить цей інструмент ефективним при розробці додатків (використовуючи компілятор C Math, можна вбудовувати будь-які процедури Matlab у готові додатки).

Для візуалізації моделювання система Matlab має бібліотеку Image Processing Toolbox, що забезпечує широкий спектр функцій, які підтримують візуалізацію проведених обчислень безпосередньо із середовища Matlab, збільшення та аналіз, а також можливість побудови алгоритмів обробки зображень. Систему MatLab можна використовувати для обробки зображень, сконструювавши власні алгоритми, які будуть працювати з масивами графіки, як з матрицями даних. Оскільки мова Matlab оптимізована для роботи з матрицями, у результаті забезпечується простота використання, висока швидкість і економічність проведення операцій над зображеннями [18].

Серед інших бібліотек системи MatLab можна також відзначити System Identification Toolbox – набір інструментів для створення математичних моделей динамічних систем, заснованих на спостережуваних вхідних/вихідних даних. Особливістю цього інструменту є наявність гнучкого користувацького інтерфейсу, що дозволяє організувати дані й моделі. Бібліотека System Identification Toolbox підтримує як параметричні, так і непараметричні методи.

Інтерфейс системи полегшує попередню обробку даних, роботу з ітеративним процесом створення моделей для одержання оцінок і виділення

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

найбільш значимих даних. Що стосується математичних обчислень, то Matlab надає доступ до величезної кількості підпрограм, що містяться в бібліотеці NAG Foundation Library компанії Numerical Algorithms Group Ltd. Це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики, створених за останні три десятиріччя. Лише додану до системи велику кількість документації цілком можна розглядати як фундаментальний багатотомний електронний довідник з математичного забезпечення.

Сьогодні система MatLab широко використовується в техніці, науці та освіті, але все-таки вона більше підходить для аналізу даних і організації обчислень, ніж для чисто математичних викладок. У пакеті прикладних програм Matlab міститься чимало тулбоксів, розглянемо коротко ті, що стосуються розв'язування інженерних задач.

Тулбокси для системи управління: Control Systems Toolbox, μ -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 і ідентифікацію систем управління, включаючи сучасні алгоритми управління.

Тулбокси, що відповідають за цифрову обробку сигналів, зображень та даних: DSP Toolbox, Image Processing Toolbox, Wavelet Toolbox, Communication Toolbox, Filter Design Toolbox - набори функцій, що дозволяють вирішувати широкий спектр задач обробки сигналів, зображень, проектування цифрових фільтрів і систем зв'язку.

Нейронні мережі: Neural Network Toolbox - інструменти для синтезу та аналізу штучних нейронних мереж.

Нечітка логіка: Fuzzy Logic Toolbox - інструменти для побудови і аналізу нечітких множин, нечітких систем, бази знань, нечіткого висновку на основі алгоритмів нечіткого виведення.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3 Структура та зміст лабораторної роботи

Кожна розроблена лабораторна робота зазначеного вище комплексу робіт складається з таких частин:

- тема;
- мета;
- теоретичні відомості;
- хід виконання роботи;
- зміст звіту;
- контрольні питання.

Для прикладу, розглянемо лабораторну роботу №1, де враховано усі вищевказані складові. При захисті лабораторних робіт, студент повинен продемонструвати практичну частину виконання та володіти теоретичним матеріалом, тобто знати відповіді на контрольні запитання.

Лабораторна робота №1.

Тема: Ознайомлення із середовищем пакету прикладних програм Matlab для моделювання комп'ютерних систем.

Мета: Вивчення основних тулбоксів середовища пакету прикладних програм Matlab. Ознайомлення із графічним середовищем Matlab GUI.

Теоретичні відомості

До складу MatLab входить середовище GUIDE (GUI) для створення додатків з графічним інтерфейсом користувача. Робота в цьому середовищі досить проста - елементи управління (кнопки, списки, що розкриваються і т.д.) розміщуються за допомогою миші, а потім програмуються події, які виникають при зверненні користувача до даних елементів управління [7].

Додаток може складатися як з одного основного вікна, так і декількох вікон і здійснювати вивід графічної і текстової інформації, в основне вікно програми і в окремі вікна. Ряд функцій MatLab призначений для створення стандартних діалогових вікон відкриття і збереження файлу, друку, вибору шрифту, вікна для

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

введення даних і ін., Якими можна користуватися у власних додатках.

Що необхідно знати для створення додатків з графічним інтерфейсом? По-перше, як програмуються файл-функції з підфункціями, файл-функції зі змінним числом вхідних і вихідних аргументів. По-друге, потрібно мати уявлення про ієрархічну структуру і властивості графічних об'єктів. Додаток з графічним інтерфейсом може бути написано і без застосування середовища GUIDE. Як приклад, можна навести bspligui, що входить до складу Spline ToolBox.

Перейти в середовище GUIDE можна, виконавши команду:

```
>> guide
```

При цьому з'являється діалогове вікно GUIDE Quick Start, екранну форму якого зображено на рисунку 2.1.

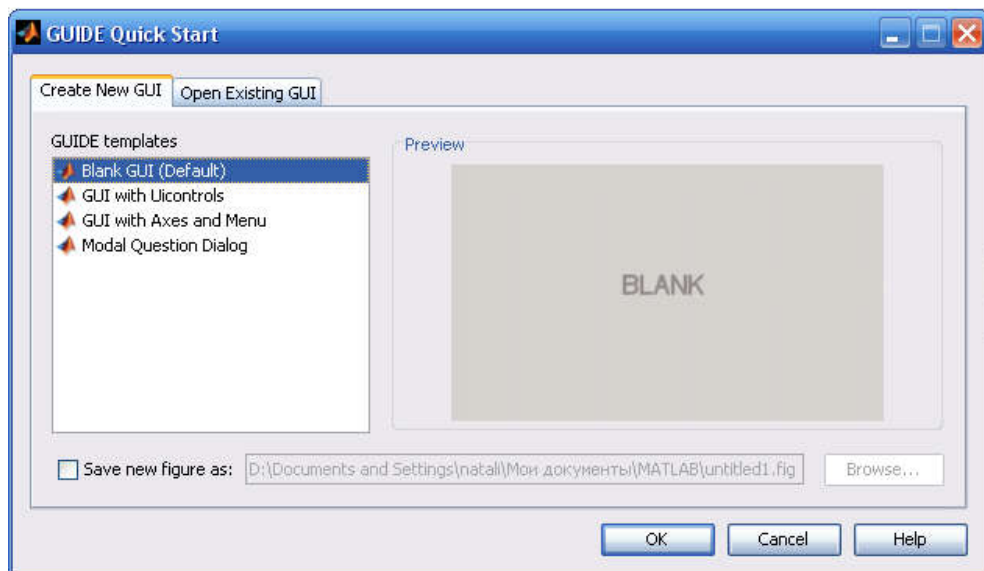


Рисунок 2.1 - Діалогове вікно GUIDE Quick Start

На екранній формі рисунка 3.1 дві вкладки:

– вкладка Create New GUI (створення нового додатка), яка нам знадобиться. На ній можна вибрати чотири заготовки: Blank GUI (пусте вікно додатка), GUI with Uicontrols (заготовка з кнопками, перемикачами і областями введення), GUI with Axes and Menu (заготовка з осями, меню, кнопкою і списком), Modal Question Dialog (заготовка для модального вікна);

- вкладка Open Existing GUI (відкриття наявної програми).

Крім того, внизу вкладки Create New GUI є прапорець, установка якого дозволяє відразу задати ім'я файлу, в якому буде зберігатися графічний інтерфейс. Але додаток завжди можна буде зберегти в процесі редагування, тому цей прапорець встановлювати не обов'язково.

При виборі Blank GUI, з'являється вікно середовища графічного редактора GUI, екранну форму якого проілюстровано на рисунку 2.2.

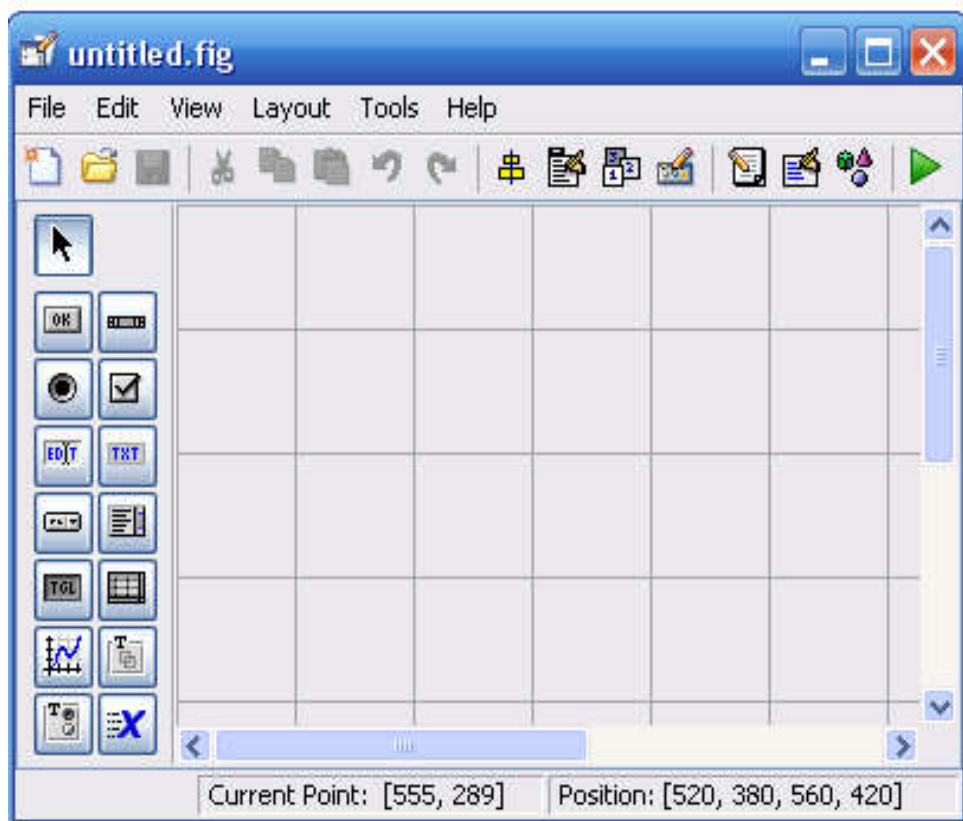


Рисунок 2.2 - Середовище GUIDE для створення нового додатку

Панель інструментів головного вікна графічного середовища GUI, як бачимо на рисунку 2.2, містить такі піктограми:

- push Button - кнопка;
- slider – полоса слайдера;
- radio Button - перемикач;
- check Box - прапорець;
- edit Text – область введення тексту;

- static Text – сталий текст;
- pop-up Menu – випадаючий список;
- list Box - список;
- toggle Button – кнопка-перемикач;
- table - таблиця;
- axes - графік;
- panel - панель;
- button Group - група кнопок;
- activeX Control.

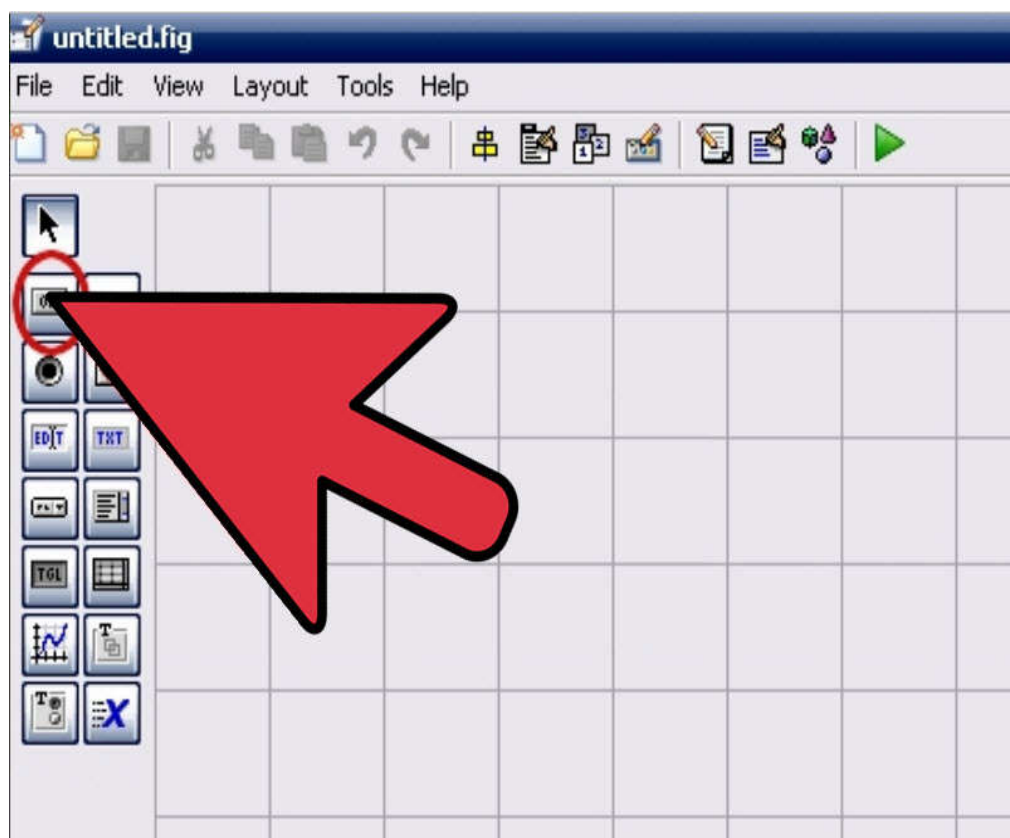


Рисунок 2.3 - Екранна форма вікна графічного середовища GUI із вказівкою на інструменти

Екранну форму головного вікна графічного середовища GUIDE, на якому створено елемент, наведено на рисунку 2.4. Як бачимо, створено елемент, який відповідає за певну дію (натискання). Таким чином можна створити будь-який інший елемент із вищенаведеного переліку.

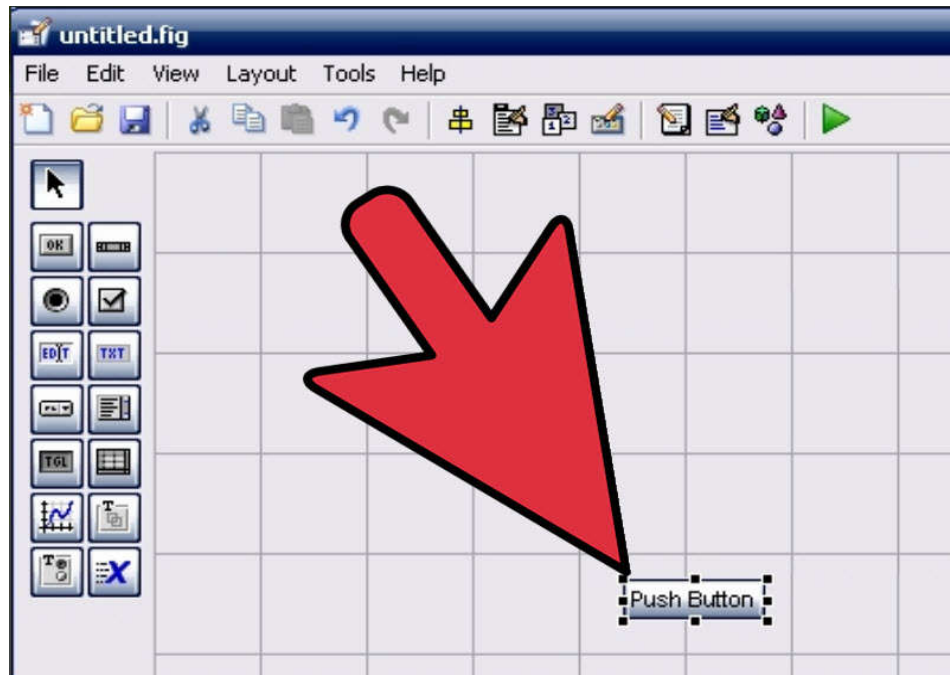


Рисунок 2.4 - Екранна форма вікна графічного середовища GUI із вказівкою на створений елемент

Для редагування розмірів елемента достатньо клікнути кнопкою миші по створеному елементу. При цьому з'являться контури елемента. Зазначений випадок зображено на рисунку 2.5.

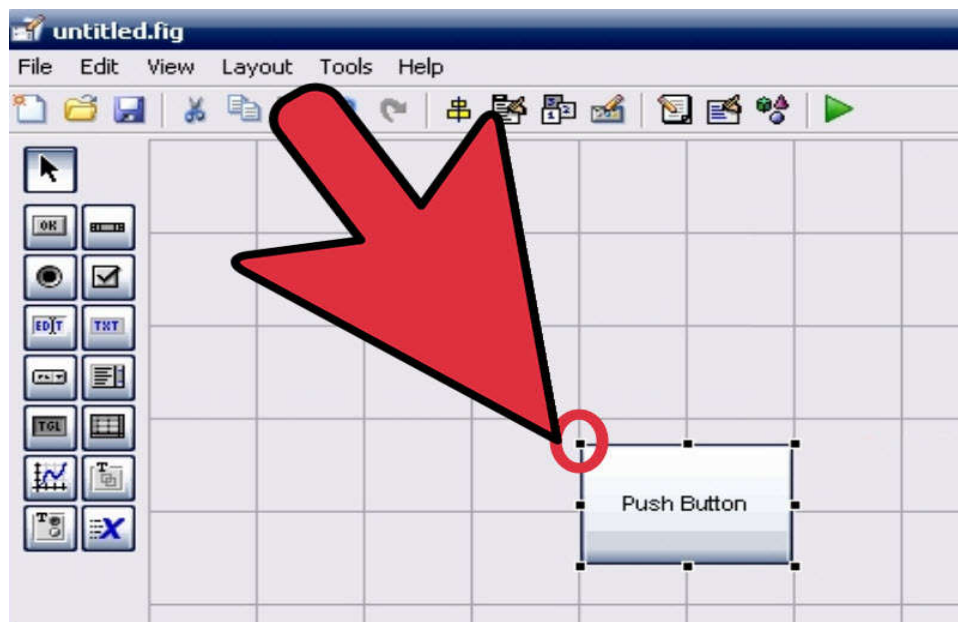


Рисунок 2.5 - Екранна форма вікна графічного середовища GUI із вказівкою на редагування розмірів елемента

Двічі клікнувши кнопкою миші на створеній піктограмі, з'явиться менеджер властивостей, в якому можна змінити параметри об'єктів (елементів) графічного додатку (назву піктограми, розмір, шрифт, колір, фон...).

На рисунку 2.6 проілюстровано екранну форму середовища GUI із активним менеджером властивостей.

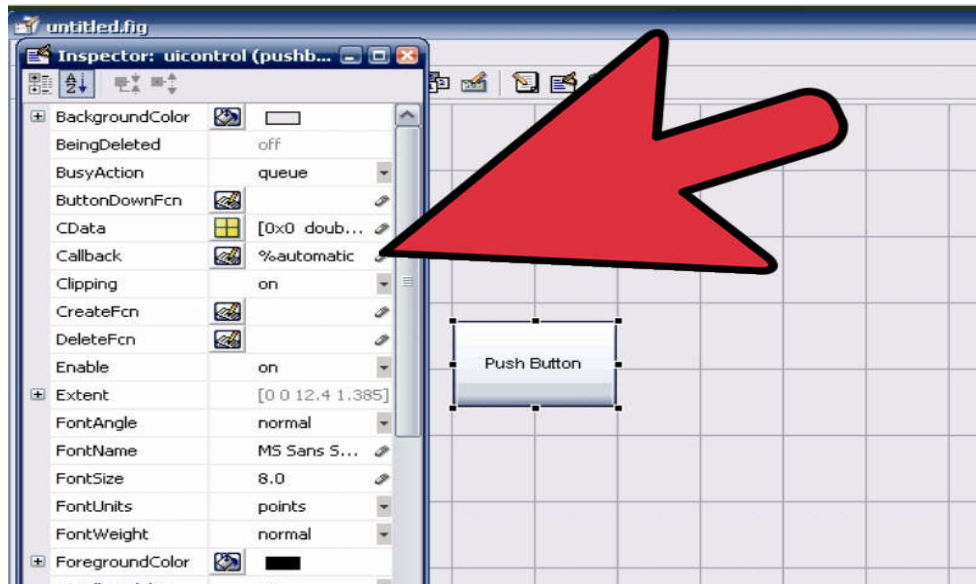
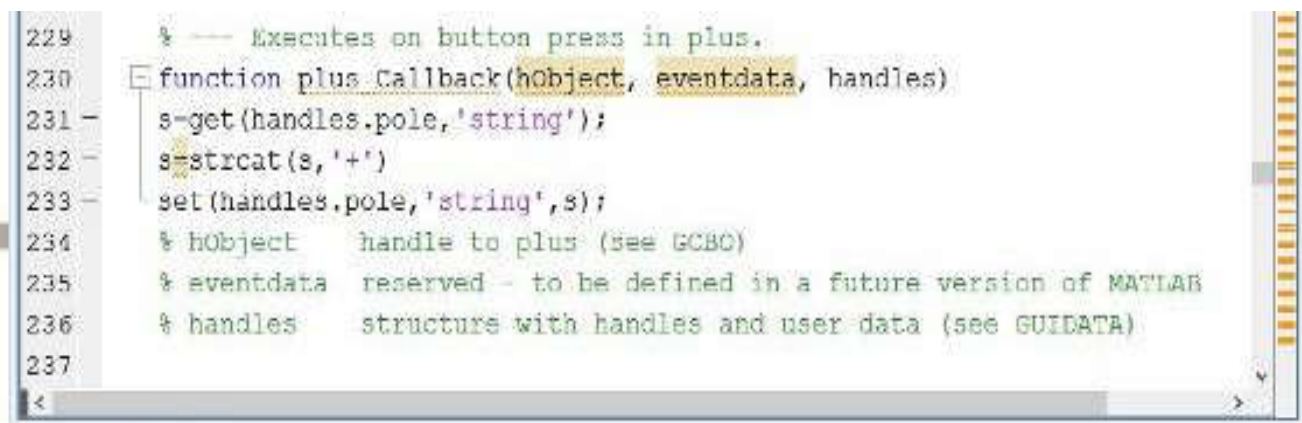


Рисунок 2.6 - Екранна форма вікна графічного середовища GUI із вказівкою на менеджер властивостей

Після додавання елемента необхідно задати його тег (Tag), який буде ідентифікувати цей об'єкт серед всіх інших об'єктів. Тег об'єкта знадобиться для отримання і встановлення його властивостей і програмування подій, що виникають при зверненні користувача до елемента, наприклад, при натисканні на кнопку. Завжди краще давати об'єктам змістовні теги (для кнопки - btn<назва>, для статичного тексту – txtWin, якщо пуста стрічка та txt<назва> для сталого напису). Зауважте, що тег і напис на елементі (його назва) не одне і те ж. За назву елемента відповідає String,

Якщо створення графічного додатку завершено, то зберігаємо файл із довільною назвою та розширенням *.fig. Після цього з'явиться згенерований код вашої програми.

Екранну форму частини згенерованого коду у середовищі Matlab, що відповідає створеному графічному середовищу, зображено на рисунку 2.7.



```
229 % --- Executes on button press in plus.
230 function plus_Callback(hObject, eventdata, handles)
231     s=get(handles.pole,'string');
232     s=strcat(s, '+')
233     set(handles.pole,'string',s);
234 % hObject    handle to plus (see GCBO)
235 % eventdata  reserved - to be defined in a future version of MATLAB
236 % handles    structure with handles and user data (see GUIDATA)
237
```

Рисунок 2.7 - Екранна форма вікна згенерованого коду у середовищі Matlab

Після цього можна програмувати події Callback будь-яких потрібних елементів. Для програмування частин згенерованого коду в контекстному меню середовища GUIDE в пункті View Callbacks обираємо підпункт Callback. При цьому відбувається перехід в редактор m-файлів до підфункції обробки події Tag_Callback, заголовок якої і коментарі генерується автоматично.

Наприклад:

```
% --- Executes on button press in btnHi.
function btnHi_Callback(hObject, eventdata, handles)
% hObject handle to btnHi (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% Handles structure with handles and user data (see GUIDATA)
```

Після заголовка підфункції (function btnHi_Callback) потрібно розмістити ті оператори, які будуть виконуватися при виконанні певної задачі.

Ім'я файл-функції складається з тега об'єкта (btnHi), подія (Callback) якого буде оброблятися і назви події Callback (є й інші події). Зміст вхідних аргументів функції такий:

- аргумент hObject містить покажчик на елемент "Hi", тобто об'єкт Uicontrol з тегом btnHi;

- аргумент eventdata зарезервований для використання в наступних версіях MatLab;

- аргумент handles є структурою із вказівкою на всі об'єкти програми.

Поля структури handles є тегами цих об'єктів. Так handles.btnHi містить вказівник на елемент "Hi", handles.figure1 - вказівник на вікно програми, handles.txtWin - вказівник на область перегляду тексту.

Розглянемо варіанти завдань для виконання.

1. Розробити калькулятор (у правому нижньому кутку створити піктограму «Виконав», натиснувши на яку, нижче появляється напис «Петров Петро, КІ-21» червоним кольором).

2. Розробити цифрову клавіатуру ноутбука (внизу по центру розмістити піктограму із прізвищем та групою виконавця, натиснувши на яку, зверху по центру з'являється назва роботи чорними літерами).

3. Розробити клавіатуру ноутбука (до кнопки f7). При цьому натиснувши на кнопку "Пробіл" на клавіші "Shift" з'явиться прізвище, назва та група виконавця.

4. Розробити графічний інтефейс для обчислення значень функцій, що зображено на рисунку 2.8. При цьому натиснувши на кнопку "Help", зверху з'явиться напис червоними літерами "Шановний (прізвище і група студента)".

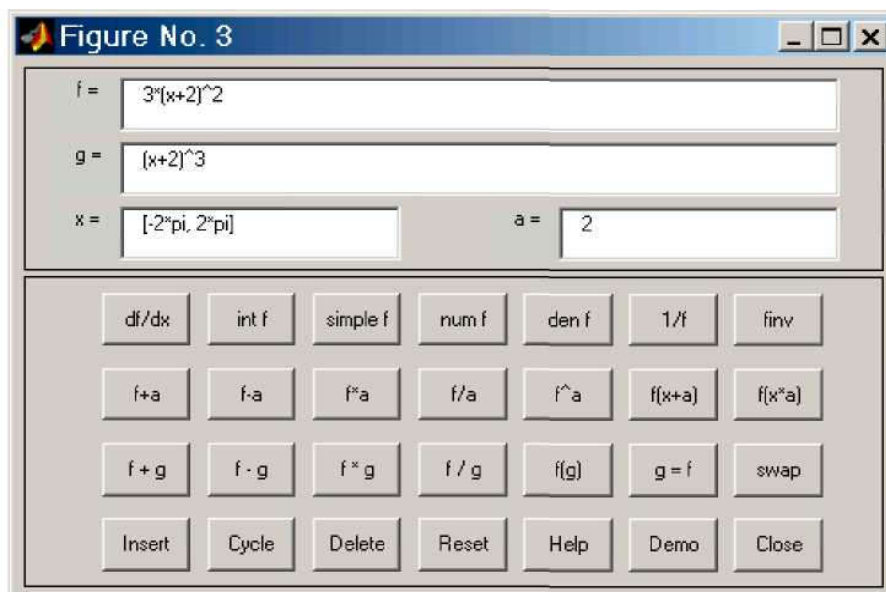


Рисунок 2.8 – Екранна форма графічного інтерфейсу для обчислення значень функції

5. Розробити графічний інтерфейс для обчислення значень функції $y_t = a_0 + a_1t + a_2t^2 + a_3t^3$. При натисканні на кнопку “Виконав студент групи КІ-22 Петренко Іван», зверху по центру з’являється назва функції великими чорними літерами.

6. Розробити графічний інтерфейс для обчислення значення функції $Y_i = \beta_0 + \beta_1X_i^2 + \beta_2X_i + \beta_3X_i^3 + u_i$. При нажиманні на кнопку “Обчислити”, з’являється напис – “Назва функції”.

7. Розробити графічний інтерфейс для обчислення значення функції $y = w_0 + w_1f_1(\bar{x} - \bar{c}_1) \cdot \sigma^{-2} + w_2f_2(\bar{x} - \bar{c}_2) \cdot \sigma^{-2} + \dots + w_8f_8(\bar{x} - \bar{c}_8) \cdot \sigma^{-2}$. Натиснувши на кнопку “Математична модель”, зверху по центру з’явиться напис “Студента групи КІ-21 Антонюка Олега”.

8. Розробити графічний інтерфейс, який зображено на рисунку 2.9. На клавіші “Cancel” зробити напис “Виконав”. Network Name - Іваненко Сергій, КІ-22 синіми літерами.

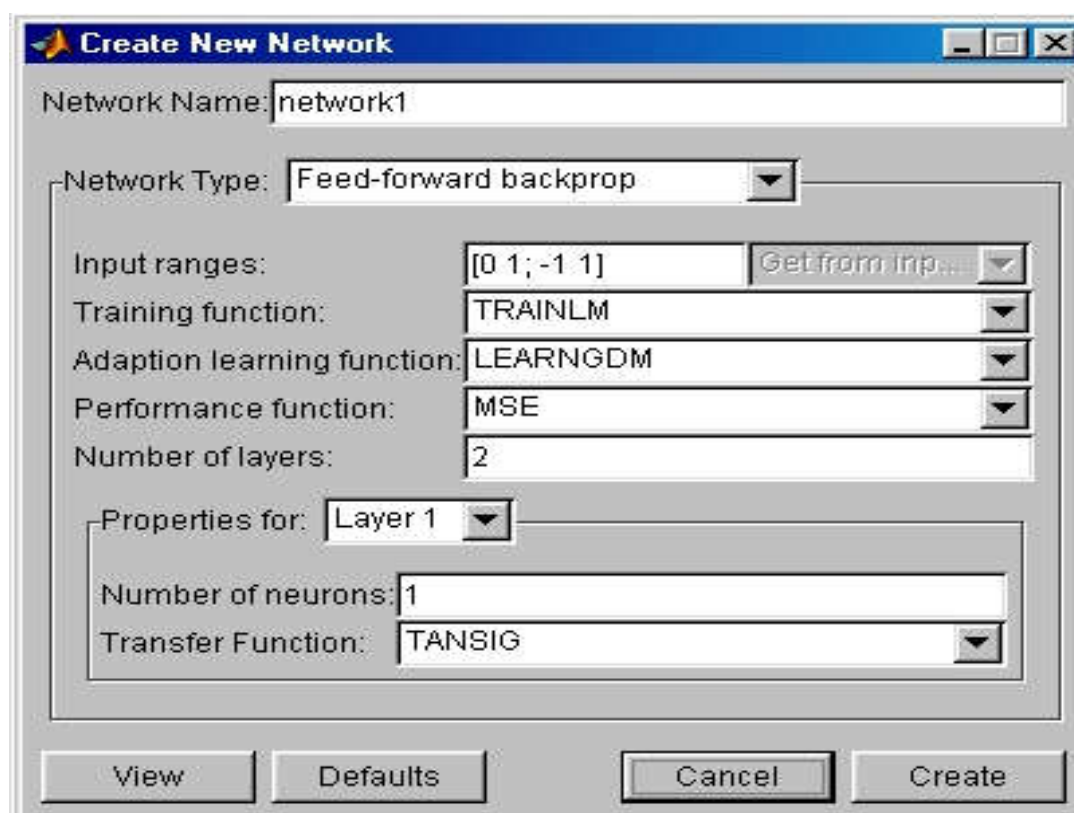


Рисунок 2.9 - Екранна форма графічного інтерфейсу для створення мережі

9. Розробити графічний інтерфейс для введення даних, який наведено на рисунку 2.10. При натисканні на клавішу “Help” внизу з’явиться напис червоними літерами - “Шановний (прізвище і група студента)”.

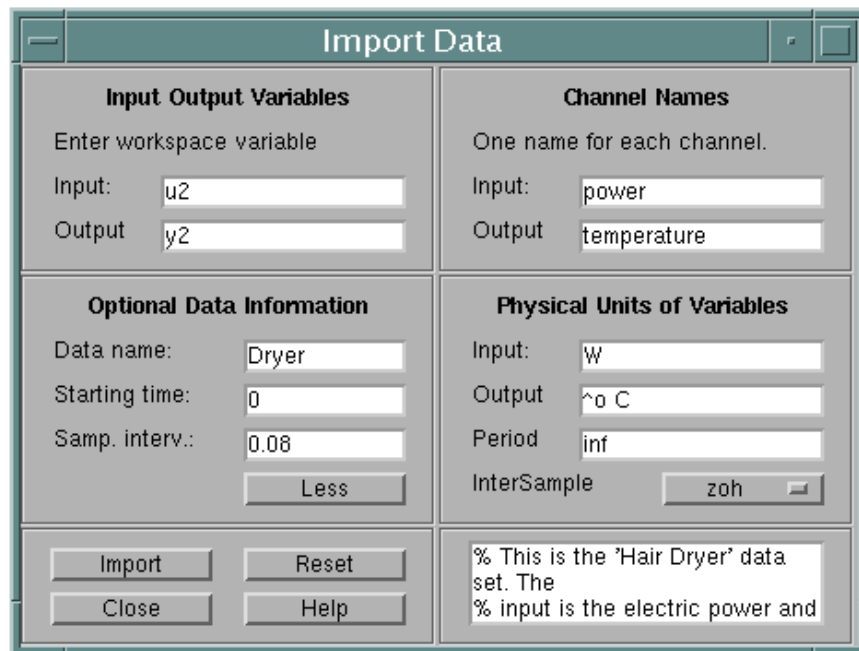


Рисунок 2.10 – Екранна форма графічного інтерфейсу для введення даних

10. Розробити клавіатуру ноутбука (після кнопки f7). При цьому, натиснувши на кнопку “Enter”, внизу по центру з’явиться прізвище, назва та група виконавця синіми літерами.

Оформлення звіту.

1. Тема, мета та коротенькі теоретичні відомості до лабораторної роботи.
2. Екранні форми результатів виконання індивідуального завдання.
3. Висновки.

Контрольні запитання.

1. Що таке ППП Matlab?
2. Що таке тулбокс?
3. Функції тулбоксів?
4. Основні властивості графічного середовища GUI?

3 РЕАЛІЗАЦІЯ КОМПЛЕКСУ ЛАБОРАТОРНИХ РОБІТ У СЕРЕДОВИЩІ MATLAB

3.1 Приклад виконання лабораторної роботи № 2

Для виконання лабораторної роботи на тему “Формування бази правил” необхідно виконати такі завдання [11, 13, 17]:

1. Проаналізувати поставлену інженерну задачу;
2. Визначити вхідні та вихідні змінні;
3. Створити вхідні та вихідні змінні та дати їм імена;
4. Визначити терми для вхідних та вихідних змінних, їх кількість, вид функції належності (трикутні, трапецевидні, гаусові та ін.), діапазон зміни;
5. Створити базу знань, що може складатися із будь-яких правил, але чим точніше сформулювати правила, тим краще буде працювати система.

Хід виконання роботи

Розробка нечіткої системи, що керує кутом повороту крана гарячої води для підтримання напору та температури на необхідному рівні.

1. У результаті аналізу бачимо, що у вказаній задачі є дві вхідні змінні температура (temp) та напір води (head) і одна вихідна змінна – кут повороту крана гарячої води (valve).

2. Створюємо змінні та даємо їм відповідні імена. Для цього задамо для вхідної змінної температура (temp) три терми: холодна (cold), середня (mid), гаряча (hot). Діапазон зміни для змінної температура задамо у вигляді [10; 80]. Задаємо відповідні значення для термів вхідної змінної температура, вибравши тип функції належності термів трикутний: cold [10 20 35], mid [30 35 40], hot [40 50 80].

Тепер задаємо терми для вхідної змінної напір води (head): малий (small), нормальний (norm) та великий (big). Діапазон зміни зазначеної вхідної змінної задамо у відносних одиницях[0;1]. Вибравши трикутну функцію належності для термів, задаємо такі числові значення зміни термів: small [0 0.1 0.3], norm [0.25 0.5

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

0.75], big [0.6 0.8 1].

3. Вихідна змінна valve характеризує, наскільки необхідно повернути кран гарячої води і в який бік – на закриття чи відкриття, щоб підтримувати потрібний напір води та її температуру. Для точнішого регулювання визначимо п'ять термів для вихідної змінної: повернути повністю (turn_h), повернути на половину (turn_f), не змінювати (norm), закрити на половину (close_h), закрити повністю (close_f). Діапазон зміни значень вихідної змінної задамо у відносних одиницях у межах [-10;10]. Функцію належності для термів вихідної змінної оберемо трикутну. При цьому числові межі для визначення термів задамо таким чином: turn_h [-10 -7 -5], turn_f [-6 -3 -1], norm [-2 0 2], close_h [1 3 6], close_f [5 7 10].

4. Створюємо базу знань, яка складається із всіх можливих нечітких правил, що визначають вихід системи при відповідних станах вхідних змінних. Сформовані правила заносимо у таблицю 3.1. Варто передбачити випадок, коли одна із вхідних змінних відсутня або не зчиталася програмою.

Таблиця 3.1 – База знань

Вхідні змінні		Вихідна змінна
Температура	Напір	
холодна	малий	повернути на половину
холодна	нормальний	закрити на половину
холодна	великий	закрити повністю
холодна	-	повернути повністю
середня	малий	повернути на половину
середня	нормальний	не змінювати
середня	великий	закрити на половину
середня	-	відкрити повністю
гаряча	малий	відкрити на половину
гаряча	нормальний	не змінювати
гаряча	великий	закрити на половину
гаряча	-	повернути повністю
-	малий	повернути на половину
-	нормальний	не змінювати
-	великий	повернути повністю

Як бачимо у таблиці 3.1, база знань для розробки нечіткої системи, що керує кутом повороту крана гарячої води для підтримання температури та напору на потрібному рівні, містить 15 нечітких правил. Усі правила типу якщо – то. Наприклад, перше правило звучить таким чином: якщо вода холодна напір води малий, то кран повернути на половину.

3.2 Приклад виконання лабораторної роботи № 3

Для виконання лабораторної роботи на тему “Створення та налаштування експертної системи з використанням Fuzzy Logic Toolbox пакета прикладних програм Matlab” необхідно виконати такі завдання:

1. Реалізувати нечітку систему, яка описана у лабораторній роботі №2. із застосуванням Fuzzy Logic Toolbox;
2. Проаналізувати нечіткий висновок системи на основі алгоритму Мамдані;
3. Проаналізувати поверхню відгуку нечіткої системи;
4. Перевірити результати нечіткої системи, змінивши значення вхідних змінних та їх функції належності.

Хід виконання роботи

Описуємо вхідні та вихідні лінгвістичні змінні із термами у Fuzzy Logic Toolbox [17]. Для того, щоб додати нову вхідну змінну, необхідно в меню Edit вибрати команду Add Variable...\Input, а вихідну - Add Variable...\ Output. Для того, щоб задати ім'я нечіткій системі в меню File обираємо підменю Export та команду To File та задаємо ім'я файлу – lab3.

Щоб перейти у редактор функцій належності, необхідно двічі натиснути на будь-якій із функцій та вибрати властивості конкретної (рисунок 3.1).

Вказуємо такі властивості функції належності:

1. Current Variable:

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- Name – ім'я функції;
- Type – тип (вхідна чи вихідна);
- Range – діапазон зміни змінної;
- Display Range – діапазон відображення.

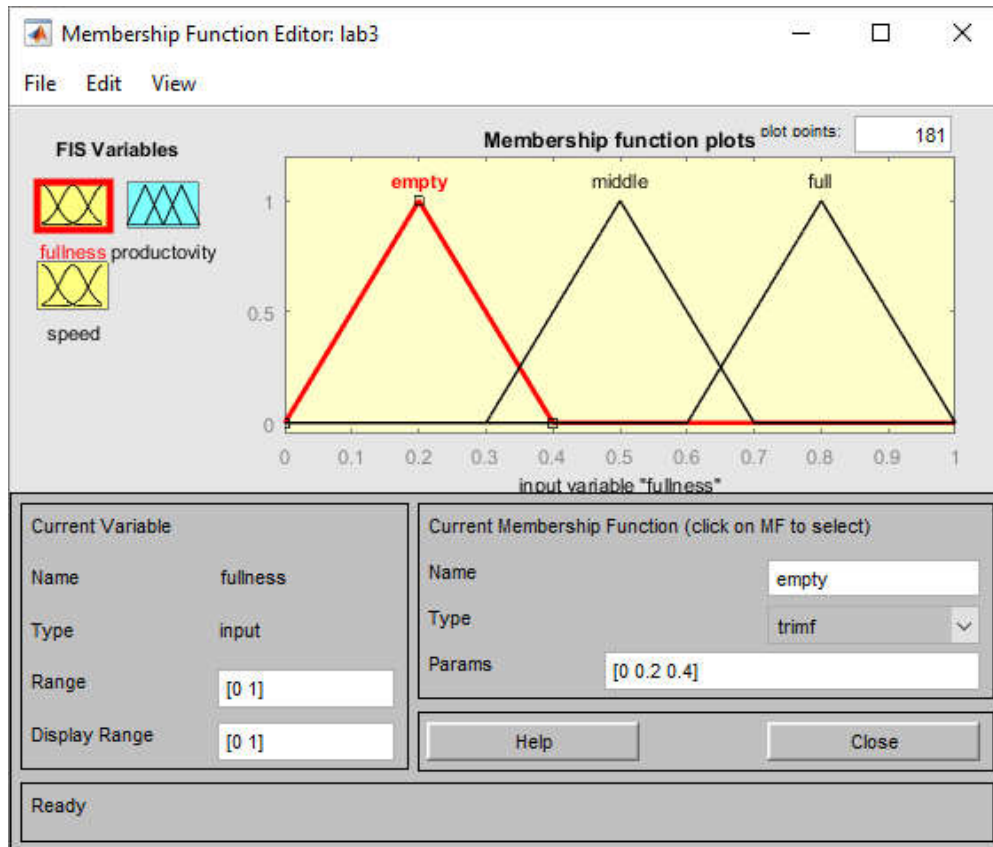


Рисунок 3.1 – Екранна форма вікна опису зміихта функцій належності

2. Current Membership Function:

- Name – ім'я поточної функції належності;
- Type – тип терму функції належності – вибирається з переліку (трикутна, трапецевидна, гаусова 1 та 2-го порядку та інші);
- Params – числові значення терму функції належності.

Для задання нових функцій належності для змінної необхідно в меню Edit вибрати команду Add MFs... У результаті з'явиться діалогове вікно (рисунок 3.2) вибору типу і кількості функцій належності.

Для формування бази знань, викликаємо редактор RuleEditor, в якому при заданні типу правил та вбору відповідних значень вхідних та вихідних змінних

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

автоматично одержуємо базу правил. Кожне правило по замовчуванню має вагу 1, проте це значення можна міняти відповідно до конкретної задачі. На рисунку 3.3 зображено вікно із сформованою базою знань.

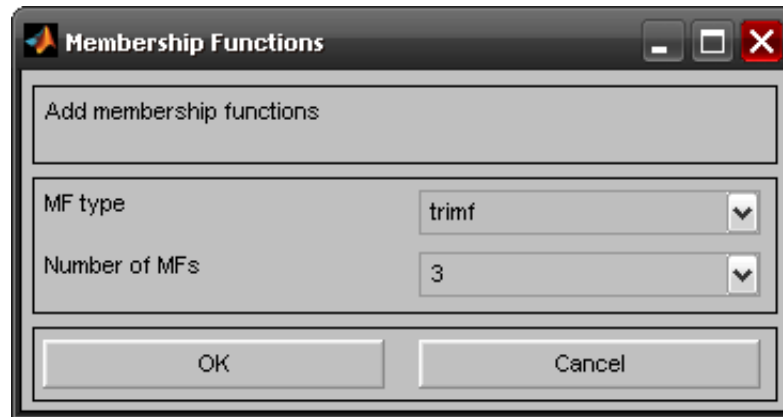


Рисунок 3.2 – Екранна форма діалогового вікна вибору типу і кількості функцій належності

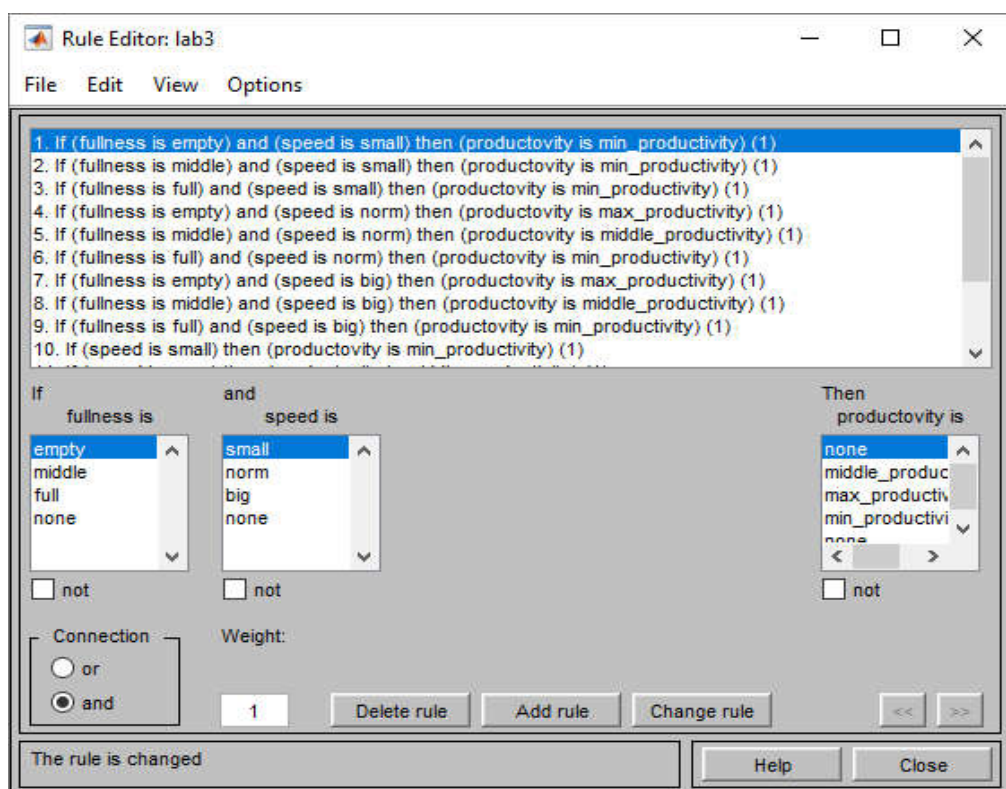


Рисунок 3.3 – Екранна форма вікна бази знань

Вікно візуалізації нечіткого логічного виведення на основі алгоритму Мамдані зображено на рисунку 3.4.

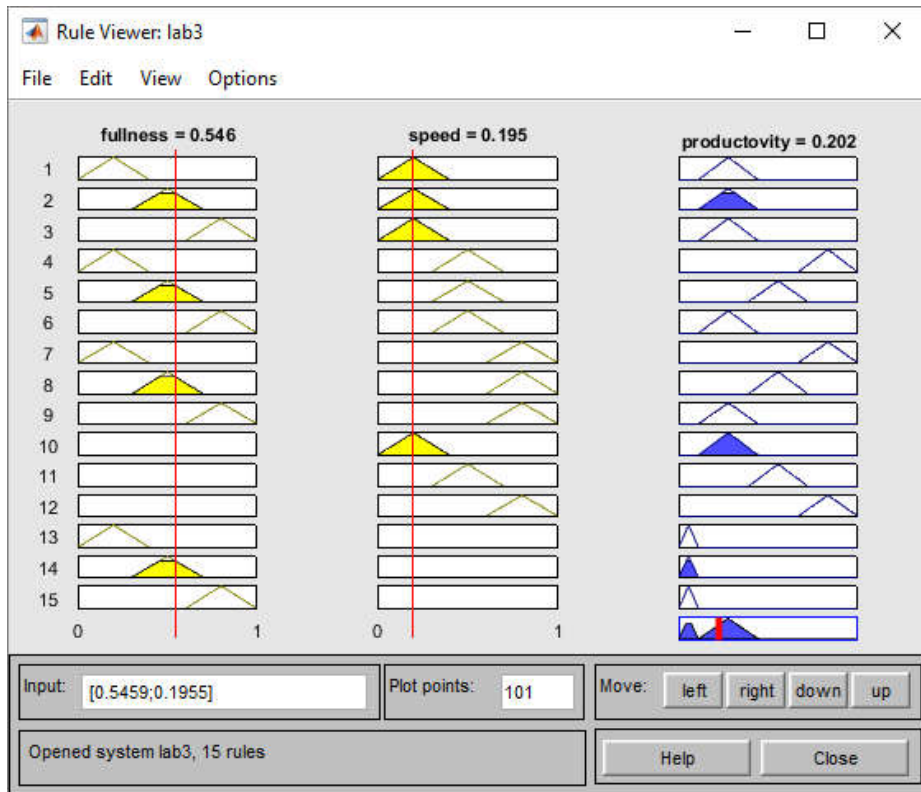


Рисунок 3.4 – Екранна форма візуалізації нечіткого логічного виведення

Поверхню “входи-вихід”, відповідну синтезованій нечіткій системі поілюстровано на рисунку 3.5.

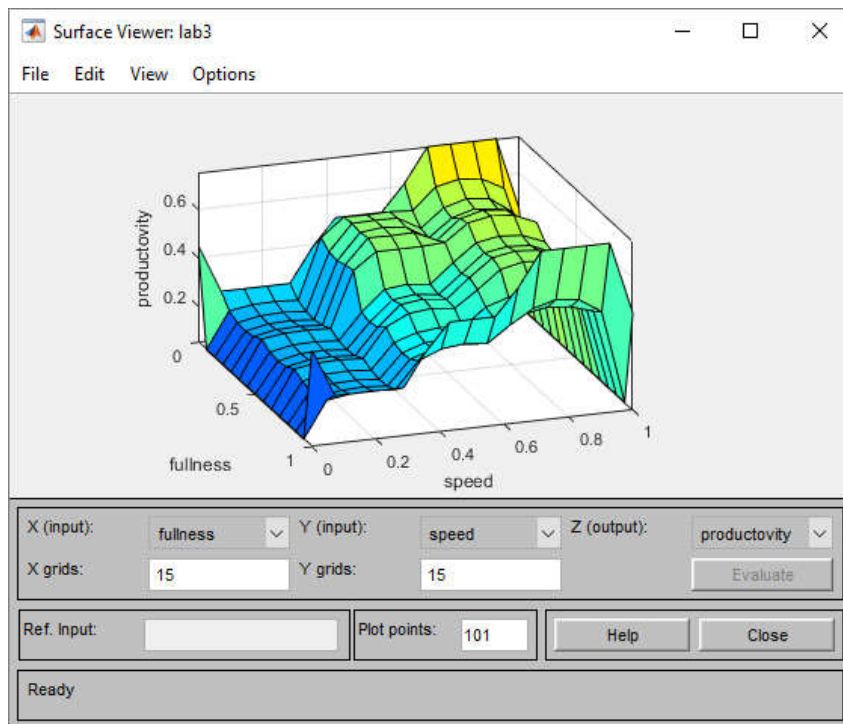


Рисунок 3.5 – Поверхня відгуку нечіткої системи

Змн.	Арк.	№ докум.	Підпис	Дата

Із результатів моделювання нечіткої системи бачимо, що система вчиться на базі правил (див. рисунок 3.4) та моделює нелінійну залежність між вхідними змінними та вихідною (див. рисунок 3.5).

3.3 Приклад виконання лабораторної роботи № 4

Для виконання лабораторної роботи на тему “порівняльне дослідження роботи алгоритмів логічного виведення мамдані та сугено” необхідно виконати такі завдання:

1. Побудувати еталонну поверхню, згідно варіанту, реалізувавши програму у Matlab;
2. Розробити нечітку систему на основі Алгоритму Сугено;
3. Проаналізувати результати нечіткого висновку, порівнявши одержану поверхню відгуку системи із еталонною.

Хід виконання роботи

Розробимо систему нечіткого логічного виведення Сугено, яка відтворює поверхню $y = \ln(x_1 + 1) \frac{1}{1 + x_2}$ при $x_1 \in [1,5]$ $x_2 \in [0,2]$.

Для побудови тривимірного зображення функції y в межах $x_1 \in [1,5]$ $x_2 \in [0,2]$ реалізувати код програми у Matlab:

```
n = 15; % кількість точок
x1 = 1:4/(n-1):5; % задання параметрів змінної x1
x2 = 0:2/(n-1):2; % задання параметрів змінної x2
y = zeros (n, n); % формування нульового масиву розміром n*n для вихідної змінної
for j = 1:n
y (j,:) = log10(x1+1)*(1/(1+x2(j)));
end
```

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

surf (x1, x2, y) % зображення поверхні функції
xlabel ('x1')
ylabel ('x2')
zlabel ('y')
title ('Target');

```

У результаті виконання програми отримаємо графічне зображення (еталонну поверхню), яке наведене на рисунку 3.6.

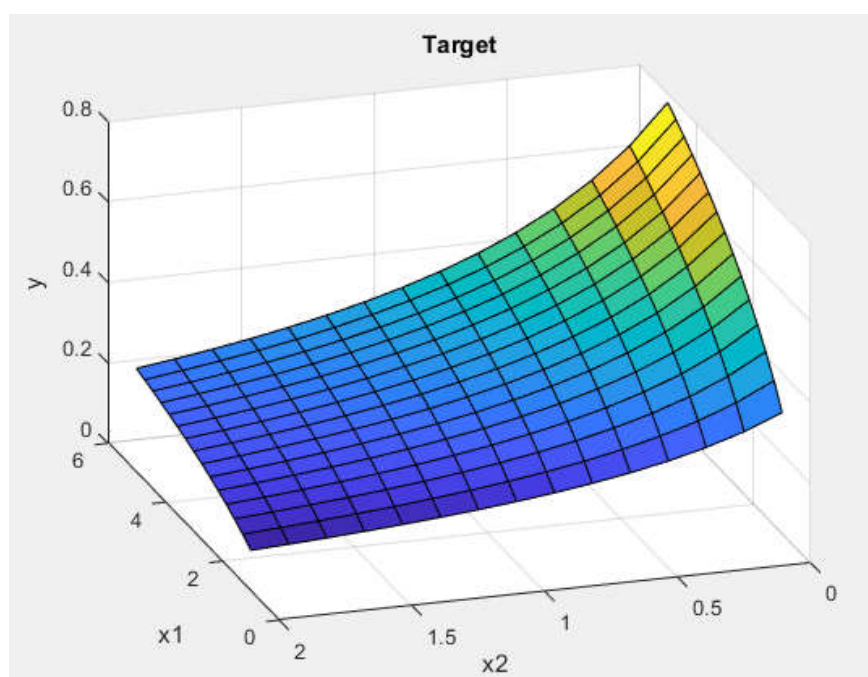


Рисунок 3.6 – Еталонна поверхня

Розробимо нечітку систему для моделювання функції на заданому діапазоні на основі алгоритму нечіткого виведення Сугено [21]. Перейдемо у редактор функцій належності і задамо діапазон зміни вхідних змінних та побудуємо для них функції належності, вказавши ім'я, тип та числові значення термів. При цьому параметри для x_1 : низький [1,5 2,6], середній [2,4 4], високий [3,9 5], а для x_2 : низький [0,001 0,4], середній [0,8 1,2], високий [1,5 2]. Функцію належності для опису термів змінних оберемо гаусову. Ілюстрації із описом змінних наведено на рисунках 3.7, 3.8.

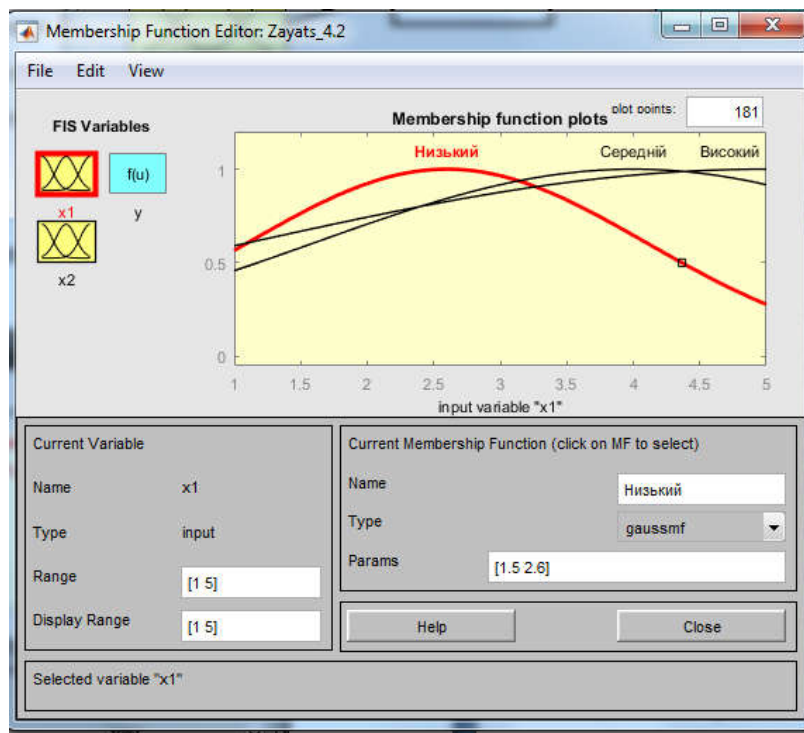


Рисунок 3.7 – Екранна форма віна візуалізації опису змінної x_1

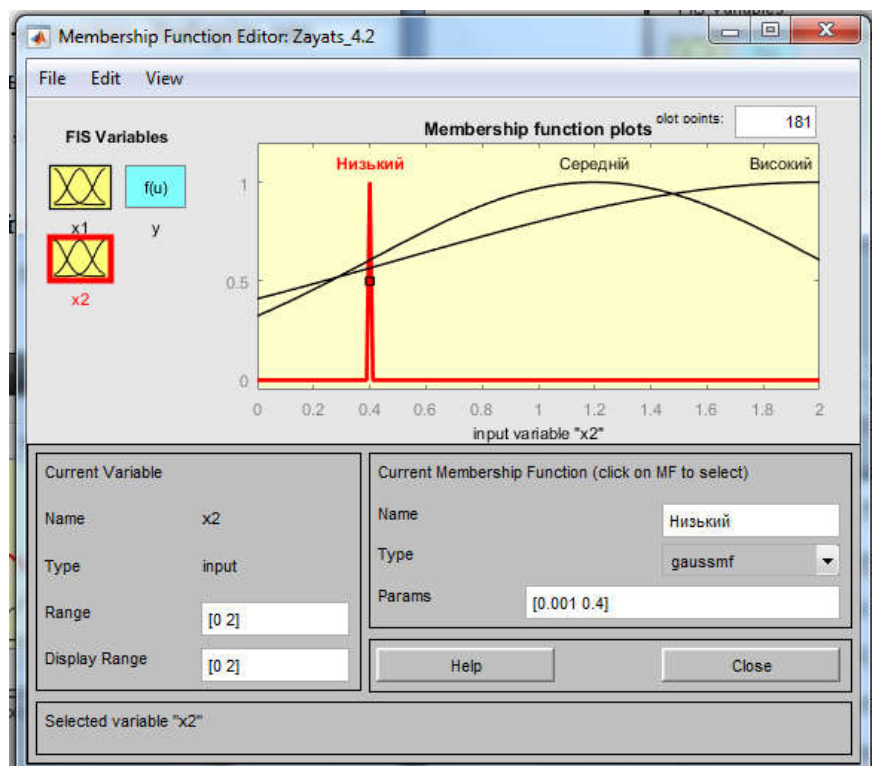


Рисунок 3.8 – Екранна форма віна візуалізації опису змінної x_2

Змн.	Арк.	№ докум.	Підпис	Дата

Для алгоритму Сугено для вихідної змінної задаються лінійні залежності між входами і виходом, що мають міститися в базі знань. У базі знань вказано 5 різних залежностей: $y=50$; $y = 4x_1 - x_2$; $y = 2x_1 + 2x_2 + 1$; $y = 8x_1 + 2x_2 + 8$; $y=0$.

Аналізуючи еталонну поверхню на рисунку 3.6, формуємо базу знань (рисунок 3.9).

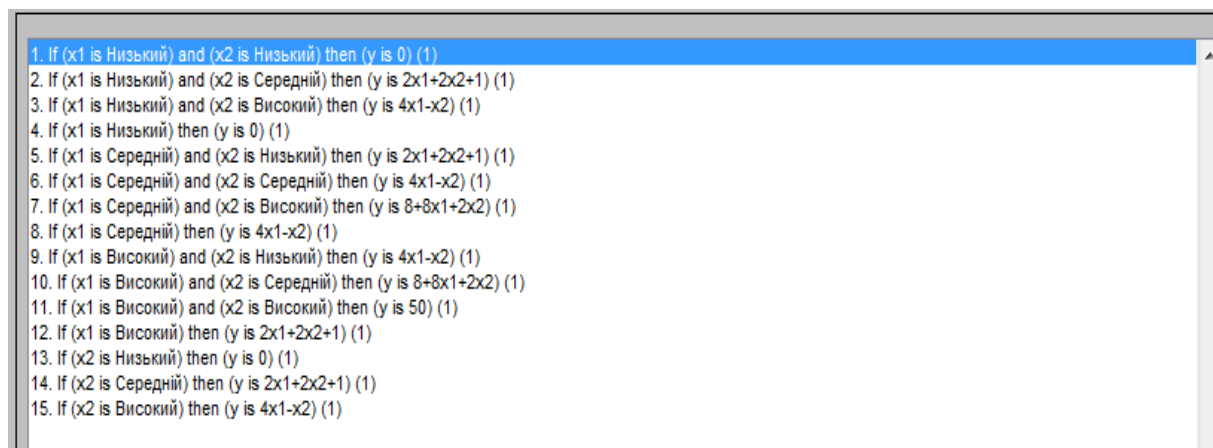


Рисунок 3.9 – Екранна форма візуалізації бази знань

Результати нечіткого логічного виведення системи та поверхню відгуку зображено на рисунках 3.10, 3.11.

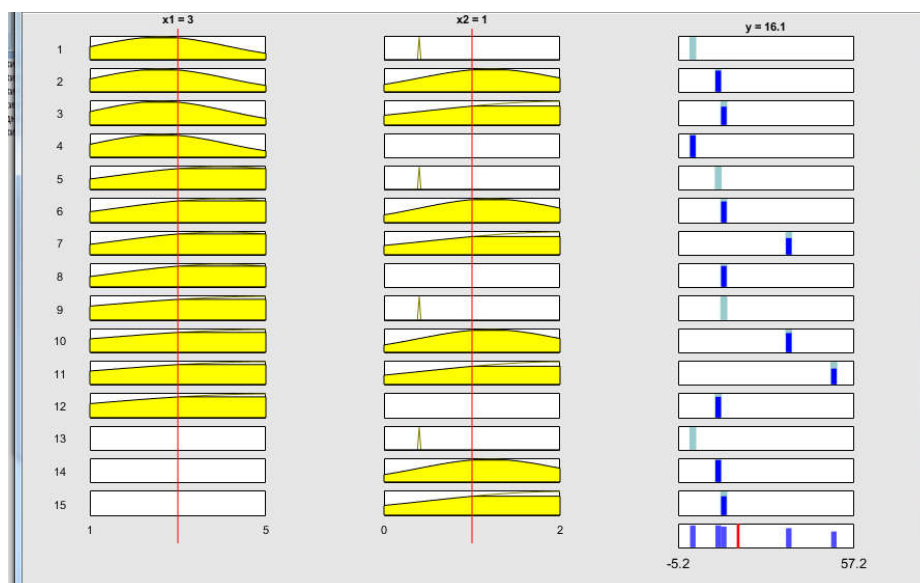


Рисунок 3.10 – Екранна форма вікна візуалізації нечіткого виведення на основі алгоритму Сугено

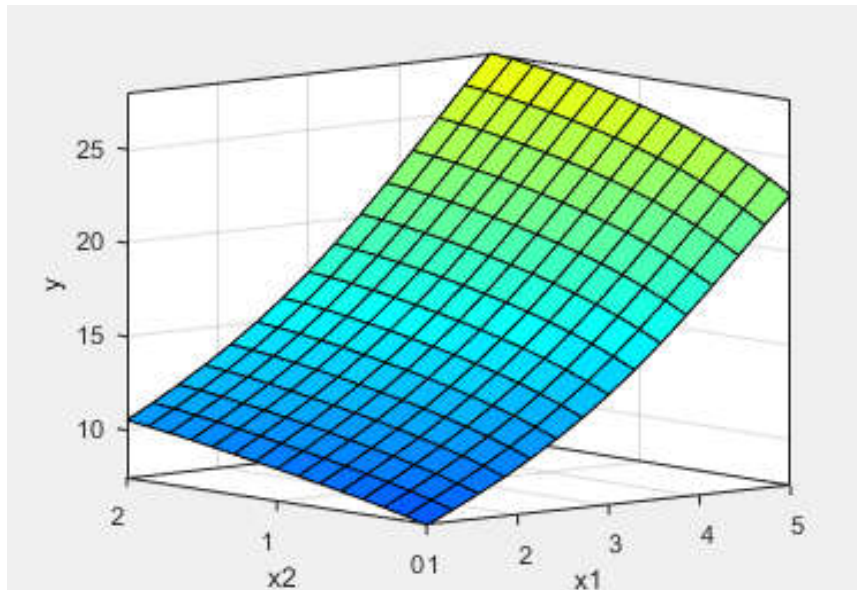


Рисунок 3.11 – Поверхня відгуку нечіткої системи

Із результатів моделювання нечіткої системи бачимо, що поверхня відгуку системи відповідає еталонній поверхні. Отже, розроблена нечітка система на основі алгоритму Сугено моделює лінійні залежності між вхідними змінними та виходом

3.3 Приклад виконання лабораторної роботи № 5

Лабоатора работа № 5 на тему: «Створення штучної нейронної мережі за допомогою Neural Networks Toolbox пакета прикладних програм Matlab» призначена для ознайомлення із тулбоксом Neural Networks [14, 15, 16]. Для того, щоб виконати зазначену роботу, потрібно:

1. Побудувати та навчити штучну нейронну мережу для обчислення $y=f(x)$ на відріжку деякому відрізьку згідно варіанту;
2. Проаналізувати результати навчання нейронної мережі;
3. Протестувати розроблену штучну нейронну мережу.

Розглянемо реалізацію лабораторної роботи на прикладі. Побудувати, використовуючи GUI-інтерфейс NNTool, штучну нейронну мережу для обчислення функції

$$y = \sin x - \ln(1 + x), [-1; 0,9],$$

де $p = [-1-0.7-0.6-0.40 \ 0.2 \ 0.3 \ 0.6 \ 0.81; -0.9-0.8-0.5-0.3-0.10.1 \ 0.3 \ 0.5 \ 0.7 \ 0.9]$ - вектор входу;

$t = [2.729 \ 1.492 \ 0.845 \ 0.347 \ 0.001 \ 0.079 \ 0.153 \ 0.595 \ 0.937 \ 1.271]$ - цільовий вектор.

Хід виконання роботи

Відкриємо за допомогою функції `nntool` основне вікно інтерфейсу, потім формуємо послідовність входів і цілей в робочій області GUI-інтерфейсу, використовуючи вікно `Create New Data`. З цією метою спочатку натиснемо кнопку `New Data` і далі в поле `Name` вікна `Create New Data` введемо спочатку ім'я змінної, потім в області значень `Value` - вектор значень i , використовуючи кнопку `Inputs` вкажемо тип змінних (`Inputs` - Входи). Введення завершимо натисканням кнопки `Create` (Створити). Аналогічну операцію виконаємо для вектора t , із зазначенням (за допомогою кнопки `Targets`), що це вектор цільових змінних.

Створимо нову штучну нейронну мережу. Для цього у вікні `Network / Data Manager` натиснемо кнопку `New Network`. У відкритому вікні `Create New Network` виберемо нейронну мережу типу `Feed-forward backprop` з прямою передачею сигналу і зворотнім поширенням помилки. При створенні мережі задамо їй ім'я, за замовчуванням (`network1`), діапазон входів визначимо (у вікні `Create New Network`) за допомогою опції `Get from input`, а кількість нейронів (`Number of neurons`) першого шару (`Layer 1`) встановимо рівним 10, функція активації у другому шарі – гіперболічний тангенс (`TANSIG`). Решту установки при створенні мережі залишимо за замовчуванням (рисунок 3.12). Створення мережі завершимо натисканням кнопки `Create`.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Інформацію про навчальну вибірку нейронної мережі та параметри навчання проілюстровано на рисунках 3.14, 3.15. Застосовуючи ці вкладки, можна задати іменавектора входу і цільовий вектор, а також значення параметрів процедури навчання. Для прикладу зазначені характеристики навчання нейронної мережі не змінюємо.

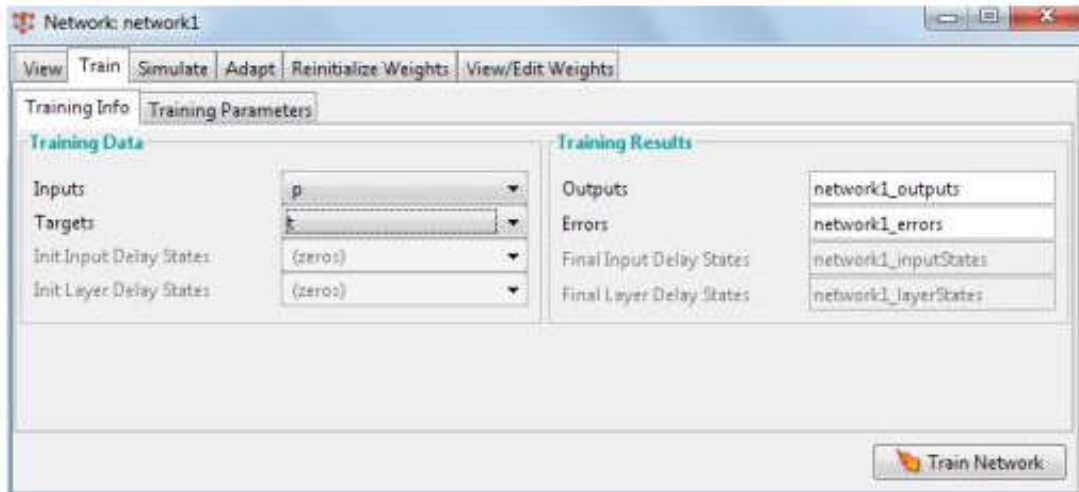


Рисунок 3.14 –Екранна форма вікна візуалізації навчальної вибірки

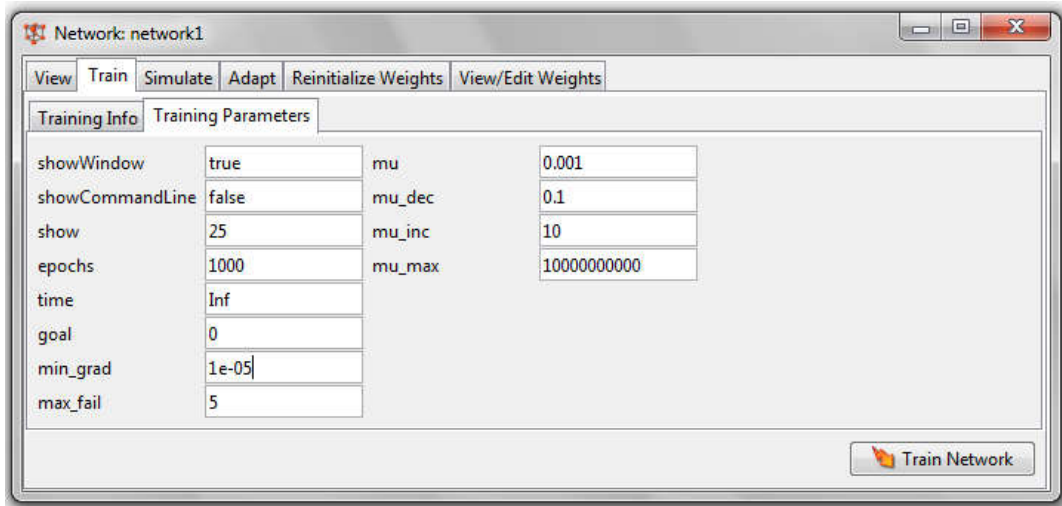


Рисунок 3.15 –Екранна форма вікна візуалізації параметрів навчання

Для навчання мережі натискаємо кнопку Train Network (див. рисунок 3.13). Процес навчання штучної нейронної мережі відображено на рисунку 3.16.

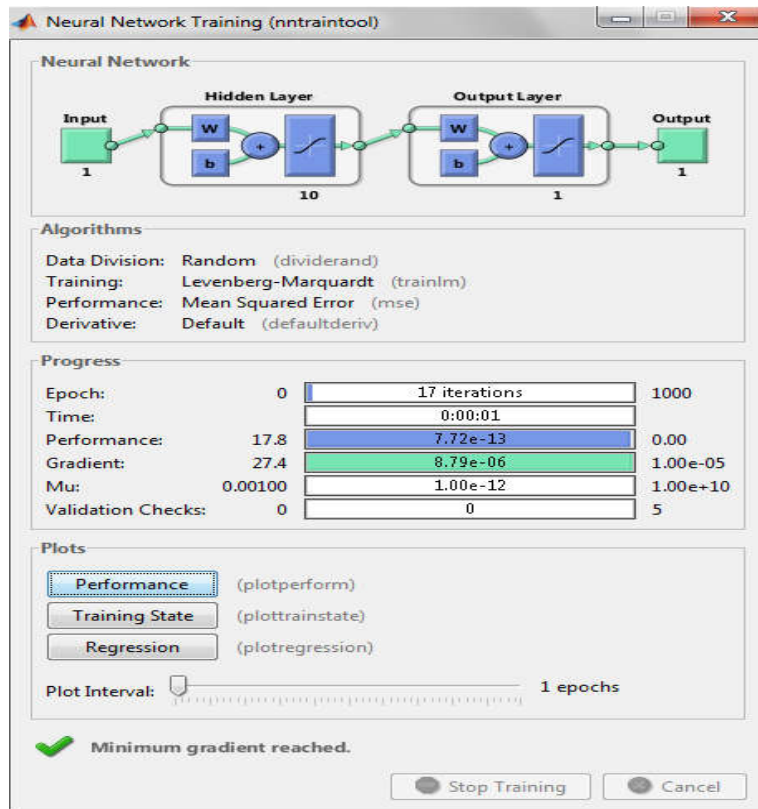


Рисунок 3.16 – Екранна форма вікна візуалізації процесу навчання штучної нейронної мережі

Якість навчання нейронної мережі для обраної навчальної вибірки даних проілюстровано на рисунках 3.17-3.18. Очевидно, що до кінця процесу навчання помилка стає дуже малою.

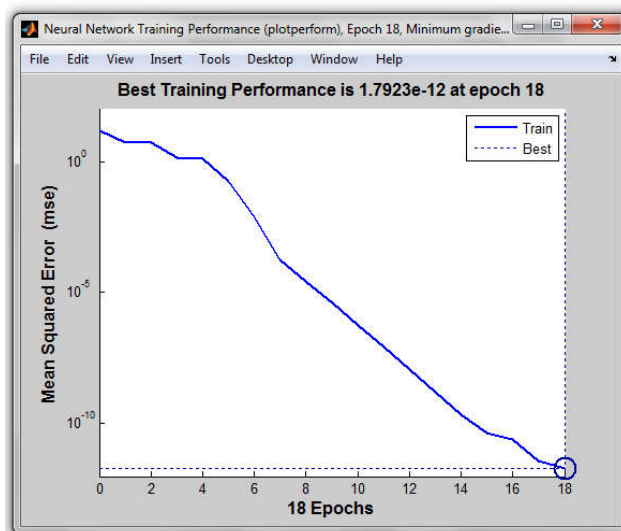


Рисунок 3.17 – Екранна форма вікна візуалізації зміни похибки мережі у процесі навчання

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

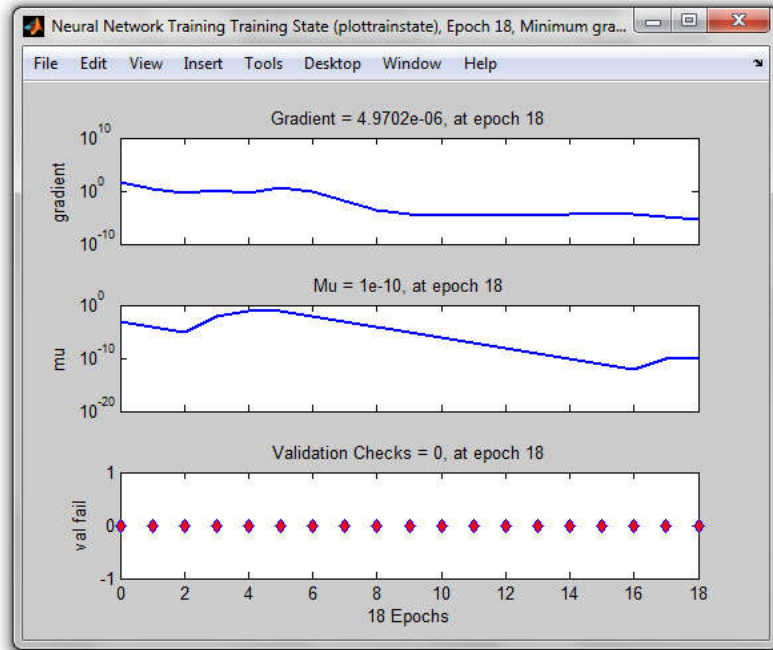


Рисунок 3.18 – Екранна форма вікна візуалізації стану навчання нейронної мережі

Значення синаптичних ваг для різних шарів створеної штучної нейронної мережі зображено на рисунку 3.19.

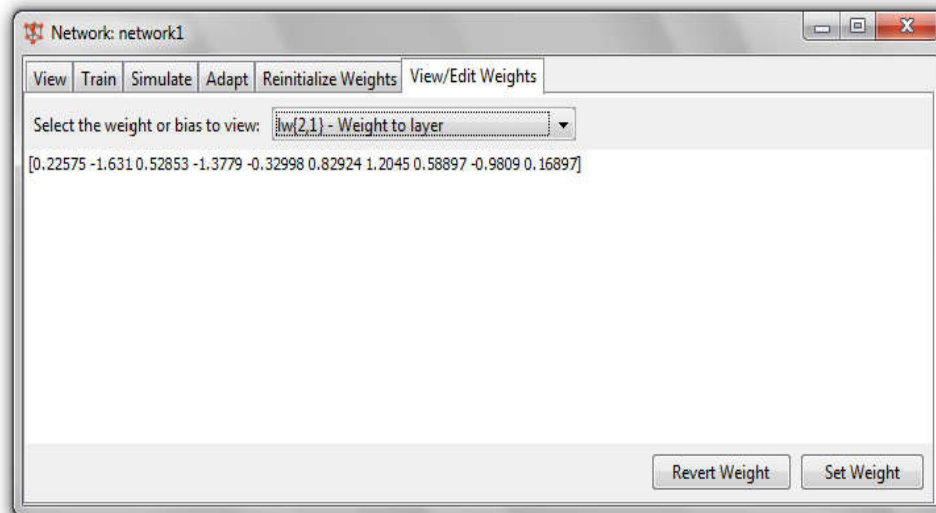
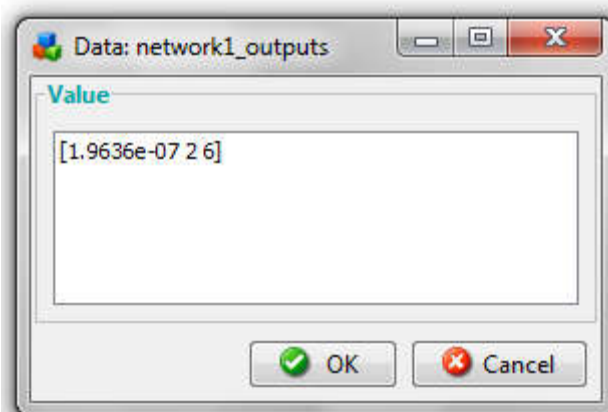


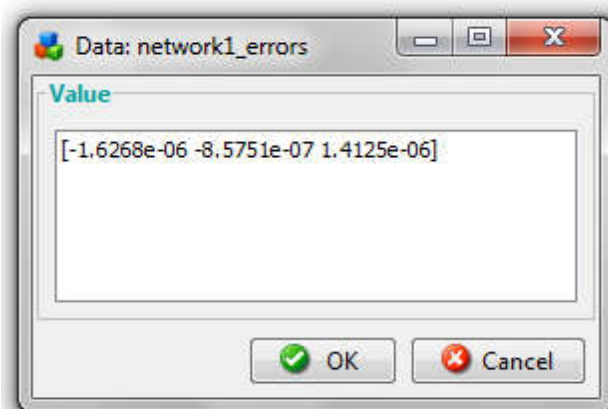
Рисунок 3.19 - Екранна форма вікна візуалізації значень налаштованих ваг у процесі навчання нейронної мережі

Останнім кроком побудови штучної нейронної мережі є тестування роботи нейронної мережі на тестовому наборі даних, використовуючи закладку Simulate.

Результати роботи нейронної мережі (вихідний вектор і помилки) можна переглянути у вікні Neural Network / Data Manager, виділивши область OutPut Data або Error Data і натиснувши кнопку Open. Результати тестування розробленої штучної нейронної мережі проілюстровано на рисунку 3.20 а) – значення виходу мережі та б) – значення похибки мережі.



а)



б)

Рисунок 3.20 – Результати тестування штучної нейронної мережі

Із результатів тестування бачимо, що розроблена штучна нейронна мережа є працездатною.

ВИСНОВКИ

У результаті виконання дипломної роботи розроблено навчально-методичний комплекс із дисципліни «Програмні засоби моделювання для комп'ютерної інженерії» При цьому одержано такі практичні результати:

1. Розроблено структуру та зміст комплексу лабораторних робіт із дисципліни, що розкриває основні практичні та теоретичні знання та вміння студентів.

2. Розроблено структуру та зміст лабораторної роботи й зазначено основні пункти, які має містити лабораторна робота для розуміння та практичного виконання.

3. Проаналізовано основні програмні середовища для моделювання комп'ютерних систем й показано, що в основному вони налаштованні на розв'язання вузького кола задач.

4. Охарактеризовано основні переваги та можливості пакету прикладних програм Matlab для розв'язування інженерних задач.

5. Реалізовано комплекс лабораторних робіт із дисципліни у середовищі Matlab, проведено аналіз результатів виконання робіт.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бахрушин В.Є. Математичне моделювання: Навчальний посібник / В. Є. Бахрушин. - Запоріжжя: ГУ "ЗІДМУ", 2004. – 140 с.
2. Бахрушин В.Є. Математичні основи моделювання систем: Навчальний посібник / В.Є. Бахрушин. - Запоріжжя: Класичний приватний університет, 2009. – 224 с.
3. Бодянский Е. В. Искусственные нейронные сети: архитектуры, обучение, применения: Монография / Е. В. Бодянский, О. Г. Руденко. – Харьков: ТЕЛЕТЕХ, 2004. – 372 с.
4. Боровиков В.П., Прогнозирование в системе STATISTICA в среде Windows. Основы теории и интенсивная практика на компьютере / В.П.Боровиков, Г. И. Ивченко. - М.: Финансы и статистика, 2000. – 320 с.
5. Брюхомицкий Ю. А. Нейросетевые модели для систем информационной безопасности / Ю. А. Брюхомицкий. – Таганрог : Изд-во ТРТУ, 2005. – 160 с.
6. Дьяконов В. Математические пакеты расширения MATLAB : специальный справочник / В. Дьяконов, В. Круглов. – СПб. : Питер, 2001. – 480 с.
7. Лебедев А. Н. Моделирование в научно-технических исследованиях / А. Н. Лебедев . - М.: Радио и связь, 1989. – 224 с.
8. Люгер Джордж Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Ф. Люгер Джордж. – 4-е изд. – Москва : “Вильямс”, 2003. – 864 с.
9. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму 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.
10. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «Бакалавр» напряму

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

підготовки 6.050102 комп'ютерна інженерія/ І.Р. Паздрій – Тернопіль: ТНЕУ, 2014. – 37 с.

11. Поспелов Д. А. Нечеткие множества в моделях управления и искусственного интеллекта / Д. А. Поспелов. – Москва : Наука, 1986. – 312 с.

12. Робоча програма з дисципліни «Програмні засоби моделювання для комп'ютерної інженерії» / Н.Я. Савка.- Тернопіль:ТНЕУ, 2017.

13. Ротштейн О.П. Проективання нечітких баз знань: лабораторний практикум та курсове проектування Навчальний посібник / О.П. Ротштейн, С.Д. Штовба. – Вінниця:ВДТУ, 1999. - 63 с.

14. Руденко О.Г. Штучні нейронні мережі: Навчальний посібник. - Харків: ТОВ “Компанія СМІТ”, 2006. – 404 с.

15. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – Москва : Телеком, 2006. – 452 с.

16. Тим Джонс М. Программирование искусственного интеллекта в приложениях / М. Тим Джонс; пер. с англ. А. И. Осипов – Москва : ДМК Пресс, 2006. – 312 с.

17. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику / С.Д. Штовба - [Электронный ресурс]. - Режим доступа: <http://matlab.exponenta.ru/fuzzylogic/book1/index.php>.

18. Штовба С.Д. Проектирование нечетких систем средствами MATLAB / С.Д. Штовба. - М.: Горячая линия–Телеком, 2007. – 288 с.

19. Ясницкий Л. Н. Введение в искусственный интеллект / Л. Н. Ясницкий. – Москва : «Академия», 2008. – 176 с.

20. Яхьяева Г. Э. Нечеткие множества и нейронные сети / Г. Э. Яхьяева. – Москва : Бином, 2006. – 316 с.

21. Timothy J. Ross. Fuzzy Logic with Engineering Applications / T.J. Ross. – USA: University of New Mexico: John Wiley & Sons Ltd, 2004. – 628 с.

					ДП.КСМ.07242/16.00.00.000.ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		