

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА "АВТОМОБИЛЬНЫЙ ТРАНСПОРТ И АВТОСЕРВИС"

**АВТОМАТИЗАЦИЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ И  
ПРОЕКТИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ для студентов  
специальности 240400 – " Организация и безопасность движения"  
Часть вторая

Курган 2004

Кафедра "Автомобильный транспорт и автосервис"

Дисциплина "Информационные технологии на транспорте "  
(специальность 240400).

Составил доцент, канд. техн. наук Борщенко Я.А.

Составлены на основе учебных планов специальности 240400

Утверждены на заседании кафедры "13" мая 2004 г.

Рекомендованы методическим советом университета  
"\_\_\_" \_\_\_\_\_ 2004 г.

## **Введение**

Автоматизированные системы управления нашли широкое применение во всех отраслях экономики. Создано и функционирует несколько тысяч АСУ различного класса и назначения.

Создание АСУ связано с анализом объекта управления, выбором критериев управления, определением структуры и функций системы. Параметры функций управления определяются особенностями объекта. Инженерная и управленческая деятельность невозможна без развития у будущего специалиста инженерного профиля способностей к анализу информации, ее обработке и принятию собственных решений.

Использование персонального компьютера для хранения и обработки информации на основе современных прикладных систем обработки данных является необходимым условием приобретения будущим инженером опыта аналитического и алгоритмического мышления в процессе учебы.

Вторая часть лабораторного практикума направлена на получение практических навыков обработки информации в современных инженерных средах таких, как MATHCAD, MATLAB.

Особое внимание обращается на освоение студентами современных методов анализа данных, построению экспертных систем и адаптивных систем управления, основанных на нейросетевых технологиях и нечеткой логике.

### **Порядок выполнения и защиты лабораторных работ**

Каждая лабораторная работа посвящена получению навыков по решению конкретных задач, перечень задач обозначен в цели и содержании лабораторной работы. После внимательного изучения поставленных задач студент, используя справку к данной программе и электронные учебники кафедры, изучает методы решения поставленных задач, после чего приступает к выполнению заданий.

После успешного выполнения заданий студент представляет результаты преподавателю в электронном виде.

После проверки результатов преподаватель допускает студента к защите, в ходе которой студенту предлагается ответить на контрольные вопросы для проверки и закрепления теоретических знаний и практических навыков по изучаемой теме.

# Лабораторная работа №1

## Автоматизация инженерных расчетов в среде MATHCAD

**Цель работы** – изучить возможности программной среды; получить навыки расчетов различной сложности и графического представления данных; изучить принципы работы с единицами измерения и оформления документов; получить навыки по программированию в данной среде.

### *1 Материальное обеспечение лабораторной работы*

1.1 ЭВМ Pentium – 166.

1.2 Программа MATHCAD.

### *2 Содержание лабораторной работы*

2.1 Освоение интерфейса и входного языка программы, определение возможностей и ограничений;

2.2 Создание документов, в которых производится расчеты и визуализация результатов;

2.3 Работа с единицами, измерения, форматирование документа, основные приемы программирования.

### *3 Общие положения*

MathCAD является математическим редактором, позволяющим проводить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов. Пользователи MathCAD - это студенты, ученые, инженеры, разнообразные технические специалисты. Благодаря простоте применения, наглядности математических действий, обширной библиотеке встроенных функций и численных методов, возможности символьных вычислений, а также превосходному аппарату представления результатов (графики самых разных типов, мощных средств подготовки печатных документов и Web-страниц), MathCAD стал наиболее популярным математическим приложением. MathCAD 2001, в отличие от большинства других современных математических приложений, построен в соответствии с принципом WYSIWYG ("What You See Is What You Get" - "что Вы видите, то и получите"). Поэтому он очень прост в использовании, в частности, из-за отсутствия необходимости сначала писать программу, реализующую те или иные математические расчеты, а потом запускать ее на исполнение. Вместо этого достаточно просто вводить математические выражения с помощью встроенного редактора формул, причем в виде, максимально приближенном к общепринятому, и тут же получать результат. Кроме того, можно изготовить на принтере печатную копию документа или создать страницу в Интернете именно в том виде, который этот документ имеет на экране компьютера при работе с MathCAD. Создатели MathCAD сделали все возможное, чтобы пользователь, не обладающий специальными знаниями в программировании (а таких большинство среди ученых и инженеров), мог в полной мере приобщиться к достижениям современной вычислительной науки и компьютерных технологий. Для эф-

фективной работы с редактором MathCAD достаточно базовых навыков пользователя. С другой стороны, профессиональные программисты могут извлечь из MathCAD намного больше, создавая различные программные решения, существенно расширяющие возможности, непосредственно заложенные в MathCAD.

В соответствии с проблемами реальной жизни, математикам приходится решать одну или несколько из следующих задач:

- ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, Web-страниц);
- проведение математических расчетов;
- подготовка графиков с результатами расчетов;
- ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах;
- подготовка отчетов работы в виде печатных документов;
- подготовка Web-страниц и публикация результатов в Интернете;
- получение различной справочной информации из области математики.

Со всеми этими (а также некоторыми другими) задачами с успехом справляется MathCAD:

- математические выражения и текст вводятся с помощью формульного редактора MathCAD, который по возможностям и простоте использования не уступает, к примеру, редактору формул, встроенному в Microsoft Word;
- математические расчеты производятся немедленно, в соответствии с введенными формулами;
- графики различных типов (по выбору пользователя) с богатыми возможностями форматирования вставляются непосредственно в документы; возможен ввод и вывод данных в файлы различных форматов;
- документы могут быть распечатаны непосредственно в MathCAD в том виде, который пользователь видит на экране компьютера, или сохранены в формате RTF для последующего редактирования в более мощных текстовых редакторах (например, Microsoft Word);
- возможно сохранение документов в формате Web-страницы, причем создание файлов с рисунками происходит автоматически;
- символьные вычисления позволяют мгновенно получить разнообразную справочную математическую информацию, а система помощи, Центр Ресурсов и встроенные электронные книги помогают быстро отыскать нужную справку или пример тех или иных расчетов.

Таким образом, следует хорошо представлять себе, что в состав MathCAD входят несколько интегрированных между собой компонентов:

- мощный текстовый редактор для ввода и редактирования как текста, так и формул,
- вычислительный процессор - для проведения расчетов согласно введенным формулам, и символьный процессор, являющийся, по сути, системой искусственного интеллекта.

Сочетание этих компонентов создает удобную вычислительную среду для разнообразных математических расчетов и одновременно, документирования результатов работы. Более подробно справку о возможностях данной среды можно найти в источниках /1-12/ и электронных учебниках кафедры в лаборатории "Информационных систем".

## **4 Выполнение лабораторной работы**

### **4.1 Освоение интерфейса и входного языка программы, анализ возможностей и ограничений**

#### **1 Запуск программы MathCAD.**

Если MathCAD запускается из главного меню Windows (с помощью кнопки **Пуск** в углу экрана), например, **Start / Programs / MathCAD 2001** (Пуск / Программы/ MathCAD 2001), то окно MathCAD появляется с открытым в нем новым пустым безымянным документом, условно называемым **Untitled:1**. Для того чтобы создать новый пустой документ, уже работая в MathCAD, следует выполнить одно из трех эквивалентных действий:

- нажмите одновременно клавиши <Ctrl>+<N>;
- нажмите кнопку **New** (Создать) на панели инструментов;
- выберите в верхнем меню **File / New** (Файл / Создать).

Создание документа можно производить с использованием шаблонов.

*Шаблон* (template) - это заготовка нового документа с введенными формулами, графиками, текстом, включая разметку, форматирование, выбор по умолчанию режима вычислений и т. д. Обычно новый документ создается на основе пустого шаблона. Для выбора другого шаблона, поставленного вместе с MathCAD или собственного, нажмите в диалоговом окне **New** (Создать) или кнопку **Browse** (Обзор). В появившемся диалоге **Browse** (Обзор) найдите местоположение нужного файла с шаблоном MathCAD. Эти файлы имеют расширение .mct (Math Cad Template). Выберите желаемый шаблон в списке файлов и нажмите кнопку **Открыть** (Open).

#### **2 Сохранение документа**

Для того чтобы сохранить документ в формате MathCAD, выберите **File / Save** (Файл / Сохранить) либо нажмите клавиши <Ctrl>+<S> или кнопку **Save** на стандартной панели инструментов. Если созданный документ сохраняется впервые, на экран будет выведено диалоговое окно **Сохранение** (Save), в котором потребуются определить его имя. Чтобы переименовать документ, сохраните его под другим именем с помощью команды **File / Save As** (Файл / Сохранить как) и появляющегося результате диалогового окна **Сохранение** (Save). В этом случае файл со старым названием не изменяется.

#### **3 Ввод математических выражений**

Ввести математическое выражение можно в любом пустом месте документа MathCAD. Для этого поместите курсор ввода в желаемое место документа, щелкнув в нем мышью, и просто начинайте вводить формулу, нажимая клавиши на клавиатуре. При этом в документе создается *математическая область*

(math region), которая предназначена для хранения формул, интерпретируемых процессором MathCAD.

1 Введите выражение  $x^{5+x}$ , для чего:

1.1 Щелкните мышью, обозначив место ввода.

1.2 Нажмите клавишу <x> - в этом месте вместо курсора ввода появится регион с формулой, содержащей один символ x, причем он будет выделен линиями ввода.

1.3 Введите оператор возведения в степень, нажав клавишу <^>, либо выбрав кнопку возведения в степень на панели инструментов **Calculator** (Калькулятор). В формуле появится местозаполнитель для введения значения степени, а линии ввода выделяют этот местозаполнитель.

1.4 Последовательно введите остальные символы <5>, <+>, <x>. Таким образом, поместить формулу в документ можно, просто начиная вводить символы, числа или операторы, например, + или /. Во всех этих случаях на месте курсора ввода создается математическая область, иначе называемая *регионом*, с формулой, содержащей и линии ввода. В последнем случае, если пользователь начинает ввод формулы с оператора, в зависимости от его типа, автоматически появляются и местозаполнители, без заполнения которых формула не будет восприниматься процессором MathCAD.

#### 4 Перемещение линий ввода внутри формул

Чтобы изменить формулу, щелкните на ней мышью, поместив таким образом в ее область линии ввода, и перейдите к месту, которое хотите исправить. Перемещайте линии ввода в пределах формулы одним из двух способов:

- щелкая в нужном месте мышью;
  - нажимая на клавиатуре клавиши - со стрелками, пробел и <Ins>:
  - клавиши со стрелками имеют естественное назначение, переводя линии ввода вверх, вниз, влево или вправо;
  - клавиша <Ins> переводит вертикальную линию ввода с одного конца горизонтальной линии ввода на противоположный;
- пробел предназначен для выделения различных частей формулы.

Осуществить перемещение линий ввода внутри формул всеми способами.

#### 5 Изменение формул

1 Вставка оператора. Операторы могут быть унарными (действующими на один операнд, как, на пример, оператор транспонирования матрицы или смены знака числа), так и бинарными (например + или /, действующими на два операнда). При вставке нового оператора в документ MathCAD определяет, сколько операндов ему требуется. Если в точке вставки оператор один или оба операнда отсутствуют, MathCAD автоматически помещает рядом с оператором один или два местозаполнителя. Для того чтобы вставить оператор не после, а перед частью формулы, выделенной линиями ввода, нажмите перед его вводом клавишу <Ins>, которая передвинет вертикальную линию ввода вперед. Это важно, в частности, для вставки оператора отрицания.

2 Научиться вставке операторов и переменных в любое место математического выражения. Операции: вырезки, копирования и вставки части формул, осуществляются через меню **Правка**.

## 2 Выполнение расчетов и визуализация результатов

### 1 Вычисление выражений

Для вычисления выражений используется знак равенства. Порядок вычислений определяется порядком написания формул!

Произвести ввод и вычисление выражений:

$$f1 = x^2 - y + \frac{\cos z}{2}; f2 = \frac{x + \sin^2 y}{z^5}; f3 = x^2 + \sqrt{y + \ln z}; f4 = \frac{|x - \sqrt{y}|}{z^2};$$
$$f5 = \frac{\operatorname{tg} x + \sqrt{y}}{z^3}; f6 = \frac{\sqrt{x + y^3}}{\ln z}; f7 = \frac{\sqrt{e^x + y^2}}{z}; f8 = \left| e^x + \frac{y^2}{z} \right|;$$
$$f9 = \log_3 z + x + \sqrt{3y}; f10 = \frac{\operatorname{tg} y + \arccos z}{x}; f11 = \frac{e^{y+z}}{|x-1,2|}; f12 = \frac{\lg x + \pi y}{z - 8,31}.$$

### 2 Ввод и редактирование текста

MathCAD - это математический редактор. Основное его назначение заключается в редактировании математических формул и расчете по ним. Вместе с тем, наряду с формульным редактором, MathCAD обладает довольно развитыми средствами по оформлению текста. Назначение текстовых областей в документах MathCAD для разных пользователей и разных задач может быть различным. Качественно стоит различать подход к тексту, используемому:

- просто в качестве комментариев;
- как элемент оформления документов для создания качественных отчетов в печатной и электронной формах.

1 Ввод текста. *Текстовую область* (или, по-другому, *регион с текстом* - text region) можно разместить в любом незанятом месте документа MathCAD. Однако, когда пользователь помещает курсор ввода в пустое место документа и просто начинает вводить символы, MathCAD по умолчанию интерпретирует их как начало формулы. Чтобы до начала ввода указать программе, что требуется создать не формульный, а текстовый регион, достаточно, перед тем как ввести первый символ, нажать клавишу "<>". В результате на месте курсора ввода появляется новый текстовый регион, который имеет характерное выделение. Курсор принимает при этом вид вертикальной линии красного цвета, которая называется *линией ввода текста* и аналогична по назначению линиям ввода в формулах.

2 Вставить текстовые комментарии о группе и исполнителе.

3 Произвести построение графиков следующих функций:

$$f(x,y)=(x^2 \cdot y^2); f(x)=x^3 - 20 \cdot x + 5; f(x)=x \cdot \cos(x); f(x)=x \cdot \sin(x); f(x)=x \cdot \sin(1/x);$$

$$G(a,b) = \begin{bmatrix} (5 + 2 \cdot \cos(a)) \cdot \cos(b) \\ (5 + 2 \cdot \cos(a)) \cdot \sin(b) \\ 2 \cdot \sin(a) \end{bmatrix}; Z(x,y) = x^2 + y^2 - 30.$$

Построение графиков выполняется при помощи специальной панели инструментов **Графика**.



### 3 Вычисление выражений с учетом единиц измерения, форматирование документа, основные приемы программирования

#### 1 Размерные переменные.

В MathCAD числовые переменные и функции могут обладать *размерностью*. Сделано это для упрощения инженерных и физических расчетов. В MathCAD встроено большое количество *единиц измерения*, с помощью которых и создаются размерные переменные.

2 Создание размерной переменной. Чтобы создать размерную переменную, определяющую, например, силу тока в 10 А:

- введите выражение, присваивающее переменной  $i$  значение ю:  $10:=10$   
Сразу после ввода ю введите символ умножения "\*";
- находясь в области местозаполнителя, выберите команду **Insert / Unit** (Вставка / Единицы), либо нажмите кнопку с изображением мерного стакана на стандартной панели инструментов, либо клавиши <Ctrl>+<U>;
- в списке **Unit** (Единица измерения) диалогового окна **Insert Unit** (вставка единицы измерений) выберите нужную единицу измерения **Ampere (A)**;
- нажмите кнопку ОК.

Если вы затрудняетесь с выбором конкретной единицы измерения, но знаете, какова размерность переменной (в нашем случае это электрический ток), то попробуйте выбрать ее в списке **Dimension** (размерность) диалогового окна **Insert Unit** (вставка единицы измерений). Тогда в списке **Unit** (единица измерения) появятся допустимые для этой величины единицы измерений. Просмотреть вставку единиц измерения можно и без выхода из диалогового окна **Insert Unit**, нажимая вместо кнопки ОК кнопку **Insert** (Вставить). В этом случае вы увидите, что единица измерений появилась в нужном месте документа, и можете поменять ее, оставаясь в диалоге **Insert Unit**.

3 Введите национальные единицы измерения и выполните расчеты выражения:

$$\sigma_{\text{evint}} = \sqrt{\left(\frac{4 \cdot Q}{\pi \cdot d1 \cdot \text{vint}^2}\right)^2 + \left(\frac{Mk}{0.2 \cdot d1 \cdot \text{vint}^3}\right)^2},$$

где Q- вертикальная нагрузка, Н, Mk - крутящий момент, Нм. d1 – диаметр винта, м.

4 Создайте скрытую зону в документе и поместите в нее определение национальных единиц измерения.

5 Произвести форматирование документа. Выровнять математические и текстовые области, привести шрифты к требованиям дипломных и курсовых проектов. Установить поля документа (20, 15, 15, 15мм), вставить номера страниц.

6 Язык программирования MathCAD. Для вставки программного кода в документы в MathCAD имеется специальная панель инструментов **Programming** (программирование), которую можно вызвать на экран нажатием кнопки **Programming Toolbar** на панели Math (математика). Большинство кнопок этой панели выполнено в виде текстового представления операторов программирования, поэтому их смысл легко понятен. Несмотря на принципиальную эквивалентность определения функций и переменных через встроенные функции MathCAD или программные модули, программирование имеет ряд существен-

ных преимуществ, которые в ряде случаев делают документ более простым и читаемым:

- возможность применения циклов и условных операторов;
- простота создания функций и переменных, требующих нескольких простых шагов;
- возможность создания функций, содержащих закрытый для остального документа код, включая преимущества использования локальных переменных и обработку исключительных ситуаций (ошибок).

7 Введите функцию  $y = \begin{cases} x^2, & \text{если } x \leq 0 \\ x, & \text{если } x > 0, x \leq 3 \\ 6 - x, & \text{если } x > 3 \end{cases}$

при помощи панели программирования.

8 Сохраните документ.

### *Контрольные вопросы*

- 1 Структура документа, понятие шаблона и их применение.
- 2 Экспорт документов и взаимодействие с другими программами.
- 3 Принципы ввода математических и текстовых выражений.
- 4 Работа с переменными и константами.
- 5 Основные методы редактирования документов, форматирование, стили.
- 6 Колонтитулы документа, формирование документа для печати.
- 7 Вычисление с размерными величинами, установка единиц измерения.
- 8 Зоны в документе, создание, назначение, установка свойств.
- 9 Графические возможности программы, принципы построения графиков.
- 10 Язык программирования MathCAD: назначение, особенности синтаксиса, ограничения.

## Лабораторная работа №2

### Автоматизация инженерных расчетов в среде MATLAB

**Цель работы** – изучить возможности программной среды; получить навыки расчетов различной сложности и графического представления данных; получение навыков по программированию в данной среде.

#### *1 Материальное обеспечение лабораторной работы*

1.1 ЭВМ Pentium – 166.

1.2 Программа MATLAB.

#### *2 Содержание лабораторной работы*

2.1 Освоение интерфейса и входного языка программы, определение возможностей и ограничений.

2.2 Расчеты и визуализация результатов.

2.3 Основные приемы программирования. Создание сценариев и функций.

#### *3 Общие положения*

Система MATLAB предлагается разработчиками (фирма Math Works, Inc.) как лидирующий на рынке, в первую очередь, в системе военно-промышленного комплекса, в аэрокосмической отрасли и автомобилестроении, язык программирования *высокого уровня* для технических вычислений с большим числом стандартных пакетов (прикладных программ).

Система MATLAB вобрала в себя не только передовой опыт развития и компьютерной реализации численных методов, накопленный за последние три десятилетия, но и весь опыт становления математики за всю историю человечества. Популярности системы способствует ее мощное расширение Simulink, предоставляющее удобные и простые средства, в том числе визуальное объектно-ориентированное программирование, для моделирования линейных и нелинейных динамических систем, а также множество других пакетов расширения системы.

Система MATLAB выполняет сложные и трудоемкие операции над векторами и матрицами даже в режиме прямых вычислений без какого-либо программирования. Ею можно пользоваться как мощнейшим калькулятором, в котором наряду с обычными арифметическими и алгебраическими действиями могут использоваться такие сложные операции, как инвертирование матрицы, вычисление ее собственных значений и принадлежащих им векторов, решение систем линейных уравнений, вывод графиков двумерных и трехмерных функций и многое другое.

Обычные числа и переменные в MATLAB рассматриваются как матрицы размером  $1 \times 1$ , что дает единообразные формы и методы проведения операций над обычными числами и массивами. Данная операция обычно называется векторизацией. Векторизация обеспечивает и упрощение записи операций, производимых одновременно над всеми элементами векторов и матриц, и существ-

венное повышение скорости их выполнения. Это также означает, что большинство функций может работать с аргументами в виде векторов и матриц. При необходимости вектора и матрицы преобразуются в массивы, и значения вычисляются для каждого их элемента.

MATLAB – расширяемая система, и ее легко приспособить к решению нужных вам классов задач. Ее огромное достоинство заключается в том, что это расширение достигается естественным путем и реализуется в виде так называемых т-файлов (с расширением .m). Иными словами, расширения системы хранятся на жестком диске компьютера и в нужный момент вызываются для использования точно так же, как встроенные в MATLAB (внутренние) функции и процедуры. Дополнительный уровень системы образуют ее пакеты расширения (toolbox). Они позволяют быстро ориентировать систему на решение задач в той или иной предметной области: в специальных разделах математики, в физике и в астрономии, в области нейтронных сетей и средств телекоммуникаций, в математическом моделировании, проектировании событийно-управляемых систем и т. д. Благодаря этому MATLAB обеспечивает высочайший уровень адаптации к решению задач конечного пользователя. Более подробно справку о возможностях данной среды можно найти в источниках /1-12/ и электронных учебниках кафедры в лаборатории "Информационных систем".

## **4 Выполнение лабораторной работы**

### **4.1 Освоение интерфейса и входного языка программы**

1 Запустить программу MATLAB.

Система MATLAB создана таким образом, что любые (подчас весьма сложные) вычисления можно выполнять в режиме *прямых вычислений*, то есть без подготовки программы.

2 Ознакомиться с основными диалоговыми панелями и меню программы.

3 В окне **Команды** вычислить выражения:  $5+80\cdot\pi$ ;  $\sin(5)$ ;  $e^{\sin(45)}+1/\cos(25)$ .

Особенности вычислений:

- для указания ввода исходных данных используется символ »;
- для блокировки вывода результата вычислений некоторого выражения после него надо установить знак;
- если не указана переменная для значения результата вычислений, то MATLAB назначает такую переменную с именем ans;
- знаком присваивания является привычный математикам знак равенства =, а не комбинированный знак :=, как во многих других языках программирования и математических системах;
- результат вычислений выводится в строках вывода (без знака »);
- встроенные функции (например, sin) записываются строчными буквами, и их аргументы указываются в *круглых скобках*;
- если вводимое математическое выражение длиннее одной строки, то в этом случае часть выражения можно перенести на новую строку с помощью знака многоточия "...".
- символьные (тестовые выражения) записываются в апострофах;

4 Ввести два вектора  $v1=[1\ 3\ 5\ 8]$  и  $v2=[5\ 67\ 44\ 45]$ . Произвести произведение и деление векторов.

Формирование упорядоченных числовых последовательностей. Такие последовательности нужны для создания векторов или значений абсциссы при построении графиков. Для этого в MATLAB используется оператор:

<Начальное значение>: <Шаг>: <Конечное значение>.

5 Сформировать вектор  $V3$  от 0 до 50 с шагом 3.

6 Сформировать таблицу  $M=[1\ 2\ 3: 4\ 5\ 6; 7\ 8\ 9]$ .

Возможен ввод элементов матриц и векторов в виде арифметических выражений, содержащих любые доступные системе функции.

7 Ввести вектор  $V = [2+2/(3+4)\ \exp(5)\ \sqrt{10}]$ .

## 4.2 Расчеты и визуализация результатов

1 Произвести вычисление выражений:

$$f1 = x^2 - y + \frac{\cos z}{2}; f2 = \frac{x + \sin^2 y}{z^5}; f3 = x^2 + \sqrt{y + \ln z}; f4 = \frac{|x - \sqrt{y}|}{z^2};$$

$$f5 = \frac{\operatorname{tg} x + \sqrt{y}}{z^3}; f6 = \frac{\sqrt{x + y^3}}{\ln z}; f7 = \frac{\sqrt{e^x + y^2}}{z}; f8 = \left| e^x + \frac{y^2}{z} \right|;$$

$$f9 = \log_3 z + x + \sqrt{3y}; f10 = \frac{\operatorname{tg} y + \arccos z}{x}; f11 = \frac{e^{y+z}}{|x-1,2|}; f12 = \frac{\lg x + \pi y}{z - 8,31}.$$

2 Выполнить построение двумерных графиков. Для построения используется команда **plot(f(x))**. Аргументы функции и диапазон их изменения должен быть определен заранее. Графики MATLAB строит в отдельных окнах, называемых графическими окнами.

3 Произвести построение графиков следующих функций:

$$f(x,y)=(x^2 \cdot y^2); f(x)=x^3 - 20 \cdot x + 5; f(x)=x \cdot \cos(x); f(x)=x \cdot \sin(x); f(x)=x \cdot \sin(1/x)$$

4. Построить несколько графиков в одном графическом окне при помощи команды: **plot(a1,f1,a2,f2,a3,f3)**, где  $a1, a2, a3$  – векторы аргументов функций,  $f1, f2, f3$  – векторы значений функций, графики которых строятся в одном окне.

5 Построить графики с помощью команды **fplot('f(x)', [xmin xmax])**.

6 Построить столбчатый график функции  $e^{-x^x}$  с помощью команды **bar(x,f(x))**;

7 Построить график с указанием погрешности каждой точки:

$$x=-2:0.1:2; y=\operatorname{erf}(x); e = \operatorname{rand}(\operatorname{size}(x))/10; \operatorname{errorbar}(x,y,e);$$

8 Построить график дискретных отсчетов:

$$x = 0:0.1:4; y = \sin(x.^2) \cdot \exp(-x); \operatorname{stem}(x,y).$$

9 Построить график в полярных координатах:

$$t=0:.01:2 \cdot \pi; \operatorname{polar}(t, \operatorname{abs}(\sin(2 \cdot t) \cdot \cos(2 \cdot t))).$$

10 Построить трехмерный график командой **plot3(x,y,z)**, где  $x,y,z$  – векторы.

Ввести:  $[x,y] = \operatorname{meshgrid}([-3:0.15:3]); z=x.^2+y.^2; \operatorname{plot3}(x,y,z)$ .

11 Постройте функцию  $z=x.^2+y.^2$  при помощи команды **mesh(z)**

12 Построить график командой **surf(z)**:

$$z=\operatorname{peaks}(25); \operatorname{surf}(z); \operatorname{colormap}(\operatorname{jet}).$$

13 Построить графики функций:

$$G(a,b) = \begin{bmatrix} (5 + 2 \cdot \cos(a)) \cdot \cos(b) \\ (5 + 2 \cdot \cos(a)) \cdot \sin(b) \\ 2 \cdot \sin(a) \end{bmatrix}; Z(x,y) = x^2 + y^2 - 30.$$

### 4.3 Создание сценариев и функций

Программами в системе MATLAB являются m-файлы текстового формата, содержащие запись программ в виде программных кодов. Язык программирования системы MATLAB имеет следующие средства:

- данные различного типа;
- константы и переменные;
- операторы, включая операторы математических выражений;
- встроенные команды и функции;
- функции пользователя;
- управляющие структуры;
- системные операторы и функции;
- средства расширения языка.

Файл-сценарий, именуемый также Script-файлом, является просто записью серии команд без входных и выходных параметров. Он имеет следующую структуру:

```
%основной комментарий;
%дополнительный комментарий;
<тело файла с любыми выражениями>.
```

Комментарии – любой текст, служащий для пояснения команд, не влияющий на результаты расчетов и начинается всегда знаком %!

Важны следующие свойства файлов-сценариев:

- они не имеют входных и выходных аргументов;
- работают с данными из рабочей области;
- в процессе выполнения не компилируются;
- представляют собой зафиксированную в виде файла последовательность операций, полностью аналогичную той, что используется в сессии.

1 Создайте новый m-файл, через меню файл(File). Введите код построения

```
графика: %Plot with color red
          %Строит график синусоиды линией красного цвета
          %с выведенной масштабной сеткой в интервале [xmin. xmax]
          x=xmin:0.1:xmax;
          plot(x.sin(x)..'r')
          grid on.
```

2 Сохраните файл-сценарий с именем на test\_skript1. Запустите на выполнение данный файл.

*M-файл-функция* является типичным объектом языка программирования системы MATLAB. Одновременно он является полноценным модулем с точки зрения структурного программирования, поскольку содержит входные и выходные параметры и использует аппарат локальных переменных. Структура такого модуля с одним выходным параметром выглядит следующим образом:

```
function var=f_name(Список_параметров)
%Основной комментарий
%Dополнительный комментарий
Тело файла с любыми выражениями
var=выражение.
```

М-файл-функция имеет следующие свойства:

- он начинается с объявления `function`, после которого указывается имя переменной `var` – выходного параметра, имя самой функции и список ее входных параметров;
- функция возвращает свое значение и может использоваться в виде `name(Список_параметров)` в математических выражениях;
- все переменные, имеющиеся в теле файла-функции, являются *локальными*, т. е. действуют только в пределах тела функции;
- файл-функция является самостоятельным программным модулем, который общается с другими модулями через свои входные и выходные параметры;
- правила вывода комментариев те же, что у файлов-сценариев;
- файл-функция служит средством расширения системы MATLAB;
- при обнаружении файла-функции он компилируется и затем исполняется, а созданные машинные коды хранятся в рабочей области системы MATLAB.

Последняя конструкция `var = выражение` вводится, если требуется, чтобы функция возвращала результат вычислений.

3 Создайте новый `m`-файл, через меню файл(File). Введите код М-файл-функции:

```
function f1=f1()
R=input('Введите одну из цифр:1,2,3');
    if R==1 % A complex plot of f(z) = z^3
        % Three maxima at the cube roots of 1
        z=cplxgrid(20);
        cplxmap(z,z.^3);
    elseif R==2
        % A complex plot of f(z) = (z^4-1)^(1/4)
        % Four zeros at the fourth roots of 1
        z=cplxgrid(20);
        cplxmap(z,(z.^4-1).^(1/4));
    elseif R==3
        % A complex plot of f(z) = z^(1/3)
        % The Riemann surface for the cube root
        cplxroot(3,15);
    end
end.
```

4 Сохраните М-файл-функцию с именем `test_function1`. Запустите на выполнение данный файл.

### *Контрольные вопросы*

- 1 Структура интерфейса программы, назначение окон.
- 2 Принципы взаимодействия с другими программами.
- 3 Принципы ввода математических и текстовых выражений.
- 4 Работа с переменными и константами, форматы представления данных.
- 5 Основные приемы вычислений с таблицами и матрицами.
- 6 Графические возможности программы.
- 7 Принципы построения графиков.
- 8 Экспорт графических изображений.
- 9 Язык программирования MATLAB: назначение, особенности синтаксиса, ограничения.
- 10 Файлы, сценарии, функции, назначение, применение, отличительные черты.



## Лабораторная работа №3

### Проектирование систем управления в среде MATLAB

**Цель работы** – познакомиться с пакетами расширений Fuzzy Logic Toolbox и Neural Networks Toolbox среды MATLAB; получить навыки построения систем управления на основе методов нечеткой логики; построить аппроксиматоры на основе нейросетей.

#### *1 Материальное обеспечение лабораторной работы*

1.1 ЭВМ Pentium – 166.

1.2 Программа MATLAB.

#### *2 Содержание лабораторной работы*

2.1 Освоение интерфейса пакета расширений Fuzzy Logic Toolbox, определение возможностей и ограничений.

2.2 Создание системы управления светофорного объекта на основе нечеткой логики.

2.3 Ознакомление с приложениями MATLABa, построенных на пакете расширений Neural Networks Toolbox.

2.4 Построение аппроксиматора на основе нейросетевой технологии.

#### *3 Общие положения*

##### **3.1 Теоретические основы применения "нечеткой логикой" (Fuzzy logic)**

Решение качественных или смешанных задач в настоящее время является наиболее сложным, т.к. необходимо решать задачи, не поддающиеся полностью или частично формализации для применения стандартной логики. Поэтому качественные задачи пока в основном решаются человеком, но есть уже и опыт решения их нейронными сетями или экспертными системами.

Управление техническими или иными объектами в большинстве своем являются задачами количественными. Аналитически управление объектом сводятся к решению одного ли системы дифференциальных уравнений.

Существуют специальные прикладные программы для моделирования динамических систем таких, как Visimm. При достаточно точном описании процесса системой уравнений ее целесообразно решать аналитическим или численным методом. Эффективно такие вычисления производятся в среде MathCAD, MATLAB.

Однако зачастую аналитическое описание объекта (математическая модель) или недостаточно верно, или решение носит сложный характер. В этих случаях прибегают к методам моделирования, анализа, построенных на основе нейросетей, нечеткой логики, генетических алгоритмах, то есть методах искусственного интеллекта (ИИ).

Нечеткая логика возникла как наиболее удобный способ построения систем управления метрополитенами и сложными технологическими процессами, а также нашла применение в бытовой электронике, диагностических и других экспертных системах. Несмотря на то, что математический аппарат нечеткой

логики впервые был разработан в США, активное развитие данного метода началось в Японии, и новая волна вновь достигла США и Европы.

Термин fuzzy (англ. нечеткий, размытый - произносится 'фаззи') является ключевым понятием. Нечеткая логика является многозначной логикой, что позволяет определить промежуточные значения для таких общепринятых оценок, как да|нет, истинно|ложно, черное|белое и т.п. Выражения подобные таким, как "слегка тепло" или "довольно холодно" становится возможно формулировать математически и обрабатывать на компьютерах. Нечеткая логика появилась в 1965 в работах Лотфи А. Задэ (Lotfi A. Zadeh), профессора технических наук Калифорнийского университета в Беркли.

Рассмотрим базовые понятия "нечеткой логики". Самым главным понятием систем, основанных на нечеткой логике, является понятие нечеткого (под)множества.

*Нечеткое множество* - это такое множество, которое образуется путем введения обобщенного понятия принадлежности, т.е. расширения двухэлементного множества значений функции принадлежности  $\{0,1\}$  до отрезка  $[0,1]$ . Это означает, что переход от полной принадлежности объекта множеству к его полной непринадлежности происходит не скачком, как в обычных "четких" множествах, а плавно, постепенно, причем степень принадлежности элемента множеству выражается числом из интервала  $[0,1]$ .

Таким образом, нечеткое множество  $A = \{(x, \mu_A(x))\}$  определяется математически как совокупность упорядоченных пар, составленных из элементов  $x$  множества  $X$  и соответствующих им степеней принадлежности  $\mu_A(x)$  или непосредственно в виде функции  $\mu_A: X \rightarrow [0,1]$ .

*Функцией принадлежности (membership function)* называется функция, которая позволяет вычислить степень принадлежности произвольного элемента универсального множества к нечеткому множеству. Графическое представление функции принадлежности называется *термом*.

*Нечеткое число* – это нечеткое подмножество универсального множества действительных чисел, имеющее *нормальную* и *выпуклую* функцию принадлежности, то есть такую, что а) существует такое значение носителя, в котором функция принадлежности равна единице, а также б) при отступлении от своего максимума влево или вправо функция принадлежности убывает.

Понятие нечеткого множества - это попытка математической формализации нечеткой информации для построения математических моделей. В основе этого понятия лежит представление о том, что составляющие данное множество элементы, обладающие общим свойством, могут обладать этим свойством в различной степени и, следовательно, принадлежать к данному множеству с различной степенью.

Из вышесказанного можно сделать следующие выводы:

- 1) нечеткие множества описывают неопределенные понятия (быстрый бегун, горячая вода, жаркая погода);
- 2) нечеткие множества допускают возможность частичной принадлежности к ним (пятница - частично выходной день (укороченный), погода скорее жаркая);
- 3) степень принадлежности объекта к нечеткому множеству определяется соответствующим значением функции принадлежности на интервале  $[0,1]$ ;

4) функция принадлежности ставит в соответствие объекту (или логической переменной) значение степени его принадлежности к нечеткому множеству.

Для наглядности приведем функцию принадлежности ( терм) множества молодых людей.

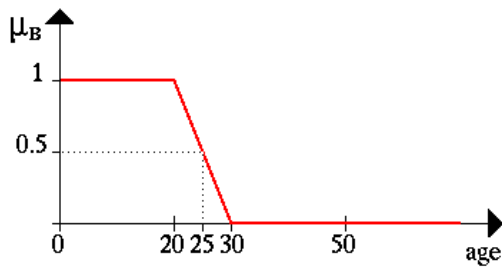


Рисунок 3.1 - Функция принадлежности  $\mu_{\tilde{A}}$  множества В

Из рисунка 3.1 видно, что 25-летние все еще молоды со степенью принадлежности 0,5.

Определим базовые операции (действия) над нечеткими множествами (числами). Аналогично действиям с обычными множествами нам потребуется определить пересечение, объединение и отрицание нечетких множеств. В своей самой первой работе по нечетким множествам Л. А. Заде предложил оператор минимума для пересечения и оператор максимума для объединения двух нечетких множеств. Легко видеть, что эти операторы совпадают с обычными (четкими) объединением и пересечением, только рассматриваются степени принадлежности 0 и 1. Чтобы пояснить это, приведем несколько примеров. Пусть А – нечеткое число (интервал) от 5 до 8 и В – нечеткое число около 4, как показано на рисунке.

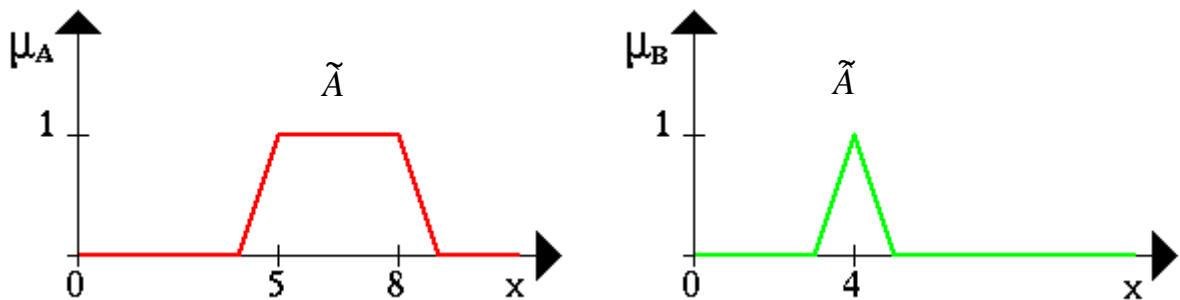


Рисунок 3.2 - Графическое представление двух нечетких чисел А и В

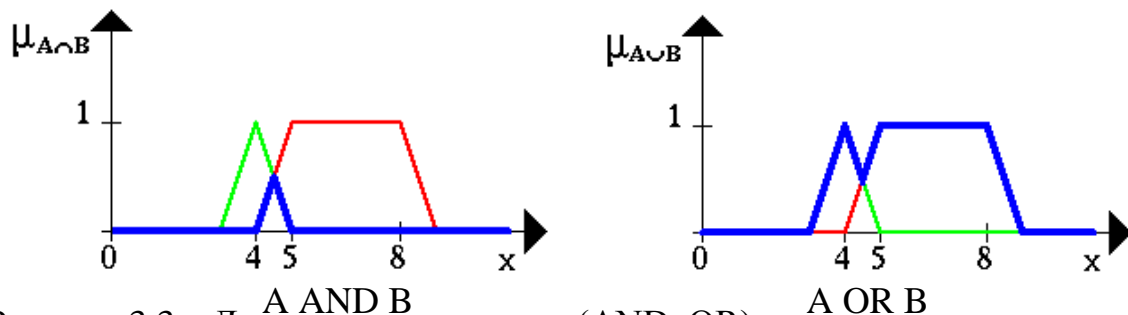


Рисунок 3.3 – Логические операции (AND, OR) с нечеткими числами А и В

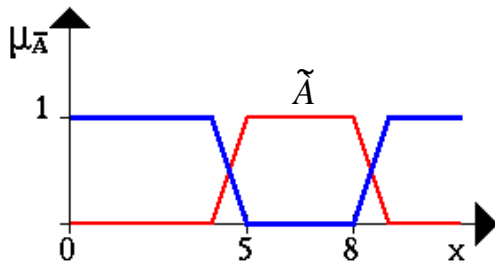


Рисунок 3.4 – Логическое отрицание (NOT) нечеткого числа A

*Фаззификацией* (fuzzification) называется процедура преобразование "четкой" точки  $\bar{O} = (\tilde{o}_1, \dots, \tilde{o}_n) \in \tilde{O}$  в нечеткое множество числа  $\tilde{A}$  из X. Другими словами нахождение значений функции принадлежности нечеткого числа в заданной точке. На этом этапе происходит установление соответствия между численным значением входной переменной и значением функции принадлежности соответствующего ей терма входной лингвистической переменной. Так, *фаззификация* числа 25 (рисунок 3.1) дает нам значение 0,5.

*Дефаззификацией* (defuzzification) называется процедура преобразования нечеткого множества в четкое число.

В теории нечетких множеств процедура дефаззификации аналогична нахождению характеристик положения (математического ожидания, моды, медианы) случайных величин в теории вероятности. Простейшим способом выполнения процедуры дефаззификации является выбор четкого числа, соответствующего максимуму функции принадлежности. Однако пригодность этого способа ограничивается лишь одноэкстремальными функциями принадлежности. Для многоэкстремальных функций принадлежности в Fuzzy Logic Toolbox запрограммированы такие методы дефаззификации:

- Centroid - центр тяжести;
- Bisector - медиана;
- LOM (Largest Of Maximums) - наибольший из максимумов;
- SOM (Smallest Of Maximums) - наименьший из максимумов;
- Mom (Mean Of Maximums) - центр максимумов.

*Нечеткой базой знаний* (fuzzy knowledge base) о влиянии факторов  $X = x_1 \dots x_n$  на значение параметра Y называется совокупность логических высказываний типа:

ЕСЛИ  $(x_1 = \mu_1)$  И  $(x_2 = \mu_2), \dots$  И  $(x_n = \mu_n)$  ИЛИ  $(x_1 = \mu_1)$ , ИЛИ  $(x_2 = \mu_2), \dots$ , ИЛИ  $(x_k = \mu_k)$ ,

ТО  $y_i = \mu_j$ ,

где  $\mu_1 \dots \mu_n$  - нечеткий терм, которым оцениваются входные переменные  $x_1 \dots x_n$ ;

$\mu_j$  - нечеткий терм, которым оцениваются выходные переменные  $y_1 \dots y_n$ .

Графически систему на основе нечеткой логики можно представить в виде блоков (рисунок 3.5).

Рассмотрим разработку подобных систем в среде MATLAB.

MATLAB – матричная лаборатория – наиболее развитая система программирования для научно-технических расчетов, дополненная к настоящему времени несколькими десятками более частных приложений, относящихся к

вычислительной математике, обработке информации, конструированию электронных приборов, экономике и ряду других разделов прикладной науки.

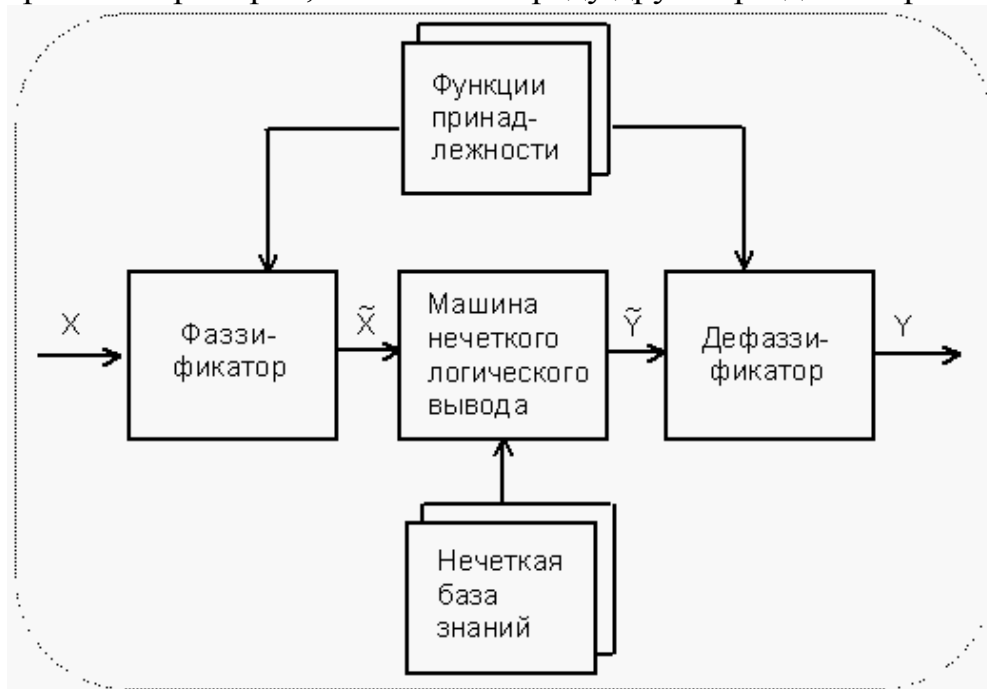


Рисунок 3.5 - Структурная схема системы, использующей "нечеткую логику"

$X$  - входной четкий вектор;  $\tilde{O}$  - вектор нечетких множеств, соответствующий входному вектору  $X$ ;  $\tilde{Y}$  - результат логического вывода в виде вектора нечетких множеств;  $Y$  - выходной четкий вектор.

MATLAB – система программирования высокого уровня, работающая как интерпретатор и включающая большой набор инструкций (команд) для выполнения самых разнообразных вычислений, задания структур данных и графического представления информации. Команды эти разбиты на тематические группы, расположенные в различных директориях системы.

Fuzzy Logic Toolbox - это пакет прикладных программ, входящих в состав среды MATLAB. Он позволяет создавать системы нечеткого логического вывода и нечеткой классификации в рамках среды MATLAB, с возможностью их интегрирования в Simulink.

Базовым понятием Fuzzy Logic Toolbox является FIS-структура - система нечеткого вывода (Fuzzy Inference System). FIS-структура содержит все необходимые данные для реализации функционального отображения "входы-выходы" на основе нечеткого логического вывода согласно схеме, приведенной на рисунке 3.5.

Fuzzy Logic Toolbox содержит следующие категории программных инструментов:

- функции;
- интерактивные модули с графическим пользовательским интерфейсом (с GUI);
- блоки для пакета Simulink;
- демонстрационные примеры.

Модуль fuzzy позволяет строить нечеткие системы двух типов - Мамдани и Сугэно. В системах типа Мамдани база знаний состоит из правил вида "Если

$x_1$ =низкий и  $x_2$ =средний, то  $y$ =высокий”. В системах типа Сугэно база знаний состоит из правил вида “Если  $x_1$ =низкий и  $x_2$ =средний, то  $y=a_0+a_1x_1+a_2x_2$ ”. Таким образом, основное отличие между системами Мамдани и Сугэно заключается в разных способах задания значений выходной переменной в правилах, образующих базу знаний. В системах типа Мамдани значения выходной переменной задаются нечеткими термами, в системах типа Сугэно - как линейная комбинация входных переменных.

### 3.2 Теоретические основы нейросетевых технологий

Искусственные нейронные сети индуцированы биологией, так как они состоят из элементов, функциональные возможности которых аналогичны большинству элементарных функций биологического нейрона. Эти элементы затем организуются по способу, который может соответствовать (или не соответствовать) анатомии мозга. Несмотря на такое поверхностное сходство, искусственные нейронные сети демонстрируют удивительное число свойств, присущих мозгу. Например, они обучаются на основе опыта, обобщают знание и извлекают существенные свойства из поступающей информации.

Искусственные нейронные сети могут менять свое поведение в зависимости от внешней среды. Этот фактор в большей степени, чем любой другой, ответствен за тот интерес, который они вызывают. После предъявления входных сигналов (возможно, вместе с требуемыми выходами) они самонастраиваются, чтобы обеспечивать требуемую реакцию.

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона. На рисунке 3.6 представлена модель, реализующая эту идею. Хотя сетевые парадигмы весьма разнообразны, в основе почти всех их лежит эта конфигурация. Здесь множество входных сигналов, обозначенных  $x_1, x_2, \dots, x_n$ , поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором  $\mathbf{X}$ , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес  $w_1, w_2, \dots, w_n$ , и поступает на суммирующий блок, обозначенный  $\Sigma$ . Каждый вес соответствует «силе» одной биологической синаптической связи, (множество весов в совокупности обозначается вектором  $\mathbf{W}$ ). Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, который мы будем называть NET. В векторных обозначениях это может быть компактно записано следующим образом:

$$\text{NET} = \mathbf{XW}.$$

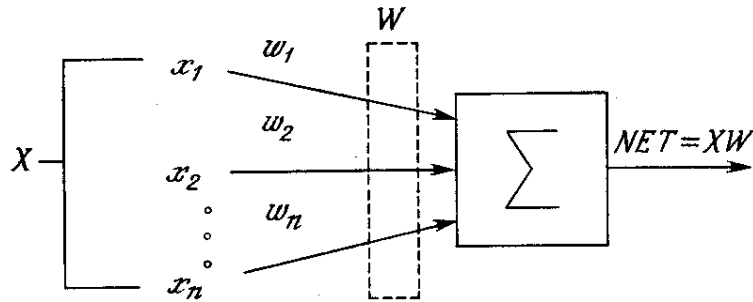


Рисунок 3.6 - Искусственный нейрон

Сигнал NET далее, как правило, преобразуется активационной функцией  $F$  и дает выходной нейронный сигнал OUT. Активационная функция может быть обычной линейной функцией

$$\text{OUT} = K(\text{NET}),$$

где  $K$  – постоянная, пороговой функции;

$$\text{OUT} = 1, \text{ если } \text{NET} > T,$$

$$\text{OUT} = 0 \text{ в остальных случаях,}$$

где  $T$  – некоторая постоянная пороговая величина, или же функцией, более точно моделирующей нелинейную передаточную характеристику биологического нейрона и представляющей нейронной сети большие возможности.

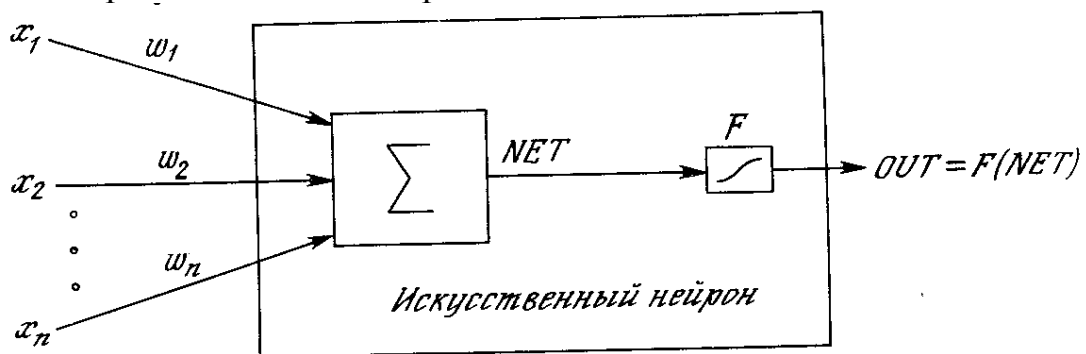


Рисунок 3.7- Искусственный нейрон с активационной функцией

На рисунке 3.7 блок, обозначенный  $F$ , принимает сигнал NET и выдает сигнал OUT. Если блок  $F$  сужает диапазон изменения величины NET так, что при любых значениях NET значения OUT принадлежат некоторому конечному интервалу, то  $F$  называется «сжимающей» функцией. В качестве «сжимающей» функции часто используется логистическая или «сигмоидальная» (S-образная) функция. Эта функция математически выражается как  $F(x) = 1/(1 + e^{-x})$ . Таким образом,

$$\text{OUT} = \frac{1}{1 + e^{-\text{NET}}}.$$

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его небольшому приращению величины NET.

Признаки, при наличии которых задачу целесообразно решать с применением нейросетевых технологий:

- отсутствие алгоритма и математической модели решения задачи;
- имеются данные экспериментальных исследований;

- проблема характеризуется большими объемами входной информации, причем данные носят вероятностный или нечеткий характер, зашумлены, искажены или противоречивы.

Любая нейронная сеть может быть представлена в виде схемы, (рисунок 3.8).

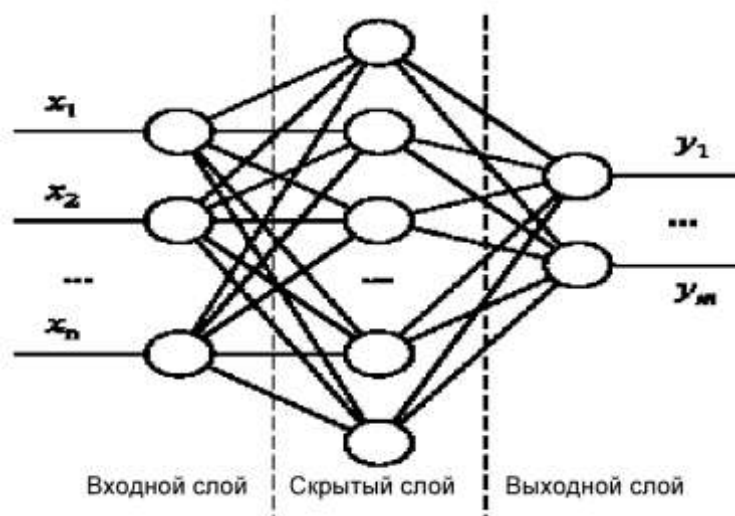


Рисунок 3.8 – Структура трехслойной нейронной сети

После построения сети требуется ее обучение. Схема обучения представлена на рисунке 3.9.

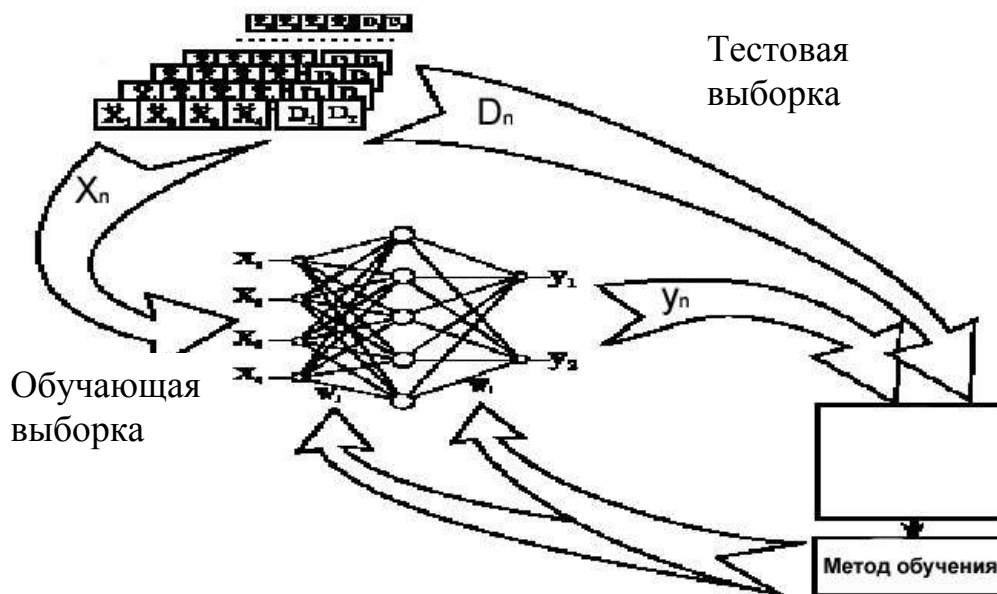


Рисунок 3.9 – Структура обучения нейронной сети

Методы (алгоритмы) обучения НС делятся на два класса:

- **обучение с учителем** - сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей.
- **обучение без учителя** - выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

После обучения сеть готова к работе, то есть решения задач классификации, аппроксимации и др.



#### 4 Выполнение лабораторной работы

4.1. Создания системы управления светофорного объекта. Обычно для таких задач выбирают систему типа Мамдани.

*Постановка задачи.* Система управления должна регулировать время работы светофора в режиме зеленого света в зависимости от количества подъезжающих к перекрестку машин, (рисунок 3.10).

*Решение задачи.* Для работы нечеткого светофора на перекрестке улиц Север-Юг (СЮ) и Запад-Восток (ЗВ) необходимо установить 8 датчиков (рисунок 3.10), которые считают проехавшие мимо них машины.

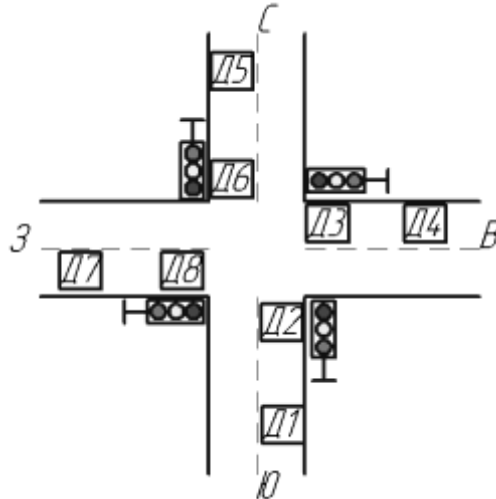


Рисунок 3.10 - Расположение датчиков на перекрестке

Светофор использует разности показаний четырех пар датчиков: (Д1-Д2), (Д3-Д4), (Д5-Д6) и (Д7-Д8). Таким образом, если для улицы СЮ горит зеленый свет, машины проезжают перекресток и показания двух пар датчиков равны: Д1=Д2, Д5=Д6, а, следовательно, их разность равна нулю. В это же время на улице ЗВ перед светофором останавливаются машины, которые успели проехать только Д4 и Д7. В результате можно рассчитать суммарное количество автомобилей на этой улице следующим образом:

$$(Д4-Д3)+(Д7-Д8)=(Д4-0)+(Д7-0)=Д4+Д7.$$

Поскольку работа светофора зависит от числа машин на обеих улицах и текущего времени зеленого света, для нашей системы предлагается использовать 3 входа:

- число машин на улице СЮ по окончании очередного цикла (интенсивность СЮ);
- число машин на улице ЗВ по окончании цикла (интенсивность ЗВ);
- время зеленого света нечеткого светофора, сек. (фаза светофора).

*Шаг 1.* Для загрузки основного fis-редактора напечатаем слова fuzzy в командной строке MATLAB. После этого откроется нового графическое окно, показанное на рисунке 3.11.

*Шаг 2.* Добавим вторую и третью входную переменную. Для этого в меню Edit выбираем команду Add input.

*Шаг 3.* Переименуем первую входную переменную. Для этого сделаем один щелчок левой кнопкой мыши на блоке input1, введем новое обозначение **фаза-**

**светофора** в поле редактирования имени текущей переменной и нажмем Enter>.

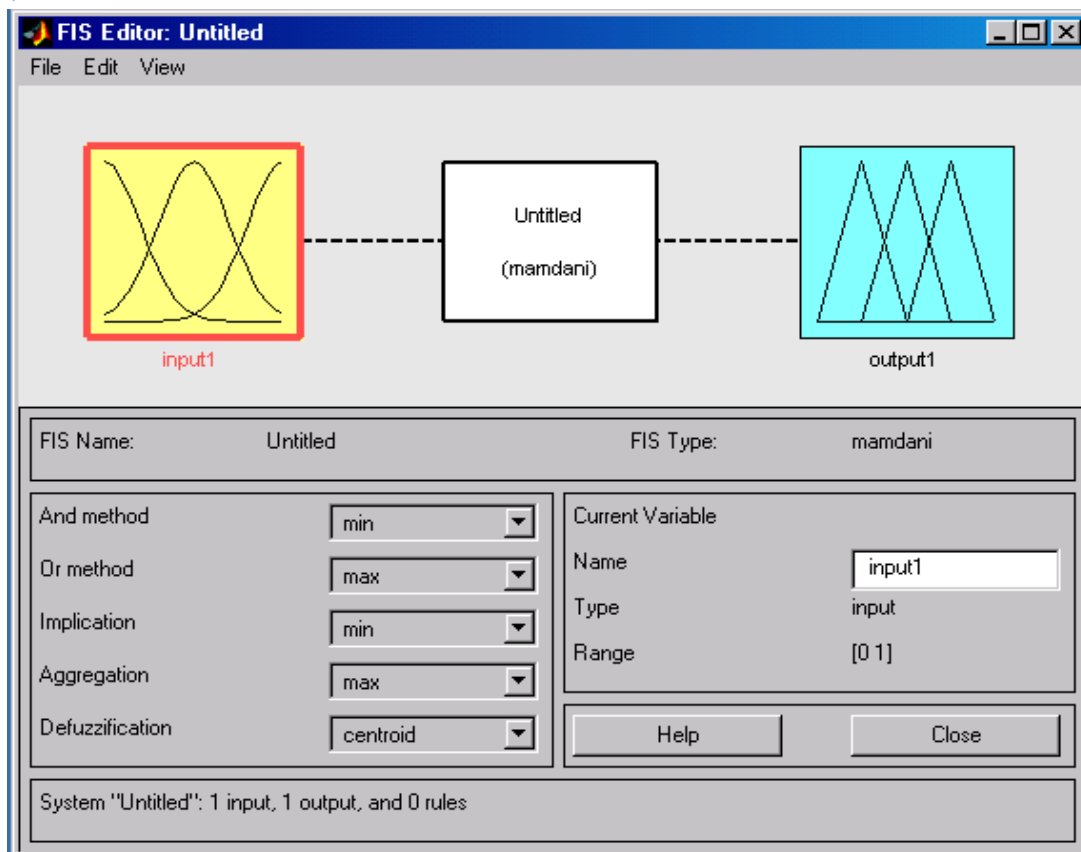


Рисунок 3.11 - Окно редактора FIS-Editor

*Шаг 4.* Переименуем остальные входные переменные. Для этого сделаем один щелчок левой кнопкой мыши на блоке input2, введем новое обозначение **интенсивностьСЮ** в поле редактирования имени текущей переменной и нажмем <Enter>. Аналогично переименовываем третью переменную.

*Шаг 5.* Переименуем выходную переменную. Для этого сделаем один щелчок левой кнопкой мыши на блоке output1, введем новое обозначение **изменить-фазу** в поле редактирования имени текущей переменной и нажмем <Enter>.

*Шаг 6.* Зададим имя системы. Для этого в меню File выбираем в подменю Export команду To disk и вводим имя файла, например, svetofor.

*Шаг 7.* Перейдем в редактор функций принадлежности. Для этого сделаем двойной щелчок левой кнопкой мыши на блоке **фаза-светофора**.

*Шаг 8.* Зададим диапазон изменения переменной **фаза-светофора**. Для этого напечатаем 0 50 в поле Range (рисунок 3.12) и нажмем <Enter>.

*Шаг 9.* Зададим функции принадлежности переменной **фаза-светофора**. Для лингвистической оценки этой переменной будем использовать 3 термина с трапециидальными функциями принадлежности. Для этого в меню Edit выберем команду Add MFs... В результате появится диалоговое окно выбора типа и количества функций принадлежности. По умолчанию, это 3 термина с треугольными функциями принадлежности. Поэтому просто нажимаем <Enter>.

*Шаг 10.* Зададим наименования термов переменной **фаза-светофора**. Для этого делаем один щелчок левой кнопкой мыши по графику первой функции принадлежности (рисунок 3.12). Затем вводим наименование термина, например,

малое, в поле Name и нажмем <Enter>. Затем делаем один щелчок левой кнопкой мыши по графику второй функции принадлежности и вводим наименование термина, например, Среднее, в поле Name и нажмем <Enter>. Еще раз делаем один щелчок левой кнопкой мыши по графику третьей функции принадлежности и вводим наименование термина, например, Высокое, в поле Name и нажмем <Enter>. В результате получим графическое окно, изображенное на рисунке 3.12.

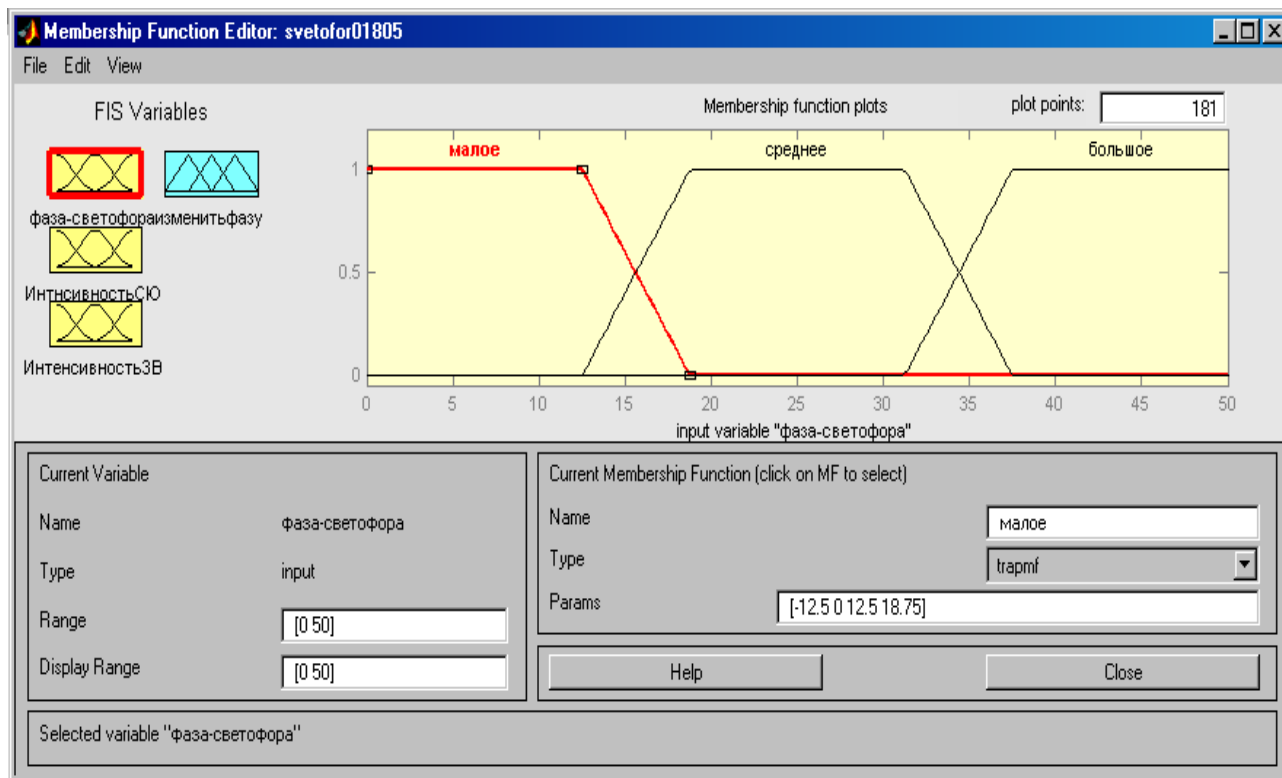


Рисунок 3.12 - Функция принадлежности фазы зеленого сигнала светофора

*Шаг 11.* Зададим функции принадлежности переменной **интенсивностьСЮ**. Для лингвистической оценки этой переменной будем использовать 5 термов с трапециидальными функциями принадлежности. Для этого активизируем переменную **интенсивностьСЮ** с помощью щелчка левой кнопки мыши на блоке **интенсивностьСЮ**. Зададим диапазон изменения переменной **интенсивностьСЮ**. Для этого напечатаем 0 90 в поле Range (рисунок 3.13) и нажмем <Enter>. Затем в меню Edit выберем команду Add MFs.... В появившемся диалоговом окне выбираем тип функции принадлежности trapmf в поле MF type и 5 термов в поле Number of MFs. После этого нажимаем <Enter>.

*Шаг 12.* По аналогии с шагом 10 зададим следующие наименования термов переменной **интенсивностьВЗ**: Очень малое, малое, среднее, большое, очень большое.

*Шаг 13.* Зададим функции принадлежности переменной **изменить-фазу**. Для лингвистической оценки этой переменной будем использовать 3 термина с гауссовкой функциями принадлежности. Для этого активизируем переменную **изменить-фазу** с помощью щелчка левой кнопки мыши на блоке **изменить-фазу**. Зададим диапазон изменения переменной **изменить-фазу**. Для этого напечатаем -20 20 в поле Range (см. рисунке 3.14) и нажмем <Enter>. Затем в меню Edit выберем команду Add MFs.... В появившемся диалоговом окне выбираем 3 термина в поле Number of MFs. После этого нажимаем <Enter>.

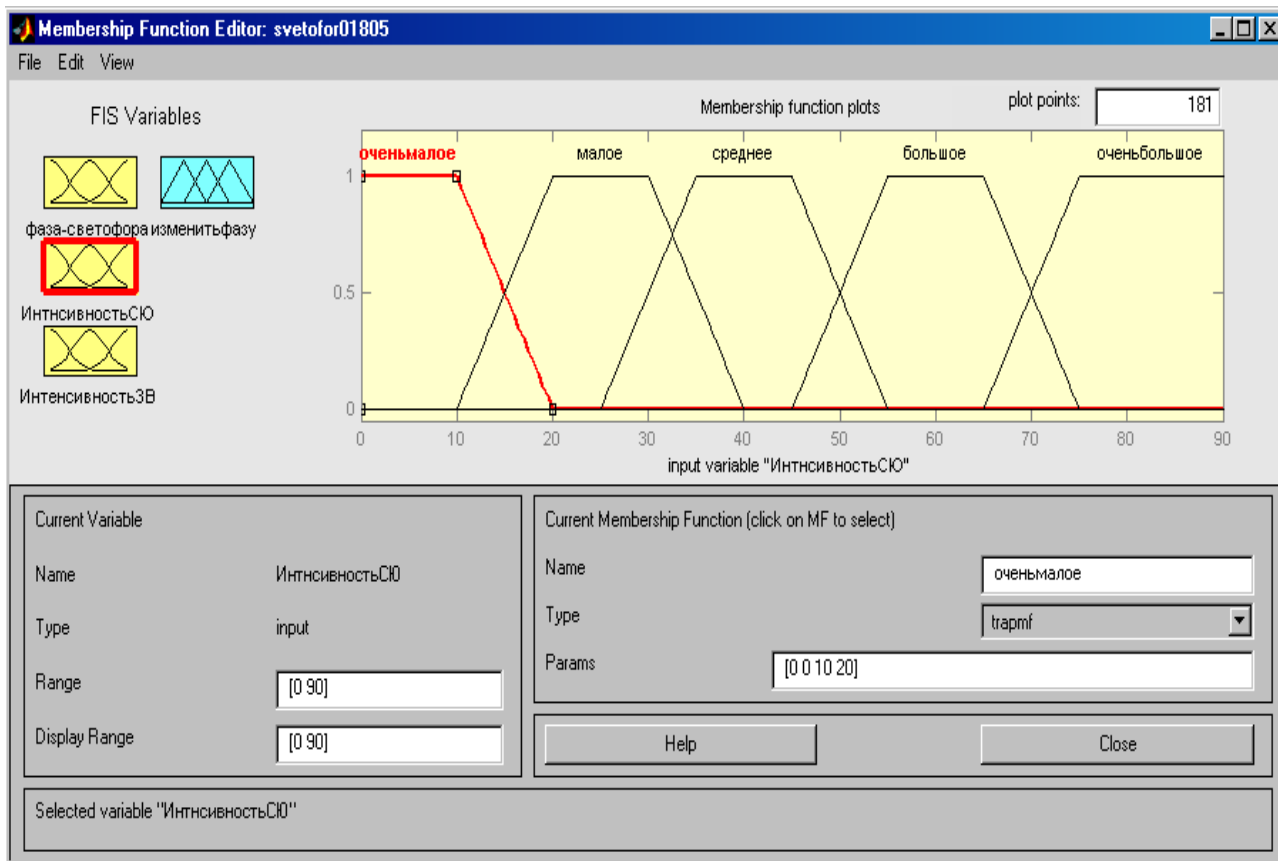


Рисунок 3.13 - Функция принадлежности интенсивности движения по направлению СЮ

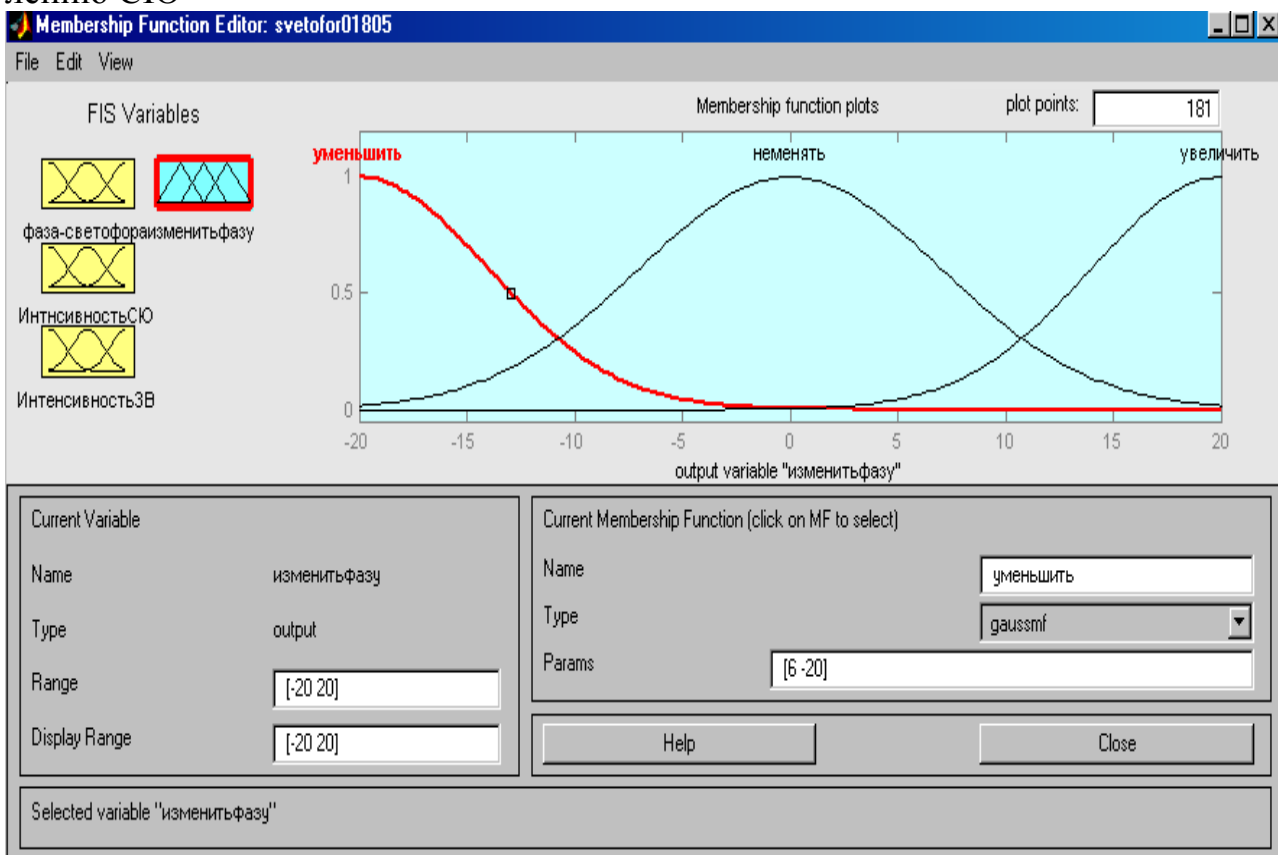


Рисунок 3.14 - Функция принадлежности выхода системы (изменение фазы зеленого сигнала светофора, сек.)

*Шаг 14.* По аналогии с шагом 10 зададим следующие наименования термов переменной у: уменьшить, не менять, Увеличить. В результате получим графическое окно, изображенное на рисунок 3.14.

*Шаг 15.* Перейдем в редактор базы знаний RuleEditor. Для этого выберем в меню Edit, команду Edit rules....

*Шаг 16.* На основе теоретических и экспериментальных данных о работе светофорных объектов сформулируем правила работы системы:

Если время зеленого света на улице СЮ=большое, & число машин на улице СЮ=малое, & число машин на улице ЗВ=большое, ТО время зеленого света = уменьшить.

Аналогично формулируем остальные правила. Число правил должно быть достаточным для эффективной работы системы в заданных условиях. Для нашей задачи это количество составляет 50 правил рисунок 3.15.

Для ввода правила необходимо выбрать в меню соответствующую комбинацию термов и нажать кнопку Add rule. На рис. 3.15. изображено окно редактора базы знаний после ввода всех девяти правил. Число, приведенное в скобках в конце каждого правила представляет собой весовым коэффициент соответствующего правила.

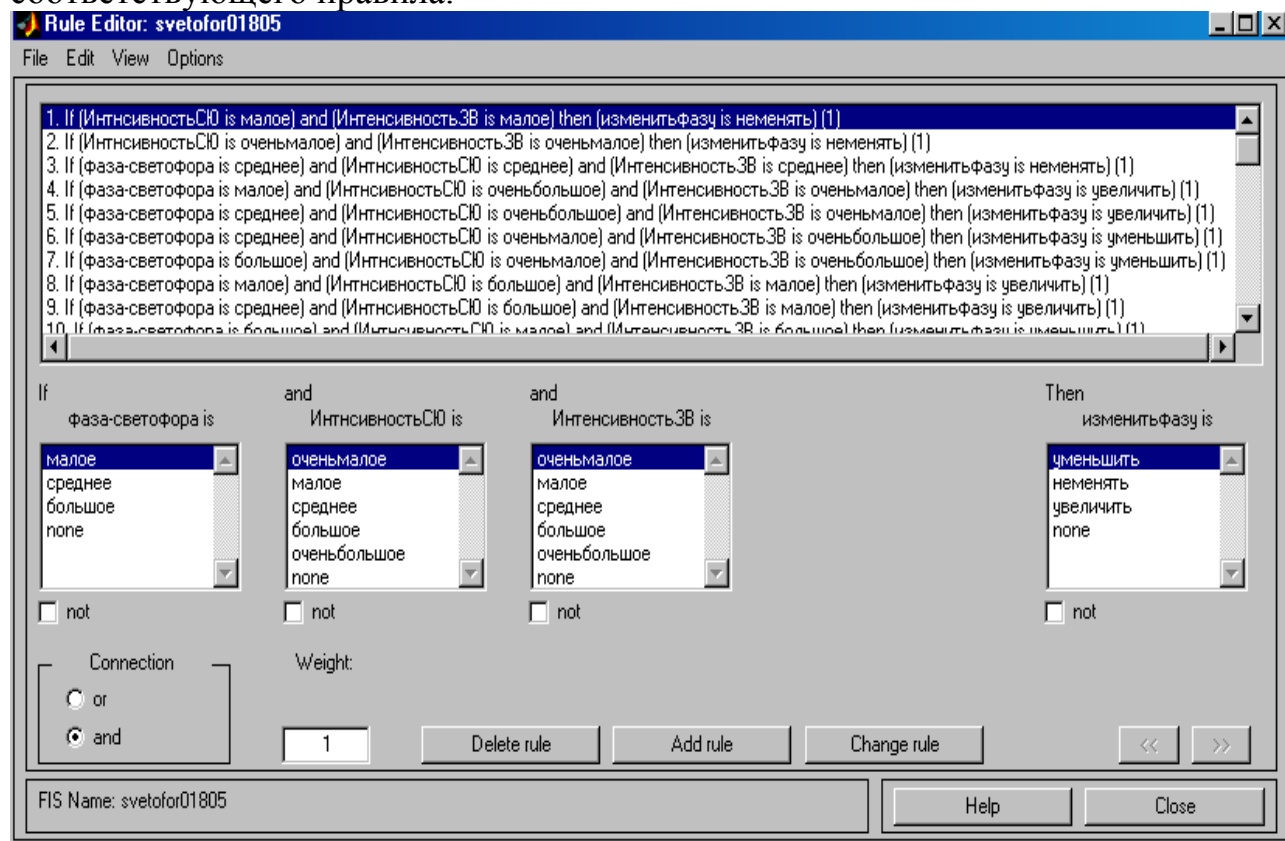


Рисунок 3.15 – Редактор базы логических правил

*Шаг 17.* Сохраним созданную систему. Для этого в меню File выбираем в подменю Export команду To disk.

На рисунке 3.16 приведено окно визуализации нечеткого логического вывода. Это окно активизируется командой View rules... меню View. В поле Input указываются значения входных переменных, для которых выполняется логический вывод.

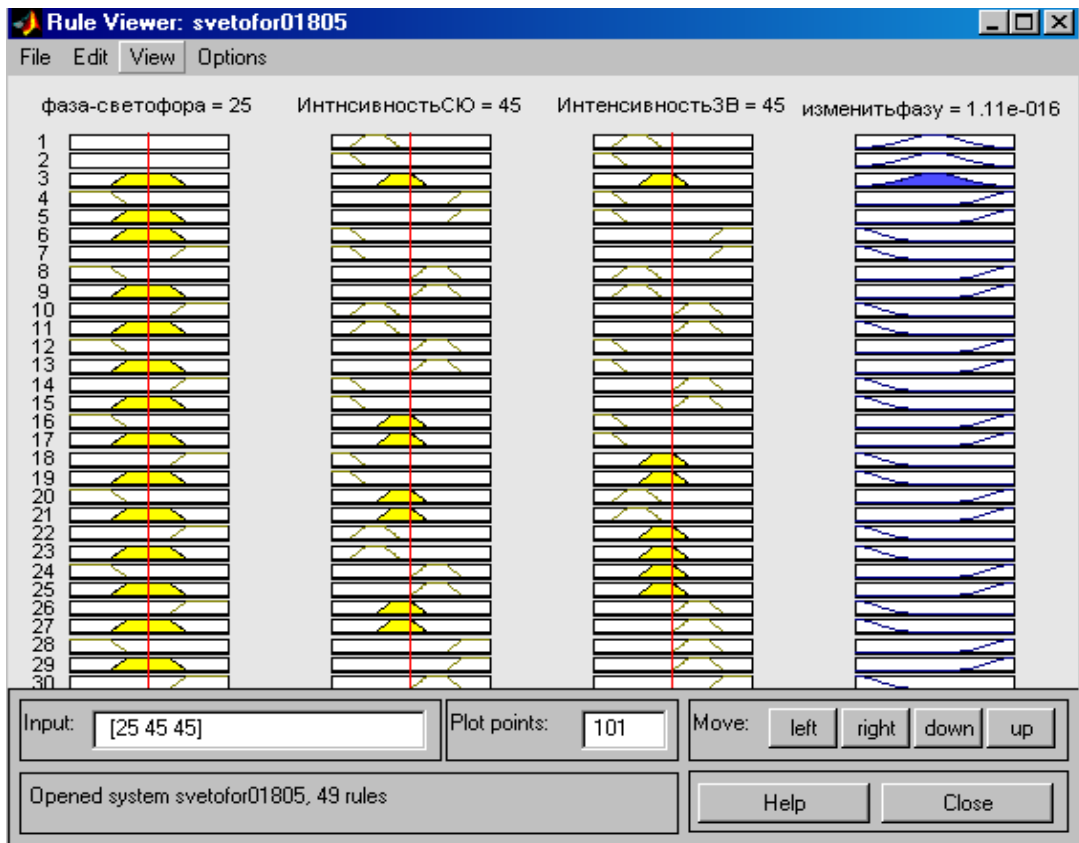


Рисунок 3.16 – Просмотр логических правил в браузере FIS редакторе

На рисунке 3.17 приведена поверхность “входы-выход”, соответствующая синтезированной нечеткой системе. Для вывода этого окна необходимо использовать команду View surface... меню View.

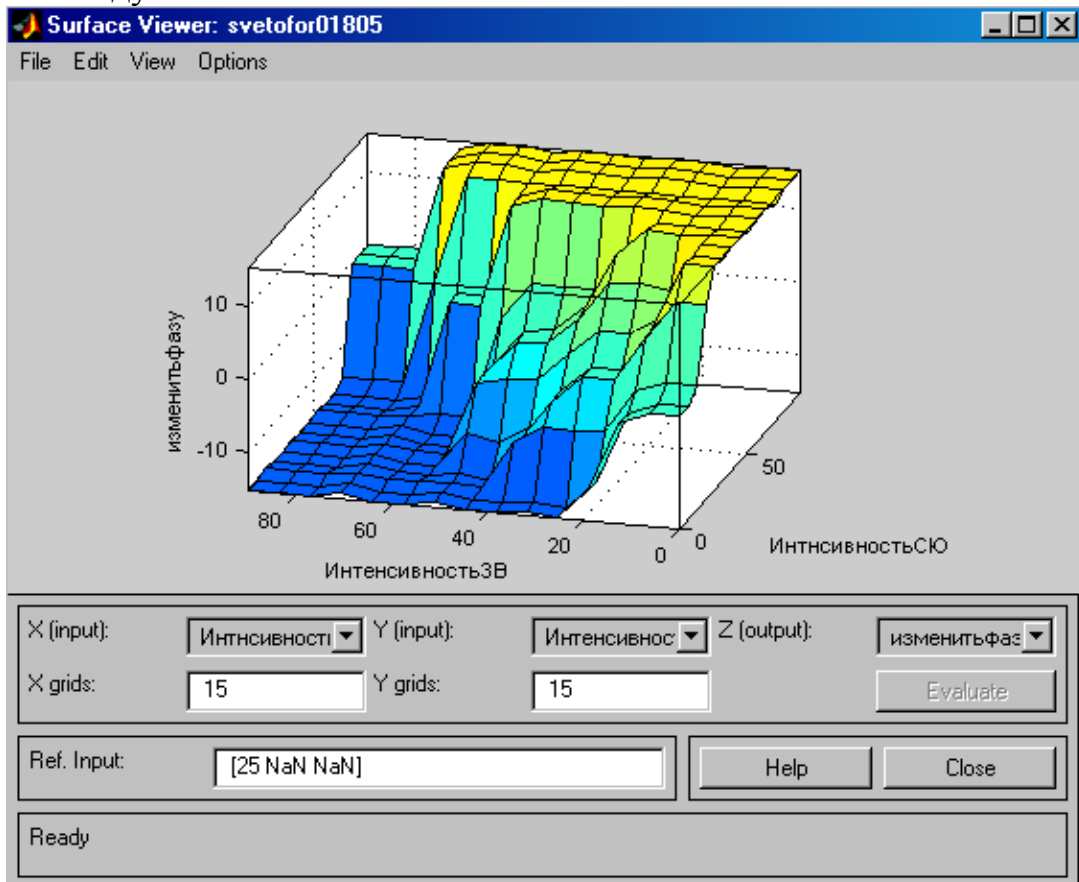


Рисунок 3.17 – Характеристика нечеткого регулятора по параметрам интенсивность движения по направлениям СЮ - ВЗ

## 4.2 Ознакомление с приложениями MATLABa построенных на пакете расширений Neural Networks Toolbox

Ознакомьтесь с демонстрационными примерами использования пакета Neural Networks Toolbox. В окне **Редактор(Editor)** выберите в списке **демонстрации**, затем **инструменты (instrument)**, далее **Neural Networks Toolbox**.

### 4.3 Построение нейросетевого аппроксиматора

1 Создайте новый m-файл, через меню файл(File). Введите код нейросети по аппроксимации функции:

```
%определение функций
time = 1:0.01:2.5;
X = sin(sin(time).*time*10);
P = con2seq(X);
T = con2seq(2*[0 X(1:(end-1))] + X);
%графическое представление исходных сигналов
plot(time,cat(2,P{:}),time,cat(2,T{:}),'--');
title('Input and Target Signals');
xlabel('Time');
legend({'Input','Target'});
%создание нейросети
net = newlin([-3 3],1,[0 1],0.1);
%обучение сети и вывод результата
[net,Y,E,Pf]=adapt(net,P,T);
plot(time,cat(2,Y{:}),'b', ...
      time,cat(2,T{:}),'r', ...
      time,cat(2,E{:}),'g',[1 2.5],[0 0],'k');
legend({'Output','Target','Error'});
%обучение сети и вывод результата
[net,Y,E,Pf]=adapt(net,P,T);
plot(time,cat(2,Y{:}),'b', ...
      time,cat(2,T{:}),'r', ...
      time,cat(2,E{:}),'g',[1 2.5],[0 0],'k');
legend({'Output','Target','Error'});
```

2 Сохраните файл-сценарий с именем на test\_NN1. Запустите на выполнение данный файл.

#### *Контрольные вопросы*

- 1 Структура интерфейса пакета Fuzzy Logic Toolbox.
- 2 Принципы построения систем моделирования и экспертных систем на основе нечеткой логики в пакете Fuzzy Logic Toolbox.
- 3 Этапы построения нечетких регуляторов.
- 4 Принципы построения базы правил нечетких систем.
- 5 Принципы построения систем на базе нейросетей.
- 6 Структура, виды нейросетей, сферы назначения.
- 7 Порядок построения нейросетевого аппроксиматора.

## Лабораторная работа №4

### Проектирование систем управления на основе гибридных нейро-нечетких систем в среде MATLAB

**Цель работы** – познакомиться с пакетами расширений Adaptive Neuro-Fuzzy Inference System ANFIS (адаптивная нейро-нечеткая система) среды MATLAB; получить навыки построения систем прогнозирования на основе нейро-нечетких систем.

#### *1 Материальное обеспечение лабораторной работы*

1.1 ЭВМ Pentium – 166.

1.2 Программа MATLAB.

#### *2 Содержание лабораторной работы*

2.1 Освоение интерфейса пакета расширений ANFIS, определение возможностей и ограничений.

2.2 Ознакомление с приложениями MATLAB, построенными в пакете расширений ANFIS.

2.3 Создание системы прогнозирования на основе ANFIS.

#### *3 Общие положения*

Основным компонентом рассмотренных ранее средств системы MATLAB в рамках пакета Fuzzy Logic Toolbox является нечеткая база (знаний) правил, которая занимает центральное место в процедурах нечеткого вывода. В то же время существуют целые классы прикладных задач, в которых выявление и построение правил невозможно или связано с серьезными трудностями концептуального характера. К таким задачам относятся задачи распознавания образов, экстраполяции и интерполяции функциональных зависимостей, классификации и прогнозирования, нелинейного и ситуационного управления, а также интеллектуального анализа данных (Data Mining).

Общей особенностью подобных задач является существование некоторой зависимости или отношения, связывающего входные и выходные переменные модели системы, представляемой в форме так называемого "черного ящика". При этом выявление и определение данной зависимости в явном теоретико-множественном или аналитическом виде не представляется возможным либо по причине недостатка информации о моделируемой проблемной области, либо сложности учета многообразия факторов, оказывающих влияние на характер данной взаимосвязи.

Для конструктивного решения подобных задач разработан специальный математический аппарат, получивший название нейронных сетей. Достоинством моделей, построенных на основе нейронных сетей, является возможность получения новой информации о проблемной области в форме некоторого прогноза. При этом построение и настройка нейронных сетей осуществляется посредством их обучения на основе имеющейся и доступной информации.

Недостатком нейронных сетей является представление знаний о проблемной области в специальном виде, которое может существенно отличаться



от возможной содержательной интерпретации существующих взаимосвязей и отношений.

Нечеткие нейронные сети или гибридные сети по замыслу их разработчиков призваны объединить в себе достоинства нейронных сетей и систем нечеткого вывода. С одной стороны, они позволяют разрабатывать и представлять модели систем в форме правил нечетких продукций, которые обладают наглядностью и простотой содержательной интерпретации. С другой стороны, для построения правил нечетких продукций используются методы нейронных сетей, что является более удобным и менее трудоемким процессом для системных аналитиков. В последнее время аппарат гибридных сетей повсеместно признается специалистами как один из наиболее перспективных для решения слабо или плохо структурированных задач прикладного системного анализа.

Система MATLAB имеет в своем составе ANFIS-редактор. ANFIS является аббревиатурой Adaptive Neuro-Fuzzy Inference System – (адаптивная нейро-нечеткая система). ANFIS-редактор позволяет автоматически синтезировать из экспериментальных данных нейро-нечеткие сети. Нейро-нечеткую сеть можно рассматривать как одну из разновидностей систем нечеткого логического вывода типа Сугэно. При этом функции принадлежности синтезированных систем настроены (обучены) так, чтобы минимизировать отклонения между результатами нечеткого моделирования и экспериментальными данными. Применяя эту систему можно эффективно разрабатывать системы прогнозирования.

Внешний вид ANFIS-редактора представлен на рисунок 4.1.

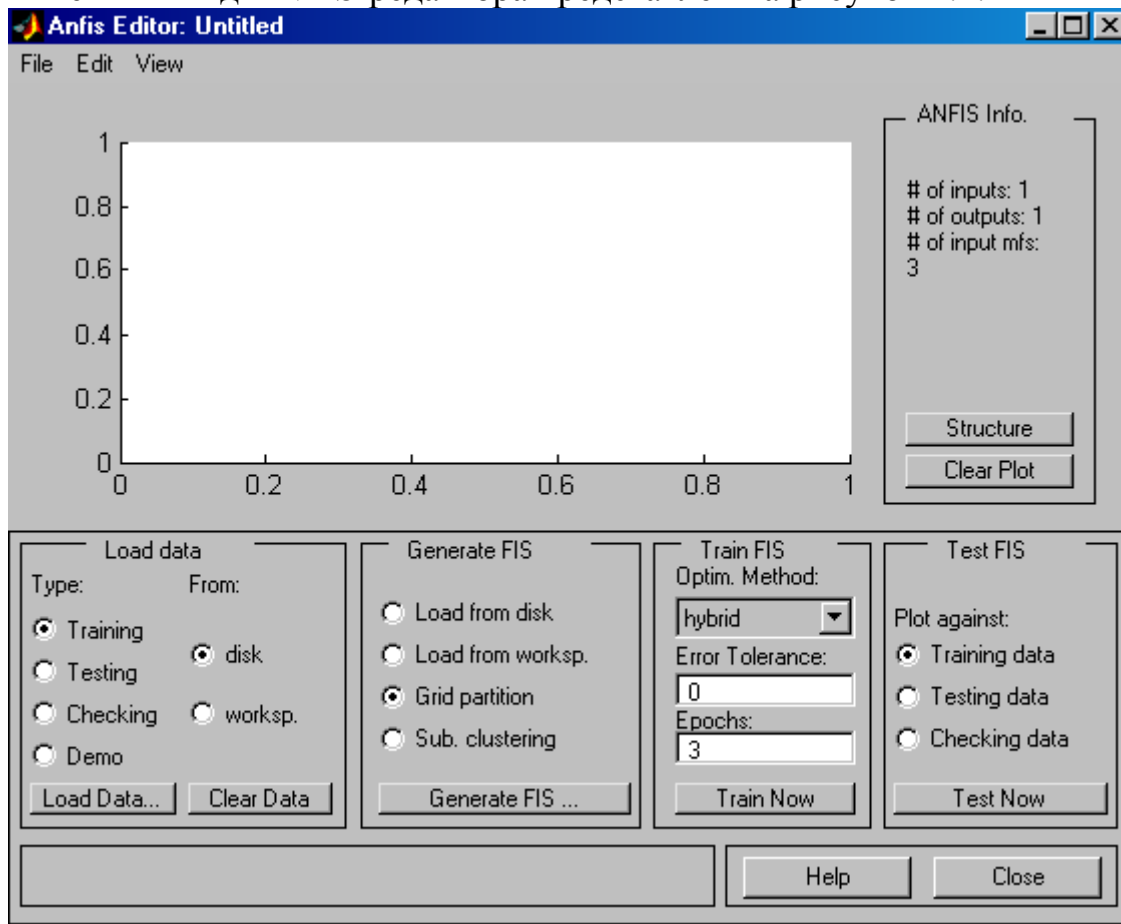


Рисунок 4.1 - ANFIS-редактор в среде MATLAB

## 4 Выполнение лабораторной работы

### 4.1 Освоение интерфейса ANFIS-редактора.

1 Запустить MATLAB.

2 Загрузить ANFIS-редактор с помощью команды **anfisedit**.

В результате выполнения этой команды появится графическое окно, изображенное на рисунке 4.1.

3 Изучить меню и области интерфейса редактора ANFIS.

### 4.2 Ознакомление с демонстрационными примерами MATLAB, построенными в пакете расширений Fuzzy Logic

1. Ознакомьтесь с демонстрационными примерами использования пакета **Fuzzy Logic**. В окне **Редактор (Editor)** выберите в списке **демонстрации**, затем **инструменты (instrument)**, далее **Fuzzy Logic**.

### 4.3 Проектирование с системы прогнозирования

1. Загрузить ANFIS-редактор с помощью команды **anfisedit**.

Проектирование систем прогнозирования производится в несколько этапов:

- загрузка данных с диска (обучающая выборка);
- генерирование структуры нечеткой системы (FIS);
- обучение нейросети системы;
- тестирование и дообучение системы прогнозирования.

2 Загрузить обучающую выборку через область **загрузка данных (Load data)**, файл **anfis1.dat**.

3 Сохранить созданную систему. Для этого в меню **File** выбираем в подменю **Export** команду **To disk**.

4 Произвести генерирование нечеткой (FIS) системы через область **генерация нечеткой структуры (Generate FIS)**. Необходимо выбрать число термов входных переменных, их вид, а также тип выходной переменной.

5 Произвести обучение нейросети гибридной системы, при этом нужно выбрать метод обучения, ошибку обучения и количество эпох при обучении. Первые два параметра подбираются по результатам обучения сети, а количество эпох должно быть достаточными для обучения сети с заданными параметрами. Следует начинать с 50 эпох. В дальнейшем увеличивают количество эпох до величины, при которой ошибка обучения не изменяется.

6 Просмотреть построенную структуру системы, используя область **свойств системы (ANFIS info)**.

7. Изучить правила нечеткой системы прогнозирования и кривую прогноза через меню (**edit** и **view**).

8. Произвести анализ полученной системы и сделать выводы.

### *Контрольные вопросы*

- 1 Гибридные нейро-нечеткие системы, назначение, сферы применения.
- 2 Структура интерфейса Adaptive Neuro-Fuzzy Inference System, Fuzzy Logic Toolbox.
- 3 Принципы построения систем прогнозирования в пакете Adaptive Neuro-Fuzzy Inference System, Fuzzy Logic Toolbox.
- 4 Этапы построения гибридных систем и их тестирование.

### **Список литературы**

1. В.Дьяконов, И.Абраменкова, В.Круглов. MATLAB с пакетами расширений под ред. Проф. В.П. Дьяконова. – М.: Нолидж. – 2001. – 880с.
2. Дьяконов В. MATHCAD 8/2000: Специальный справочник – СПб.: Питер, 2001. – 592 с.
3. Дьяконов В., Круглов В. Математические пакеты расширения MATLAB: Специальный справочник. – СПб.: Питер, 2001. – 480 с.
4. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ – Петербург, 2003. – 736 с.
5. Макаров Е.Г. Инженерные расчеты в MathCAD: Учебный курс. – СПб.: Питер, 2003. – 448 с.
6. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А.Поспелова. - М.: Наука, 1986. -312 с.
7. Потемкин В.Г.Вычисления в среде MATLAB.– М.: Диалог - МИФИ, 2004. – 720 с.
8. Трахтенгерц Э.А. Компьютерная поддержка принятия решений. -М.: СИНТЕГ, 1998. -376 с.
9. [www.mathworks.com](http://www.mathworks.com) [Электронный ресурс]: Фирма The Mathworks, Inc. (USA) – Режим доступа: [http// www.mathworks.com](http://www.mathworks.com), свободный. – Загл. с экрана. – Яз. англ.
10. [www.matlab.ru](http://www.matlab.ru) [Электронный ресурс]: Консультационный Центр MATLAB – Режим доступа: [http//www.matlab.ru](http://www.matlab.ru), свободный. – Загл. с экрана. – Яз. русский, англ.
11. [www.mathsoft.com](http://www.mathsoft.com) [Электронный ресурс]: Фирма Mathsoft Engineering & Education, Inc. (USA) – Режим доступа: <http://support.mathsoft.com>, свободный. – Загл. с экрана. – Яз. англ.
12. [www.basegroup.ru](http://www.basegroup.ru). [Электронный ресурс]: Фирма BASEGROUP Labs (Россия) – Режим доступа: <http://www.basegroup.ru>, свободный. – Загл. с экрана. – Яз. русский, англ.

**Ярослав Анатольевич Борщенко**

**АВТОМАТИЗАЦИИ ИНЖЕНЕРНЫХ РАСЧЕТОВ И  
ПРОЕКТИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
к выполнению лабораторных работ для студентов  
специальности 240400 – " Организация и безопасность движения"  
Часть вторая

Компьютерный набор: Я.А. Борщенко

Редактор: Т.В. Тимофеева

---

Подписано к печати		Бумага тип. № 1
Формат 60x84 1/16	Усл.п.л. 2,25	Уч. изд. л. 2,25
Заказ	Тираж 100	Цена свободная

---

Издательство Курганского государственного университета.

640669 г. Курган, ул. Гоголя, 25.

Курганский государственный университет, ризограф.