

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Емельянов Сергей Геннадьевич
Должность: ректор
Дата подписания: 16.12.2021 20:54:46
Уникальный программный ключ:
9ba7d3e34c012eba476ffd2d064c2181935be730da2374d16f5c0e538b6c6

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Доктионова
« _____ » _____ 2018 г.



СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА
ИНФОРМАЦИИ (ТЕХНИЧЕСКИЕ И МЕДИЦИНСКИЕ
СИСТЕМЫ)

Методические рекомендации по организации и выполнению
самостоятельной работы для аспирантов направления подготовки
09.06.01 и 12.06.01

Курск 2018

УДК 004.93:61

Составители: О.В. Шаталова., С.А.Филист

Рецензент

Доктор технических наук, профессор А.Ф. Рыбочкин

Системный анализ, управление и обработка информации (технические и медицинские системы): методические рекомендации по организации и выполнению самостоятельной работы / Юго-Зап. гос. ун-т; сост.: О.В.Шаталова. - Курск, 2018. - 37 с.

Методические указания соответствуют требованиям программ дисциплин «Системный анализ, управление и обработка информации (технические и медицинские системы)».

Предназначены для обучающихся по программам высшего образования по направлениям 12.04.04 «Биотехнические системы и технологии», 12.06.01 «Фотоника, приборостроение, оптические и биотехнические системы и технологии (Приборы, системы и изделия медицинского назначения)» и 09.06.01 «Информатика и вычислительная техника (Системный анализ, управление и обработка информации (технические и медицинские системы))»

Текст печатается в авторской редакции

Подписано в печать 1.03.18 . Формат 60x84 1/16.

Усл.печ.л. 1,7. Уч.-изд.л. 1,6 Тираж 100 экз. Заказ: 1440. Бесплатно. Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Самостоятельная работа №1

«Методы и принципы системного исследования»

1.1 Краткие теоретические сведения

В широком смысле слова под системным исследованием предметов и явлений окружающего нас мира понимают такой метод, при котором они рассматриваются как части и элементы определенного целостного образования. Эти части или элементы, взаимодействуя друг с другом, определяют новые, целостные свойства системы, которые отсутствуют у отдельных ее элементов. Главное, что определяет систему, — это взаимосвязь и взаимодействие частей в рамках целого. Для системного исследования характерно именно целостное рассмотрение, установление взаимодействия составных частей или элементов совокупности, несводимость свойств целого к свойствам частей.

Учение о системах возникло в середине XIX в., но приобрело особенно важное значение в XX в. Его иначе называют еще системным подходом к изучаемым объектам, или системным анализом.

Система— это такая совокупность элементов, или частей, в которой существует их взаимное влияние и взаимное качественное преобразование. С этой точки зрения современное естествознание приблизилось к тому, чтобы стать настоящей системой, потому что все его части ныне взаимосвязаны, в нем нет уже ни одной естественной науки в рафинированно чистом виде.

Под системой понимают совокупность компонентов и устойчивых, повторяющихся связей между ними. Процесс системного рассмотрения объектов широко применяется в самых различных областях общественных, естественных и технических наук, в практике социального планирования и управления в обществе, при решении комплексных социальных проблем, при подготовке и реализации разнообразных целевых программ.

Основными свойствами систем являются следующие:

- всеобщий характер, поскольку в качестве системы могут рассматриваться все без исключения предметы и явления окружающего мира;
- невещественность;

- внутренняя противоречивость (конкретность и абстрактность, целостность и дискретность, непрерывность и прерывность);
- способность к взаимодействию;
- упорядоченность и целостность;
- устойчивость и взаимообусловленность.

Способность процессов и явлений мира образовывать системы, наличие систем, системного строения материальной действительности и форм ее познания получила название системности. Понятие системности отражает одну из характерных признаков действительности: способность вступать в такого рода взаимодействия, в результате которых образуются новые качества, не присущие исходным объектам взаимодействия.

Система — это множество объектов вместе с отношениями между объектами, между их свойствами, которые взаимодействуют между собой таким образом, что обуславливают возникновение новых, целостных, системных свойств. Для лучшего понимания природы систем рассмотрим их строение, структуру и классификацию.

Строение системы характеризуется теми компонентами, из которых она образована. Такими компонентами являются: подсистемы, части или элементы системы. Подсистемы составляют наибольшие части системы, которые обладают определенной автономностью, но в то же время они подчинены и управляются системой. Элементами называют наименьшие единицы системы.

Структурой системы называют совокупность тех специфических взаимосвязей и взаимодействий, благодаря которым возникают новые целостные свойства, присущие только системе и отсутствующие у отдельных ее компонентов.

Классификация систем может производиться по самым разным основаниям деления. Прежде всего все системы можно разделить на материальные и идеальные. К материальным системам относится подавляющее большинство систем неорганического, органического и социального характера. Материальными системами называют их потому, что их содержание и свойства не зависят от познающего субъекта. Содержание и свойства идеальных систем зависят от субъекта. Наиболее простой классификацией систем является деление их на статические и динамические. Среди динамических систем обычно выделяют детерминистические и вероятностные

системы. Такая классификация основывается на характере предсказания динамики поведения систем. По характеру взаимодействия с окружающей средой различают системы открытые и закрытые. Обычно выделяют те системы, с которыми данная система взаимодействует непосредственно и которые называют окружением или внешней средой системы. Все реальные системы в природе и обществе являются, как мы уже знаем, открытыми и, следовательно, взаимодействующими с окружением путем обмена веществом, энергией и информацией. Системы классифицируют также на простые и сложные. Простыми системами называют системы с небольшим числом переменных, взаимоотношения между которыми поддаются математической обработке и выведению универсальных законов. Сложная система состоит из большого числа переменных и большого количества связей между ними. Сложная система имеет свойства, которых нет у ее частей и которые являются следствием эффекта целостности системы.

1.2 Цель работы

убедившись во всеобщности кибернетических законов и принципов, научиться находить их проявления в функционировании конкретных систем; понять различия между законами управления и принципами исследования в системном подходе.

1.3 Задание

Исследовать заданные системы с помощью применения принципа «черного ящика», а именно – определить по 6-7 входов и выходов каждой из систем и выделить по 3 наиболее существенных.

Контрольные вопросы

1. «Система» как основное понятие системного подхода.
2. Классификация систем.
3. Этапы системного анализа.
4. В чем заключается закон необходимого разнообразия ?
5. В чем различия между обратной связью и принципом этой связи ?

6. В каких ситуациях используется принцип «черного ящика»?

Самостоятельная работа №2 «Модель. Моделирование систем»

2.1 Краткие теоретические сведения

При использовании метода моделирования свойства и поведение объекта изучают путем применения вспомогательной системы – модели, находящейся в определенном объективном соответствии с исследуемым объектом.

Под объектом исследования понимается либо некоторая система, элементы которой в процессе достижения конечной цели реализуют один или несколько процессов, либо некоторый процесс, реализуемый элементами одной или нескольких систем. В связи с этим в дальнейшем тексте термины «модель объекта», «модель системы», «модель процесса» следует воспринимать как эквивалентные.

Представления о тех или иных свойствах объектов, их взаимосвязях формируются исследователем в виде описания этих объектов на обычном языке, в виде рисунков, графиков, формул или реализуются в виде макетов и других устройств. Подобные способы описания обобщаются в едином понятии – модель, а построение и изучение моделей называется моделированием.

Заслуживает предпочтения следующее определение: **модель** – объект любой природы, который создается исследователем с целью получения новых знаний об объекте-оригинале и отражает только существенные (с точки зрения разработчика) свойства оригинала.

Модель считается адекватной объекту-оригиналу, если она с достаточной степенью приближения на уровне понимания моделируемого процесса исследователем отражает закономерности процесса функционирования реальной системы во внешней среде.

Модели позволяют вынести упрощенное представление о системе и получить некоторые результаты намного проще, чем при изучении реального объекта. Более того, гипотетически модели объекта могут быть исследованы и изучены перед тем, как объект будет создан.

В практике исследования производственно-экономических объектов модели могут применяться для самых разных целей, что вызывает использование моделей различных классов. Построение одной-единственной математической модели для сложной

производственной системы практически не представляется возможным без разработки вспомогательных моделей. Поэтому, как правило, при создании конечной математической модели исследуемого объекта строят частные вспомогательные модели, отражающие ту или иную информацию об объекте, имеющуюся у разработчика на данном этапе построения модели.

В основе моделирования лежит теория подобия, которая утверждает, что абсолютное подобие может иметь место лишь при замене одного объекта другим точно таким же. При моделировании абсолютное подобие не имеет места и стремятся к тому, чтобы модель достаточно хорошо отображала исследуемую сторону функционирования объекта.

Классификационные признаки. В качестве одного из первых признаков классификации видов моделирования можно выбрать степень полноты модели и разделить модели в соответствии с этим признаком на полные, неполные и приближенные. В основе полного моделирования лежит полное подобие, которое проявляется как во времени, так и в пространстве. Для неполного моделирования характерно неполное подобие модели изучаемому объекту. В основе приближенного моделирования лежит приближенное подобие, при котором некоторые стороны функционирования реального объекта не моделируются совсем.

2.2 Цель

На основе применения системных принципов научиться моделировать поведение и функционирование реальных объектов.

2.3 Задание

Для заданных систем построить модели состава и модели структуры. В модели структуры объяснить основные связи между элементами (или подсистемами) и определить цель, достигнутую в ходе структурного моделирования.

Контрольные вопросы

1. Понятие модели ? Типы моделей (основные).
2. Является копия объекта моделью ? Почему ?
3. Основные компоненты моделей состава и структуры

системы.

4. Соответствие объект-модель : различие и сходство
5. Моделирование систем
6. Характеристика моделей

Самостоятельная работа №3

Проектирование системы нечеткого вывода

3.1 Цель работы

Освоить проектирование нечетких систем в пакете Fuzzy Logic Toolbox и Symbolic Math Toolbox вычислительной среды MATLAB.

3.2 Задание

Получить задание по одному из вариантов таблицы 3.1

Подготовить нечеткие правила в соответствии с графическим представлением функций задания.

Трехмерное изображение выбранной зависимости построить, воспользовавшись программными средствами MATLAB.

При этом можно руководствоваться программой нижеприведенного примера, которая моделирует зависимость $y = x^2 \cdot \sin(x - 1)$.

3.3 Порядок выполнения работы

Пример

```
% Построение графика функции  $y = x_1^2 \cdot \sin(x_2 - 1)$   
% в области  $x_1 = [-6, 4]$ ,  $x_2 = [-4.4, 1.7]$ .  
n=15; % количество точек дискретизации  $x_1 = \text{linspace}(-6, 4, n)$ ;  
 $x_2 = \text{linspace}(-4.4, 1.7, n)$ ;  $y = \text{zeros}(n, n)$ ; for j=1:n  
y(j,:)= $x_1.^2 \cdot \sin(x_2(j)-1)$ ; end surf(x1, x2, y)  
xlabel('x_1'); ylabel('x_2'); zlabel('y'); title('Искомая зависимость')  
Сформируем нечеткие правила, соответствующие зависимости  
выходной переменной от входных по полученной поверхности  
рисунок 3.1.
```

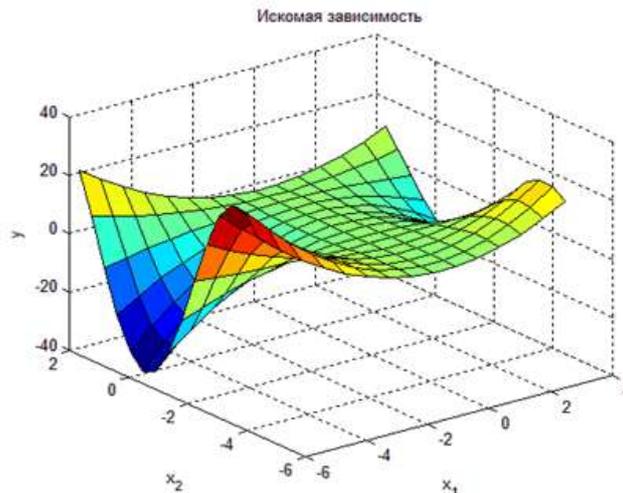


Рисунок 3.1- Графическое представление функции $y=x_1^2\sin(x_2-1)$

Можно воспользоваться и возможностями пакета Symbolic Math Toolbox, введя, например, следующие команды:

```
syms x y % создаем символьные переменные
z=x^2*sin(y-1); ezsurf(z,[-2,2])
```

Для лингвистической оценки входных переменных x_1 и x_2 выберем по три термина - низкий, средний, высокий; для выходной y пять термов - низкий, ниже среднего, средний, выше среднего, высокий. Получим, например, следующие правила:

- ЕСЛИ x_1 = низкий И x_2 = низкий, ТО y = высокий;
- ЕСЛИ x_1 = низкий И x_2 = средний, ТО y = низкий;
- ЕСЛИ x_1 = низкий И x_2 = высокий, ТО y = высокий;
- ЕСЛИ x_1 = средний, ТО y = средний;
- ЕСЛИ x_1 = высокий И x_2 = низкий, ТО y = выше среднего;
- ЕСЛИ x_1 = высокий И x_2 = средний, ТО y = ниже среднего;
- ЕСЛИ x_1 = высокий И x_2 = высокий, ТО y = выше среднего.

Спроектировать нечеткую систему, выполнив следующую последовательность шагов.

Открыть FIS-редактор, напечатав слово `fuzzy` в командной строке.

В появившемся графическом окне FIS Editor добавим вторую входную переменную. Для этого в меню Edit выбираем команду Add input.

Переименуем первую входную переменную. Для этого сделаем щелчок левой кнопкой мыши на блоке Input 1, введем новое обозначение x_1 в поле редактирования имени текущей переменной и нажмем <Enter>.

Переименуем вторую входную переменную, введя x_2 на блоке Input 2.

Переименуем выходную переменную. Для этого щелкнем мышкой на блоке Output 1. Введем новое обозначение y в поле редактирования имени текущей переменной; нажмем <Enter>.

Зададим имя системы. Для этого в меню File выберем в подменю Export команду To File и введем имя файла, например, Lab_2.

Перейдем в редактор функций принадлежности. Для этого сделаем двойной щелчок левой кнопкой мыши на блоке x_1 и зададим диапазон изменения переменной x_1 , напечатав -6 4 в поле Range (рис. 2).

Зададим функции принадлежности переменной x_1 . Для лингвистической оценки этой переменной будем использовать три термина с треугольными функциями принадлежности. Эти функции установлены по умолчанию, поэтому переходим к следующему шагу.

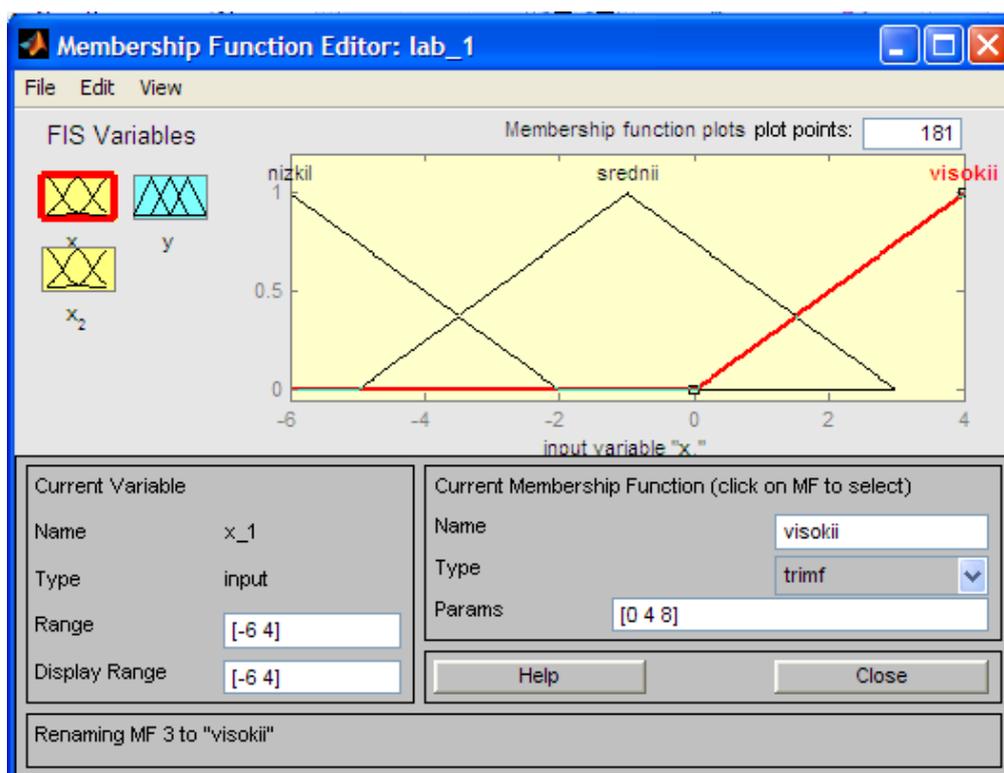


Рисунок 3.2 - Функции принадлежности переменной x_1

Зададим наименования термов переменной x_1 . Для этого щелкнем мышкой по графику первой функции принадлежности рисунок 3.2. График активной функции принадлежности выделяется красной жирной линией. Затем введем наименование термина Низкий в поле Name и нажмем <Enter>. Щелкнем мышкой по графику второй

функции принадлежности, введем наименование термина Средний в поле Name и нажмем <Enter>. Щелкнем мышкой по графику третьей функции принадлежности, введем наименование термина Высокий в поле Name и нажмем <Enter>.

Зададим функции принадлежности переменной x_2 . Для этого активизируем переменную x_2 щелчком мышкой по блоку x_2 . Зададим диапазон изменения переменной x_2 . Для этого напечатаем -4.4 1.7 в поле Range и нажмем <Enter>. Для лингвистической оценки этой переменной будем использовать, как и ранее, три термина с треугольными функциями принадлежности. Они установлены по умолчанию, поэтому переходим к следующему шагу.

По аналогии с шагом 9 зададим следующие наименования термов переменной x_2 : Низкий, Средний, Высокий.

Зададим функции принадлежности переменной y . Для лингвистической оценки этой переменной будем использовать пять термов с гауссовыми функциями принадлежности. Для этого щелчком мыши по блоку y активизируем переменную y . Зададим диапазон изменения переменной y . Для этого напечатаем -50 50 в поле Range (рис. 1.3) и нажмем <Enter>. Затем в меню Edit выберем команду Remove All MFs для удаления установленных по умолчанию функций принадлежности. После этого в меню Edit выберем команду Add MF. В появившемся диалоговом окне выберем тип функции принадлежности gaussmf в поле MF type и пять термов в поле Number MFs. После ввода функций принадлежности редактор активизирует первую входную переменную, поэтому для продолжения работы щелкнем мышкой по пиктограмме y .

По аналогии с шагом 9 зададим следующие наименования термов переменной y : Низкий, Ниже среднего, Средний, Выше среднего, Высокий. В результате получим графическое окно, изображенное на рисунке 3.3.

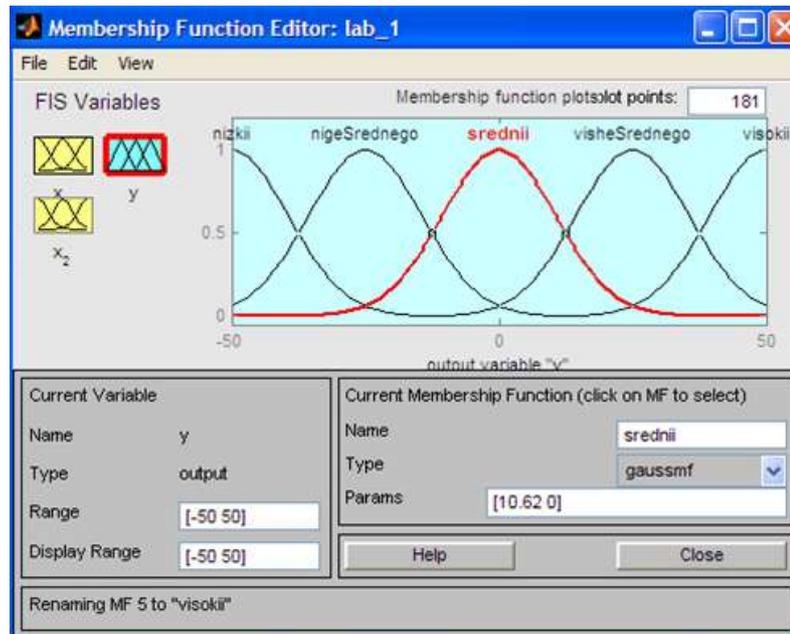


Рисунок 3.3.- Функции принадлежности переменной у

Перейдем в редактор базы знаний Rule Editor. Для этого в меню Edit выберем команду Rules... Для ввода правила выбираем в меню соответствующую комбинацию термов и нажимаем кнопку Add rule. На рис. 1.4 изображено окно редактора базы знаний после ввода всех семи правил. В конце правил в скобках указаны весовые коэффициенты.

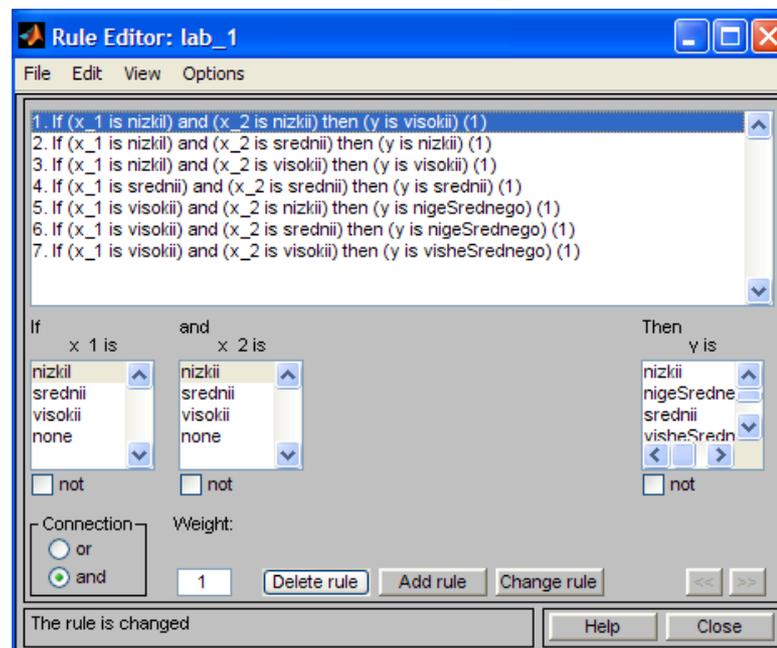


Рисунок 3.4.- Нечеткая база знаний Mamdani

На рисунке 3.5 приведено окно визуализации нечеткого вывода. Окно активизируется командой Rules меню View. В поле Input указываются значения входных переменных, для которых выполняется нечеткий логический вывод.

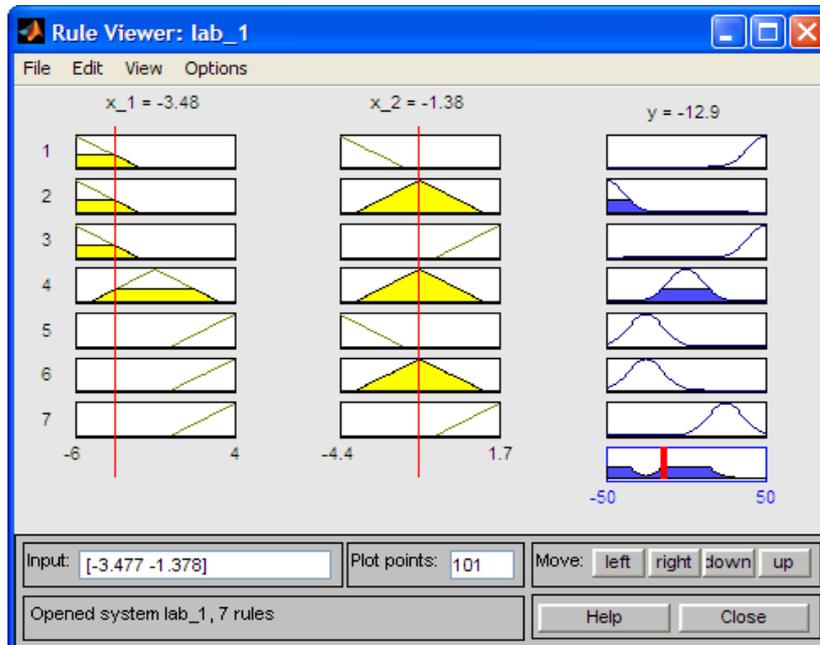


Рисунок 3.5. Визуализация нечеткого вывода Mamdani

На рисунке 3.6 приведена поверхность «входы - выход», соответствующая синтезированной нечеткой системе. Окно выводится по команде Surface меню View.

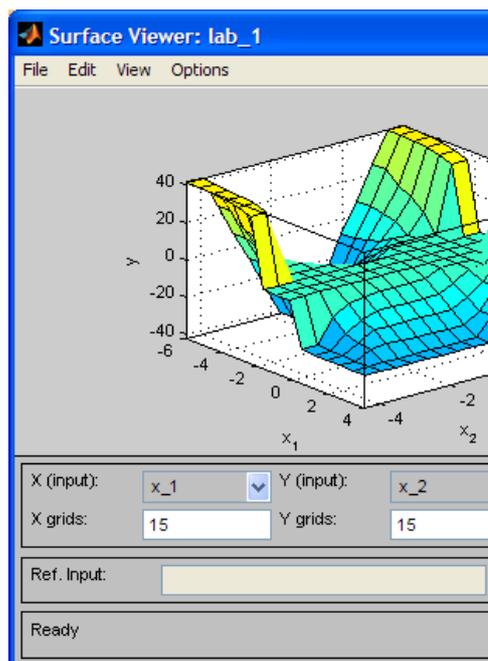


Рисунок 3.6- Поверхность «входы - выход» для базы знаний в Surface Viewer

Сохраним созданную систему. Для этого в меню File выберем в подменю Export команду To File.

При сравнении поверхностей на рисунке 1 и 6 можно сделать выводы относительно качества описания нечеткими правилами моделируемой нелинейной зависимости.

Варианты заданий

Таблица 3.1- Варианты заданий

№	Вид зависимости	Диапазон изм.
1	2	3
1	$Z = x.^2 - y.^2$	x,y [-1 1]
2	$Z = x.^3 + y.^2$	x,y [-1 1]
3	$Z = \exp(-x.^2 - y.^2)$	x,y [-1 1]
4	$Z = \exp(-x.^2 + y.^2)$	x,y [-2 2]
5	$Z = x * y * \sin(x^2 + y^2)$	x,y [-2 2]
6	$Z = x.^2 * \sin(y - 1)$	x,y [-2 2]
7	$Z = y.^2 * \sin(x)$	x,y [-2 2]
8	$Z = y.^2 * \cos(x)$	x,y [-2 2]
9	$Z = y.^2 * \cos(x)^2$	x,y [-1, 1]
10	$Z = 4 * \cos(x) / y$	x,y [0.5 3.14]
11	$Z = (x - y) / (x + y)$	x,y [1 10]
12	$Z = \exp(-x.^2) + \exp(-y.^2)$	x,y [-1 1]
13	$Z = x.^2 + y.^2$	x,y [-1, 10]
14	$Z = 5 * x.^2 * \cos(y)$	x,y [-1, 1]
15	$Z = -x * y + y.^2$	x,y [0, 5]
16	$Z = 5 * x.^2 * \sin(y)$	x,y [-1, 1]
17	$Z = y.^2 * \sin(x).^2$	x,y [-1, 1]
18	$Z = 2 * x.^2 - (y - 1).^2$	x,y [-1, 1]
19	$Z = y.^2 * \cos(x)$	x,y [-2, 2]
20	$Z = x.^3 + y.^2$	x,y [-1, 1]

Содержание отчета

Исходная функция варианта задания и ее графическое представление.

Лингвистические правила решений.

Описание последовательности действий при проектировании нечеткой системы.

Анализ результатов нечеткого вывода.

Оценка качества описания нечеткими правилами моделируемой нелинейной зависимости.

Контрольные вопросы

1. Типовые оптимизационные модели.
2. Дайте понятие моделирование с позиции системного анализа.
3. Основные методы моделирования систем
4. Нечеткие модели и нечеткие алгоритмы принятия решений.
5. Нечеткая логика.

Самостоятельная работа №4

Построение нечеткой аппроксимирующей системы в пакете FUZZY LOGIC TOOLBOX

4.1 Цель работы

Изучение основных функций пакета Fuzzy Logic Toolbox программной среды MATLAB, а также приобретение навыков построения нечеткой аппроксимирующей системы.

4.2 Порядок выполнения работы

Командой (функцией) Fuzzy из режима командной строки запускается основная интерфейсная программа пакета Fuzzy Logic – редактор нечеткой системы вывода (Fuzzy Inference System Editor, FIS Editor, FIS-редактор).

Вид открывающегося при этом окна приведен на рисунке 4.1. Главное меню редактора содержит позиции:

File – работа с файлами моделей (их создание, сохранение, считывание и печать);

Edit – операции редактирования (добавление и исключение входных и выходных переменных);

View – переход к дополнительному инструментарию.

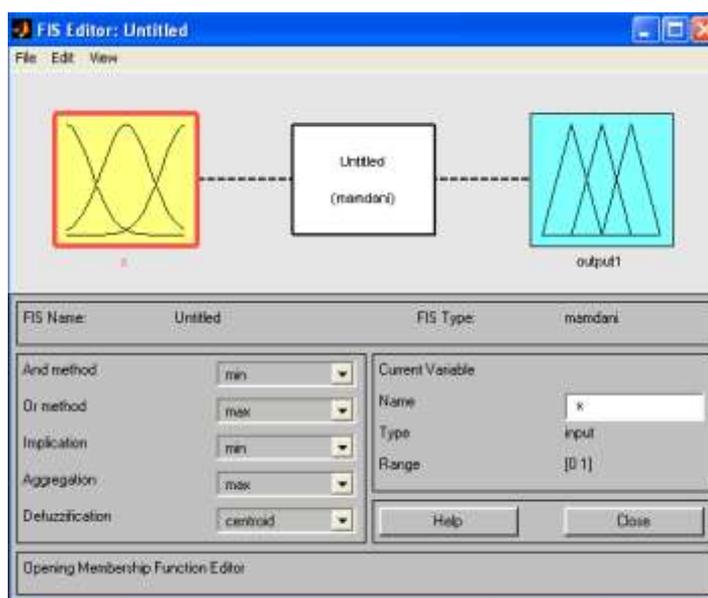


Рисунок 4.1- Вид окна Fis Editor

Попробуем сконструировать нечеткую систему, отображающую зависимость между переменными x и y , заданную с помощью таблицы 4.1 (легко видеть, что представленные в таблице данные отражают зависимость $y = x^2$).

x	-1	0.6	0	0.4	1
y	1	0.36	0	0.16	1

Требуемые действия отобразим следующими пунктами.

В позиции меню File выбираем опцию New Sugeno FIS (новая система типа Sugeno), при этом в блоке, отображаемом белым квадратом, в верхней части окна редактора появится надпись Untitled2 (Sugeno).

Щелкнем левой кнопкой мыши по блоку, озаглавленному input 1 (вход 1). Затем в правой части редактора в поле, озаглавленном Name (Имя), вместо input 1 введем обозначение нашего аргумента, т.е. x . Обратим внимание, что если теперь сделать где-нибудь (вне блоков редактора) однократный щелчок мыши, то имя отмеченного блока изменится на x ; то же достигается нажатием после ввода клавиши Enter.

Дважды щелкнем по этому блоку. Перед нами откроется окно редактора функций принадлежности – Membership Function Editor рисунок 4.2. Войдем в позицию меню Edit данного редактора и выберем в нем опцию Add MFs (Add Membership Functions – Добавить функций принадлежности). При этом появится диалоговое окно (рисунок 1.3), позволяющее задать тип (MF type) и количество (Number of MFs) функций принадлежности (в данном случае все относится к входному сигналу, т. е. к переменной x). Выберем гауссовы функции принадлежности (gaussmf), а их количество зададим равным пяти – по числу значений аргумента в таблице 4.1. Подтвердим ввод информации нажатием кнопки ОК, после чего произойдет возврат к окну редактора функций принадлежности.

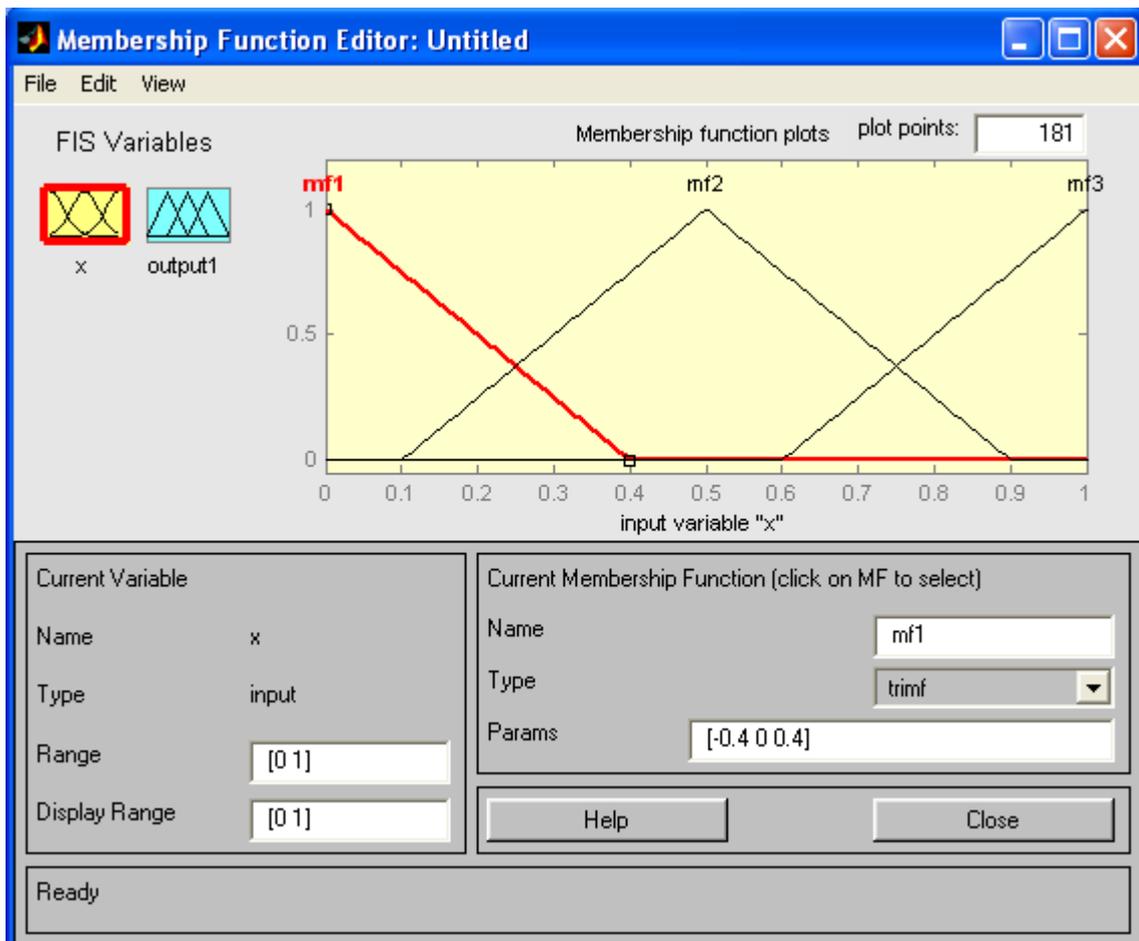


Рисунок 4.2-Окно редактора функций принадлежности

В поле Range (Диапазон) установим диапазон изменения x от -1 до $+1$, т.е. диапазон, соответствующий таблице 1. Щелкнем затем левой кнопкой мыши где-нибудь в поле редактора (или нажмем клавишу ввода Enter). Обратим внимание, что после этого произойдет соответствующее изменение диапазона в поле Display Range (Диапазон дисплея).

Обратимся к графикам заданных нами функций принадлежности, изображенным в верхней части окна редактора функций принадлежности. Заметим, что для успешного решения поставленной задачи необходимо чтобы ординаты максимумов этих функций совпадали с заданными значениями аргумента x . Для левой, центральной и правой функций такое условие выполнено, но две другие необходимо «подвинуть» вдоль оси абсцисс. «Передвижка» делается весьма просто: подводим курсор к нужной кривой и щелкаем левой кнопкой мыши. Кривая выбирается, окрашиваясь в красный цвет, после чего с помощью курсора ее и можно подвинуть в нужную сторону (более точную установку можно провести, изменяя числовые значения в поле Params (Параметры) – в данном случае

каждой функции принадлежности соответствуют два параметра, при этом первый определяет размах кривой, а второй – положение ее центра). Для выбранной кривой, кроме этого, в поле Name можно изменять имя (завершая ввод каждого имени нажатием клавиши Enter).

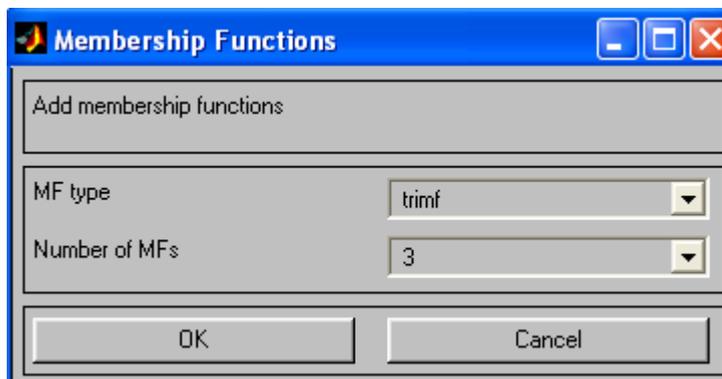


Рисунок 4.3- Диалоговое окно задания типа и количества функций принадлежности

Проделаем требуемые перемещения кривых и зададим всем пяти кривым новые имена, например:

- самой левой – bn,
- следующей – n,
- центральной – z,
- следующей за ней справа – p,
- самой правой – br.

Нажмем кнопку Close и выйдем из редактора функций принадлежности, возвратившись при этом в окно редактора нечеткой системы (FIS Editor).

Сделаем однократный щелчок левой кнопкой мыши по голубому квадрату (блоку), озаглавленному output 1 (выход 1). В окошке Name заменим имя output 1 на u (как в пункте 2).

Дважды щелкнем по отмеченному блоку и перейдем к программе – редактору функций принадлежности. В позиции меню Edit выберем опцию Add MFs. Появляющееся диалоговое окно вида рис. 3 позволяет задать теперь в качестве функций принадлежности только линейные (linear) или постоянные (constant) – в зависимости от того, какой алгоритм

Sugeno (1-го или 0-го порядка) мы выбираем. Если в вашем компьютере установлена версия, в которой нет данных функций принадлежности, то можно оставить по умолчанию – trimf. Это, конечно, повлияет на результат, поэтому можно поэкспериментировать, изменяя тип функций принадлежности.

В рассматриваемой задаче необходимо выбрать постоянные функции принадлежности с общим числом 4 (по числу различных значений y в таблице 4.2). Подтвердим введенные данные нажатием кнопки ОК, после чего произойдет возврат в окно редактора функций принадлежности.

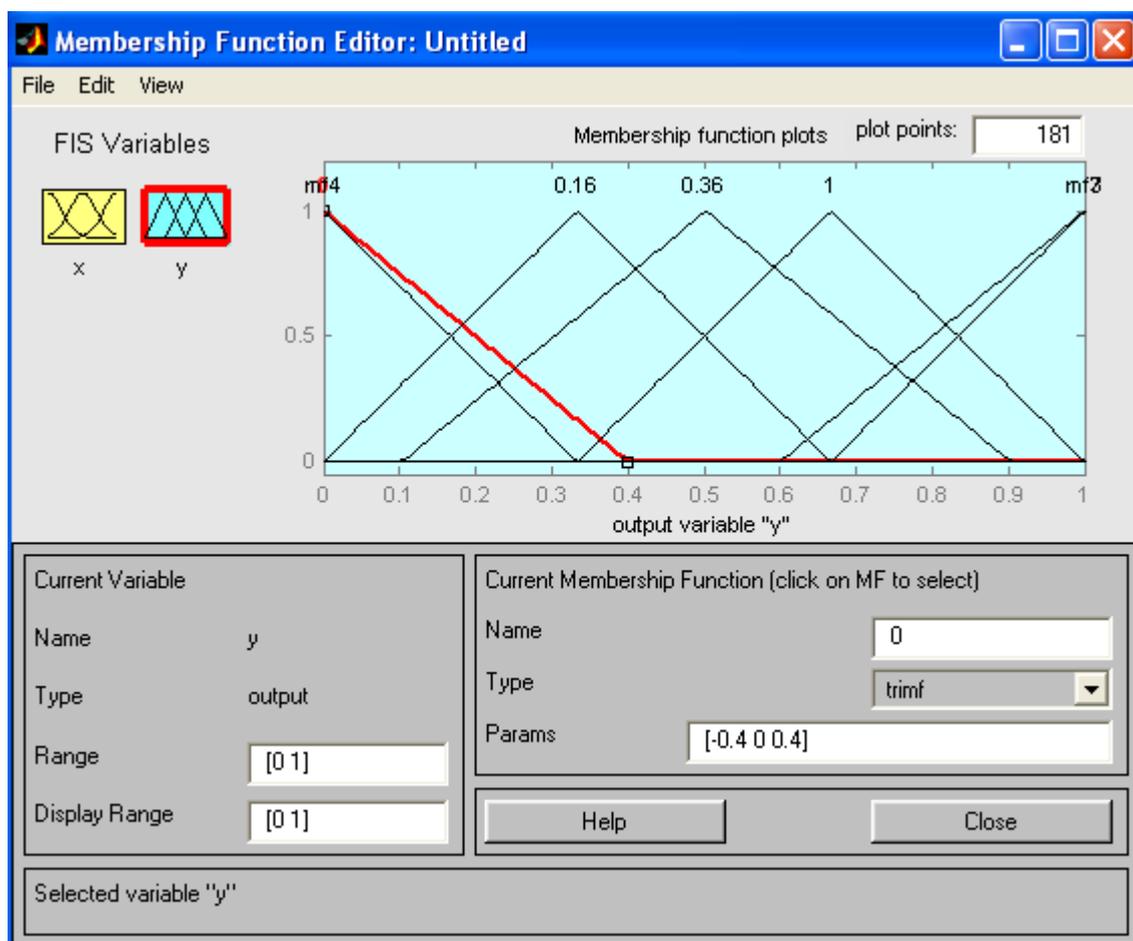


Рисунок 4.4-Параметры функций принадлежности переменной y

Обратим внимание, что здесь диапазон (Range) изменения, устанавливаемый по умолчанию – $[0, 1]$, менять не нужно. Изменим лишь имена функций принадлежности (их графики при использовании алгоритма Sugeno для выходных переменных не приводятся), например, задав их как соответствующие числовые значения y , т.е. 0, 0.16, 0.36, 1; одновременно эти же числовые значения введем в поле

Params рисунок 4.4. Затем закроем окно нажатием кнопки Close и вернемся в окно FIS-редактора.

Дважды щелкнем левой кнопкой мыши по среднему (белому) блоку, при этом раскроется окно еще одной программы – редактора правил (Rule Editor). Введем соответствующие правила. При вводе каждого правила необходимо обозначить соответствие между каждой функцией принадлежности аргумента x и числовым значением y . Кривая, обозначенная нами bn , соответствует $x = -1$, т.е. $y = 1$. Выберем, поэтому в левом поле (с заголовком x is bn), а в правом 1 и нажмем кнопку Add rule (Добавить правило). Введенное правило появится в окне правил и будет представлять собой запись:

If (x is bn) then (y is 1)

Аналогично поступим для всех других значений x , в результате чего сформируется набор из 5 правил рисунок 4.5. Закроем окно редактора правил и возвратимся в окно FIS-редактора. Построение системы закончено и можно начать эксперименты по ее исследованию. Заметим, что большинство опций выбиралось нами по умолчанию

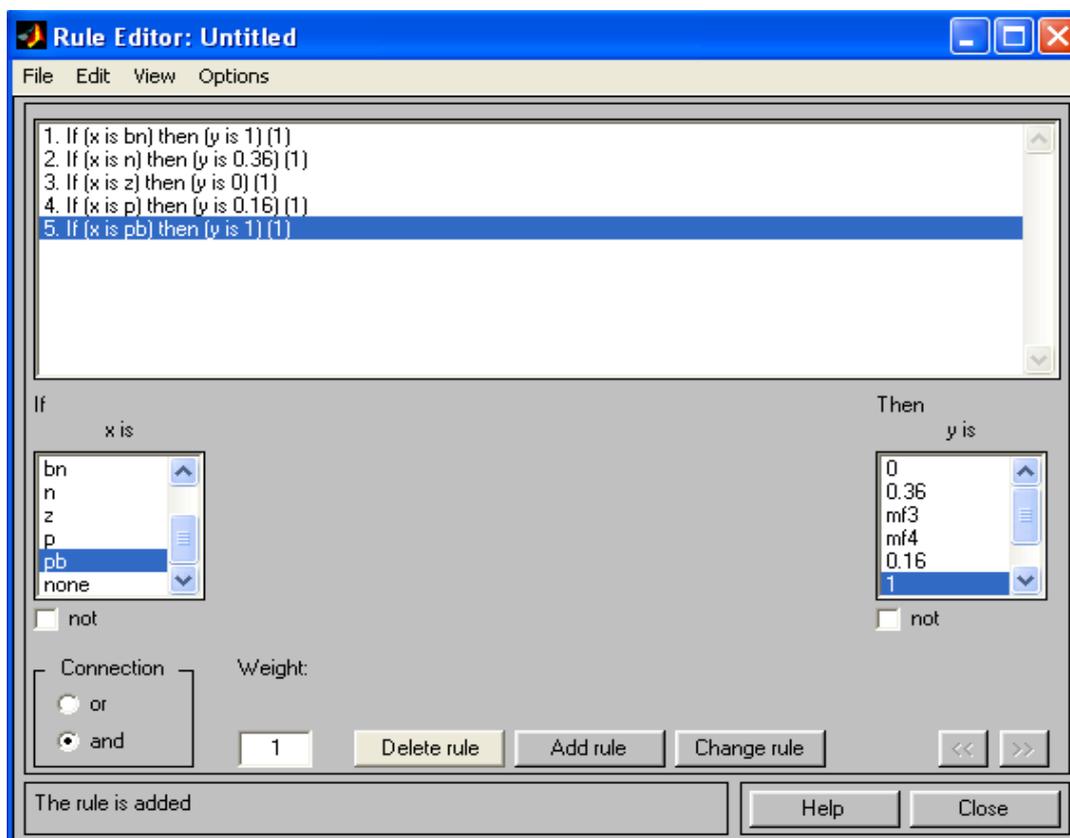


Рисунок 4.5- Окно редактора правил

Предварительно сохраним на диске (используя пункты меню File/Save to disk as...) созданную систему под каким-либо именем, например, Proba.

Выберем позицию меню View. Как видно из выпадающего при этом подменю, с помощью пунктов Edit membership functions и Edit rules можно совершить переход к двум выше рассмотренным программам – редакторам функций принадлежности и правил (то же можно сделать и нажатием клавиш Ctrl+2 или Ctrl+3). Но сейчас нас будут интересовать два других пункта – View rules (Просмотр правил) и View surface (Просмотр поверхности). Выберем пункт View rules, при этом откроется окно рисунок 4.6 еще одной программы – просмотра правил (Rule Viewer)

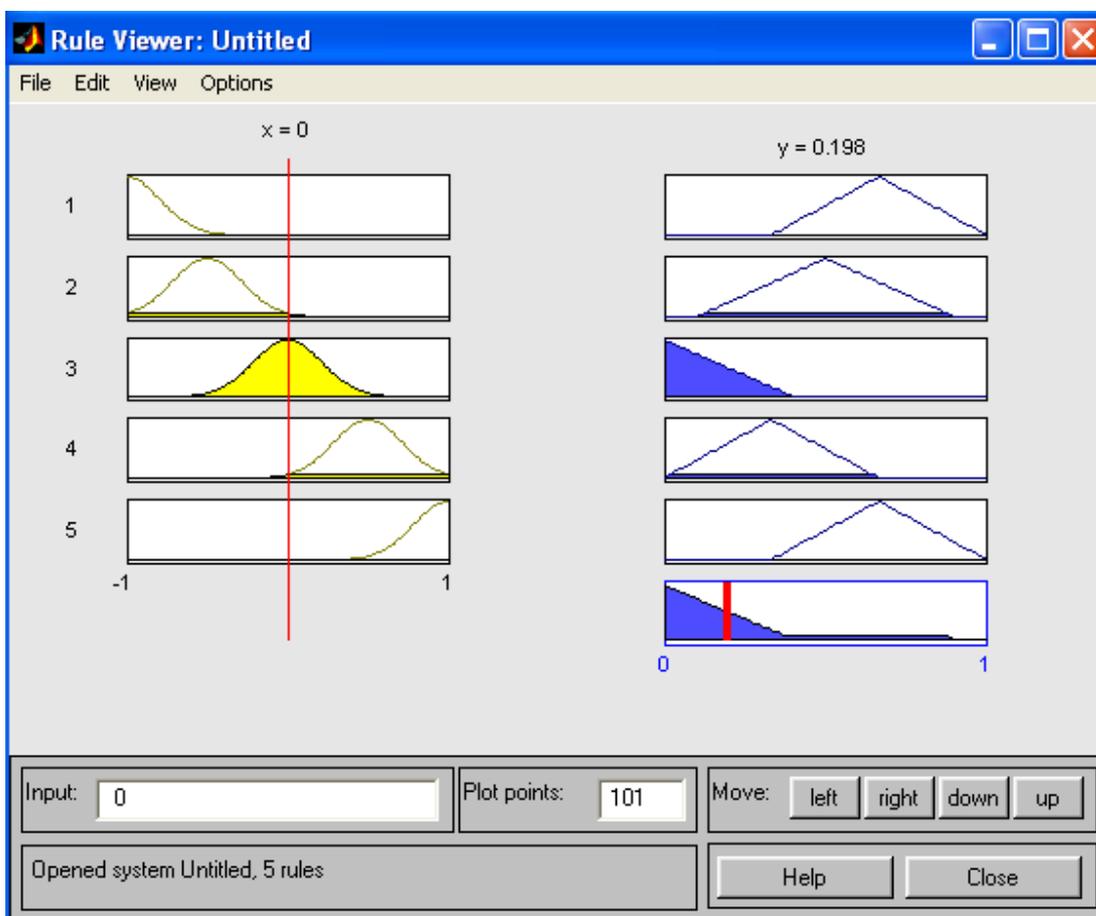


Рисунок 4.6- Окно просмотра правил

В правой части окна в графической форме представлены функции принадлежности аргумента ж, в левой – переменной выхода u с пояснением механизма принятия решения. Красная вертикальная черта, пересекающая графики в правой части окна, которую можно перемещать с помощью курсора, позволяет изменять значения пере-

менной входа (это же можно делать, задавая числовые значения в поле Input (Вход)), при этом соответственно изменяются значения y в правой верхней части окна. Зададим, например, $x = 0.5$ в поле Input и нажмем затем клавишу ввода (Enter). Значение y сразу изменится и станет равным 0.202. Таким образом, с помощью построенной модели и окна просмотра правил можно решать задачу интерполяции, т.е. задачу, решение которой и требовалось найти. Изменение аргумента путем перемещения красной вертикальной линии очень наглядно демонстрирует, как система определяет значения выхода.

Закроем окно просмотра правил и выбором пункта меню

View/View surface перейдем к окну просмотра поверхности отклика (выхода), в нашем случае – к просмотру кривой $y(x)$ рисунок 4.7. Видно, что смоделированное системой по таблице данных таблице 4.1 отображение не очень-то напоминает функцию x^2 . Ну что ж, ничего удивительного в этом нет: число экспериментальных точек невелико, да и параметры функций принадлежности (для x) выбраны, скорее всего, неоптимальным образом. Ниже мы рассмотрим возможность улучшения качества подобной модели.

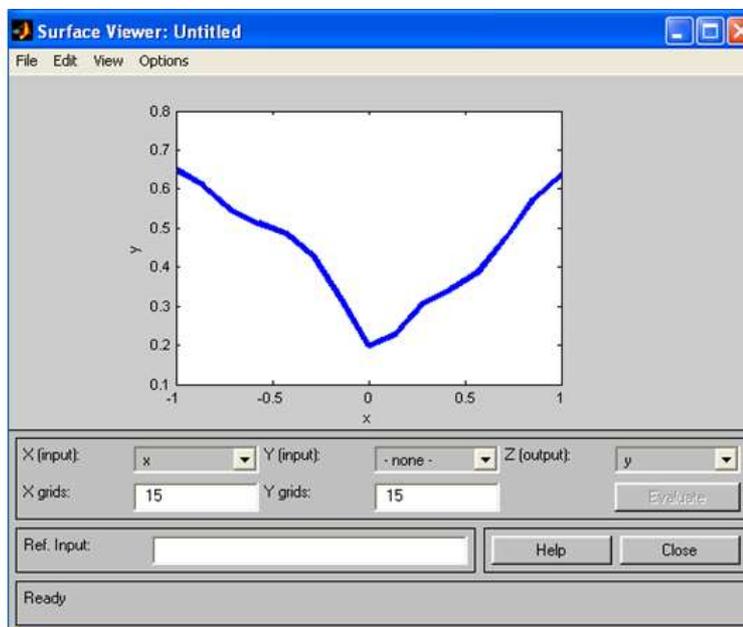


Рисунок 4.7- Окно просмотра поверхности отклика

В заключение рассмотрения примера отметим, что с помощью вышеуказанных программ - редакторов на любом этапе проектирования нечеткой модели в нее можно внести необходимые

Из опций, устанавливаемых в FIS-редакторе по умолчанию при использовании алгоритма Sugeno, можно отметить:

логический вывод организуется с помощью операции умножения (prod);

композиция – с помощью операции логической суммы (вероятностного ИЛИ, probor);

4.3 Задание

Сконструируйте нечеткую систему, отображающую зависимость между переменными x и y , заданную с помощью таблице 4.3. По результатам работы определить тип кривой.

Таблица 4.3.-Варианты заданий

Варианты	Значение аргумента и функции					
	3	4	5	6	7	
1	-1	-0.5	0	0.2	1	
1	1	0.25	0	0.4	1	
2	-1	-0.6	0.2	0.4	1	
2	-1	-1.67	5	2.5	1	
3	-1	-0.5	0	0.3	1	
3	-1	-0.13	0	0.27	1	
4	-1	-0.6	0	0.3	1	
4	0	0.8	1	0.95	0	
5	-1	-0.5	0	0.2	1	
5	1	-0.125	0	0.008	1	
6	-1	-0.6	0.2	0.4	1	
6	0	-0.64	-0.96	-0.84	0	
7	-1	-0.5	0	0.3	1	
7	-3	-2	-1	-0.4	1	
8	-1	-0.6	0	0.3	1	
8	0.5	0.09	0	0.0225	0.5	
9	-1	-0.5	0	0.2	1	
9	0.5	0.03125	0	0.0008	0.5	
10	-1	-0.6	0.2	0.4	1	
10	-1	2.78	25	6.25	1	
11	-1	-0.5	0	0.3	1	

Контрольные вопросы

1. Назовите разные постановки задачи многокритериального выбора, приводящие к различным решениям.
2. Построение нечеткой аппроксимирующей системы
3. Аппроксимирующая система в пакете FUZZY LOGIC TOOLBOX
4. Характеристика аппроксимирующей системы
5. На каком множестве осуществляется выбор в случае статистической неопределенности.

Самостоятельная работа № 5

Цифровая обработка изображений средствами MATLAB

5.1 Цель работы

Изучение операций по сегментации изображений, функций, реализующих операции по сегментации изображения, и приобретение практических навыков использования этих функций.

5.2

Краткие теоретические сведения

Сегментация изображения представляет собой разделение изображения на области по сходству свойств (признаков) в их точках. Признаки подразделяются на естественные и искусственные. Естественные признаки устанавливаются простым (визуальным) анализом изображения, а искусственные – в результате специальной обработки различных измерений. Примерами естественных признаков являются структура, текстура, яркость объекта. Примеры искусственных признаков: гистограммы распределения яркости, спектр и др.

К основным видам сегментации изображений относится сегментация по яркости, цветовым координатам, контурам, форме.

Методы сегментации

Сегментация методом выращивания областей

Для сегментации изображения можно использовать метод выращивания областей – группирование пикселей или подобластей в более крупные области по заранее заданным критериям роста. Берутся «центры кристаллизации», а затем на них наращиваются области путем добавления к каждому центру тех соседних пикселей, которые по своим свойствам близки к центру кристаллизации (например, имеют яркость или цвет в определенном диапазоне). Ниже приведена функция `regiongrow`, которая выполняет выращивание областей.

Синтаксис

`[g, NR, SI, TI] = regiongrow(f, S, T),`

где f – это сегментируемое изображение, а параметр S – массив (с размерами как у f) или скаляр. Если S – массив, то он содержит 1 в

тех позициях, где расположены центры кристаллизации и 0 во всех остальных местах. Если S является скаляром, то он задает значение яркости пикселей, которые становятся центрами кристаллизации. Аналогично, T может быть массивом (с размерами, как у f) или скаляром. Если T – массив, то его элементы являются локальными пороговыми значениями для f. Скаляр T определяет глобальный порог.

5.3 Задание

Задание 1. Создать функцию regiongrow.

```
function [g, NR, SI, TI] = regiongrow(f, S, T)
if numel(S) == 1
SI = f == S;
```

```

S1 = S
else
SI = bwmorph(S, 'shrink', Inf);
J = find(SI);
S1 = f(J);
end
TI = false(size(f));
for K = 1:length(S1)
seedvalue = S1(K);
S = abs(f - seedvalue) <= T;
TI = TI | S;
end
[g, NR] = bwlabel(imreconstruct(SI, TI));
```

Задание 2. Выполнить сегментацию наращиванием областей для изображения, хранящегося в файле Finance.bmp, используя функцию regiongrow.

```
[x,map] =
imread('c:\image\Finance.bmp'); I =
im2double(ind2gray(x,map));
figure,imshow(I)
S = 0.9783; T = 0.0651; % эти значения находятся
экспериментально (взяты из изображения)
[g, NR, SI, TI] = regiongrow(I, S, T);
figure,imshow(TI)
```

Также для сегментации используется метод разделения. Функция MatLab, реализующая подобный алгоритм, приведена ниже.

Сегментация методом разделения

Изображение разбивается на непересекающиеся блоки, которые с помощью некоторого критерия проверяются на однородность.

Функция сегментации полутоновых изображений методом разделения

qtdecomp

Синтаксис

$A = \text{qtdecomp}(I, \text{threshold}, \text{mindim})$

Функция qtdecomp осуществляет сегментацию полутоновых изображений методом разделения. В функции qtdecomp каждый блок разбивается на 4 непорекрывающихся блока одинакового размера. На первом шаге алгоритма блоком считается все изображение. Мельчайшим по размерам является блок, который нельзя разделить на 4 блока одинакового размера, т. е. такой блок, у которого число строк или число столбцов нечетное. Таким образом, в функции qtdecomp рекомендуется использовать изображения с размерами, равными степеням двух. Функция $A = \text{qtdecomp}(I, \text{threshold}, \text{mindim})$ осуществляет сегментацию полутонового изображения I методом разделения и помещает результат в разреженный массив A (тип данных sparse MatLab). Элементам матрицы $A(r,c)$, соответствующим координатам левых верхних углов блоков на исходном изображении I , присваиваются значения, определяющие размеры каждого блока. Блок считается однородным, если разница между максимальным и минимальным значением пикселей блока меньше параметра threshold. Параметр mindim определяет минимальный размер блока.

Функция получения блоков из quadro-дерева результатов сегментации qtgetblk

Синтаксис

$[\text{vals}, \text{idx}] = \text{qtgetblk}(I, A, \text{dim})$

Функция возвращает в массив vals все блоки размером dim, получившиеся в результате сегментации полутонового изображения I с помощью функции qtdecomp. В параметре A передается разреженный массив,

описывающий quadro-дерево с результатами сегментации. Координаты левых верхних углов блоков, помещенных в массив `vals`, находятся в векторе `idx`. Если нет ни одного блока размером `dim`, то всем возвращаемым параметрам присваиваются значения пустых `vals` матриц.

Функция замены блоков – результатов сегментирования `qtsetblk`

Синтаксис

$I_D = \text{qtsetblk}(I_S, A, \text{dim}, \text{vals})$

Функция создает новое полутоновое изображение I_D , заменяя в исходном полутоновом изображении все блоки размера `dim`, полученные в результате сегментации с помощью функции `qtdecomp`, на блоки из массива `vals`. В параметре `A` передается разреженный массив, описывающий quadro-дерево с результатами сегментации. Данная функция используется для преобразования изображения в соответствии с результатами сегментации методом разделения.

Рассмотрим работу функции `qtdecomp` совместно с функциями `qtgetblk` и `qtsetblk` для полутонового изображения размера 8×8 пикселей. Формат представления данных – `uint8`. Будем считать, что блок изображения является однородным, если величина разброса яркостей пикселей в блоке не превышает 10 градаций яркости. Установим минимально возможный размер блока. В нашем случае он равен двум.

Будем считать, что к объекту относятся блоки, средняя яркость которых не превышает 50. Требуется изменить исходное изображение так, чтобы пикселям блоков, относящихся к объекту, было присвоено значение 1, а пикселям блоков, не относящихся к объекту, 0.

Задание 3. Выполнить сегментацию небольшого текстового изображения методом разделения.

Исходное изображение:

$I = [$
10 11 10 15 20 25 47 51
11 14 17 13 27 29 52 55
12 13 11 10 24 47 56 60
13 14 11 13 49 54 74 77
15 16 43 48 79 82 87 86
17 18 45 50 85 80 80 84

```
29 51 50 59 80 83 83 85
59 61 58 61 81 85 86 88 ];
```

Сегментация методом разделения: размер минимального блока 2×2 ; блок считается однородным, если в его пределах яркость изменяется меньше, чем на 10 градаций.

```
>> A = qtdecomp(I,10,2);
```

Для удобства визуального анализа предварительно преобразуем разреженную матрицу A в обычную матрицу M с помощью функции full.

```
>> M =
full(A) M =
 4 0 0 0 2 0 2 0
 0 0 0 0 0 0 0 0
 0 0 0 0 2 0 2 0
 0 0 0 0 0 0 0 0
 2 0 2 0 4 0 0 0
 0 0 0 0 0 0 0 0
 2 0 2 0 0 0 0 0
 0 0 0 0 0 0 0 0
```

В результате сегментации получили 2 блока размером 4×4 (левая верхняя и правая нижняя части изображения) и 8 блоков размером 2×2 .

Переберем в цикле все возможные размеры блоков: 8, 4, 2.

```
>> dim = 8;
>> while dim >= 2
% получить в переменной blocks все блоки
размера dim. [blocks,idx] = qtgetblk(I,A,dim);
[x y n] = size(blocks);
% если блоки такого размера есть в quadro-дереве,
if n > 0
% то перебираем все блоки размера dim
for j = 1:n
% если среднее значение яркости пикселей в пределах блока
меньше 50
if (mean2(blocks(:,:,j)) < 50)
% то заменяем значения всех пикселей
блока на 1, blocks(:,:,j) = ones(dim,dim);
else
```

```

%   иначе заменяем значения всех
пикселей на 0. blocks(:,:,j) = zeros(dim,dim);
    end;
    end % end for
%   устанавливаем новые значения всех пикселей
размера dim I = qtsetblk(I,A,dim,blocks);
    end; % end if
    dim = dim/2;
    end % end while      % ссылка на 1 из списка литературы
%   получившееся
изображение I >> I
    I =
    1 1 1 1 1 1 0 0
    1 1 1 1 1 1 0 0
    1 1 1 1 1 1 0 0
    1 1 1 1 1 1 0 0
    1 1 1 1 0 0 0 0
    1 1 1 1 0 0 0 0
    0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0

```

Задание 4. Выполнить сегментацию реального изображения из файла cotton3.bmp

```

rgb = imread('c:\Image\cotton3.bmp');
I = im2double(rgb2gray(rgb));
figure,imshow(I)
T=graythresh(I);
A = qtdecomp(I,0.1,2);
dim = 8;
while dim >= 2
[blocks,idx] = qtgetblk(I,A,dim);
[x y n] = size(blocks);
if n>0
for j = 1:n
if (mean2(blocks(:,:,j))<T)
blocks(:,:,j) = ones(dim,dim);
else
blocks(:,:,j) = zeros(dim,dim);

```

```

end
end
I1 = qtsetblk(I,A,dim,blocks);
end
dim = dim/2;
end
figure,imshow(I1)

```

Функция выбора интересующей области по цвету roicolor

Синтаксис

```
BW = roicolor(S,low,high)
```

```
BW = roicolor(S,v)
```

Для любого варианта вызова функции roicolor бинарное изображение формируется по следующему алгоритму: пикселу бинарного изображения BW(r, c) присваивается значение 1, если яркость пиксела S(r, c) исходного полутонового изображения или индекс S(r, c) палитрового изображения принадлежит диапазону [low, high] или любому из значений вектора v. В противном случае BW(r, c) присваивается значение 0.

Задание 5. Выбрать цветовые области из изображения файла chip.bmp, задавая индексы с помощью гистограммы.

```
>> [x,map]=imread('C:\Image\chip.bmp');
```

```

>>
>> figure,imhist(x,map),title('histogramma')
>> figure,imshow(x,map),
>> bw=roicolor(x,9,12);
>> figure,imshow(bw),title('9 - 12')
>> x1=immultiply(bw,x);
>> figure,imshow(x1,map),title('9 - 12')
>> bw=roicolor(x,3,8);
>> figure,imshow(bw),title('3 - 8')
>> x1=immultiply(bw,x);
>> figure,imshow(x1,map),title('3 - 8')

```

Синтаксис

```
P = impixel(S,c,r)
```

Функция `imread` возвращает значения красной, зеленой и синей составляющих цвета для заданных координат – `c` и `r` – векторов значений столбцов и строк.

Задание 6. Выполнить яркостный срез полноцветного изображения файла `cotton3.bmp`.

```
[img] = imread('c:\Image\cotton3.bmp');
[m,n,k] = size(img)
img = im2double(img);
R = zeros(m,n,3);
z = [0.1,0.8;0.1,0.8; 0.1,0.9];
for y = 1:m
for x = 1:n
b = imixel(img,x,y);
if ((b(1)>= z(1,1))&(b(1)<= z(1,2)))& ((b(2)>= z(2,1))&(b(2)<=
z(2,2))) ... & ((b(3)>= z(3,1))&(b(3)<= z(3,2)))
R(x,y,1) = b(1); R(x,y,2) = b(2);
R(x,y,3) = b(3); else
R(x,y,1) = 0; R(x,y,2) = 0; R(x,y,3) = 0;
end
end
end
end
figure, imshow(img)
figure, imshow(R)
```

Задания для самостоятельного решения

1. Выполнить сегментацию изображений файла `Clouds.bmp` методом разделения.

2. Выполнить сегментацию изображений файла `Construc.bmp` методом выращивания областей.

3. Выбрать цветовые области из изображения файла `bike.bmp`, задавая индексы с помощью гистограммы.

3. Выполнить яркостный срез полноцветного изображения файла `bike.bmp`, задавая диапазон `r` от 0.2 до 0.8; `g` от 0.2 до 0.7; `b` от 0.1 до 0.7.

Контрольные вопросы

1. В чем заключается сегментация изображения?
2. Какие признаки используются для сегментации?
3. В чем заключается метод выращивания областей, использующийся для сегментации изображения?
4. В чем заключается метод разделения, использующийся для сегментации изображения?
5. Что является входными параметрами функции сегментации методом разделения?
6. В чем заключается преобразование яркостного среза?
7. Какие параметры возвращает функция `imrixel`?