1. Программирование в MATLAB. Подготовка тестовых данных

1.1. В режиме командной строки задать матрицы A и B размером 3Х4 с произвольными элементами

A=rand(3,4)

A =

0.8147 0.9134 0.2785 0.9649

0.9058 0.6324 0.5469 0.1576

0.1270 0.0975 0.9575 0.9706

>> B=rand(3,4)

B =

0.9572 0.1419 0.7922 0.0357

0.4854 0.4218 0.9595 0.8491

0.8003 0.9157 0.6557 0.9340

1.2. Определить матрицу С как разность матриц A и B

>> C=A-B

C =

-0.1424 0.7715 -0.5137 0.9292

0.4204 0.2106 -0.4126 -0.6915

-0.6733 -0.8182 0.3018 0.0366

1.3. Определить матрицу D как матрицу, каждый элемент которой равен модулю соответствующего элемента матрицы C

>> D=abs(C)

D =

0.1424 0.7715 0.5137 0.9292

0.4204 0.2106 0.4126 0.6915

0.6733 0.8182 0.3018 0.0366

1.4. Найти число M как максимальный элемент матрицы D

>> M=max(max(D))

M =

0.9292

1.5. Найти число S как среднюю сумму элементов матрицы D

>> S=sum(sum(D))/12

S =

0.4935

1.6. Реализовать пункты 1.2-1.5 в виде функции от 2-х переменных, возвращающую матрицу D и числа M и S

function [ D1, M1, S1 ] = universalFunc( A,B )

D1 = abs(A-B);

M1 = max(max(D1));

t = size(D1);

S1 = sum(sum(D1))/(t(1)\*t(2));

end

1.7. Проверить работу функции на введенных ранее матрицах A и B.

[d,m,s] = universalFunc(A,B)

d =

0.1424 0.7715 0.5137 0.9292

0.4204 0.2106 0.4126 0.6915

0.6733 0.8182 0.3018 0.0366

m =

0.9292

s =

0.4935

1.8. Используя циклы, создать функцию, возвращающую матрицу вида 10201x2, содержащую все всевозможные сочетания значений от 0 до 10 с шагом 0.1

function result = Cicle

x=1;

for j =0:0.1:10

for i = 0:0.1:10

result(x,1)= i;

result(x,2)= j;

x=x+1;

end

end

end

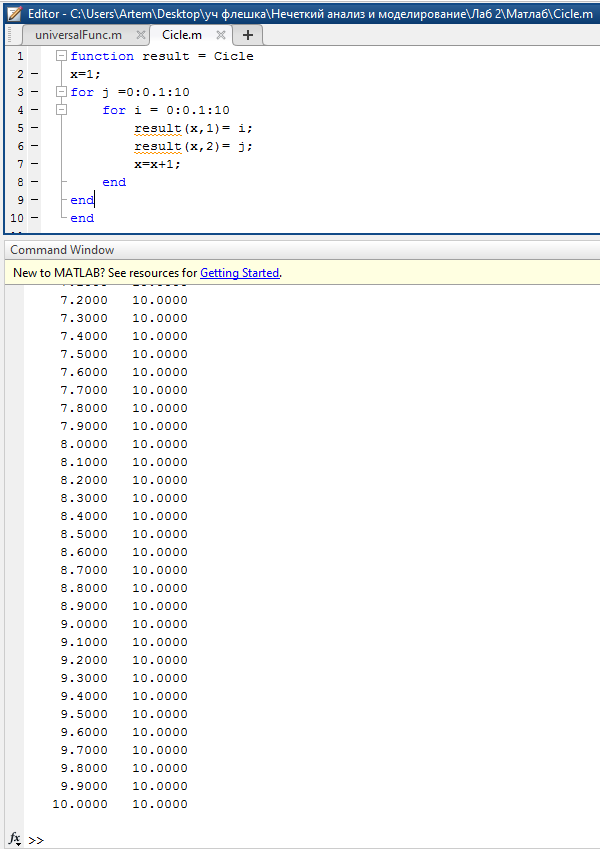


Рисунок 1 – Матрица вида 10201x2

2. Вычисление результата логического вывода

2.1. Используя функцию readfis или пункт меню File->Export-> To\_Work\_Space FIS-редактора, загрузить в рабочее пространство MATLAB FIS-структуру из файла tipper.fis

x1=readfis

x1 =

name: 'tipper'

type: 'mamdani'

andMethod: 'min'

orMethod: 'max'

defuzzMethod: 'centroid'

impMethod: 'min'

aggMethod: 'max'

input: [1x2 struct]

output: [1x1 struct]

rule: [1x3 struct]

2.2. Используя функцию mam2sug преобразовать её в FIS-структуру типа Сугэно (используя переменную с другим именем).

x2=mam2sug(x1)

name: 'tipper'

type: 'sugeno'

andMethod: 'min'

orMethod: 'max'

defuzzMethod: 'wtaver'

impMethod: 'prod'

aggMethod: 'max'

input: [1×2 struct]

output: [1×1 struct]

rule: [1×3 struct]

2.3. Загрузить в FIS-редактор полученную структуру из рабочей области

Fuzzy

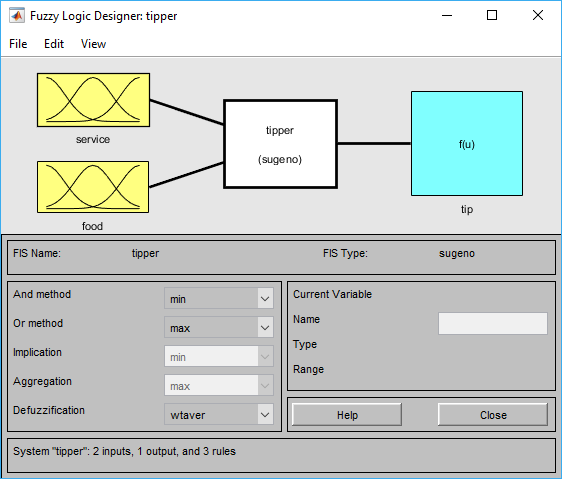


Рисунок 2 – Структура tipper (x2)

2.4. Сохранить FIS-структуру под новым именем

Tippernew.fis

2.5. Зафиксировать в отчёте значения выходных констант, изображения логического вывода (при прежних значениях входных переменных) и поверхности вывода.

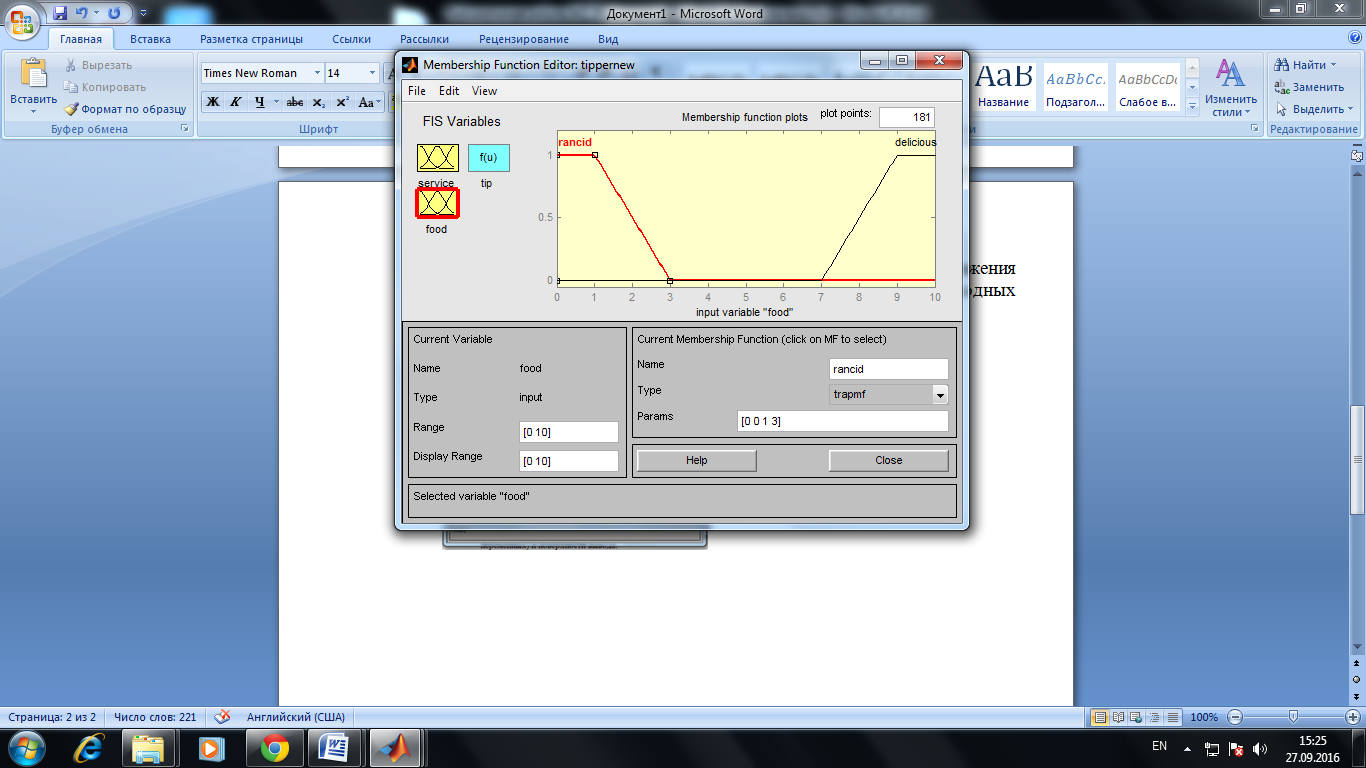
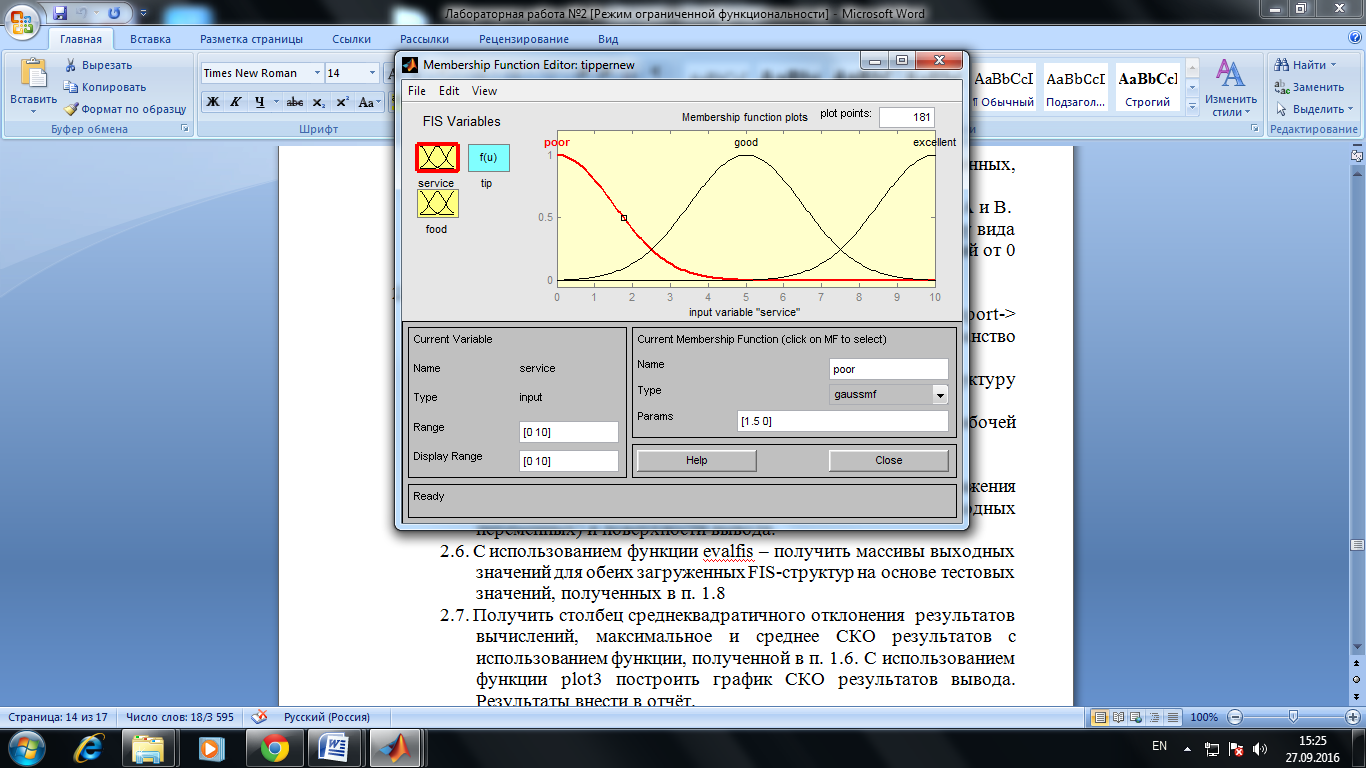


Рисунок 3 – Service, food

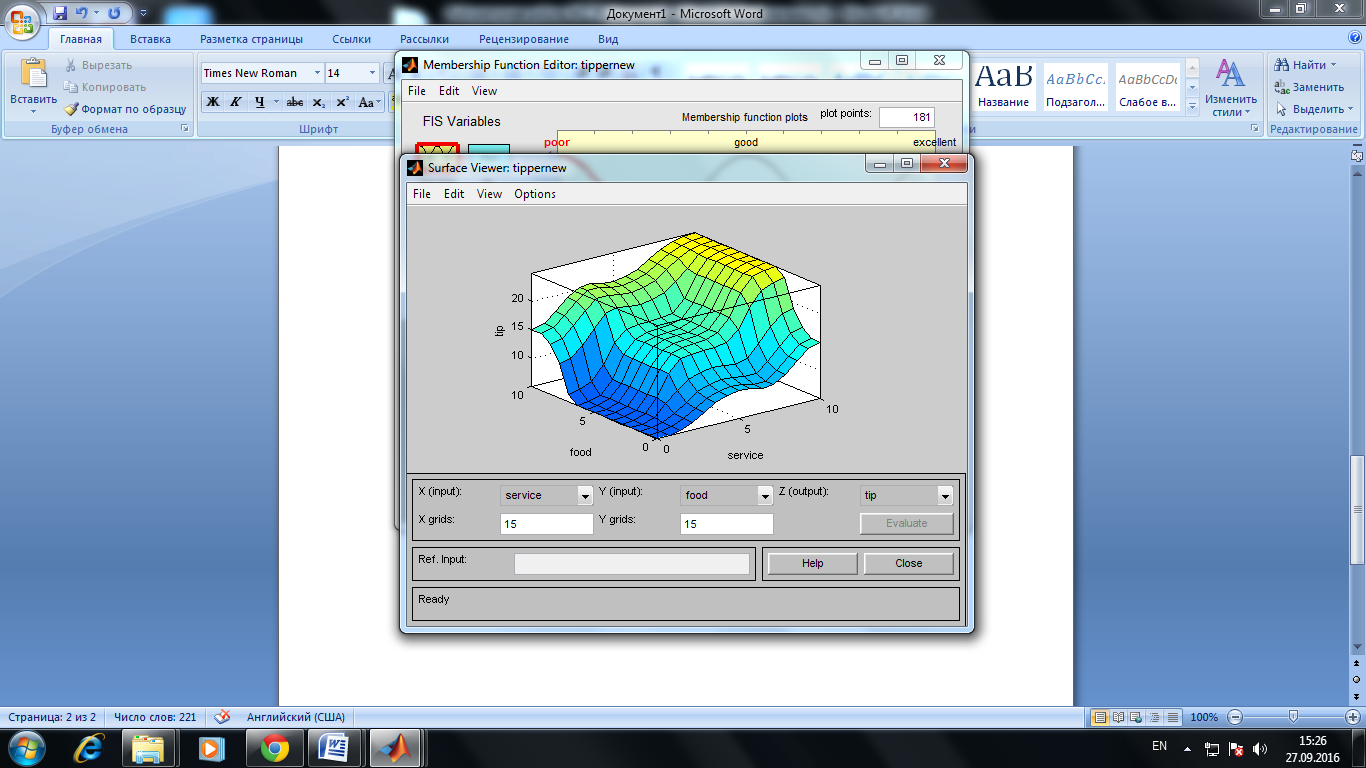
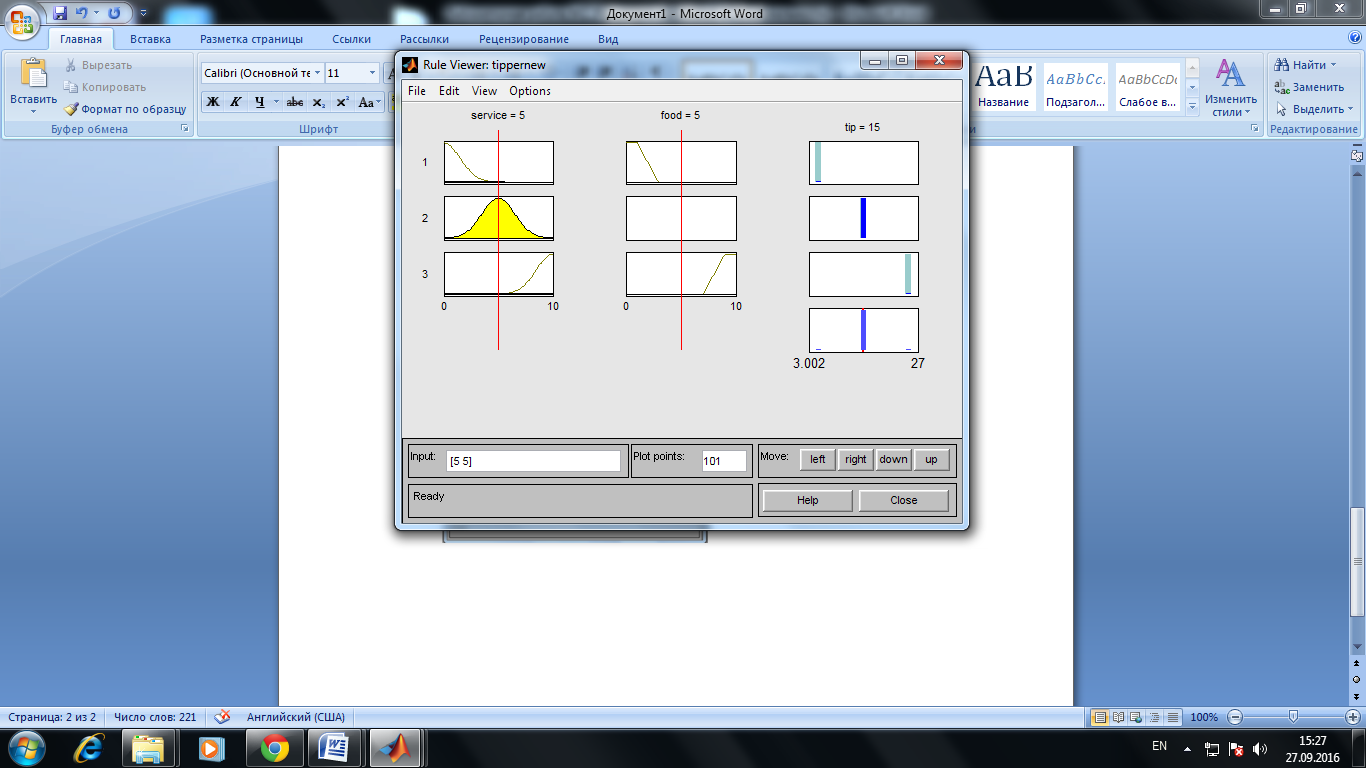
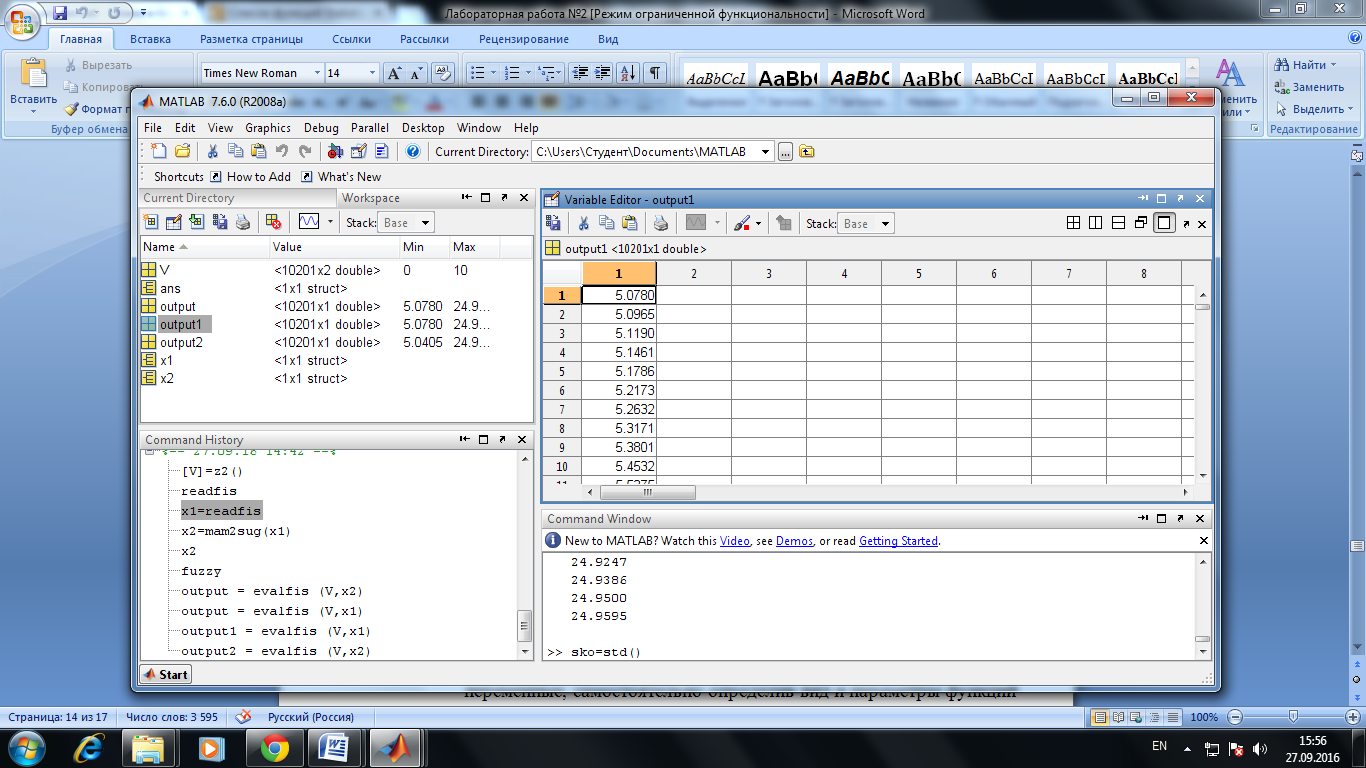


Рисунок 4 - Изображения логического вывода и поверхности вывода

2.6. С использованием функции evalfis – получить массивы выходных значений для обеих загруженных FIS-структур на основе тестовых значений, полученных в п. 1.8

Для tipper

output1 = evalfis (V,x1)

  
Рисунок 5 - Массив выходных значений для первой FIS-структуры

Для tippernew (Сегуно)

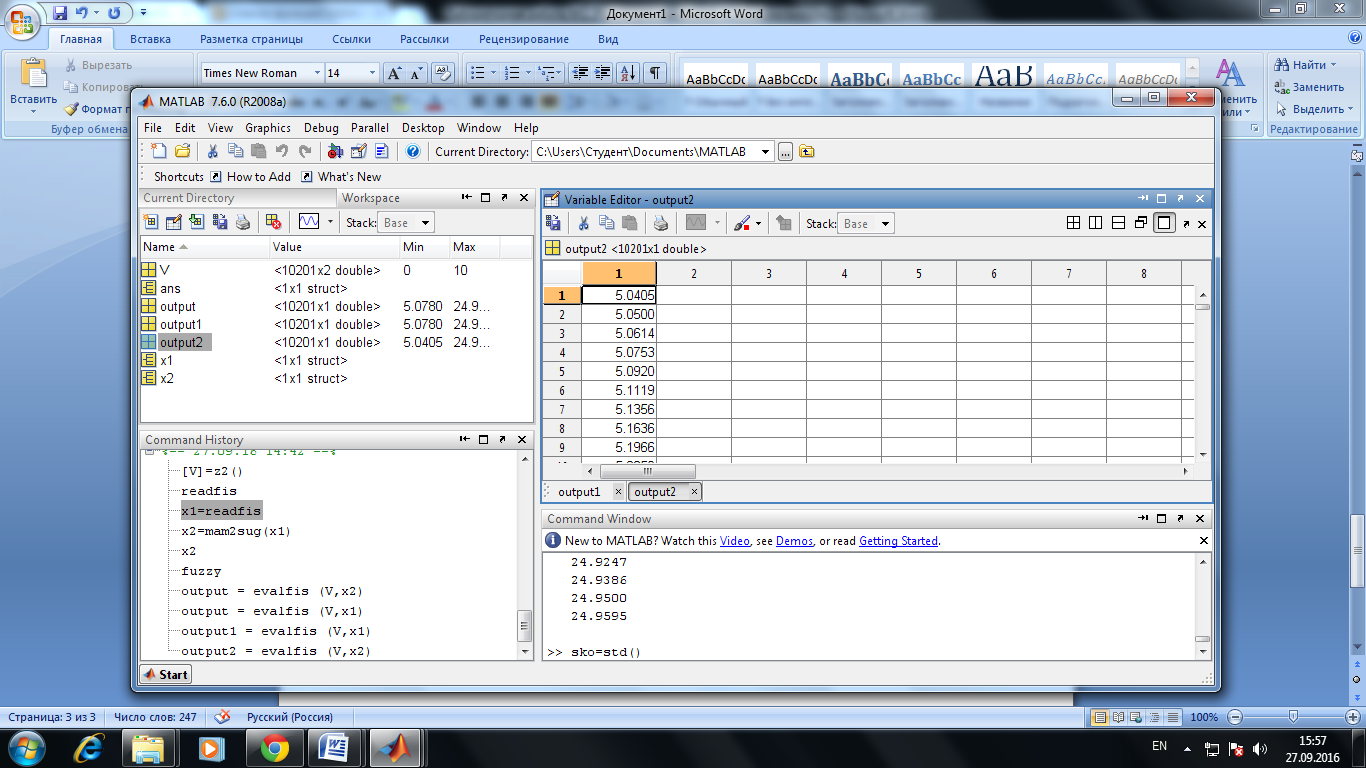


Рисунок 6 - Массив выходных значений для второй FIS-структуры

2.7. Получить столбец среднеквадратичного отклонения результатов вычислений, максимальное и среднее СКО результатов с использованием функции, полученной в п. 1.6. С использованием функции plot3 построить график СКО результатов вывода. Результаты внести в отчёт.

[Sk,m,sr]=lb2(output1,output2)

m = 2.1416

sr = 0.4527

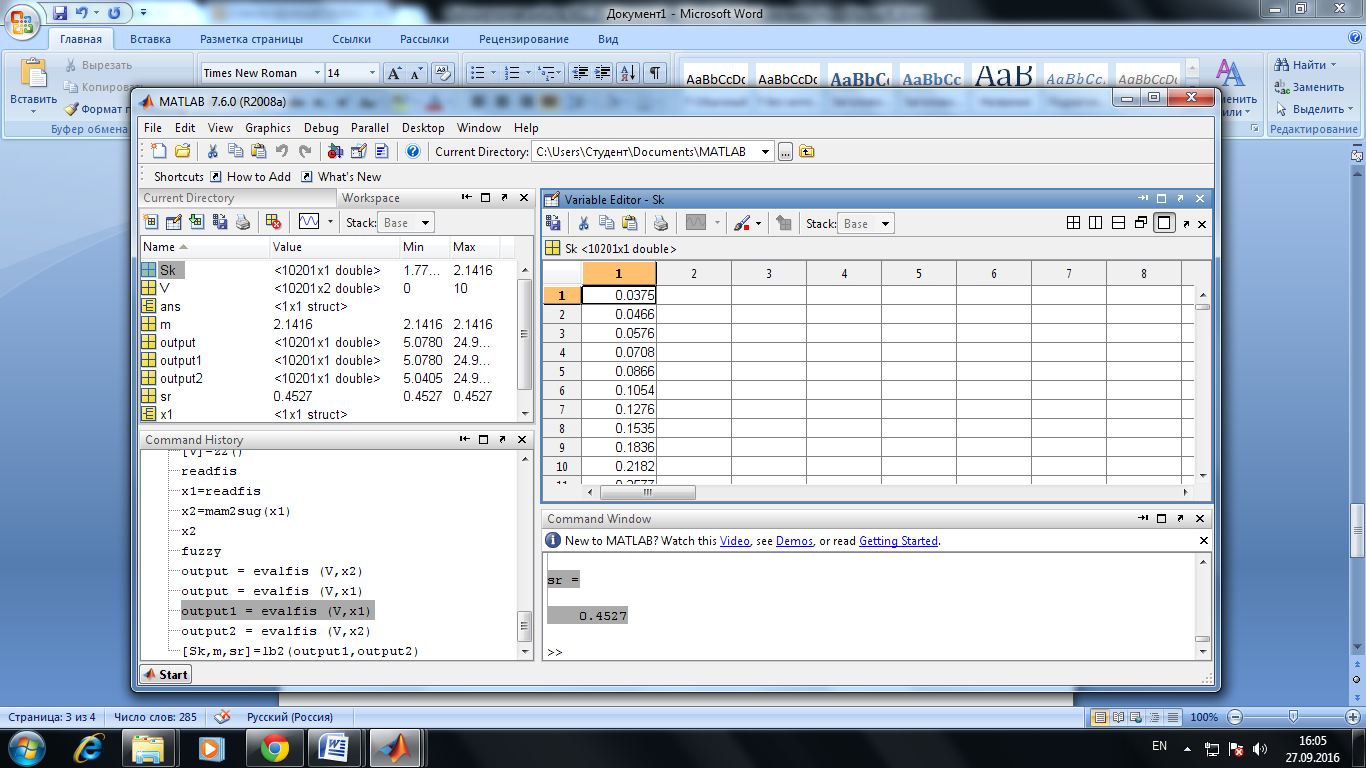


Рисунок 7 - Столбец среднеквадратичного отклонения результатов вычислений

c=V(1:10201,1)

d=V(1:10201,2)

x=Sk

plot3(c,d,x)

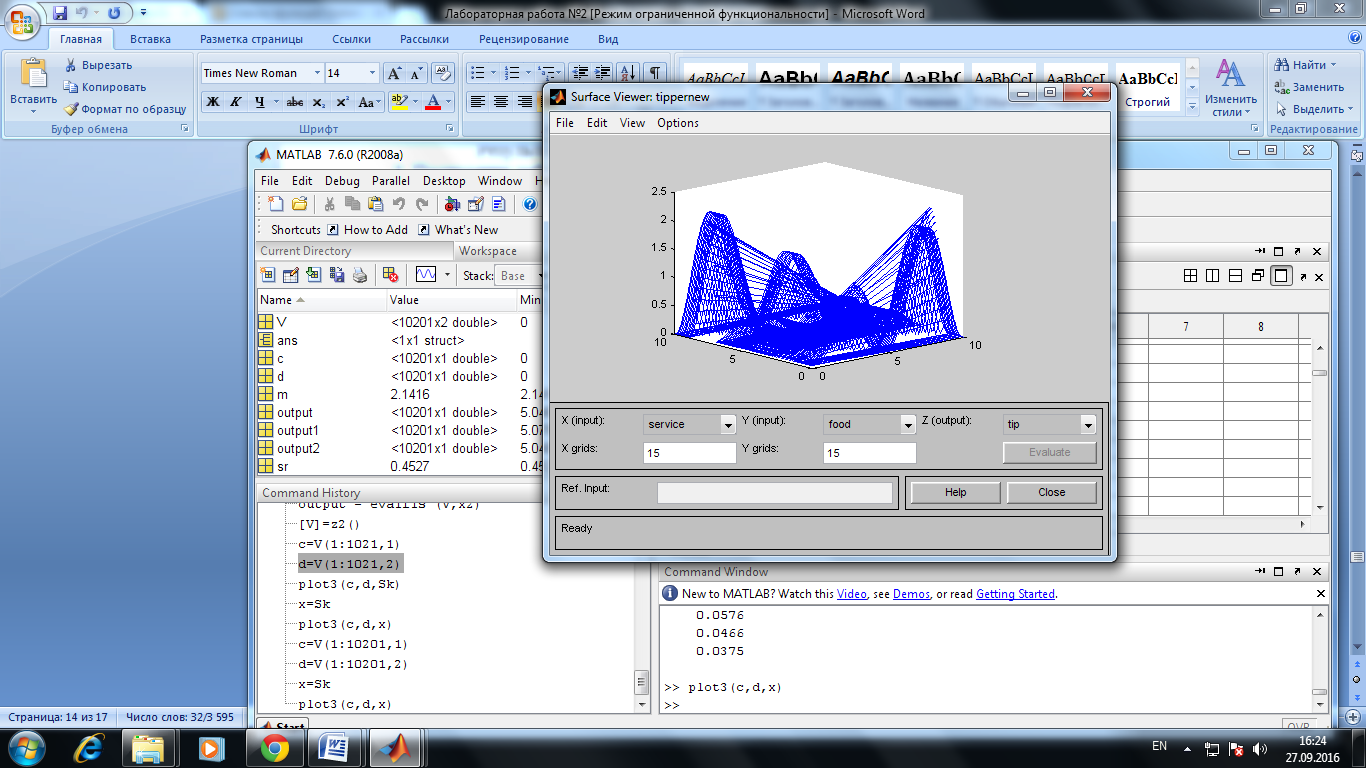


Рисунок 8 - График СКО результатов вывода

3. Построение собственной нечёткой системы типа Сугэно

3.1. Запустить FIS-редактор и создать новую FIS-структуру типа Сугэно

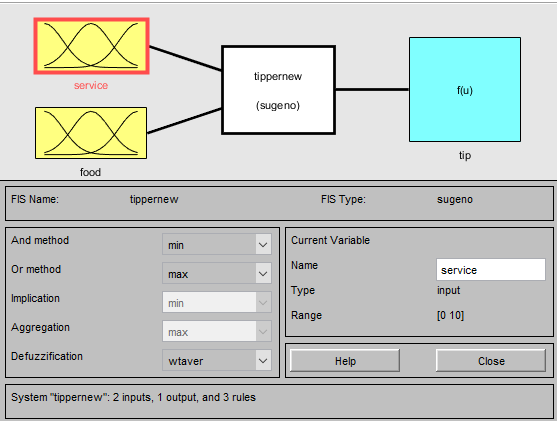


Рисунок 9 - новая FIS-структуру типа Сугэно

3.2. В соответствии с заданием по бригаде задать входные и выходные переменные, самостоятельно определив вид и параметры функций принадлежности.

Тема 3. Зависимость загрузки пассажирского автобуса от времени суток и дальности от начальной остановки.

Таблица 1 – Переменная time (диапазон значений 0-24)

|  |  |  |
| --- | --- | --- |
| Имя терма | Тип | Параметры |
| morning | gaussmf | [2.5 0] |
| lunch | gaussmf | [2.5 12] |
| eve | gaussmf | [2.5 24] |

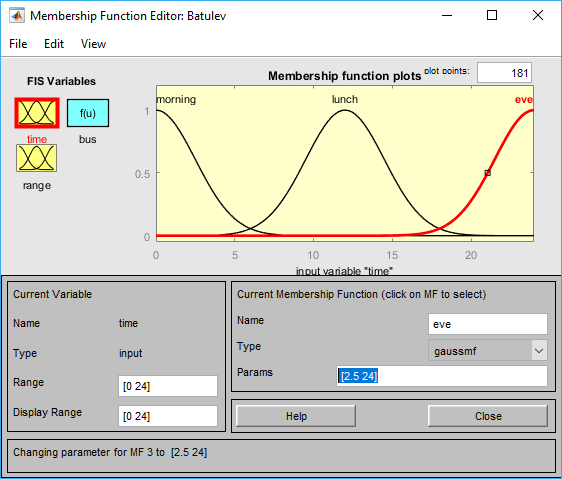


Рисунок 10 – данные о входной переменной time

Таблица 2 – Переменная range (диапазон значений 0-100)

|  |  |  |
| --- | --- | --- |
| Имя терма | Тип | Параметры |
| near | trapmf | [0 0 10 25] |
| middle | trapmf | [10 40 60 90] |
| far | trapmf | [75 90 100 100] |

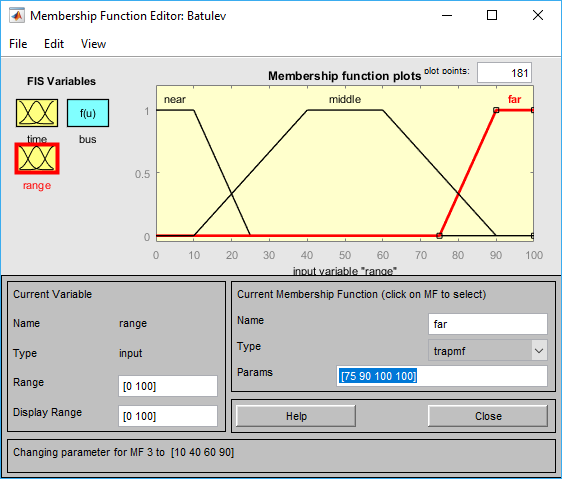


Рисунок 11 – данные о входной переменной range

Таблица 3 - Выходная переменная bus (диапазон значений 0-30)

|  |  |  |
| --- | --- | --- |
| Имя терма | Тип | Параметры |
| few | constant | 5 |
| normal | constant | 15 |
| many | constant | 25 |

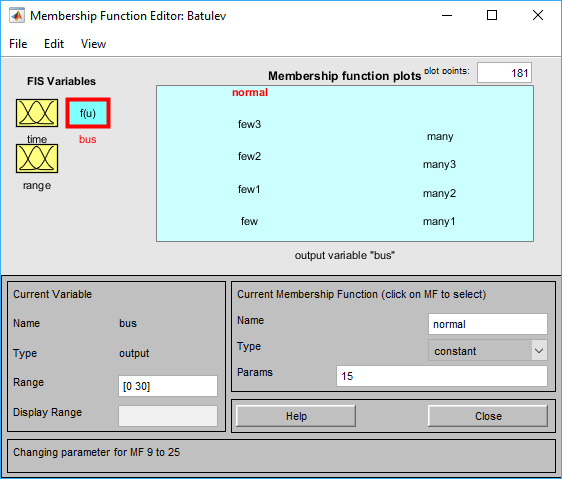


Рисунок 12 – данные о выходной переменной bus

3.3. Задать базу правил на основе знаний о характере системы. Для связи нечётких значений входных переменных использовать связку “и”. Сформировать выборку, основывающуюся на априорных знаниях (априорную выборку), представляющую собой матрицу размером N×3, где N – число правил в базе знаний (подробнее см. приложение).

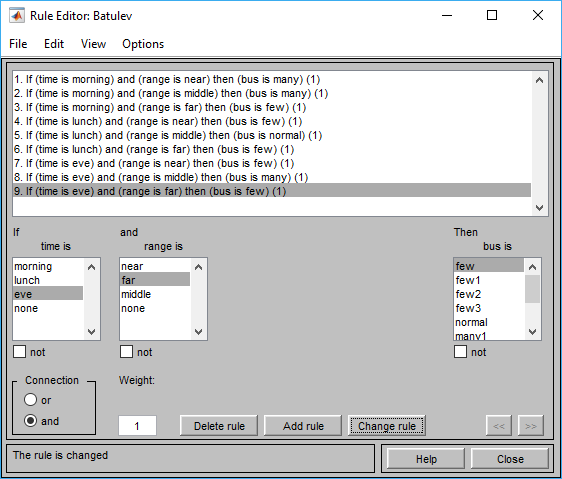


Рисунок 13 – Правила

Априорная выборка

M = [0 0 25; 0 50 25; 0 100 5;

12 0 5; 12 50 15;12 100 5;

24 0 5; 24 50 25; 24 100 5];

0 0 25

0 50 25

0 100 5

12 0 5

12 50 15

12 100 5

24 0 5

24 50 25

24 100 5

3.4. Для исходных настроек (методы “и”, дефаззификации) получить изображения логического вывода и поверхности вывода. Результаты зафиксировать в отчёте

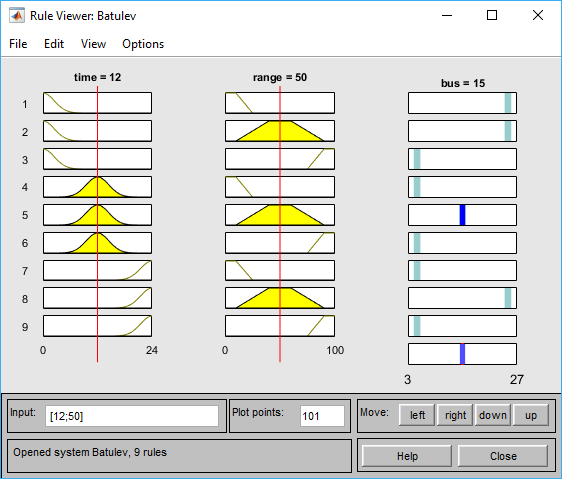


Рисунок 14 – Изображение логического вывода

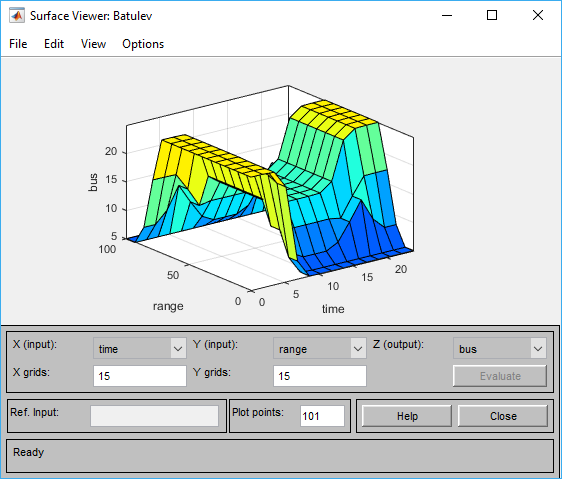


Рисунок 15 – Изображение поверхности вывода(Imp = prod, Agg = sum)

3.5. Установить метод импликации ‘prod’, а метод агрегации ‘sum’ (вручную через командное окно Matlab).

Они были установлены ранее поэтому устанавливаем в min и мах.

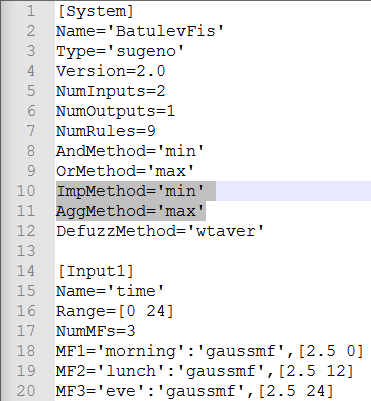


Рисунок 15.1 - BatulevFis.fis

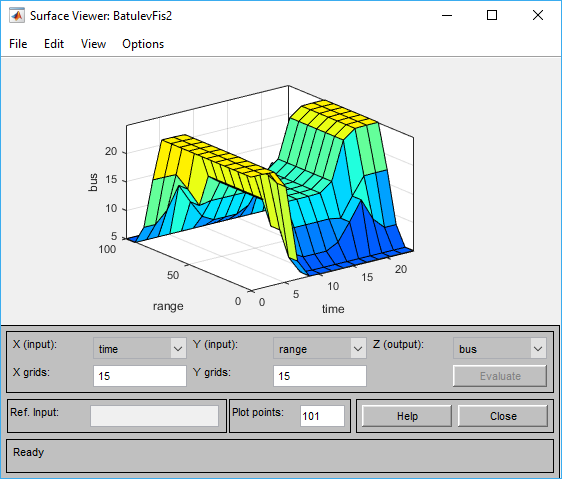


Рисунок 15.2 - поверхность вывода (Imp = min, Agg = max )

3.6. Изменяя способы определения логической связки “и” и методы дефаззификации, определить наиболее приемлемое сочетание параметров.

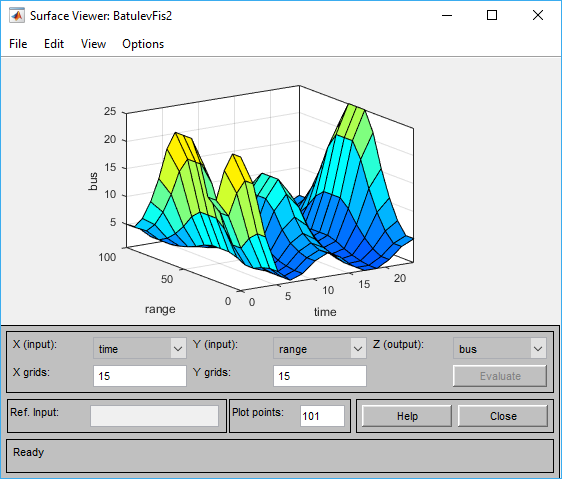


Рисунок 15.3 - поверхность вывода (and=prob,def=wtsum)

Как видно изменений нет.

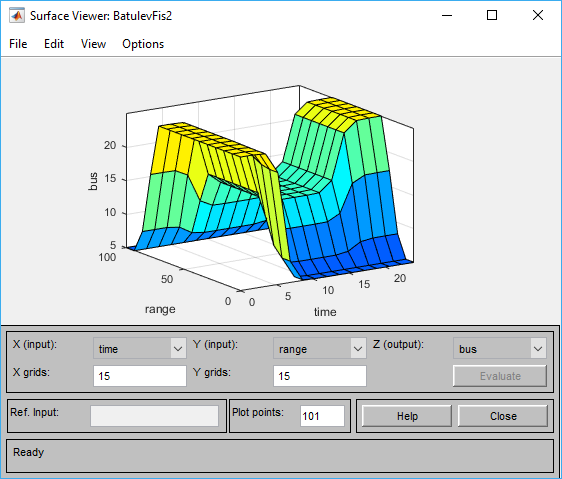


Рисунок 15.4 - поверхность вывода (and=prob,def=wtaver)

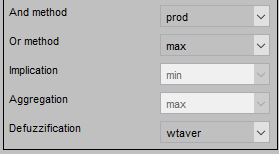


Рисунок 15.5 - Наиболее приемлемое сочетание параметров

Тема 4. Зависимость шанса спортсмена на победу на олимпиаде в зависимости от его рейтинга за год и текущей формы.

Таблица 4 - Переменная rating (диапазон значений 0-1)

|  |  |  |
| --- | --- | --- |
| Имя терма | Тип | Параметры |
| low | gaussmf | [0.2 0] |
| middle | gaussmf | [0.2 0.5] |
| high | gaussmf | [0.15 0.9] |

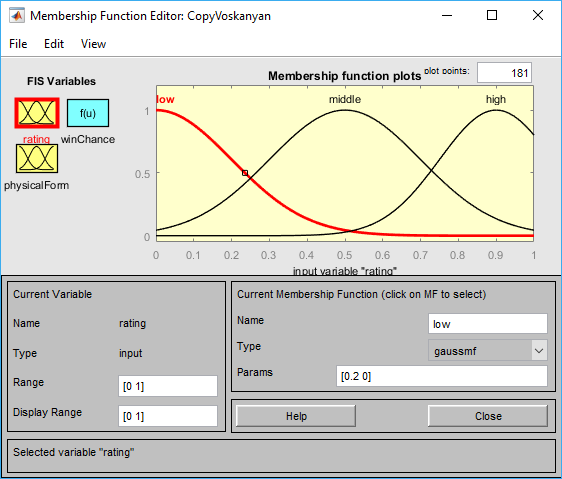


Рисунок 16 - данные о входной переменной rating

Таблица 5 – Переменная physicalForm (диапазон значений 0-1)

|  |  |  |
| --- | --- | --- |
| Имя терма | Тип | Параметры |
| bad | trapmf | [0 0 0.2 0.4] |
| good | trimf | [0.3 0.55 0.8] |
| perfect | trapmf | [0.7 0.9 1 1] |

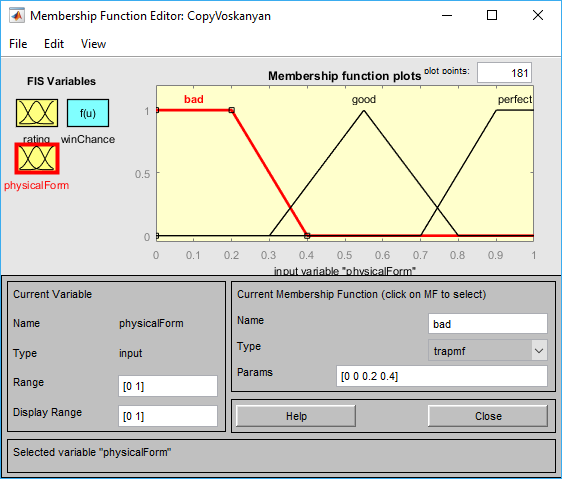


Рисунок 17 – данные о входной переменной physicalForm

Таблица 6 - Выходная переменная winChance (диапазон значений 0-1)

|  |  |  |
| --- | --- | --- |
| Имя терма | Тип | Параметры |
| minimal, nearZero, low | constant | 0.3 |
| belowTheAverage, average, aboveAverage | constant | 0.5 |
| high, nearMaximal, maximal | constant | 0.8 |

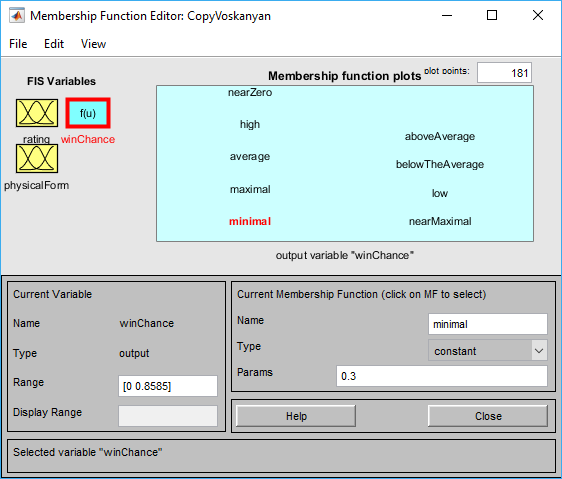


Рисунок 18 – данные о выходной переменной winChance

3.3. Задать базу правил на основе знаний о характере системы. Для связи нечётких значений входных переменных использовать связку “и”. Сформировать выборку, основывающуюся на априорных знаниях (априорную выборку), представляющую собой матрицу размером N×3, где N – число правил в базе знаний (подробнее см. приложение).

>> M=[0 0 0.3; 0 0.55 0.3; 0 1 0.5;

0.5 0 0.3; 0.5 0.55 0.5; 0.5 1 0.8;

0.9 0 0.5; 0.9 0.55 0.8; 0.9 1 0.8 ]

M =

0 0 0.3000

0 0.5500 0.3000

0 1.0000 0.5000

0.5000 0 0.3000

0.5000 0.5500 0.5000

0.5000 1.0000 0.8000

0.9000 0 0.5000

0.9000 0.5500 0.8000

0.9000 1.0000 0.8000

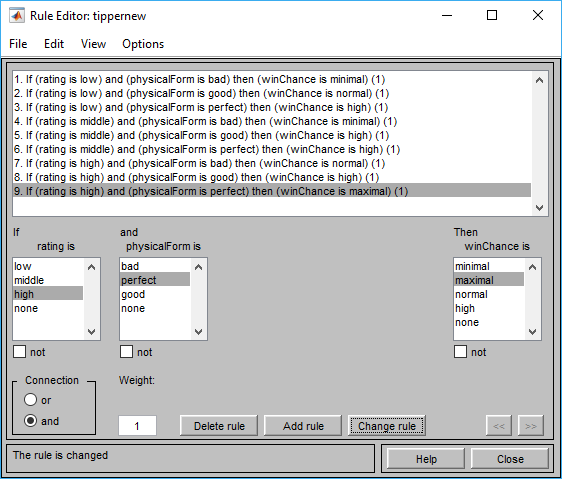


Рисунок 19 – Правила

3.4. Для исходных настроек (методы “и”, дефаззификации) получить изображения логического вывода и поверхности вывода. Результаты зафиксировать в отчёте

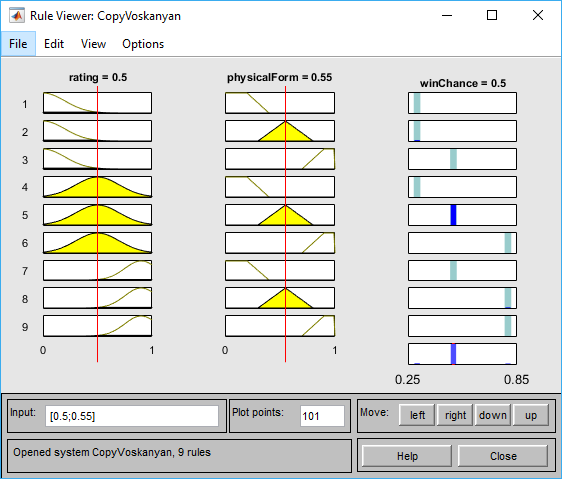


Рисунок 20 – Изображение логического вывода

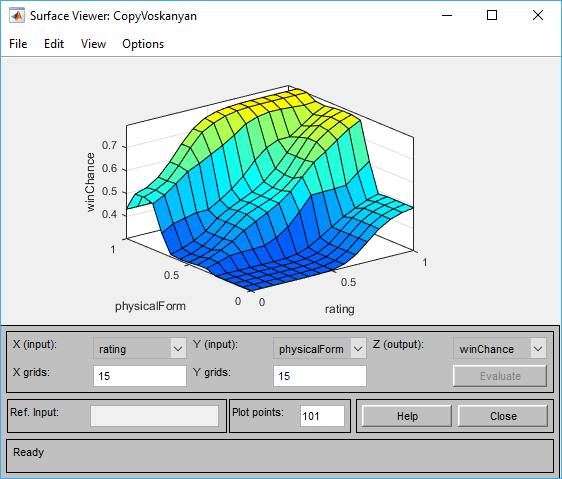


Рисунок 21 – Изображение поверхности вывода(Imp = prod, Agg = sum)

3.5. Установить метод импликации ‘prod’, а метод агрегации ‘sum’ (вручную через командное окно Matlab).

Они были установлены ранее поэтому устанавливаем в min и мах.

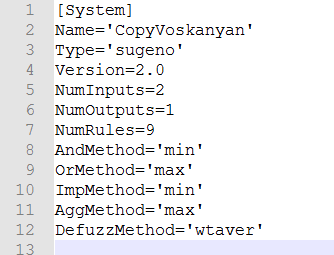


Рисунок 22 – CopyVoskanyan.fis

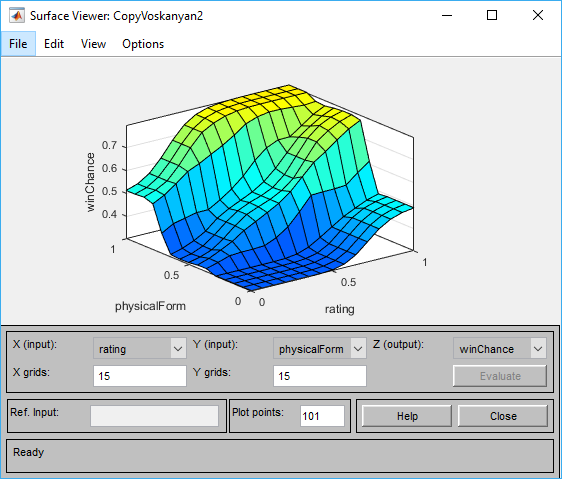


Рисунок 23 - поверхность вывода (Imp=min, Agg=max, And=min, Def=wtaver)

3.6. Изменяя способы определения логической связки “и” и методы дефаззификации, определить наиболее приемлемое сочетание параметров.

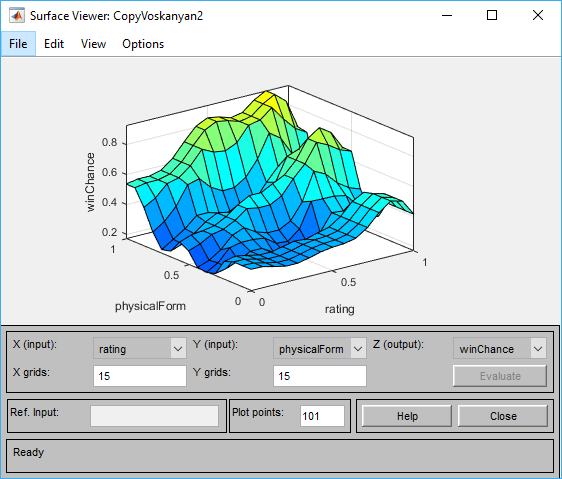
****

Рисунок 24.1 - поверхность вывода (And=min, Def=wtsum)

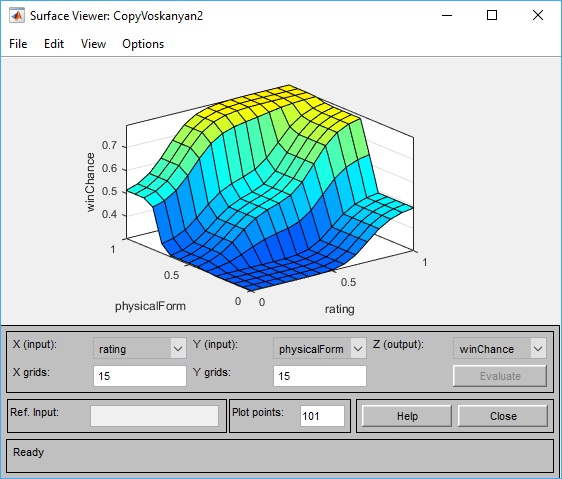
****

Рисунок 24.2 - поверхность вывода (And=prod, Def=wtaver)

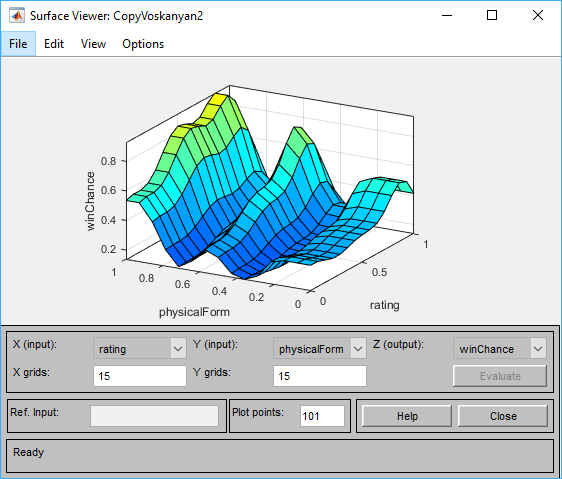
****

Рисунок 24.3 - поверхность вывода (And=prod, Def=wtsum)

Ввиду отсутствия неадекватно резких перепадов выходных значений, наиболее приемлемым сочетанием способа определения логической связки “и” и метода диффазификации является данное:

AndMethod = ‘prod’

DefuzzMethod = ‘wtaver’

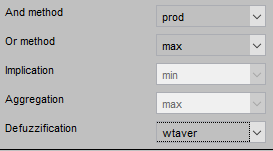


Рисунок 24.4 - Наиболее приемлемое сочетание параметров