Под аппроксимацией обычно подразумевается описание некоторой, порой не заданной явно, зависимости или совокупности представляющих ее данных с помощью другой, обычно более простой или более единообразной зависимости. Часто данные находятся в виде отдельных узловых точек, координаты которых задаются таблицей данных.

**Результат аппроксимации может не проходить через узловые точки.** Напротив, **задача интерполяции — найти данные в окрестности узловых точек.** Для этого используются подходящие функции, значения которых в узловых точках совпадают с координатами этих точек. Например, при линейной интерполяции зависимости у(х) узловые точки соединяются друг с другом отрезками прямых и считается, что искомые промежуточные точки расположены на этих отрезках.

Для повышения точности интерполяции применяют параболы (квадратичная интерполяция) или полиномы более высокой степени (полиномиальная интерполяция). Для обработки данных MATLAB использует различные функции интерполяции и аппроксимации данных.

Объект или система, полученная в результате проектирования, может быть исследована, в диапазоне функционирования, как черный ящик, на который воздействуют входными данными и анализируют отклик. Полученная зависимость описывает поведение объекта или системы и формализована в виде модели через аппроксимацию и интерполяцию данных.

**Y=f(X)**

**ЧЯ**

X Y

**Подготовка исходных данных поведения исследуемого объекта, описанного в виде функции от одной переменной.**

Для генерации исходных данных используем программу на языке MatLab, например такого вида.

% количество точек, в которых исследуем объект

% **определяется в зависимости от функции варианта в приложении №1.**

L=**200**;

% генерация значений аргумента X в диапазоне от -5 до 5 с шагом 10/L.

% диапазон **определен номером варианта в приложении №1**.

X=**-5:(10/L):5**;

x=-5:5;

% расчет точек функции, зашумленной линейным шумом

% **вводится для имитации неточности поведения**

%при измерении значения выхода

Noise=0.5\*rand(1,(L+1))\*2-1.0;

% расчет значений вектора Y зашумленного Noise

% здесь необходимо использовать функцию из приложения №1,

% **в соответствии с номером варианта**

Y=**X.^2.\*sin(2\*X+pi/2)**+Noise;

y= x.^2.\*sin(2\*x+pi/2)

% рисуем точками зашумленный график

plot(X,Y,'LineStyle','none','Marker','.','MarkerSize',5);

hold on;

% рисуем десять точек с одинаковым интервалом

plot(x,y,'o');

hold on;

**Лабораторная работа №1**

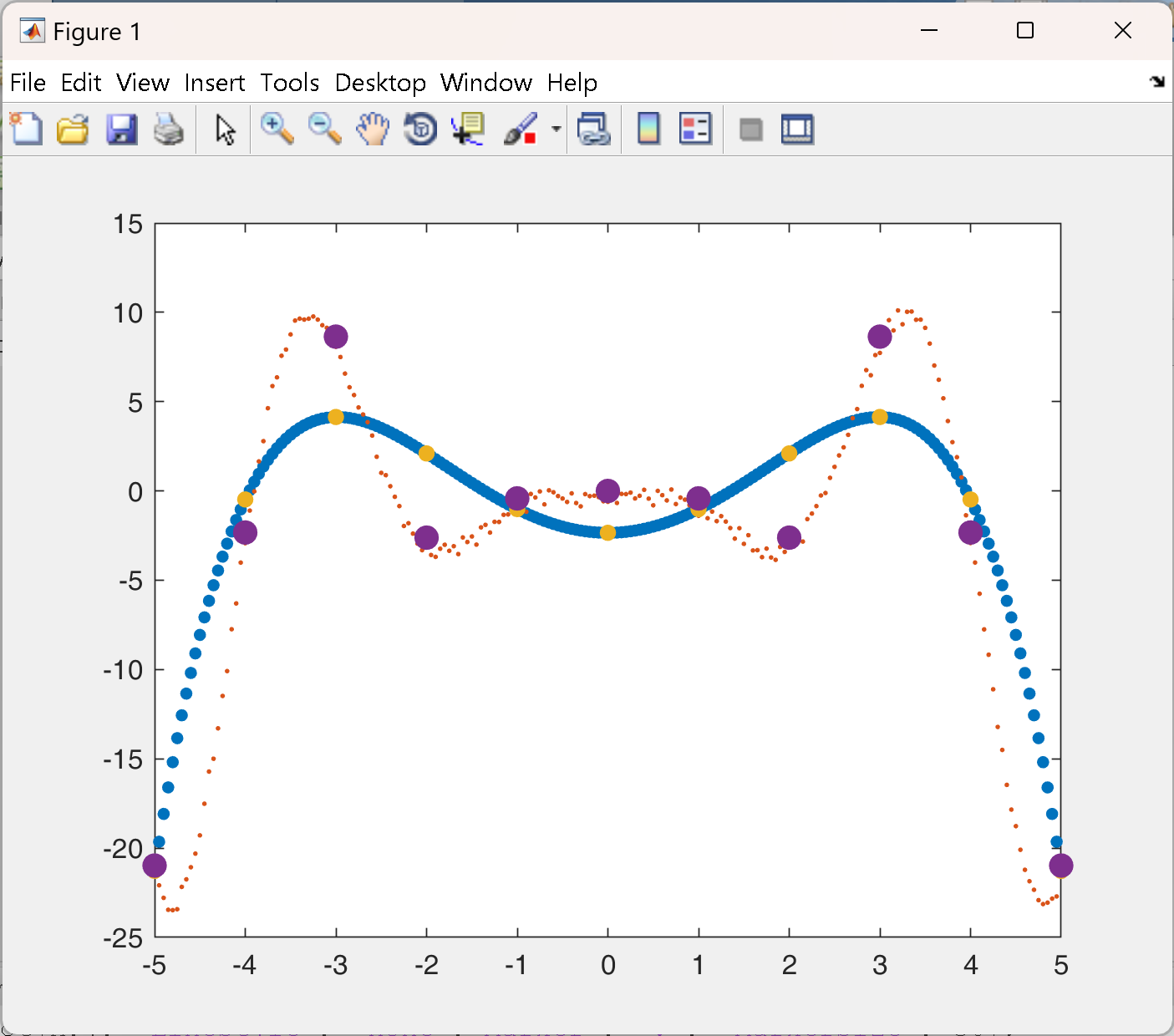
**интерполяция данных полиномами на основе метода наименьших квадратов и рядом Фурье.**

1. **Полиномиальная регрессия**

Одна из наиболее известных аппроксимаций — полиномиальная. В системе MATLAB определены функции аппроксимации данных полиномами по методу наименьших квадратов — **полиномиальной регрессии**. Это выполняет функция **polyfit(x,y,n)** которая возвращает вектор коэффициентов полинома р(х) степени п, который с наименьшей среднеквадратичной погрешностью аппроксимирует некоторую функцию у(х). Результатом является вектор-строка длиной n+1, содержащий коэффициенты полинома в порядке уменьшения степеней х и у равно n+1, то реализуется обычная полиномиальная аппроксимация, при которой график полинома точно проходит через узловые точки с координатами (х.у), хранящиеся в векторах х и у. В противном случае точного совпадения графика с узловыми точками не наблюдается.

· [p.S] = polyflt(x.y.n) — возвращает коэффициенты полинома р и структуру S для использования вместе с функцией **polyval** с целью оценивания или предсказания погрешности;

· [p.S] = polyf1t(x,y,n,mu) возвращает коэффициенты полинома р и структуру S для использования вместе с функцией polyval с целью оценивания или предска-зния погрешности, но так, что присходит центрирование (нормирование) и масштабирование х, xnorm = (х - mu(l))/mu(2), где mu(l) = mean(x) и mu(2) = std(x). Центрирование и масштабирование не только улучшают свойства степенного многочлена, получаемого при помощи **polyval**, но и значительно повышают качественные характеристики самого алгоритма аппроксимации.



**Рис. 2.** Пример использования функции polyfit

% Получаем полином 7 степени для данных исследования зависимости x от y

% p – коэффициенты полученного полинома **на диапазоне** **[-5,5], 10 точек**

p=polyfit(x,y,7);

% Используем полиномом –p для расчета **Y\_NEW** **на диапазоне 200 точек**

Y\_NEW =polyval(p, X);

% Используем полиномом –p для расчета **y на диапазоне 10 точек**

y\_new =polyval(p, x);

% Визуализация результатов аппроксимации на исходном диапазоне

plot(X, Y\_NEW, 'LineStyle', 'none','Marker', '.', 'MarkerSize', 15);

plot(X,Y,'LineStyle','none','Marker','.','MarkerSize',5);

plot(x, y\_new, 'LineStyle', 'none','Marker', '.', 'MarkerSize', 20);

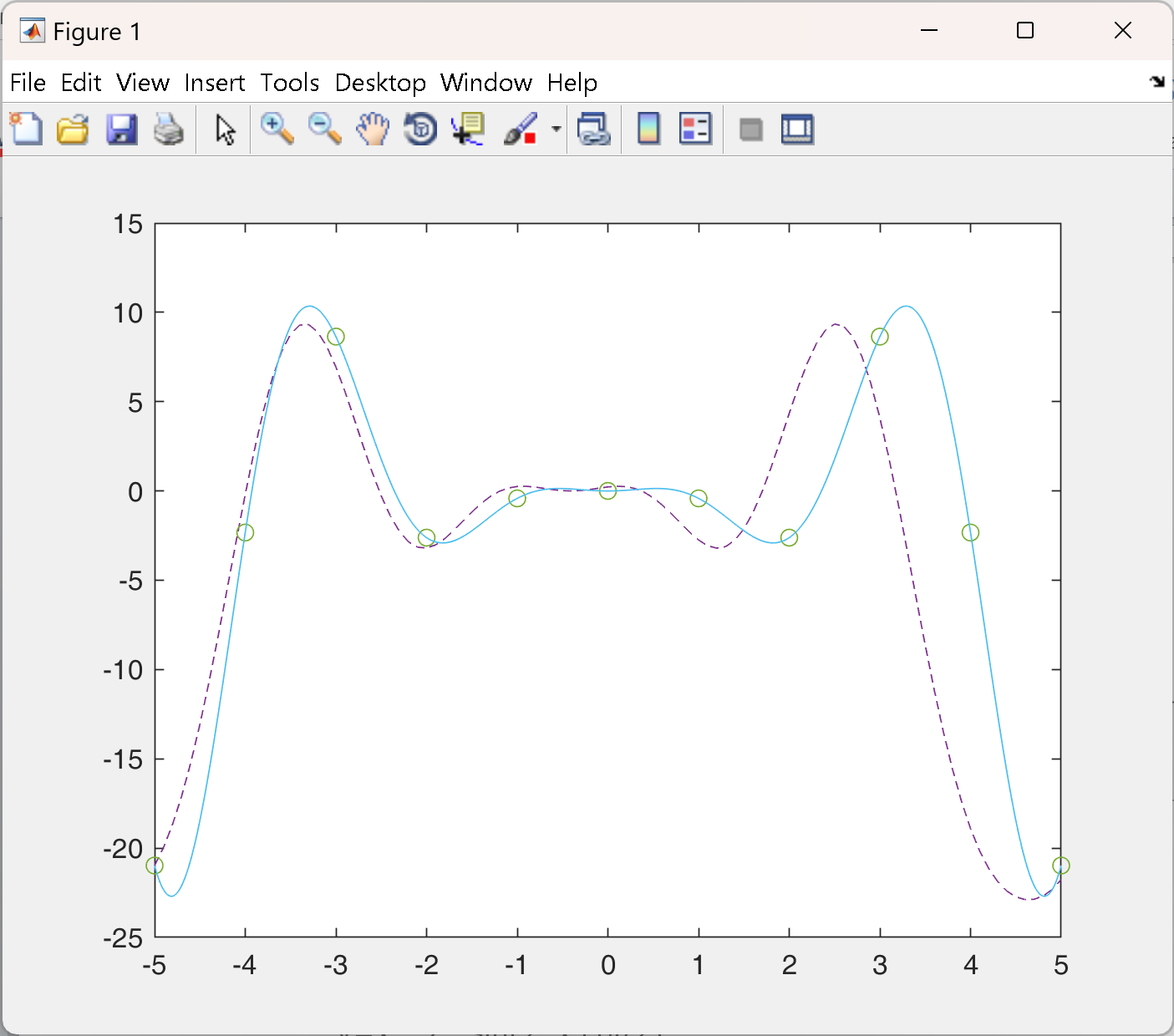
plot(x,y, 'LineStyle', 'none','Marker', '.', 'MarkerSize', 30);

График аппроксимирующего полинома седьмой степени на рис. 2. показан красным цветом, а точки исходной зависимости синим. К сожалению, при степени полинома свыше 5 погрешность полиномиальной регрессии (и аппроксимации) сильно возрастает и ее применение без центрирования и масштабирования становится рискованным. Обратите внимание на то, что при полиномиальной регрессии узловые точки не ложатся точно на график полинома, поскольку их приближение к нему является наилучшим в смысле минимального среднеквадратического отклонения.

**Интерполяция периодических функций рядом Фурье**

Под интерполяцией обычно подразумевают вычисление значений функции *f(x)*в промежутках между узловыми точками. Линейная, квадратичная и полиномиальная интерполяция реализуются при полиномиальной аппроксимации. А вот для периодических (и особенно для гладких периодических) функций хорошие результаты может дать их интерполяция тригонометрическим рядом Фурье. Для этого используется следующая функция:

· interpft(x,n) — возвращает вектор у, содержащий значения периодической Функция yp = interpft(y, n) возвращает одномерный массив чисел, который является периодической функцией, определенной в n точках и аппроксимирующей одномерный массив y. Если length(x) = m, а интервал дискретности dx, то интервал дискретности для y определяется по формуле dy = dx \* m/n, причем n всегда превышает m.

****

**Рис. 3.** Пример использования функции interpft. Пунктир апроксимация, линия исходная функция.

% Создаем 11 значений на диапазоне [-5,5]

x=-5:5;

% Исходная функция

y=x.^2.\*sin(2\*x+pi/2);

%Создаем 101 значение на диапазоне [-5,5]

x1=-5:0.1:5;

%Интерполяция рядом Фурье

y1=interpft(y,101);

x2=-5:0.01:5;

%Расчитываем 1001 значение y2

y2=x2.^2.\*sin(2\*x2+pi/2);

plot(x1,y1, '--');

hold on;

plot(x,y, 'o' ,x2, y2);

Рис. 3. иллюстрирует эффективность данного вида интерполяции на примере функции, которая представляет собой сильно искаженную синусоиду.

**Лабораторная работа №2**

**Построение нечеткой аппроксимирующей системы**

**Цель работы:** изучить основные понятия и определения теории нечетких множеств, операции с ними, назначение и возможности пакета нечеткой логики Fuzzy Logic Matlab построить нечеткую аппроксимирующую систему для заданной функции.

**Пример аппроксимации функции с помощью Fuzzy Inference System Editor.**

1. 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 рис. 4.

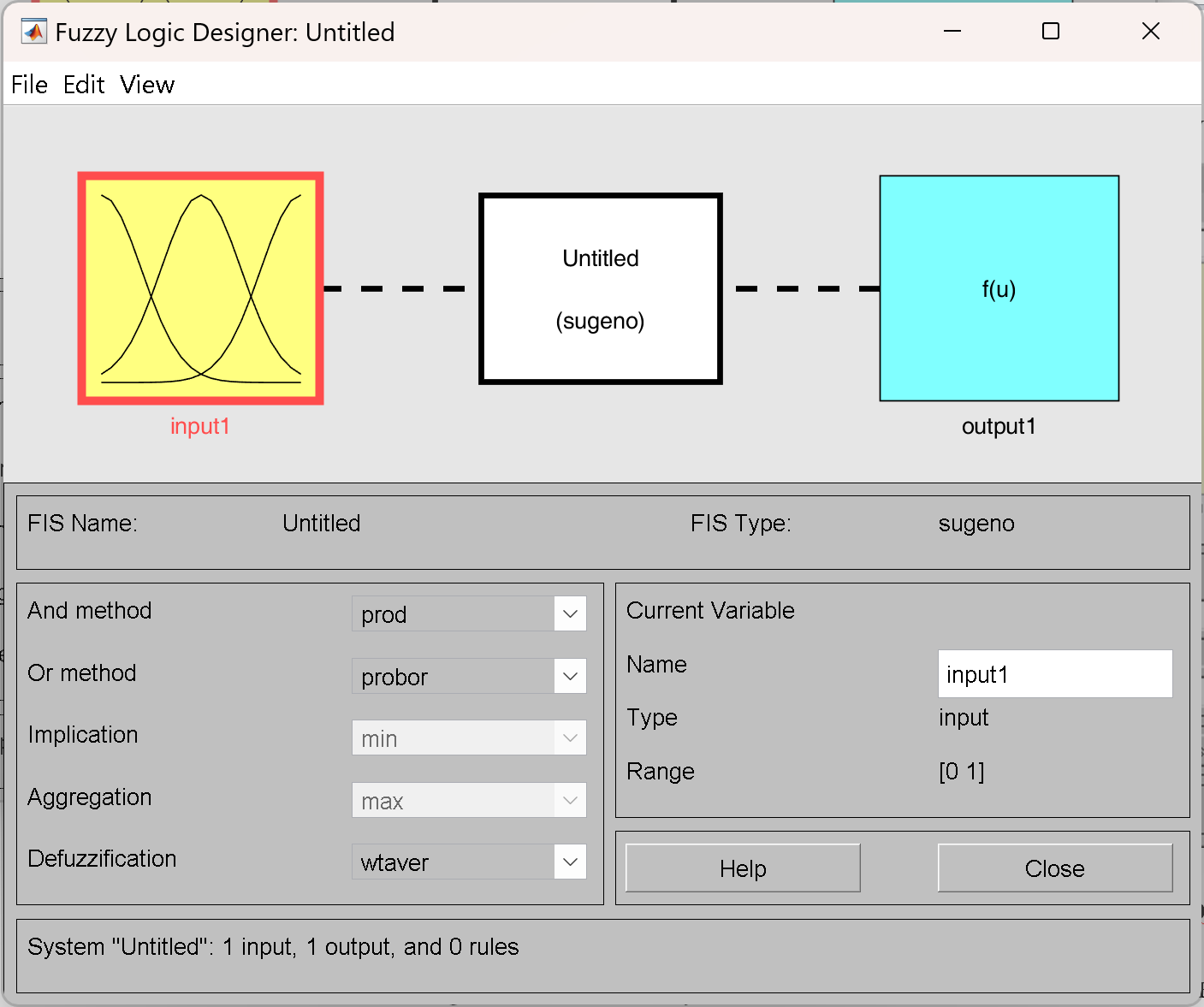


Рис. 4. Вид окна FIS Editor

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).

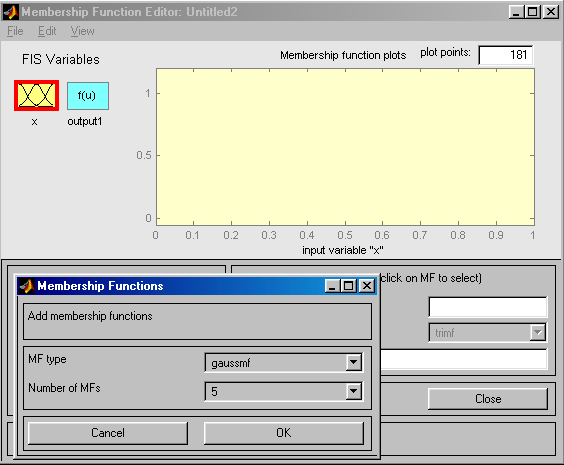
View -переход к дополнительному инструментарию.

Пример:

Для аппроксимации функции y= x2\*sin(2\*x+pi/2) составим таблицу 11 значений х и у на диапазоне [-5,5]:

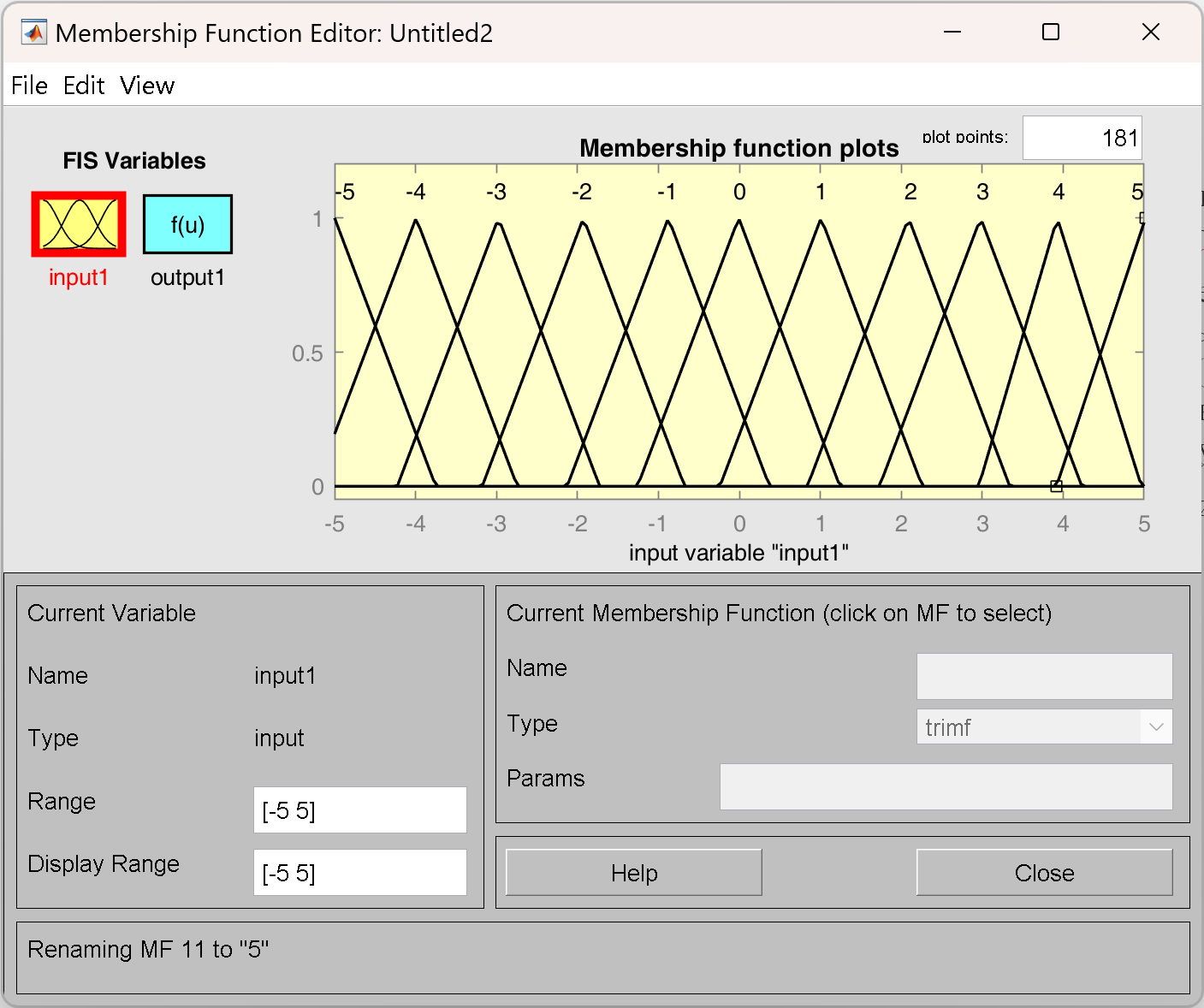
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 |
| y | -20,97 | -2,32 | 8,64 | -2,61 | -0,41 | 0 | -0,41 | -2,61 | 8,64 | -2,32 | -20,97 |

1. /В позиции меню File выбирается опция **New Sugeno FIS** (новая система типа Sugeno), при этом в блоке, отображаемом белым квадратом в верхней части окна редактора, появится надпись Untitled2 (**sugeno**).
2. Щелчком левой кнопкой мыши по блоку, озаглавленному inputl (вход1) выбирается редактор входных переменных. В правой части редактора, в поле, озаглавленном Name (Имя), вместо inputl вводится обозначение аргумента, т.е. х.
3. Двойной щелчок по этому блоку вызывает окно редактора функций принадлежности - Membership Function Editor (см. рис. 5). В меню Edit данного редактора н выбирается опция Add MF» (AddMejBbership Funcions - Добавить функций принадлежности). При этом появится диалоговое окно, позволяющее задать тип (MF type) и количество {Number of MFs) функций принадлежности (в данном случае все относится « входному сигналу, т.е. к переменной х). Выбираются гауссовы функции принадлежности, а их количество задается равным 11 - по числу значений аргумента в таблице.



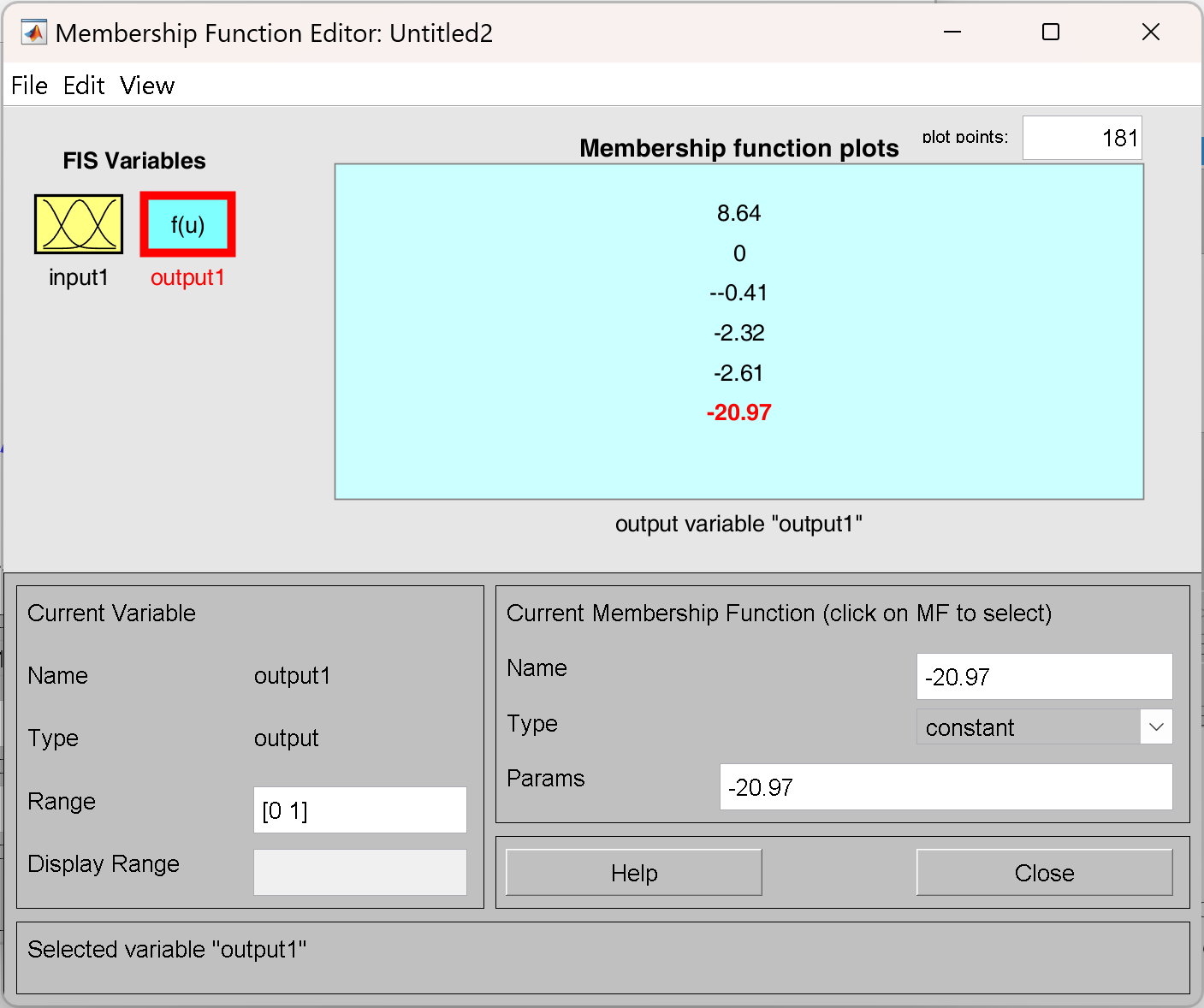
**Рис. 5. Окно редактора функций принадлежности - Membership Function Editor**

1. В поле Range (Диапазон) устанавливается диапазон изменения х от -5 до 5,-т.е. диапазон, соответствующий таблице. После этого произойдет соответствующее изменение диапазона в поле Display Range (Диапазон дисплея).
2. Изменяются значения параметров заданных функций принадлежности, например, необходимо, чтобы ординаты максимумов этих функций совпадали с заданными значениями аргумента х. Числовые значения в поле Params (Параметры) – соответствуют размаху кривой и положению ее центра. Поле Name задает имя функции. Полученный результат приведен на рис.6.

****

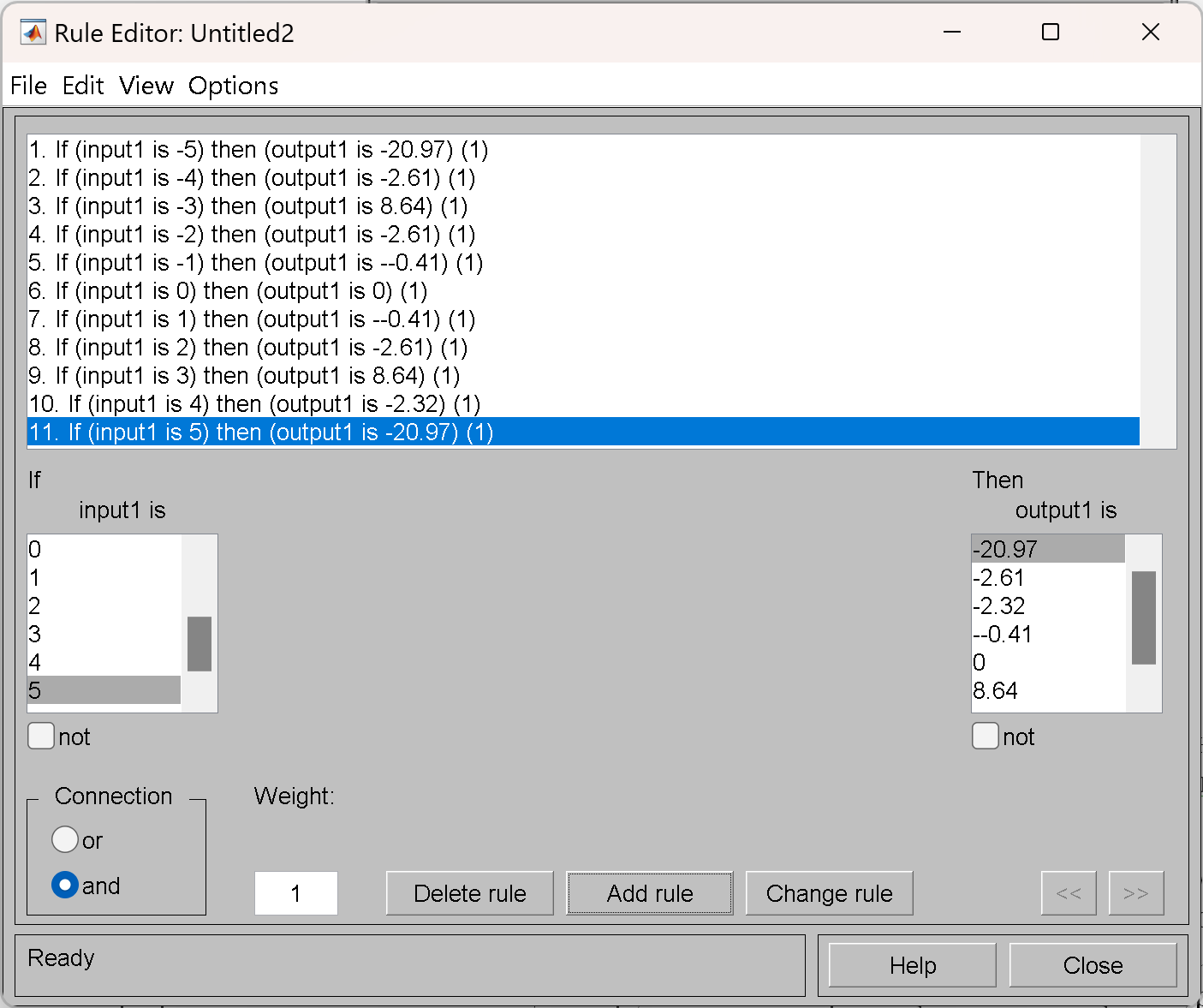
**Рис.6. Заданные функций принадлежности**

1. Редактирование выходных функций принадлежности осуществляется двойным щелчком кнопкой мыши по голубому квадрату (блоку), озаглавленному outputl (выход1).
2. В позиции меню Edit выбирается опция Add MFs. Появляющееся затем диалоговое окно позволяет задать в качестве функций принадлежности только линейные (linear) или постоянные (constant) - в зависимости от того, какой алгоритм Sugeno (1-го или 0-го порядка) выбирается. В рассматриваемой задаче необходимо выбрать постоянные функции принадлежности с общим числом 4 (по числу различных значений у в таблице).
3. Изменяются имена функций принадлежности (их графики при использовании алгоритма Sugeno для выходных переменных не приводятся), например, имена могут задаваться так же, как и их соответствующие числовые значения у, из таблицы; одновременно эти же числовые значение вводятся в поле Params (рис.7).

****

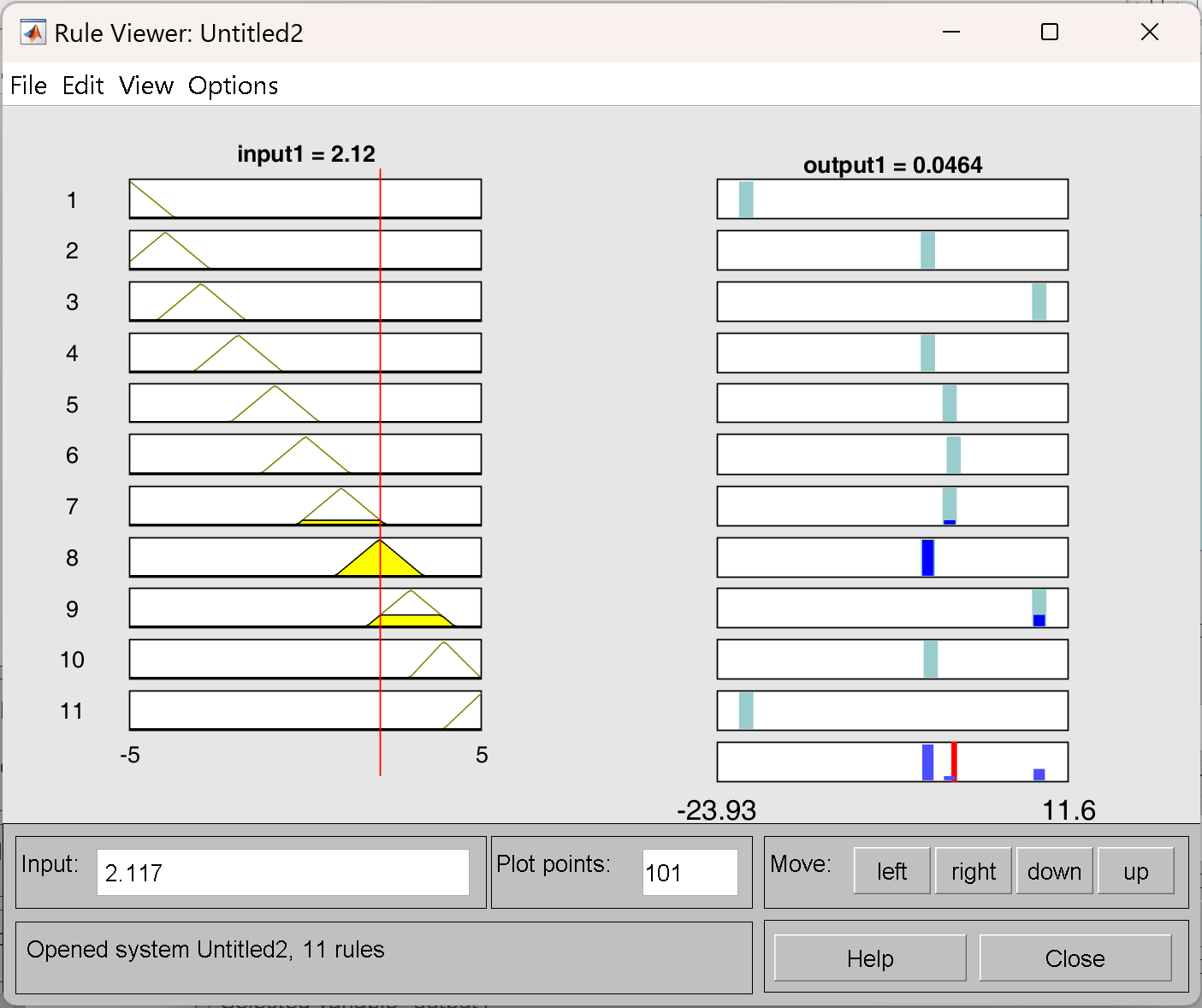
**Рис.7. Редактирование выходных функций**

1. Двойной щелчок по среднему (белому) блоку вызывает редактор правил (Rule Editor). При вводе каждого правила необходимо обозначить соответствие между каждой функцией принадлежности аргумента х и числовым значением у. Кривая, обозначенная function-1, соответствует х=-1, т.е. у=1. В левом поле (с заголовком х is) выбирается function-1, а в правом 1 и нажимается кнопка Add rule (Добавить правило). Аналогично добавляются правила для всех других значений х. Построение системы закончено, и можно начать эксперименты по ее исследованию.



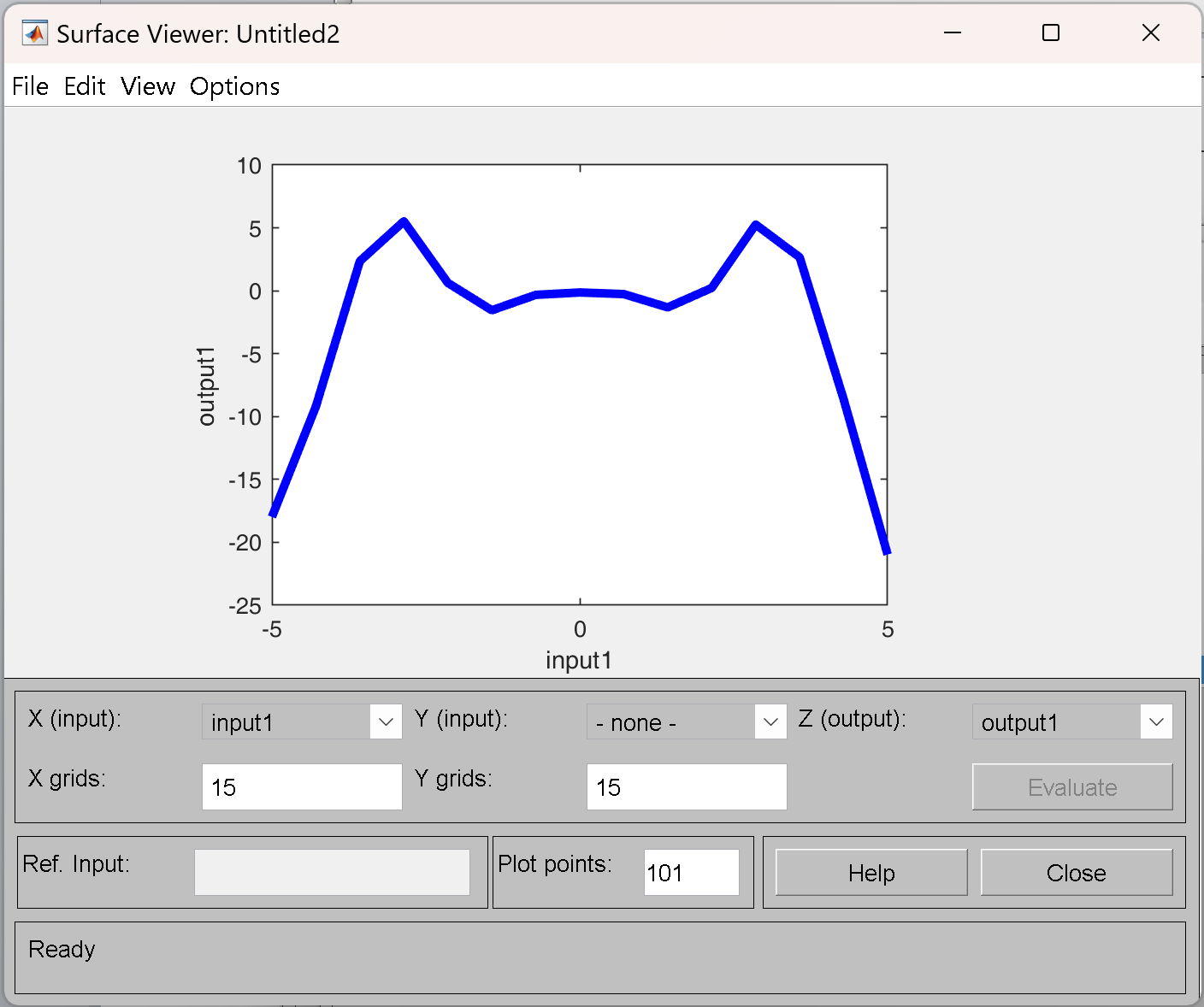
**Рис.8. Редактор правил (Rule Editor)**

1. C помощью пунктов Edit membership functions и Edit rules меню View можно возвращаться к выше рассмотренным программам - редакторам функций принадлежности и правил (то же можно сделать и нажатием клавиш Ctrl+2 или Ctrl+3).
2. View rules (Просмотр правил). В правой части окна в графической форме представлены функции принадлежности аргумента х, в левой - переменной выхода у с пояснением механизма принятия решения. Красная вертикальная черта, пересекающая графики в повой части окна, которую можно перемещать с помощью курсора, позволяет изменять значения переменной входа (это же можно делать задавая числовые значения в поле Input(Вход)), при этом соответственно изменяются значения у в правой верхней части окна. Таким образом, с помощью построенной модели и окна просмотра правил можно решать задачу интерполяции, т.е. задачу, решение которой в требовалось найти. Изменение аргумента путем перемещения красной вертикальной линии очень наглядно демонстрирует, как система определяет значения выхода.



**Рис.9. Просмотр правил (View rules)**

1. View surface (Просмотр поверхности отклика(выхода)). В данном примере поверхностью отклика является кривая у=х2 Как видно, полученная кривая весьма приближенно напоминает параболу. Это вызвано тем, что число экспериментальных точек невелико, да и параметры функций принадлежности (для х) выбраны скорее всего неоптимальным образом.



**Рис.10. Просмотр поверхности отклика(выхода) (View surface)**

С помощью вышеуказанных программ-редакторов на любом этапе проектирования нечеткой модели в нее можно внести необходимые коррективы, вплоть до задания какой-либо особенной пользовательской функции принадлежности. 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);
* приведение к четкости - дискретным вариантом центроидного метода (взвешенным средним, wtaver).

Используя соответствующие поля в левой нижней части окна FIS-редактора, данные опции можно, при желании, изменить.

**Порядок выполнения работы**

В ходе выполнения лабораторной работы необходимо выполнить аппроксимацию заданной функции используя Fuzzy Logic - редактор нечеткой системы вывода.

.

1. Ознакомиться с теоретической частью и примером.
2. **Построить график функции в соответствии с номером варианта.**
3. Построить нечеткую аппроксимирующую систему используя алгоритм Сугено для заданной функции и получить поверхность отклика.
4. Скорректировать количество и тип задаваемых функций принадлежности, стараясь добиться максимального приближения полученной поверхности отклика к исходному графику.
5. Оценить точность аппроксимации.

**Лабораторная работа №3**

**АППРОКСИМАЦИЯ ДАННЫХ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ**

**Цель работы.**

Создать сеть прямого распространения с одним скрытым слоем, предназначенную для аппроксимации массива экспериментальных данных и построение нейросетевой модели.

**Подготовка данных для обучения.**

Для генерации обучающей выборки используем программу на языке MatLab, например такого вида.

% количество точек

L=200;

% генерация значений аргумента X в диапазоне от -5 до 5 с шагом 10/L.

X=-5:(10/L):5;

% расчет точек функции, зашумленной линейным шумом

Noise=0.5\*rand(1,(L+1))\*2-1.0;

% расчет значений вектора Y зашумленного Noise

Y=X.^2.\*sin(2\*X+pi/2)+Noise;

**Создание и обучение ИНС для аппроксимации функции**

Создадим искусственную нейронную сеть с именем **Net** для аппроксимации сгенерированных данных при помощи команды **feedforwardnet** (сеть прямого распространения с одним скрытым слоем). Запуск обучения командой **train.** Готовую сеть тестируют на новом диапазоне значений, используя функцию **sim**.

% Включаем режим добавления новых графиков

hold on;

% Вывод на экран обучающей выборки

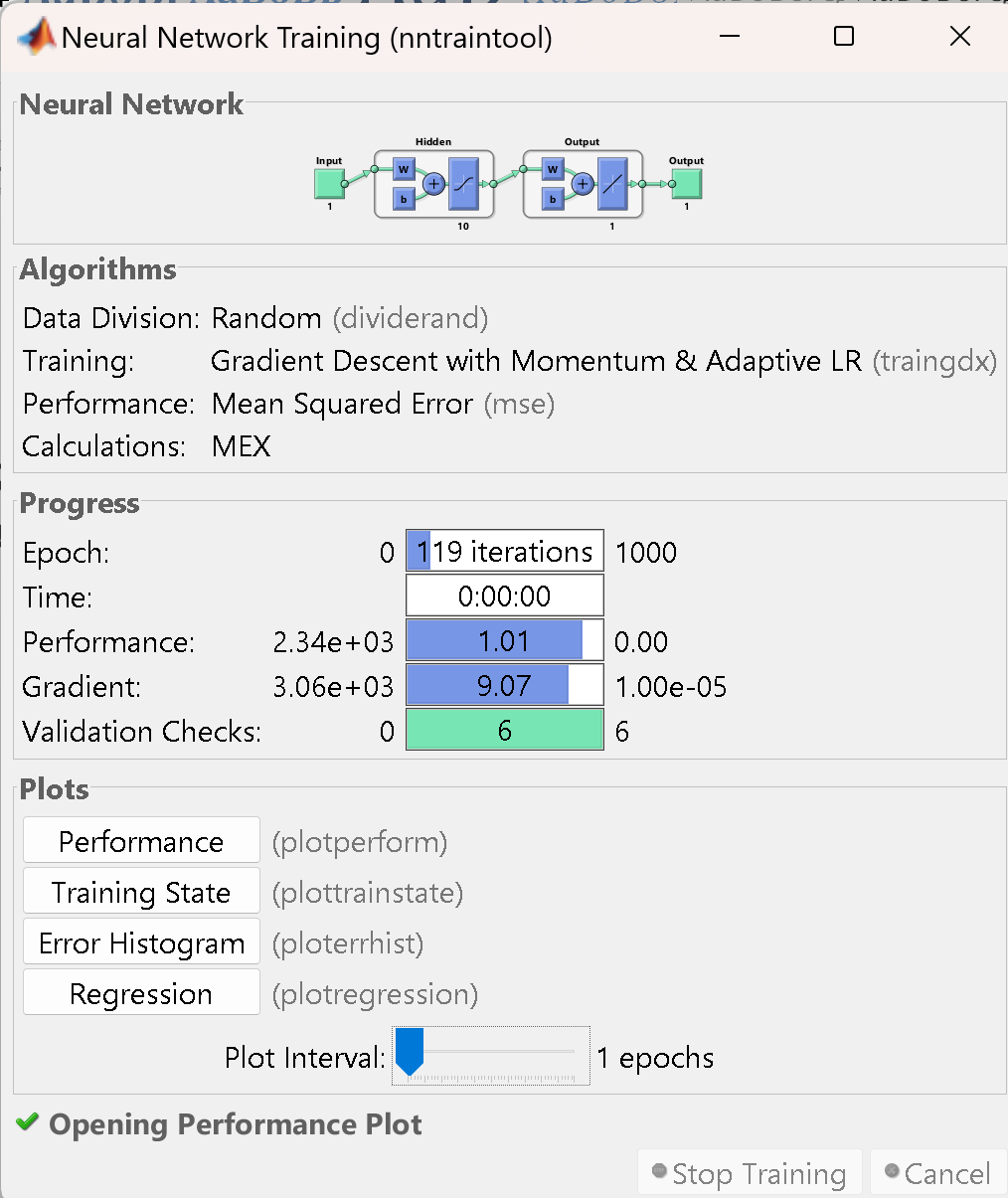
plot(X,Y,'LineStyle','none','Marker','.','MarkerSize',5);

% Создание ИНС прямого распространения. Сеть имеет один скрытый слой с 10 нейронами и выходной слой. алгоритм обучения 'traingdx')

Net = feedforwardnet(10,'traingdx');

% обучение нейросети на данных Y от X.

Net=train(Net,X,Y);



**Рис.11. Интерфейсная форма нейросети**

В процессе обучения **MatLab** выводит интерфейсную форму Нейросети. Архитектура системы – количество слоев **(2)** и нейронов в них **10 и 1**, типы функций активации **('tansig' и 'purelin'),** алгоритм обучения (модифицированный градиентный спуск), тип функции ошибки среднеквадратическая. Кнопка **Performance** содержит график динамики погрешности сети по тактам обучения. Останов процесса обучения по умолчанию происходит, когда значение функционала ошибки ИНС возрастает шесть эпох подряд.

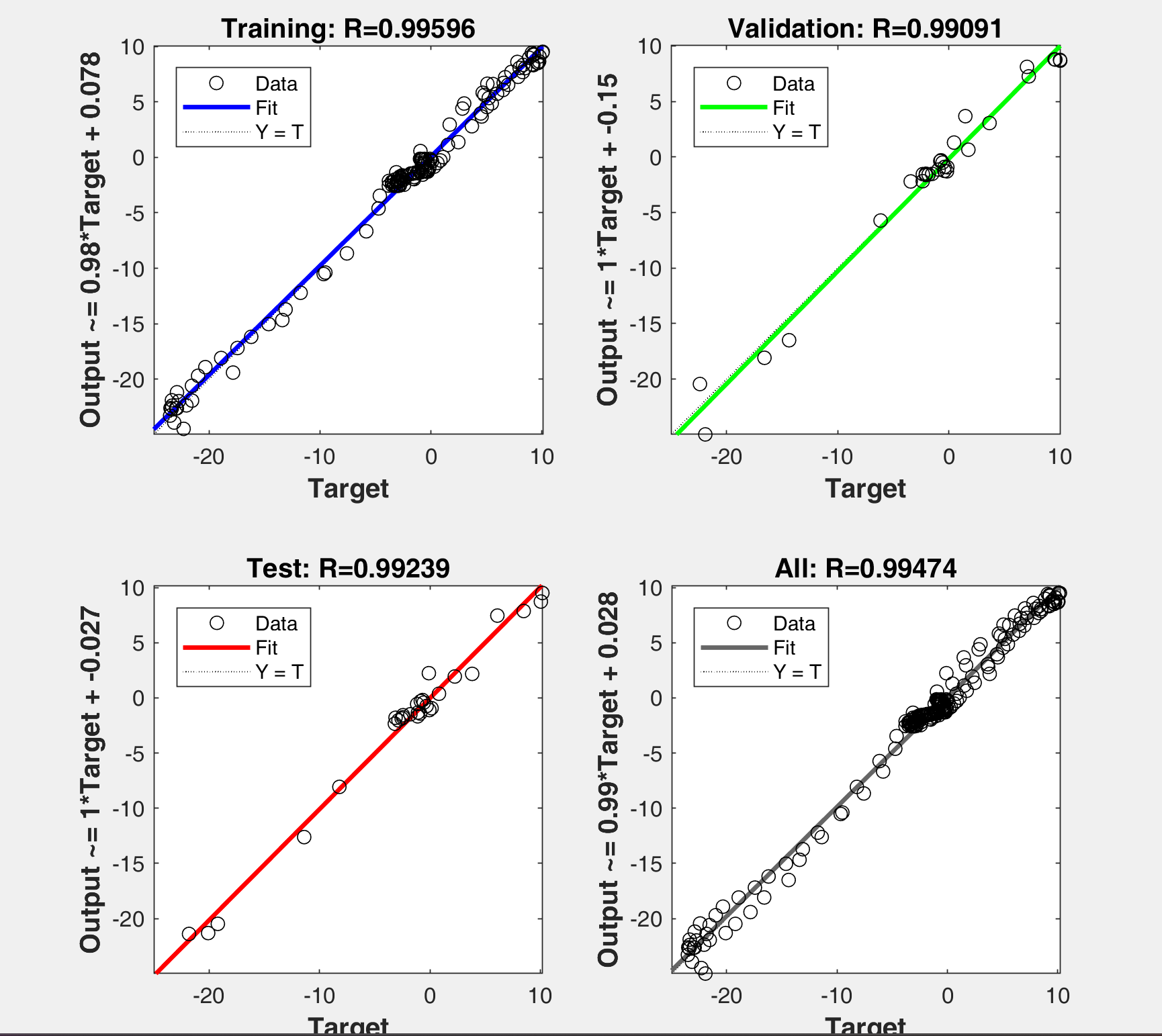
В данном примере результат обучения сети адекватен:

1. Заключительная среднеквадратичная ошибка (**MSE**) мала.
2. Ошибка проверочного (**Validation**) и тестового (**Test**) наборов

имеют подобные характеристики.

1. Переобучения не произошло (после точки остановки происходит лишь увеличение MSE проверочного набора до 101-й эпохи).

На графике видно что в ближайшей окрестности (порядка 10 % диапазона обучающей выборки) ИНС обладает экстраполирующей способностью воспроизводя поведение исходной функции. Для оценки результата обучения ИНС является построение функций регрессии выходных значений ИНС (**Output**) от целевых значений (**Target**), которые были заданы в обучающей выборке. Диаграмма регрессионного анализа вызывается кнопкой **Regression**.



**Рис.12. Функции регрессии - вход и цель**

Если построенная ИНС аппроксимирует исходные данные неудовлетворительно, улучшить решение можно попытаться с помощью следующих приемов:

1. Изменение структуры сети – увеличение числа слоев и числа нейронов;
2. Увеличение размера обучающей выборки;
3. Увеличение размерности пространства входных параметров,

если таковое имеется;

1. Изменение алгоритма обучения нейронной сети.

**Порядок выполнения работы**

1. Для заданной функции,из раздела индивидуальных заданий, построить ее табличные.
2. Настроить нейронную сеть для аппроксимации этих табличных значений, используя обучающую выборку и подбирая количество нейронов в скрытом слое
3. Вывести на график табличные значения и аппроксимированную функцию.
4. Оценить значение погрешности результата аппроксимации.

**Лабораторная работа №4**

**Построение нейронечеткой модели.**

**Цель работы.**

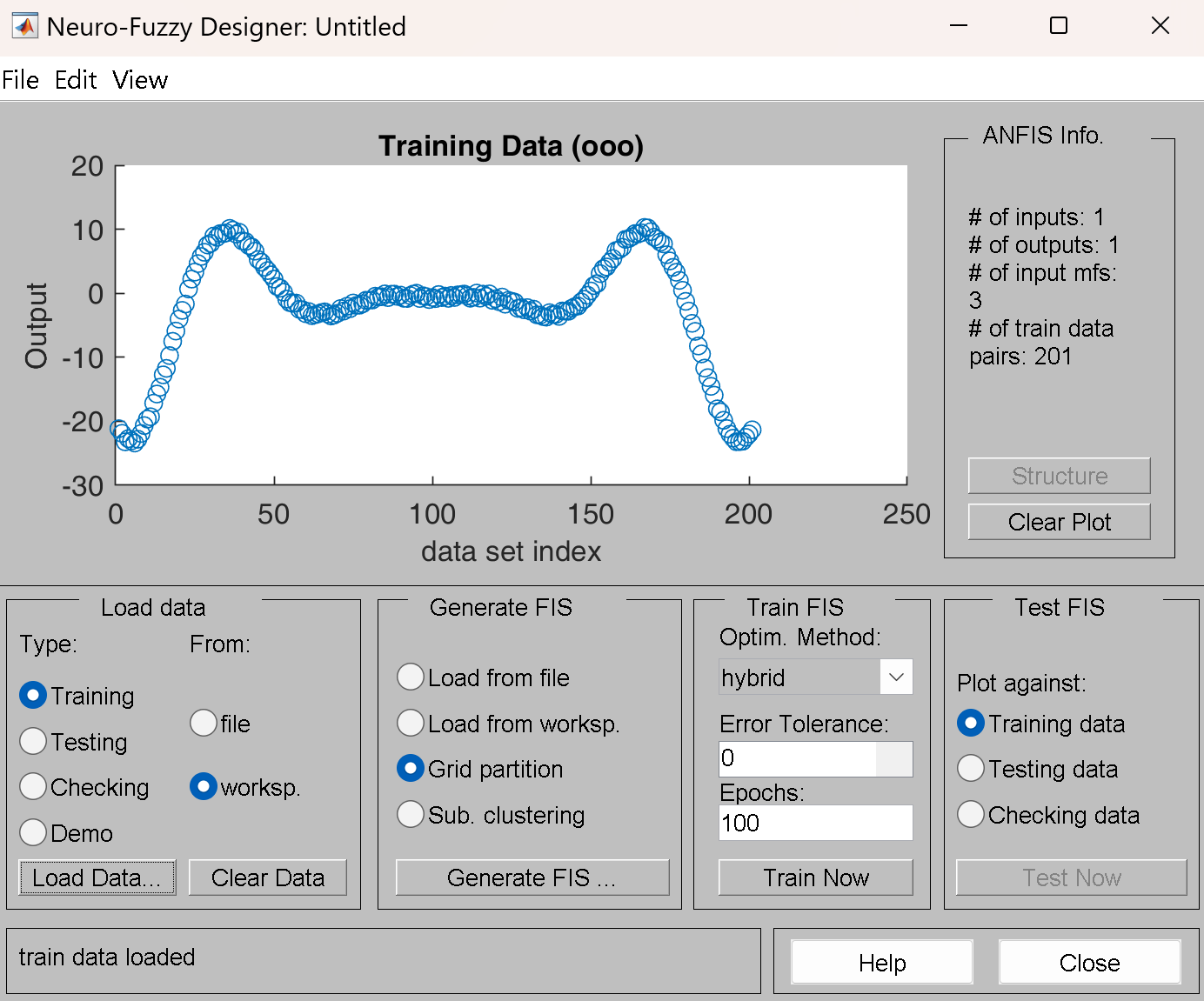
Построить модель, на основе экспериментальных данных с использованием нечеткой нейронной сети, сконструированной в ANFIS редакторе среды MatLab. Оценить качество модели.

**Порядок выполнения работы**

Для работы с ANFIS редактором необходимо подготовить исходные данные в виде обучающей выборки исследуемого процесса в 200 точках и тестовую выборку для проверки модели.

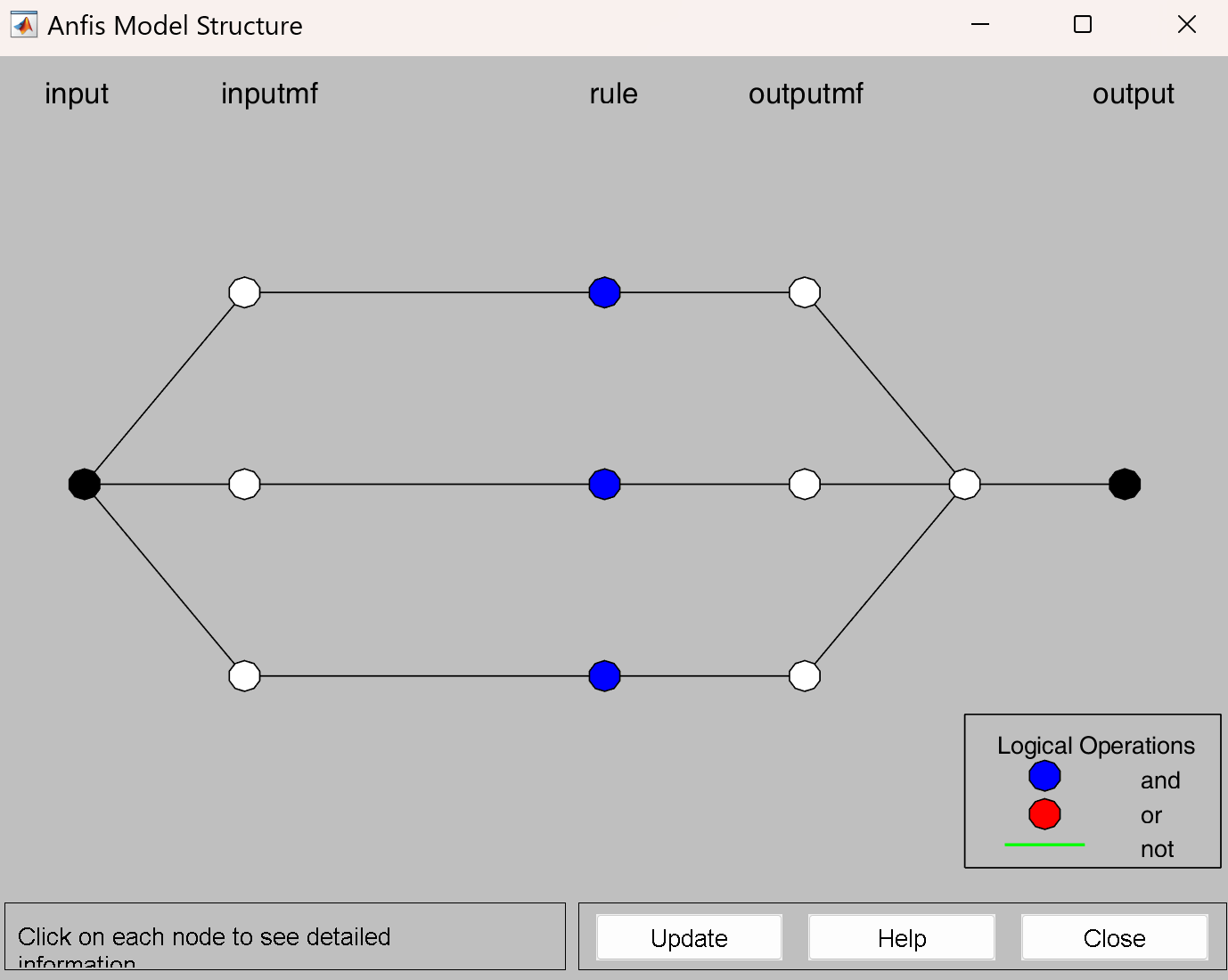
Командой anfisedit загружаем ANFIS-редактор, в котором сгенерируем и

обучим нечеткую систему типа Сугено.



**Рис.13. Интерфейс ANFIS редактора, исходная функция**

1. В области **Load data, type – Training** имя массива данных, на котором будет производиться обучение нечеткой нейросети. Тип загрузки **файл** или **рабочее пространство MatLab**. Аналогично загружаются тестовые и проверочные массивы данных.
2. После подготовки исходных данных настраиваем нечеткую систему. Выбираем **File -> New Fis -> Sugeno.** Появится окно нечеткой системы с одним входом и одним выходом, система правил не заполнена. Число нечетких функций входа выхода равно трем, это чистая, необученная нечеткая система.
3. Следующий этап – генерация и настройка нечеткой системы. Он заключается в выборе опции **Grid partition (**равномерное разбиение**)** в окне **Generate FIS**  и запуске генерации кнопкой  **Generate FIS.**

****

**Рис.14. Нейронечеткая сеть.**

1. После создания нечеткой нейронной сети в окне **Train FIS** можно обучить сеть настроив параметры обучения – количество циклов обучения **Apochs** и предельный размер ошибки, при которой обучение остановится **Error Tolerance**. Нажав кнопку Train Now, увидим процесс обучения сети и уменьшение ошибки сети на исходных данных.
2. Окно **Test Fis** предназначено для графической визуализации результатов обучения сети на обучающей выборке, тестовых примерах и контрольной выборке, из окна **Load data.**
3. В окне Anfis Info можно визуализировать полученную сеть.
4. В массиве **Outdate=Evalfis(testdate,fis);** будут хранится данные полученные в результате работы сети **fis** обработавшей тестовые данные **testdate.**
5. Постройте график, оценки точности аппроксимации данных **Y** от **Outdate**. Оцените точность модели.

**Индивидуальные задания**

**1. (1,85–х)\*cos(3,5x–0,5), x∈[–10,10]**

**2. cos(exp(x))/sin(ln(x)), x∈[2,4]**

**3. sin(x)/x2, x∈[3.1,20]**

**4. sin(2x)/x2, x∈[–20,–3,1]**

**5. cos(2x)/x2, x∈[–20,–2,3]**

**6. (x–1)cos(3x–15), x∈[–10,10]**

**7. ln(x)cos(3x–15), x∈[1,10]**

**8. cos(3x–15)/abs(x)=0, x∈[–10,–0.3), (0.3,10] x∈[–0.3,0.3]**

**9 cos(3x–15)\*x, x∈[–9,6, 9,1]**

**10. sin(x)/(1+exp(–x)), x∈[0,5,10]**

**11. cos(x)/ (1+exp(–x), x∈[0,5,10]**

**12. (exp(x)–exp(–x))cos(x)/ (exp(x)+exp(–x)), x∈[–5,5]**

**13. (exp(–x)–exp(x))cos(x)/ (exp(x)+exp(–x)), x∈[–5,5]**

**14. cos(x–0,5)/abs(x) x∈[–10,0),(0,10], min**

**15. cos(2x)/abs(x–2) x∈[–10,2),(2,10], max.**