

Федеральное агентство железнодорожного транспорта  
Уральский государственный университет путей сообщения

Кафедра «Мехатроника»

**В. С. Тарсян**

**ПАКЕТ  
FUZZY LOGIC TOOLBOX  
FOR MATLAB**

Учебное пособие  
по курсу «Методы искусственного интеллекта»  
для студентов специальности 220401 — «Мехатроника»  
направления 220400 — «Мехатроника и робототехника»

Екатеринбург  
Издательство УрГУПС  
2013

УДК 004.89, 007.52  
Т19

**Тарасян, В. С.**

Т19 Пакет Fuzzy Logic Toolbox for Matlab : учеб. пособие / В. С. Тарасян. — Екатеринбург : Изд-во УрГУПС, 2013. — 112 с.  
ISBN 978-5-94614-248-9

Учебное пособие составлено в соответствии с учебным планом по дисциплине «Методы искусственного интеллекта» для студентов специальности 220401 — «Мехатроника» направления 220400 — «Мехатроника и робототехника».

Дано описание пакета прикладных программ Fuzzy Logic Toolbox for MATLAB. Приведены подробные рекомендации по работе с пакетом в интерактивном и командном режимах. Дополнительно описаны встроенные функции пакета. Пособие предназначено для студентов, обучающихся специальности «Мехатроника».

УДК 004.89, 007.52

*Печатается по решению  
редакционно-издательского совета университета*

*Автор:* В.С. Тарасян, доцент кафедры «Мехатроника»,  
канд. физ.-мат. наук, УрГУПС

*Рецензент:* Ю.Ф. Долгий, профессор кафедры механики  
и математического моделирования ИМКН УрФУ,  
д-р физ.-мат. наук, профессор

---

**Тарасян Владимир Сергеевич**

**ПАКЕТ FUZZY LOGIC TOOLBOX FOR MATLAB**

Учебное пособие по курсу «Методы искусственного интеллекта»  
для студентов специальности 220401 — «Мехатроника»  
направления 220400 — «Мехатроника и робототехника»

Редактор *С. В. Пилюгина*

Подписано в печать 28.02.13. Формат 60×84/16. Бумага офсетная. Усл. печ. л. 6,5.  
Тираж 65 экз. Заказ 19

Издательство УрГУПС  
620034, Екатеринбург, ул. Колмогорова, 66

---

ISBN 978-5-94614-248-9 © Уральский государственный университет  
путей сообщения (УрГУПС), 2013

# Оглавление

Введение .....	4
1. Процесс разработки системы нечеткого вывода в интерактивном режиме .....	5
1.1. Редактор систем нечеткого вывода FIS .....	6
1.2. Редактор функций принадлежности .....	12
1.3. Редактор правил системы нечеткого вывода.....	15
1.4. Программа просмотра правил системы нечеткого вывода .....	17
1.5. Программа просмотра поверхности системы нечеткого вывода .....	20
2. Пример разработки системы нечеткого вывода в интерактивном режиме.....	22
3. Процесс разработки системы нечеткого вывода в режиме командной строки .....	35
4. Описание функций пакета Fuzzy Logic Toolbox for MatLab.....	44
5. Алфавитный указатель функций пакета Fuzzy Logic Toolbox for MatLab .....	111
Библиографический список .....	112

# Введение

**Н**ечеткая логика и теория нечетких множеств лежат в основе многих методов исследования и моделирования систем, относящихся к области искусственного интеллекта. Для реализации процесса нечеткого моделирования в среде MATLAB предназначен специальный пакет расширения Fuzzy Logic Toolbox. В рамках этого пакета, который расположен в папке ... \MATLAB\toolbox\fuzzy, пользователь может выполнять необходимые действия по разработке и использованию нечетких моделей в одном из следующих режимов:

- в интерактивном режиме с помощью графических средств редактирования и визуализации всех компонентов систем нечеткого вывода;
- режиме команд с помощью ввода имен соответствующих функций с необходимыми аргументами непосредственно в окно команд системы MATLAB.

В пособии рассматриваются особенности разработки систем нечеткого вывода в каждом из этих режимов и даются рекомендации по выполнению необходимой последовательности действий. Пособие также содержит описание всех функций пакета Fuzzy Logic Toolbox for MATLAB, предназначенных для работы с нечеткими множествами, создания и редактирования систем нечеткого вывода, выполнения процедур нечеткого вывода. Функции перечислены в алфавитном порядке, в необходимых случаях даны рисунки и примеры. Для каждой функции приводится ее назначение, а также описаны общий синтаксис и все формальные параметры, используемые в качестве аргументов.

Пособие написано на основе встроенной системы помощи пакета Fuzzy Logic Toolbox for MATLAB и предназначено для студентов, обучающихся специальности «Мехатроника».

# 1

## Процесс разработки системы нечеткого вывода в интерактивном режиме

Для разработки и дальнейшего применения систем нечеткого вывода в интерактивном режиме могут быть использованы следующие графические средства, входящие в состав пакета Fuzzy Logic Toolbox.

- Редактор систем нечеткого вывода FIS (FIS Editor).
- Редактор функций принадлежности системы нечеткого вывода (Membership Function Editor).
- Редактор правил системы нечеткого вывода (Rule Editor).
- Программа просмотра правил системы нечеткого вывода (Rule Viewer).
- Программа просмотра поверхности системы нечеткого вывода (Surface Viewer).

Кроме этих графических средств в состав пакета Fuzzy Logic Toolbox также входят следующие специальные программы.

- Редактор адаптивных систем нейро-нечеткого вывода (Adaptive Neuro-Fuzzy Inference System Editor).
- Программа нечеткой кластеризации методом нечетких С-средних (fuzzy C-means clustering).

Функции MATLAB, которые могут быть использованы для вызова соответствующих графических средств, представлены в табл. 1.1.

Таблица 1.1

Функции графического интерфейса пользователя

Функция	Назначение
<code>anfisedit</code>	Редактор гибридных сетей ANFIS
<code>findcluster</code>	Программа нечеткой кластеризации

Функция	Назначение
<b>fuzzy</b>	Редактор системы нечеткого вывода FIS
<b>mfedit</b>	Редактор функций принадлежности
<b>ruleedit</b>	Редактор правил нечеткого вывода
<b>ruleview</b>	Программа просмотра правил и диаграммы нечеткого вывода
<b>surfview</b>	Программа просмотра поверхности нечеткого вывода

Рассмотрим особенности каждого из графических средств, которые следует использовать для разработки и исследования систем нечеткого вывода в среде MATLAB.

### 1.1. Редактор систем нечеткого вывода FIS

Редактор систем нечеткого вывода FIS (или просто редактор FIS) является основным средством, которое используется для создания или редактирования систем нечеткого вывода в графическом режиме. Редактор FIS может быть открыт с помощью ввода функции **fuzzy** или **fuzzy('FISfile')** в командной строке (рис. 1.1). Эта функция предоставляет пользователю возможность задавать и редактировать на высоком уровне свойства системы нечеткого вывода, такие, как число входных и выходных переменных, тип системы нечеткого вывода, используемый метод дефаззификации и т. д.

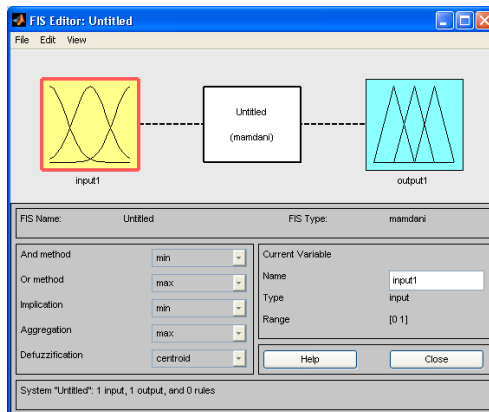


Рис. 1.1. Графический интерфейс редактора FIS, вызываемый функцией **fuzzy**

Если функция **fuzzy** вызывается без аргументов, то редактор FIS вызывается для вновь создаваемой системы нечеткого вывода с именем **Untitled** по умолчанию. По умолчанию также задается ряд таких параметров, таких как тип системы нечеткого вывода (Мамдани), нечеткие логические операции, методы импликации, агрегирования и дефаззификации и некоторые другие. Пользователь может согласиться с этими значениями или изменить их.

Если функция **fuzzy** вызывается с аргументом в форме **fuzzy ('FISfile')**, где **FISfile** — имя внешнего файла с расширением .fis с заранее разработанной системой нечеткого вывода, то редактор вызывается с уже загруженной системой FIS с именем **FISfile**. Например, на рис. 1.2 в редакторе открыт файл **mam21.fis**.

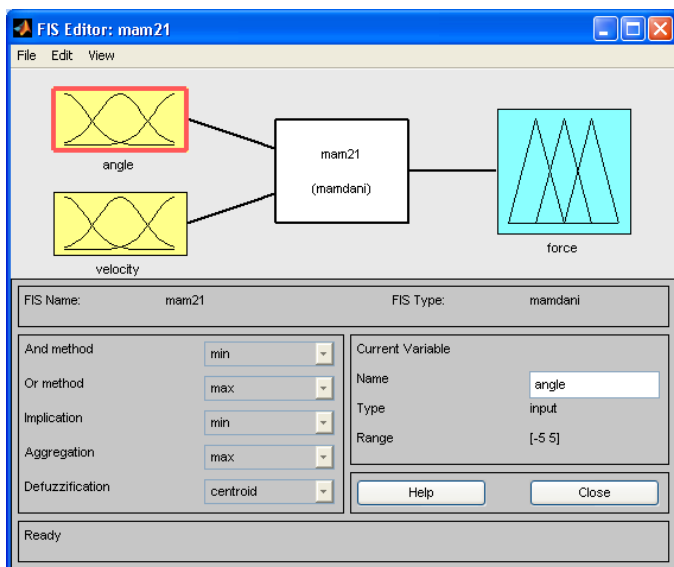


Рис. 1.2. Окно графического интерфейса редактора FIS

Возможен также вызов редактора FIS с помощью этой же функции в формате **fuzzy(fisname)**, где **fisname** — имя структуры FIS в рабочей области MATLAB. В этом случае соответствующая структура нечеткого вывода должна быть предварительно создана либо загружена в рабочую область с помощью функции **fisname=readfis('FISfile')**, где **FISfile** — имя внешнего fis-файла с разработанной ранее системой нечеткого вывода.

Редактор FIS обладает графическим интерфейсом и позволяет вызывать все другие редакторы и программы просмотра систем нечеткого вывода. Графический интерфейс редактора обладает достаточным удобством и гибкостью, необходимой для интерактивной работы с отдельными компонентами системы нечеткого вывода.

В верхней части рабочего интерфейса редактора FIS изображается диаграмма, представляющая в визуальной форме входы и выходы системы нечеткого вывода, в центре которых находится *редактор нечетких правил* Rule Editor. Щелчок на прямоугольнике с изображением входа или выхода выделяет соответствующую переменную и делает ее текущей. Изображение текущей переменной при этом выделяется красным цветом.

Двойной щелчок на прямоугольнике с изображением входной или выходной переменной вызывает редактор функций принадлежности с загруженной в него соответствующей переменной. Двойной щелчок на изображении процессора нечетких правил вызывает редактор правил для соответствующей системы нечеткого вывода. Если некоторая переменная существует в системе нечеткого вывода, но не используется в правилах вывода, то связь ее с процессором нечетких правил изображается не сплошной, а пунктирной линией.

Редактор FIS имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д. Рассмотрим назначение пунктов меню редактора FIS.

- Пункт меню **File** (Файл) редактора FIS содержит следующие операции:
  - **New FIS...** — позволяет выбрать тип задаваемой новой системы нечеткого вывода: **Mamdani** или **Sugeno** — типа Мамдани или Саджено соответственно. Задаваемая система нечеткого вывода не имеет ни входных, ни выходных переменных, а ее имя задается по умолчанию как **Untitled**;
  - **Import** — позволяет загрузить в редактор FIS существующую систему нечеткого вывода одним из двух способов: **From Workspace...** или **From Disk...** — из рабочего пространства MATLAB или из внешнего файла соответственно. В последнем случае вызывается стандартное диалоговое окно открытия внешнего файла с диска;
  - **Export** — позволяет сохранить редактируемую систему нечеткого вывода одним из двух способов: **To Workspace...** или



**To Disk...** — в рабочем пространстве программы MATLAB или во внешнем файле соответственно. В последнем случае вызывается стандартное диалоговое окно сохранения файла на диске;

- **Print** — позволяет распечатать на принтере редактируемую систему нечеткого вывода. В этом случае вызывается стандартное диалоговое окно настройки свойств печати на подключенном к данному компьютеру принтере;
- **Close** — закрывает редактор FIS, при этом вызывается диалоговое окно с предложениями сохранить или отказаться от сохранения редактируемой системы нечеткого вывода.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** — отменяет выполнение последнего действия;
  - **Add Variable...** — позволяет добавить в редактируемую систему нечеткого вывода входную (**Input**) или выходную (**Output**) переменную;
  - **Remove Selected Variable** — удаляет предварительно выбранную переменную из редактируемой системы нечеткого вывода;
  - **Membership Functions...** — вызывает редактор функций принадлежности;
  - **Rules** — вызывает редактор правил нечеткого вывода.
- Пункт меню **View** (Просмотр) содержит следующие операции:
  - **Rules** — вызывает программу просмотра правил нечеткого вывода;
  - **Surface** — вызывает программу просмотра поверхности нечеткого вывода.

В левой нижней части рабочего интерфейса редактора FIS имеется пять всплывающих меню:

- **And method** (Метод логической конъюнкции) — позволяет задать один из следующих методов для выполнения логической конъюнкции в условиях нечетких правил:
  - **min** — метод минимального значения
$$T(\mathcal{A} \wedge \mathcal{B}) = \min\{T(\mathcal{A}), T(\mathcal{B})\};$$
  - **prod** — метод алгебраического произведения
$$T(\mathcal{A} \wedge \mathcal{B}) = T(\mathcal{A}) \cdot T(\mathcal{B});$$
  - **Custom** — метод, определенный пользователем.

- **Or method** (Метод логической дизъюнкции) — позволяет задать один из следующих методов для выполнения логической дизъюнкции в условиях нечетких правил:

- **max** — метод максимального значения

$$T(\mathcal{A} \vee \mathcal{B}) = \max\{T(\mathcal{A}), T(\mathcal{B})\};$$

- **probor** — метод алгебраической суммы

$$T(\mathcal{A} \vee \mathcal{B}) = T(\mathcal{A}) + T(\mathcal{B}) - T(\mathcal{A}) \cdot T(\mathcal{B});$$

- **Custom** — метод, определенный пользователем.

- **Implication method** (Метод вывода заключения) — позволяет задать один из следующих методов для выполнения (активизации) логического заключения в каждом из нечетких правил:

- **min** — метод минимального значения

$$\mu'(y) = \min\{c_i, \mu(y)\},$$

где  $c_i$  — степень истинности заключения правила  $R_i$ , входящего в базу правил системы нечеткого вывода,  $\mu(y)$  — функция принадлежности терма, который является значением выходной переменной правила  $R_i$ ;

- **prod** — метод алгебраического произведения

$$\mu'(y) = c_i \cdot \mu(y);$$

- **Custom** — метод, определенный пользователем. Это меню не используется для систем нечеткого вывода типа Саджено.

- **Aggregation method** (Метод агрегирования) — позволяет задать один из следующих методов для агрегирования значений функции принадлежности каждой из выходных переменных в заключениях нечетких правил:

- **max** — метод максимального значения

$$T(\mathcal{A} \vee \mathcal{B}) = \max\{T(\mathcal{A}), T(\mathcal{B})\};$$

- **sum** — метод граничной суммы

$$T(\mathcal{A} \vee \mathcal{B}) = \min\{T(\mathcal{A}) + T(\mathcal{B}), 1\};$$

- **probor** — метод алгебраической суммы

$$T(\mathcal{A} \vee \mathcal{B}) = T(\mathcal{A}) + T(\mathcal{B}) - T(\mathcal{A}) \cdot T(\mathcal{B});$$

- **Custom** — метод, определенный пользователем. Это меню не используется для систем нечеткого вывода типа Саджено.
- **Defuzzification method** (Метод дефаззификации) — позволяет задать один из следующих методов для выполнения дефаззификации выходных переменных в системе нечеткого вывода типа Мамдани:
  - **centroid** — метод центра тяжести для дискретного множества значений функции принадлежности

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)},$$

где  $n$  — число одноточечных нечетких множеств, каждое из которых характеризует единственное значение рассматриваемой выходной лингвистической переменной;

- **bisector** — метод центра площади

$$\int_{\min}^y \mu(x) dx = \int_y^{\max} \mu(x) dx;$$

- **mom** (middle of maximum) — метод среднего максимума, определяемый как среднее арифметическое левого и правого модальных значений;
- **som** (smallest of maximum) — метод наименьшего (левого) модального значения;
- **lom** (largest of maximum) — метод наибольшего (правого) модального значения;
- **Custom** — метод, определенный самим пользователем.

Для систем нечеткого вывода типа Саджено можно выбрать один из следующих методов дефаззификации:

- **wtaver** (weighted average) — метод взвешенного среднего

$$y = \frac{\sum_{i=1}^n c_i \cdot w_i}{\sum_{i=1}^n c_i},$$

где  $n$  — общее количество активных нечетких продукционных правил, в подзаключениях которых присутствует получаемая выходная лингвистическая переменная;

- **wtsum** (weighted sum) — метод взвешенной суммы

$$y = \sum_{i=1}^n c_i \cdot w_i,$$

где  $n$  — общее количество активных нечетких продукционных правил, в подзаключениях которых присутствует полу-чаемая выходная лингвистическая переменная.

Методы, определенные пользователем, должны быть заданы либо в существующих m-файлах, либо в форме разработанных пользователем отдельных m-файлов, размещенных в папке ...\`MATLAB\toolbox\`fuzzy или в папке ...\`MATLAB\work`. При размещении m-файлов с пользовательскими методами работы с нечеткими переменными в текущей рабочей папке возможны проблемы с использованием нужных файлов.

В правом нижнем углу находятся кнопка вызова встроенной справочной системы MATLAB (**Help**) и кнопка закрытия редактора FIS (**Close**).

## 1.2. Редактор функций принадлежности

Редактор функций принадлежности, как следует из его названия, предназначен для задания и редактирования функций принадлежности отдельных термов системы нечеткого вывода в графическом режиме. Редактор функций принадлежности может быть открыт с помощью ввода функции `mfedit`, а также `mfedit('FISfile')` или `mfedit(fisname)` в командной строке, либо с помощью главного меню Редактора FIS (командой меню **Edit -> Membership Functions...** или нажатием клавиш `<Ctrl>+<2>`).

Эта функция, записанная в файле `mfedit`, просто вызывает редактор функций принадлежности, без загрузки какой бы то ни было системы нечеткого вывода. Функция в формате `mfedit('FISfile')` вызывает редактор функций принадлежности, который позволяет пользователю в графическом режиме анализировать и модифицировать все функции принадлежности некоторой структуры FIS, сохраненной во внешнем файле с именем `FISfile.fis`. Функция в формате `mfedit(fisname)` работает с переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `fisname`. Для каждой функции принадлежности можно изменить ее имя, тип и параметры. Редактор предоставляет пользователю не только возможность выбрать любую из 11 встроенных функций принадлежности, но и задать собственную функцию принадлежности.

Результат вызова редактора функций принадлежности с помощью функции `mfedit('mam21')` изображен на рис. 1.3.

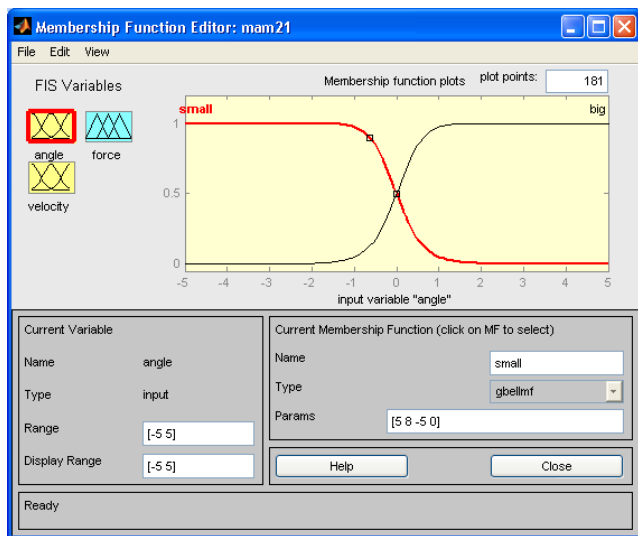


Рис. 1.3. Окно редактора функций принадлежности

Для отображения графиков функций принадлежности следует выбрать необходимую переменную в левой части графического интерфейса редактора под заголовком **FIS Variables** (Переменные FIS). Чтобы выбрать нужную функцию принадлежности, следует щелкнуть на ней или ее метке в основном окне графиков функций принадлежности.

Редактор функций принадлежности имеет главное меню программы, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д.

- Пункт меню **File** (Файл) редактора функций принадлежности содержит такие же операции, что и одноименный пункт меню редактора FIS.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** — отменяет выполнение последнего действия;
  - **Add MF...** — позволяет добавить встроенную функцию принадлежности термов для выбранной переменной;

- **Add Custom MF...** — позволяет добавить пользовательскую функцию принадлежности для отдельной переменной;
- **Remove Current MF** — позволяет удалить отдельную функцию принадлежности;
- **Remove All MFs** — позволяет удалить все функции принадлежности для отдельной переменной;
- **FIS Properties...** — вызывает редактор FIS;
- **Rules...** — вызывает редактор правил нечеткого вывода.
- Пункт меню **View** (Вид) содержит следующие операции:
  - **Rules** — вызывает программу просмотра правил нечеткого вывода;
  - **Surface** — вызывает программу просмотра поверхности нечеткого вывода.

Раскрывающийся список типов функций принадлежности позволяет выбрать одну из 11 встроенных функции принадлежности. Используя соответствующие поля ввода, можно изменить имена термов выбранной переменной в поле ввода **Name**, модифицировать параметры встроенных функций принадлежности в поле ввода **Params**.

Поскольку данный редактор не позволяет задать в диалоговом режиме функцию принадлежности, определенную пользователем, в случае подобной необходимости следует воспользоваться соответствующими функциями командного режима. Тем не менее, встроенных типов функций принадлежности оказывается вполне достаточно для большинства практических приложений.

Изменить вид функции принадлежности можно также с помощью мыши: следует выделить изменяемую функцию принадлежности на графике (она будет изображена красным цветом) и, не отпуская нажатую левую кнопку мыши, перемещать маркер в нужную сторону. При этом будут изменяться график соответствующей функции принадлежности и ее параметры. Такой возможностью следует пользоваться с большой осторожностью, поскольку выполненные таким способом изменения функции принадлежности уже не удастся отменить после экспорта системы нечеткого вывода.

В правом нижнем углу находятся кнопка вызова встроенной справочной системы MATLAB (**Help**) и кнопка закрытия редактора функций принадлежности (**Close**).

### 1.3. Редактор правил системы нечеткого вывода

Редактор правил системы нечеткого вывода, как следует из его названия, предназначен для задания и редактирования отдельных правил системы нечеткого вывода в графическом режиме. Редактор правил может быть открыт с помощью ввода функции `ruleedit ('FISfile')` или `ruleedit(fisname)` в командной строке, либо с помощью главного меню редактора FIS (командой меню **Edit -> Rules...** или нажатием клавиш `<Ctrl>+<3>`).

Эта функция, записанная в формате `ruleedit ('FISfile')`, вызывает редактор правил, который позволяет пользователю в графическом режиме анализировать и модифицировать правила продукции системы нечеткого вывода FIS, сохраненной во внешнем файле с именем `FISfile.fis`. Функция позволяет также выполнять грамматический анализ правил, которые используются в некоторой системе нечеткого вывода FIS.

Чтобы использовать данный редактор для создания правил, необходимо предварительно определить все входные и выходные переменные, для чего можно воспользоваться редактором системы нечеткого вывода FIS и редактором функций принадлежности. Задать правила можно с помощью выбора соответствующих значений термов входных и выходных переменных. Результат вызова функции `ruleedit ('mam21')` изображен на рис. 1.4.

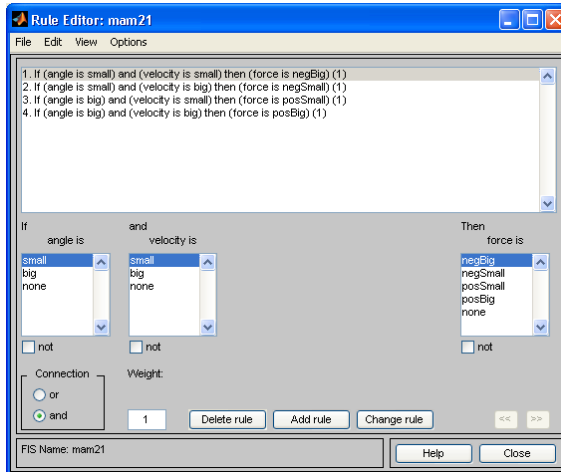


Рис. 1.4. Окно редактора правил

Функция в формате `ruleedit(fisname)` вызывает редактор правил для переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `fisname`.

Редактор правил имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д.

- Пункт меню **File** (Файл) редактора правил содержит такие же операции, что и соответствующий пункт меню редактора FIS.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** — отменяет выполнение последнего действия;
  - **FIS Properties...** — вызывает редактор FIS;
  - **Membership Functions...** — вызывает редактор функций принадлежности.
- Пункт меню **View** (Вид) содержит следующие операции:
  - **Rules** — вызывает программу просмотра правил;
  - **Surface** — вызывает программу просмотра поверхности вывода.
- Пункт меню **Options** (Сервис) содержит следующие операции:
  - **Language** — позволяет выбрать язык для записи правил в форме текста: **English** (английский), **Deutsch** (немецкий) или **Francais** (французский);
  - **Format** — позволяет выбрать формат записи правил системы нечеткого вывода: **Verbose** (в форме текста), **Symbolic** (в символической форме) или **Indexed** (в цифровой форме).

При записи правил в форме текста для создания законченных предложений используются служебные слова «if», «then», «is», «AND», «OR» и т. д. При записи правил в символической форме эти служебные слова заменяются символами соответствующих операций. Например, правило «if (A is  $\mathcal{A}$ ) and (B is  $\mathcal{B}$ ) then (C is  $\mathcal{C}$ )» преобразуется к виду « $(A = \mathcal{A}) \& (B = \mathcal{B}) \Rightarrow (C = \mathcal{C})$ ». Правила нечеткого вывода, записанные в цифровой форме, соответствуют формату их представления в структуре FIS, который рассматривается в разделе 3.

Поля ввода в средней части графического интерфейса редактора правил позволяют задать новое правило в системе нечеткого вывода. Для этого необходимо выделить имя термина соответствующей переменной, которая должна быть предварительно определена с помощью редактора функций принадлежности. Если некоторый терм не



входит в правило, то для него следует выбрать значение «none». Если в условии правила используется логическое отрицание некоторого термина, то для этого термина следует отметить соответствующий флажок с меткой «not».

Редактор правил позволяет также задать логические связки для подусловий правила (переключатель **Connection**) и вес правила (поле ввода **Weight**). Кнопки в нижней части графического интерфейса редактора правил, как следует из их названий, служат для удаления выделенного в окне правила (**Delete rule**), добавления созданного правила в систему (**Add rule**) и внесения изменений в выделенное в окне правило (**Change rule**). В правом нижнем углу находятся кнопки вызова встроенной справочной системы MATLAB (**Help**) и кнопка закрытия редактора правил (**Close**).

#### 1.4. Программа просмотра правил системы нечеткого вывода

Главное назначение программы просмотра правил заключается в возможности визуализировать результаты нечеткого вывода и получать значения выходных переменных в зависимости от исходных значений входных переменных. Графический интерфейс программы просмотра правил может быть открыт пользователем с помощью ввода функции `ruleview('FISfile')` или `ruleview(fisname)` в командном окне, либо с помощью главного меню редактора FIS, редактора функций принадлежности или редактора правил (командой меню **View -> Rules** или нажатием клавиш <Ctrl>+<5>).

Функция, записанная в формате `ruleview('FISfile')`, вызывает программу просмотра правил, которая изображает диаграмму нечеткого вывода для структуры FIS, сохраненной во внешнем файле с именем `FISfile.fis`. Функция в формате `ruleview(fisname)` вызывает программу просмотра правил для переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `fisname`.

Программа просмотра правил не позволяет редактировать правила и функции принадлежности термов переменных и используется после разработки системы нечеткого вывода на этапе ее анализа и оценки. Функцию также целесообразно использовать в том случае, когда необходимо визуально представить весь процесс нечеткого вывода от начала до конца. Пользователь имеет возможность оценить значения выходных переменных нечеткой модели и влияние каждо-

го из правил на результат нечеткого вывода посредством изменения значений входных переменных.

Графический интерфейс программы просмотра правил изображен на рис. 1.5. Программа просмотра правил имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и выполнять другие операции.

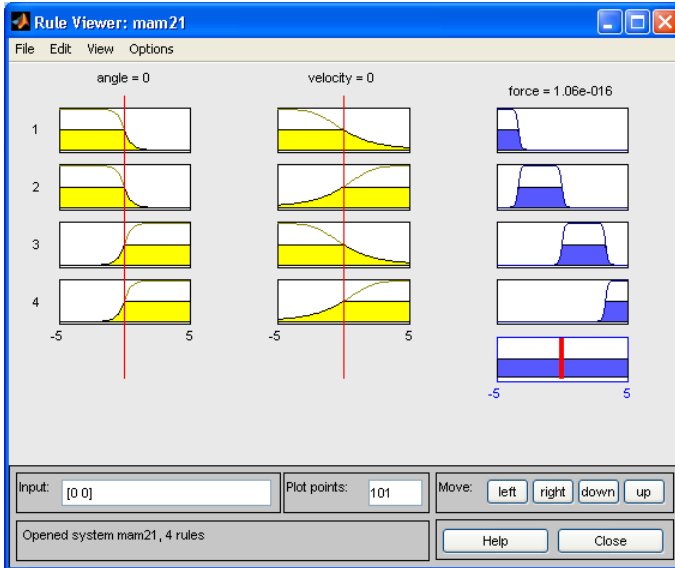


Рис. 1.5. Окно программы просмотра правил

- Пункт меню **File** (Файл) редактора правил содержит такие же операции, что и одноименный пункт меню редактора FIS.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** — отменяет выполнение последнего действия;
  - **FIS Properties...** — вызывает редактор FIS;
  - **Membership Functions...** — вызывает редактор функций принадлежности;
  - **Rules...** — вызывает программу редактирования правил.
- Пункт меню **View** (Вид) содержит следующие операции:
  - **Surface** — вызывает программу просмотра поверхности вывода.

- Пункт меню **Options** (Сервис) содержит следующие операции:
  - **Format** — позволяет выбрать формат записи правил системы нечеткого **вывода: Verbose** (в форме текста), **Symbolic** (в символической форме) или **Indexed** (в цифровой форме).

В центральной части графического интерфейса программы просмотра правил расположены прямоугольники, соответствующие отдельным входным переменным (функции принадлежности желтого цвета) и выходным переменным (функции принадлежности синего цвета) правил нечеткого вывода. Каждому правилу соответствует отдельная строка из этих прямоугольников. Номера правил указаны в левой части графического интерфейса.

В правой нижней части графического интерфейса расположен прямоугольник, изображающий нечеткое множество, соответствующее выходной переменной после аккумуляирования всех заключений правил нечеткого вывода. Полученное в результате дефаззификации значение выходной переменной указывается в верхней части столбца с именем этой выходной переменной ( $force=1.06e-016$  на рис. 1.5).

Прямоугольники входных переменных пересекает вертикальная прямая красного цвета, положение которой соответствует конкретному значению входной переменной соответствующего столбца. Задать конкретные значения входных переменных можно либо с помощью их записи в поле ввода **Input** либо с помощью мыши, перемещая вертикальные прямые в нужном направлении. В последнем случае можно щелкнуть на той или иной вертикальной прямой и, удерживая нажатой левую кнопку мыши, переместить прямую вправо или влево. Также можно просто щелкнуть в необходимой точке внутри прямоугольника соответствующей входной переменной.

Полученные после изменения значения входных переменных непосредственно отображаются в верхней части прямоугольников после имени входных переменных и в поле ввода с меткой **Input**. Более того, система MATLAB реагирует на каждое изменение значения отдельной входной переменной выполнением процедуры нечеткого вывода, получением и отображением соответствующих результирующих значений выходных переменных.

## 1.5. Программа просмотра поверхности системы нечеткого вывода

Программа просмотра поверхности системы нечеткого вывода позволяет просматривать поверхность системы нечеткого вывода и визуализировать графики зависимости выходных переменных от отдельных входных переменных. Графический интерфейс программы просмотра правил может быть открыт с помощью ввода в командной строке команды `surfview('FISfile')` или `surfview(fisname)` либо с помощью главного меню редактора FIS, редактора функций принадлежности или редактора правил (командой меню **View -> Surface** или нажатием клавиш <Ctrl>+<6>).

Функция, записанная в формате `surfview('FISfile')`, вызывает программу просмотра поверхности, которая изображает поверхность нечеткого вывода для структуры FIS, сохраненной во внешнем файле с именем `FISfile.fis`, для любой одной или двух из ее входных переменных. Функция в формате `surfview(fisname)` вызывает программу просмотра поверхности вывода для переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `fisname`.

Графический интерфейс программы просмотра поверхности изображен на рис. 1.6. Программа просмотра поверхности вывода имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д.

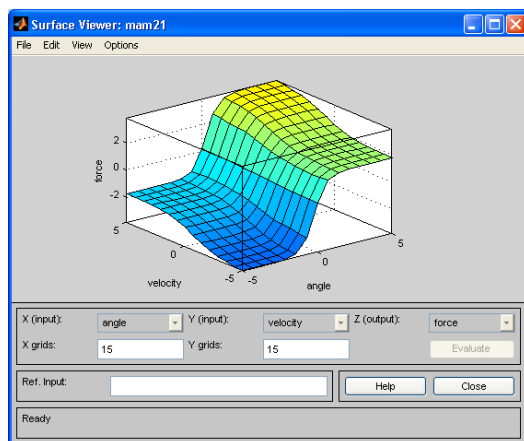


Рис. 1.6. Окно программы просмотра поверхности вывода

- Пункт меню **File** (Файл) редактора правил содержит такие же операции, что и соответствующий пункт меню редактора FIS.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** — отменяет выполнение последнего действия;
  - **FIS Properties...** — вызывает редактор FIS;
  - **Membership Functions...** — вызывает редактор функций принадлежности;
  - **Rules...** — вызывает программу редактирования правил.
- Пункт меню **View** (Вид) содержит следующие операции:
  - **Rules** — вызывает программу просмотра правил.
- Пункт меню **Options** (Сервис) содержит следующие операции:
  - **Plot** — позволяет выбрать один из восьми стилей изображения графика поверхности вывода;
  - **Color Map** — позволяет выбрать одну из четырех цветовых схем изображения графика поверхности вывода;
  - **Always evaluate** — пометка галочкой этого пункта вложенного меню приводит к автоматическому формированию новой поверхности вывода всякий раз, когда вносятся изменения в систему нечеткого вывода, влияющие на форму графика поверхности вывода (такие, как изменение количества точек сетки графика). Данное значение принято по умолчанию. Чтобы его отменить, необходимо снять галочку у этого пункта вложенного меню, щелкнув на этой позиции меню.

Программа просмотра поверхности вывода не позволяет вносить изменения в систему нечеткого вывода и соответствующую ей структуру FIS. Используя главное меню программы, пользователь может выбрать входные переменные и соответствующие им горизонтальные оси системы координат ( $X$  и  $Y$ ), а также выходную переменную, которой соответствует вертикальная ось системы координат ( $Z$ ).

Щелкнув и удерживая левую кнопку мыши на осях графика поверхности, посредством последующего перемещения курсора мыши в том или ином направлении можно изменить угол просмотра поверхности вывода. Если рассматривается система нечеткого вывода, имеющая более двух входных переменных, то для невизуализируемых входных переменных следует задать некоторые постоянные значения (константы).

Далее описываются конкретные приемы и операции с рассмотренными средствами пакета Fuzzy Logic Toolbox, используя которые, пользователь может разработать и выполнить анализ системы нечеткого вывода в графическом режиме.

# 2

## Пример разработки системы нечеткого вывода в интерактивном режиме

**В** качестве примера разработки системы нечеткого вывода в интерактивном режиме с помощью графических средств пакета Fuzzy Logic Toolbox рассмотрим следующую нечеткую модель, которая входит в число демонстрационных примеров системы MATLAB.

*Пример.* Рассмотрим ситуацию в ресторане, при которой, согласно принятым в США традициям, после окончания обслуживания посетителя принято оставлять официанту чаевые. Основываясь на устоявшихся в этой стране обычаях и интуитивных представлениях посетителей ресторанов, величина суммы чаевых не является постоянной и зависит, например, от качества обслуживания и качества приготовления заказанных блюд.

Задача состоит в том, чтобы разработать некоторую экспертную систему, которая была бы реализована в виде системы нечеткого вывода и позволяла бы определять величину чаевых на основе субъективных оценок посетителей качества обслуживания и качества приготовления заказанных блюд.

Эмпирические знания о рассматриваемой проблемной области могут быть представлены в форме следующих эвристических правил продукции:

1. Если обслуживание плохое или ужин подгоревший, то чаевые — малые.
2. Если обслуживание хорошее, то чаевые — средние.
3. Если обслуживание отличное или ужин превосходный, то чаевые — щедрые.

В качестве входных параметров системы нечеткого вывода будем рассматривать две нечеткие лингвистические переменные: «*service*»

(«качество обслуживания») и *«food»* («качество приготовления заказанных блюд»), а в качестве выходных параметров — нечеткую лингвистическую переменную *«tip»* («величина чаевых»).

В качестве терм-множества первой лингвистической переменной *«service»* будем использовать множество  $T_1 = \{\text{«poor»}, \text{«good»}, \text{«excellent»}\}$  («плохое», «хорошее», «отличное»), а в качестве терм-множества второй лингвистической переменной *«food»* будем использовать множество  $T_2 = \{\text{«rancid»}, \text{«delicious»}\}$  («подгоревший», «превосходный»). В качестве терм-множества выходной лингвистической переменной *«tip»* будем использовать множество  $T_3 = \{\text{«cheap»}, \text{«average»}, \text{«generous»}\}$  («малые», «средние», «щедрые»). При этом каждый из термов входных переменных будем оценивать по 10-балльной шкале, а термы выходной переменной — в процентах от стоимости заказанных блюд.

С учетом сделанных уточнений рассмотренная субъективная информация о величине чаевых может быть представлена в форме трех правил нечетких продукций следующего вида (система нечеткого вывода типа Мамдани):

**ПРАВИЛО\_1:** ЕСЛИ «качество обслуживания плохое» ИЛИ «ужин подгоревший» ТО «величина чаевых малая»,

**ПРАВИЛО\_2:** ЕСЛИ «качество обслуживания хорошее» ТО «величина чаевых средняя»,

**ПРАВИЛО\_3:** ЕСЛИ «качество обслуживания отличное» ИЛИ «ужин превосходный» ТО «величина чаевых щедрая».

Следует отметить, что соответствующая данному примеру система нечеткого вывода хранится во внешнем файле с именем **tipper.fis** в папке ... \MATLAB\toolbox\fuzzy\fuzdemos.

Процесс разработки системы нечеткого вывода в интерактивном режиме для рассмотренного выше примера «*Чаевые в ресторане*» состоит в выполнении следующей последовательности действий.

1. Вызвать редактор систем нечеткого вывода FIS, для чего в окне команд набрать имя соответствующей функции **fuzzy**. После выполнения этой команды на экране появится графический интерфейс редактора FIS с именем системы нечеткого вывода **Untitled** и типом системы нечеткого вывода (Мамдани), предложенными по умолчанию (см. рис. 1.1).

2. Поскольку в примере рассматривается система нечеткого вывода с двумя входами, необходимо добавить в разрабатываемую систему FIS еще одну входную переменную. Для этого следует выполнить команду меню **Edit -> Add Variable... -> Input. B** результате

выполнения данной команды на диаграмме системы нечеткого вывода появится новый желтый прямоугольник с именем второй входной переменной **input2** (рис. 2.1).

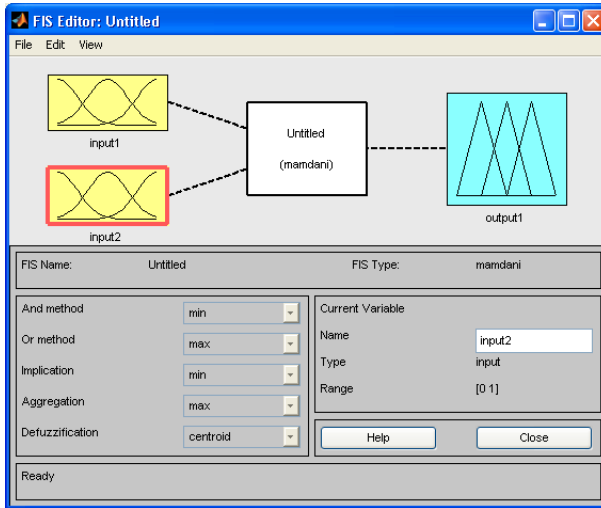


Рис. 2.1. Вид редактора FIS после добавления второй входной переменной

3. Изменим имена входных и выходных переменных, предложенных системой MATLAB по умолчанию: необходимо выделить прямоугольник с именем соответствующей переменной, выполнив щелчок на его изображении на диаграмме (стороны выделенного прямоугольника имеют красный цвет). После чего следует набрать новое имя переменной в поле ввода **Name** в правой части редактора FIS. Результат изменения имен переменных системы нечеткого вывода изображен на рис. 2.2.

При создании системы нечеткого вывода в интерактивном режиме допускается применение русскоязычных названий. В этом случае следует иметь в виду, что в некоторых версиях MATLAB имеются проблемы с кириллицей в названиях переменных и термов (символы «с» и «я»). Чтобы избежать проблем с корректным отображением символов кириллицы, следует давать такие имена переменным, которые состоят из одного слова без дополнительных служебных символов. Также следует иметь в виду, что при отображении на экране имен переменных используется LaTeX-транслятор, который вос-



принимает символ подчеркивания как преобразование следующего символа в нижний индекс. Например, имя переменной `input_11` отобразится как `input11`. Более подробно об использовании LaTeX-транслятора для записи математических выражений можно узнать в соответствующих разделах помощи по LaTeX и MATLAB.

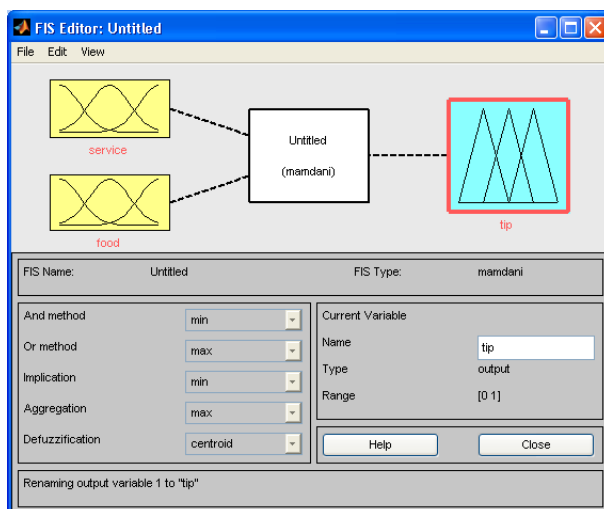


Рис. 2.2. Вид редактора FIS после изменения имен переменных, предложенных системой MATLAB по умолчанию

4. Изменим имя системы нечеткого вывода (Untitled), предложенное по умолчанию. Для этого сохраним создаваемую структуру FIS во внешнем файле с именем `mytip.fis`, выполнив команду меню **File -> Export -> To Disk...**. При этом будет вызвано стандартное диалоговое окно сохранения файла, в котором пользователю предлагается ввести имя соответствующего файла (расширение файла присывается автоматически). Оставим без изменения предложенные системой MATLAB по умолчанию: метод нечеткого логического И (**And method**) — значение «min», метод нечеткого логического ИЛИ (**Or method**) — значение «max», метод импликации (**Implication**) — значение «min», метод агрегирования (**Aggregation**) — значение «max» и метод дефаззификации (**Defuzzification**) — значение «centroid».

5. Теперь необходимо определить термы и их функции принадлежности для входных и выходных переменных нашей системы нечеткого вывода. Для этой цели следует воспользоваться редактором

функций принадлежности, который может быть вызван одним из следующих способов:

- двойным щелчком на значке прямоугольника с именем соответствующей переменной;
- командой меню **Edit -> Membership Functions...** (предварительно должен быть выделен прямоугольник с именем соответствующей переменной);
- нажатием клавиш **<Ctrl>+<2>** (предварительно также должен быть выделен прямоугольник с именем соответствующей переменной).

После вызова редактора функций принадлежности каждой из переменных по умолчанию предлагается три терма с треугольными функции принадлежности (рис. 2.3).

Сначала изменим диапазон определения значений входных переменных, для чего в полях ввода **Range** и **Display Range** изменим верхнее значение с 1 на 10 (баллов). Аналогично выполняются изменения соответствующих диапазонов для выходной переменной «чаевые», при этом верхнее значение 1 следует заменить на 30 (%). Изменения подтверждаются нажатием на клавишу **<Enter>**.

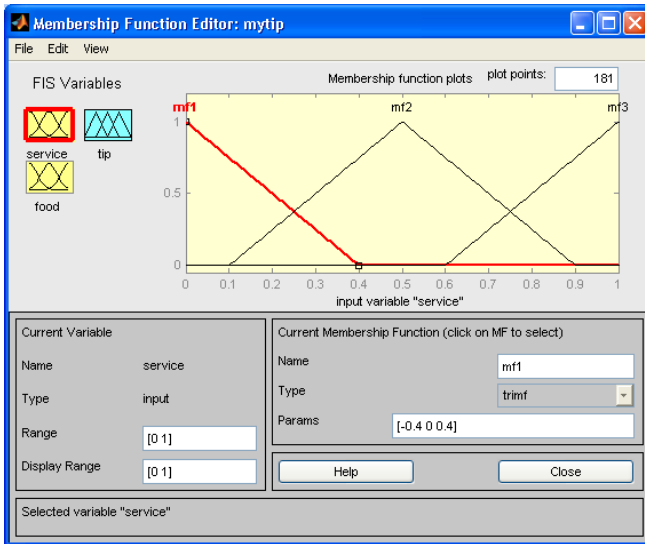


Рис. 2.3. Вид редактора функций принадлежности после его вызова с функциями принадлежности для термов переменной «*service*», предложенных системой MATLAB по умолчанию

Далее изменим названия термов первой входной переменной «*service*», предложенные системой MATLAB по умолчанию (*mf1*, *mf2*, *mf3*) на «*poor*», «*good*», «*excellent*» соответственно. После чего изменим тип функций принадлежности первой переменной, предложенный по умолчанию, на функции типа Гаусса (*gaussmf*), выбрав соответствующий пункт в поле **Type**. Параметры вновь заданных функций принадлежности оставим без изменения. Вид редактора функций принадлежности после внесенных изменений для первой из входных переменных изображен на рис. 2.4. Аналогичным образом изменим названия термов второй входной переменной «*food*» и удалим один из термов с соответствующей функцией принадлежности. Для удаления термина следует выделить удаляемую функцию принадлежности и нажать клавишу <Delete> на клавиатуре. Переход к редактированию переменной осуществляется щелчком на изображении прямоугольника с именем необходимой переменной. Для переменной «*food*» изменим тип функций принадлежности ее термов на трапециевидные функции (*trapmf*) и их параметры следующим образом: для термина «*rancid*» зададим параметры [0 0 1 3], а для термина «*delicious*» — [7 9 10 10].

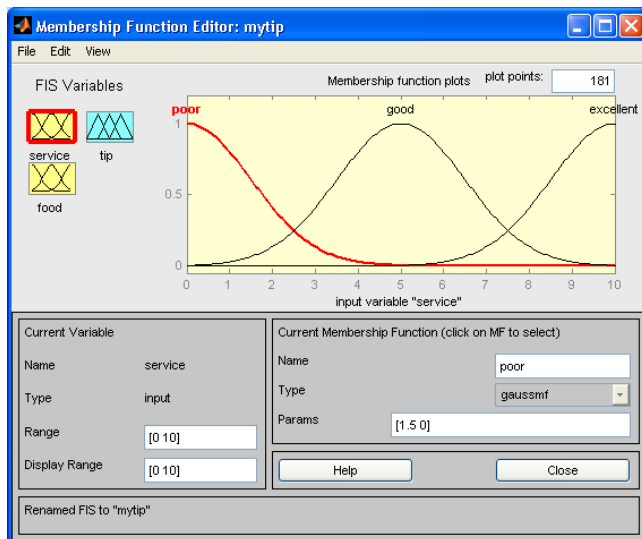


Рис. 2.4. Вид редактора функций принадлежности после изменения названий термов и типа их функций принадлежности для первой входной переменной «*service*»

Вид редактора функций принадлежности после внесенных изменений для второй входной переменной изображен на рис. 2.5.

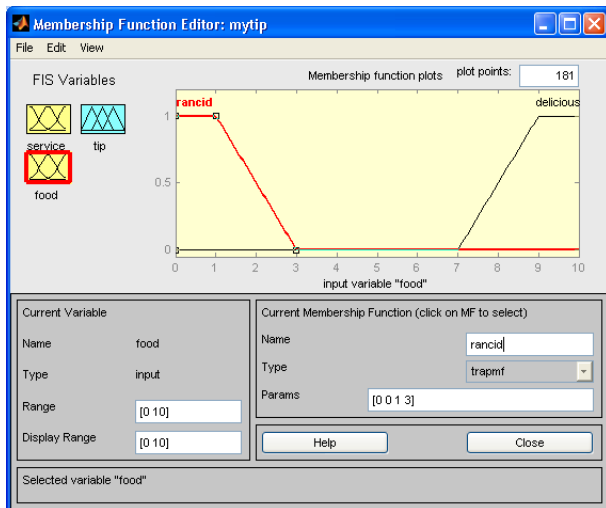


Рис. 2.5. Вид редактора функций принадлежности после изменения названия термов и типа их функций принадлежности для второй входной переменной «*food*»

Наконец, изменим названия термов и параметры функций принадлежности для выходной переменной «*tip*», оставив без изменения треугольный тип функций принадлежности, предложенный системой MATLAB. Для терма «*cheap*» зададим параметры [0 5 10], а для терма «*average*» — [10 15 20], для терма «*generous*» — [20 25 30]. Вид редактора функций принадлежности после сделанных изменений для выходной переменной «*tip*» изображен на рис. 2.6.

6. Теперь определим правила нечеткого вывода для разрабатываемой экспертной системы. Для этой цели следует воспользоваться редактором правил, который может быть вызван одним из следующих способов:

- двойным щелчком на значке квадрата в центре с именем создаваемой системы нечеткого вывода (**myfis**);
- командой меню **Edit -> Rules...**;
- нажатием клавиш <Ctrl>+<3>.

Поскольку первоначально база правил нечеткого вывода пуста, то после вызова редактора правил центральное многострочное поле

вывода не содержит никаких правил. Для их определения следует использовать поля меню и переключатели в нижней части графического интерфейса редактора правил. Для задания первого правила следует оставить выделенные по умолчанию поле с именем термина «*poor*» для первой входной переменной, поле с именем термина «*rancid*» для второй входной переменной и поле с именем термина «*poor*» для выходной переменной. Далее следует переключатель **Connection** поставить в положение **or** (логическое ИЛИ) и нажать на кнопку **Add rule**. После этого первое правило с символами кириллицы отобразится в верхнем окне.

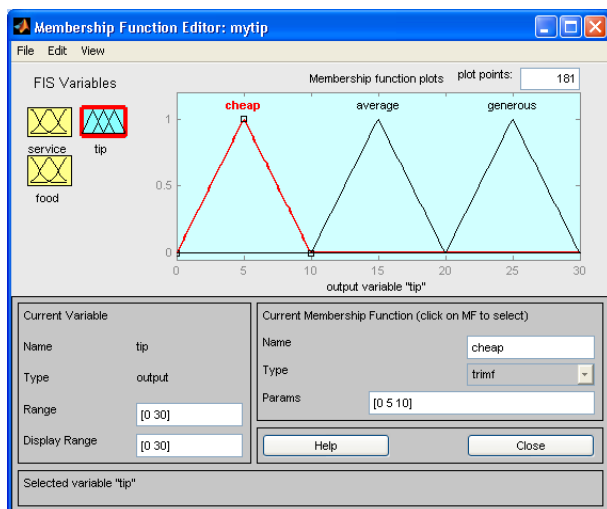


Рис. 2.6. Вид редактора функций принадлежности после изменения названия термов и типа их функций принадлежности для выходной переменной «*tip*»

Аналогичным образом задается второе правило, для которого следует выделить имена термов «*good*», «*none*» и «*average*», и третье правило с именами термов «*excellent*», «*delicious*» и «*generous*» для соответствующих переменных. Вид редактора правил после их определения для разрабатываемой экспертной системы изображен на рис. 2.7.

Заметим, что в поле ввода **Weight** отображается вес каждого правила, который можно изменять в пределах интервала  $[0, 1]$  (оставим без изменения его значение по умолчанию, равное 1 для всех правил). Этот же вес правил записывается в круглых скобках в окне правил после каждого из правил нечеткого вывода.

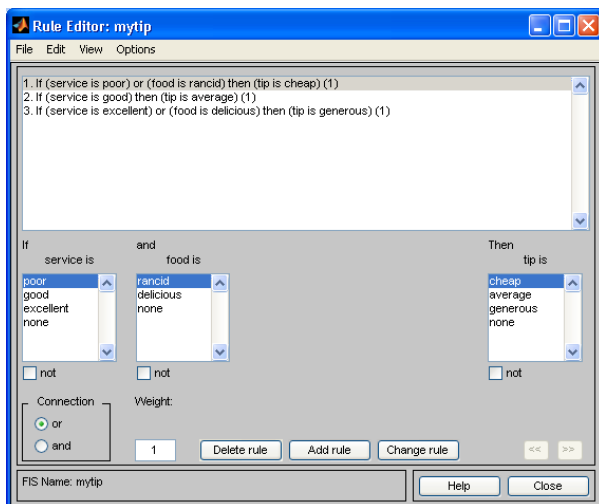


Рис. 2.7. Вид редактора правил нечеткого вывода после их определения

7. После задания правил нечеткого вывода оказывается возможным получить результат нечеткого вывода (значение выходной переменной) для конкретных значений входных переменных. С этой целью необходимо открыть программу просмотра правил одним из следующих способов:

- командой меню **View -> Rules** редактора FIS;
- командой меню **View -> Rules** редактора функций принадлежности;
- командой меню **View -> Rules** редактора правил;
- нажатием клавиш <Ctrl>+<5>.

После вызова программы просмотра правил для нашей системы нечеткого вывода по умолчанию для входных переменных предложены средние значения из интервала их допустимых значений (значения [5 5] в поле ввода **Input**). Это означает, что посетитель ресторана оценивает качество обслуживания в 5 баллов и качество ужина также в 5 баллов. Таким значениям входных переменных соответствует значение чаевых в 15 %, которое отображается выше прямоугольников правил в правой части окна программы просмотра.

Изменим значения входных переменных для другого случая, которому соответствует качество обслуживания в 0 баллов («хуже некуда») и качество ужина в 10 баллов («лучше не бывает»). Для этого кур-

сор мыши переместим в поле ввода **Input** и введем соответствующие значения входных переменных: [0 10]. Система MATLAB оставит значение чаевых без изменения (15 %), однако на диаграмме правил можно заметить результаты выполненных изменений (рис. 2.8).

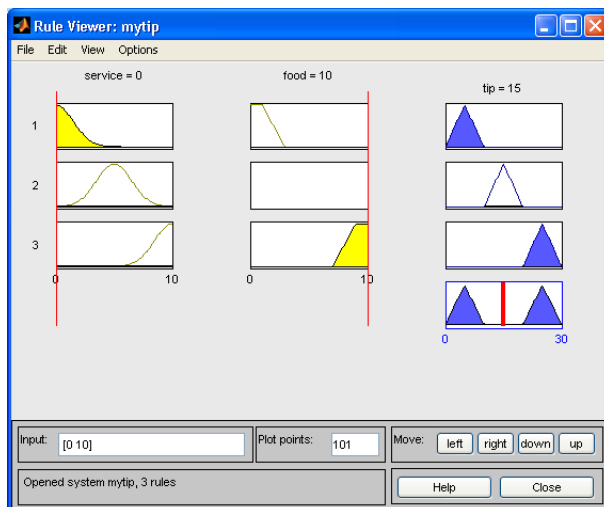


Рис. 2.8. Вид программы просмотра правил нечеткого вывода после изменения значений входных переменных на [0 10]

Поскольку процесс нечеткого моделирования предполагает анализ результатов нечеткого вывода при различных значениях входных переменных с целью установления адекватности разработанной нечеткой модели (в данном случае — экспертной системы), рассмотрим и другие случаи. Предположим, что качество обслуживания оценивается в 10 баллов («лучше не бывает»), а качество ужина — в 2 балла («бывает и хуже, но реже»). Введем соответствующие значения переменных аналогичным способом. В этом случае разработанная нами система нечеткого вывода рекомендует нам оставить чаевые в размере 16.4 %.

Если же предположить, что качество обслуживания по-прежнему отличное (10 баллов), а качество ужина несколько улучшилось и оценивается в 3 балла, то величина чаевых существенно изменится и станет равной 24.7 %. Более того, дальнейшее увеличение качества ужина не оказывает изменения величины чаевых. В частности, для значений входных переменных [10 10] величина чаевых составит по-

прежнему 24.7 %. Если некоторым из посетителей такая экспертная система покажется неадекватной (в частности, для случая значений входных переменных [10 10] можно бы оставить максимальные чаевые в 30 %), то разработанная система нечеткого вывода потребует модификации. Данная модификация может потребовать изменения существующих правил или добавления новых, а также изменения параметров функций принадлежности входных и выходной переменных. Более тонкая настройка модели может быть связана с увеличением количества термов для каждой из входных и выходных переменных, что, в свою очередь, приведет к увеличению количества правил в системе нечеткого вывода и общему усложнению модели.

8. Для окончательного анализа разработанной нечеткой модели может оказаться полезной программа просмотра поверхности нечеткого вывода, которая может быть вызвана одним из следующих способов:

- командой меню **View -> Surface** редактора FIS;
- командой меню **View -> Surface** редактора функций принадлежности;
- командой меню **View -> Surface** редактора правил;
- командой меню **View -> Surface** программы просмотра правил;
- нажатием клавиш <Ctrl>+<6>.

Графический интерфейс программы просмотра поверхности нечеткого вывода для разработанной нечеткой модели изображен на рис. 2.9.

Эта программа служит для общего анализа адекватности нечеткой модели, позволяя оценить влияние изменения значений входных нечетких переменных на значение одной из выходных нечетких переменных. В случае необходимости можно получить график зависимости выходной переменной от одной из входных переменных. Для этого необходимо выбрать нужную переменную в раскрывающемся списке **X (input)**, а в раскрывающемся списке **Y (input)** выбрать значение «none». Полученный график зависимости изображен на рис. 2.10.

Построенный график зависимости соответствует среднему значению второй входной переменной («food») в 5 баллов. Это значение может быть изменено пользователем, для чего следует ввести нужное значение в поле ввода **Ref.Input**. Заметим, что значение **NaN** для первой входной переменной соответствует ее изменению во всем интервале определения [0, 10].



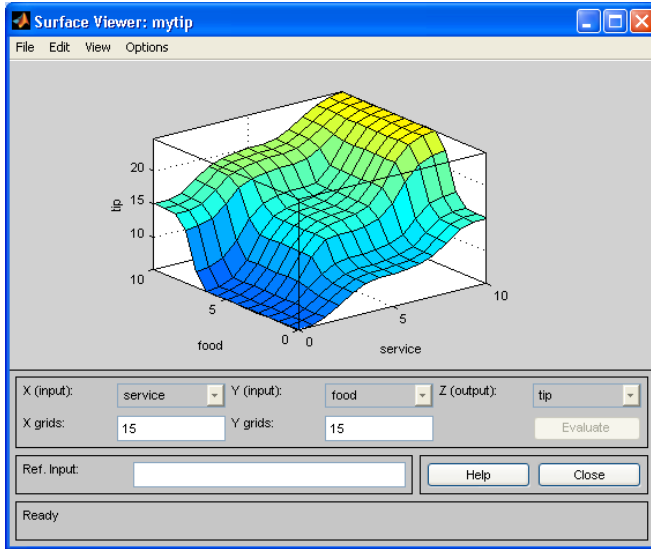


Рис. 2.9. Вид программы просмотра поверхности нечеткого вывода для разработанной нечеткой модели

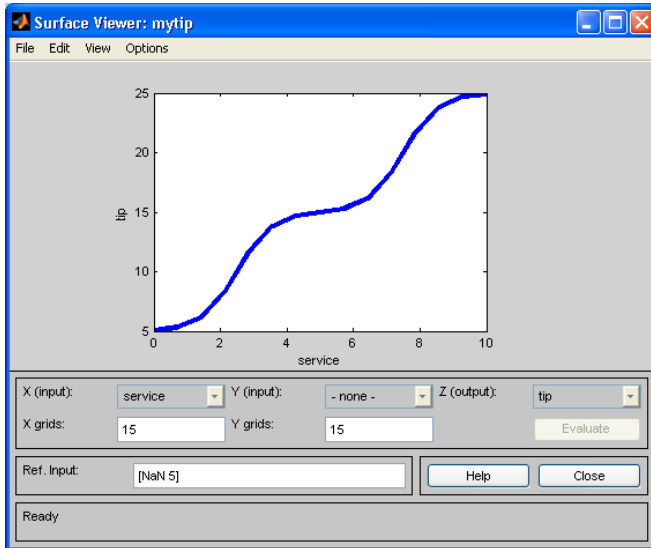


Рис. 2.10. График зависимости выходной переменной от первой из входных переменных для разработанной нечеткой модели

Заканчивая рассмотрение процесса разработки простейшей системы нечеткого вывода в интерактивном режиме, следует заметить, что наиболее эффективным этот способ оказывается для сложных нечетких моделей с большим числом переменных и правил нечеткого вывода. В этом случае задание переменных и функций принадлежности их термов в графическом режиме, а также визуализация правил позволяют существенно уменьшить трудоемкость разработки нечеткой модели, снизить количество возможных ошибок и сократить общее время нечеткого моделирования.

В процессе работы следует помнить, что количество переменных и правил в нечеткой модели, которые могут быть визуализированы, ограничено. В частности, если число входных переменных превышает 10, то их отображение в соответствующих графических редакторах происходит с искажениями.

Процесс разработки системы нечеткого вывода в режиме команд может дополнить, а в отдельных случаях и заменить процесс разработки в интерактивном режиме, предоставляя пользователю полный контроль над всеми переменными рабочей области системы MATLAB.

# 3

## Процесс разработки системы нечеткого вывода в режиме командной строки

Процесс разработки системы нечеткого вывода в режиме команд реализуется с помощью функций, входящих в состав пакета Fuzzy Logic Toolbox. В системе MATLAB реализованы следующие основные группы функций нечеткой логики:

- 11 встроенных функций принадлежности для термов нечетких переменных (табл. 3.1);
- функции управления структурой данных системы нечеткого вывода FIS (табл. 3.2);
- функции дополнительных методов и взаимодействия с пакетом Simulink (табл. 3.3).

Подробное описание функций пакета Fuzzy Logic Toolbox приводится во встроенной системе помощи и разделе 4 настоящего пособия.

Таблица 3.1

Функции принадлежности пакета Fuzzy Logic Toolbox

Функция	Назначение
<b>dsigmf</b>	Разность двух сигмоидальных функций принадлежности
<b>gauss2mf</b>	Функция принадлежности типа двухсторонней кривой Гаусса
<b>gaussmf</b>	Функции принадлежности типа кривой Гаусса
<b>gbellmf</b>	Обобщенная функция принадлежности типа колоколообразной кривой
<b>pimf</b>	П-образная функция принадлежности
<b>psigmf</b>	Произведение двух сигмоидальных функций принадлежности
<b>smf</b>	S-образная функция принадлежности
<b>sigmf</b>	Сигмоидальная функция принадлежности
<b>trapmf</b>	Трапециевидная функция принадлежности
<b>trimf</b>	Треугольная функция принадлежности
<b>zmf</b>	Z-образная функция принадлежности

Таблица 3.2

## Функции управления структурой данных системы нечеткого вывода

Функция	Назначение
<b>addmf</b>	Добавление функции принадлежности в систему нечеткого вывода FIS
<b>addrule</b>	Добавление правила в систему нечеткого вывода FIS
<b>addvar</b>	Добавление переменной в систему нечеткого вывода FIS
<b>defuzz</b>	Дефаззификация функции принадлежности
<b>evalfis</b>	Выполнение нечеткого вывода в FIS
<b>evalmf</b>	Выполнение оценки функции принадлежности
<b>gensurf</b>	Построение поверхности вывода FIS
<b>getfis</b>	Вывод свойств нечеткой системы
<b>mf2mf</b>	Преобразование параметров двух функций принадлежности
<b>newfis</b>	Создание новой системы нечеткого вывода FIS
<b>parsrule</b>	Проверка правильности правил вывода
<b>plotfis</b>	Отображение структуры входа — выхода системы нечеткого вывода FIS
<b>plotmf</b>	Изображение графиков всех функций принадлежности соответствующей лингвистической переменной
<b>readfis</b>	Загрузка системы нечеткого вывода FIS с диска
<b>rmmf</b>	Удаление функции принадлежности из системы нечеткого вывода FIS
<b>rmvar</b>	Удаление переменной из системы нечеткого вывода FIS
<b>setfis</b>	Задание свойств нечеткой системы
<b>showfis</b>	Отображение обозначения FIS
<b>showrule</b>	Отображение правил системы нечеткого вывода FIS
<b>writefis</b>	Сохранение системы нечеткого вывода FIS на диске

Таблица 3.3

## Функции дополнительных методов

Функция	Назначение
<b>anfis</b>	Обучение системы нейро-нечеткого вывода типа ANFIS
<b>fcm</b>	Нахождение кластеров для алгоритма нечетких C-средних FCM
<b>fuzarith</b>	Выполнение операций нечеткой арифметики
<b>fuzblock</b>	Библиотека нечеткой логики пакета Simulink
<b>genfis1</b>	Генерация матрицы системы нечеткого вывода с использованием «жадного» алгоритма grid
<b>genfis2</b>	Генерация матрицы системы нечеткого вывода с использованием субтрактивной кластеризации (subtractive clustering)
<b>sfis</b>	S-функция нечеткого вывода для пакета Simulink
<b>subclust</b>	Программа нахождения кластеров для алгоритма субтрактивной кластеризации

При разработке системы нечеткого вывода в режиме команд необходимо знать, что нечеткая модель в рабочей области системы MATLAB представляется в форме так называемой *структуры*.

В системе MATLAB структура представляет собой тип данных, которая состоит из *полей* и, возможно, других структур. В свою очередь, поле представляет собой простейший тип данных (число или строка). В рабочей области MATLAB структура представляется в форме массива и обозначается ключевым словом `struct array`.

Для представления собственно правил в рабочей области MATLAB используется специальный цифровой формат. Все правила системы нечеткого вывода представляются в форме матрицы, содержащей одну или несколько строк. Каждой строке соответствует отдельное правило. Если система нечеткого вывода имеет  $m$  входных переменных и  $n$  выходных переменных, то соответствующая матрица должна иметь в точности  $m+n+2$  столбцов.

Первые  $m$  столбцов относятся к входным переменным системы. При этом номер столбца должен соответствовать номеру терма для конкретной входной переменной. Следующие  $n$  столбцов относятся к выходным переменным системы вывода. Каждый столбец также имеет номер, который должен соответствовать номеру функции принадлежности для выходной переменной. Столбец с номером  $m+n+1$  содержит вес, с которым применяется данное правило. Вес может принимать любое значение между нулем и единицей. В общем случае целесообразно задавать вес равным единице (задается по умолчанию). Столбец с номером  $m+n+2$  содержит число 1, если для подусловий данного правила используется нечеткий оператор AND (нечеткое И). Этот столбец содержит число 2, если для подусловий данного правила используется нечеткий оператор OR (нечеткое ИЛИ).

Так, например, для рассмотренного выше демонстрационного примера «*Чаевые в ресторане*», правила в рабочей области MATLAB будут представлены в форме матрицы

```
1 1 1 1 2
2 0 2 1 1
3 2 3 1 2
```

Для данного примера первая строка матрицы может быть интерпретирована в форме следующего правила нечеткого вывода: «Если первая входная переменная принимает значение первого терма

(функция принадлежности с номером 1) ИЛИ вторая входная переменная принимает значение первого терма (функция принадлежности с номером 1), то выходная переменная принимает значение первого терма (функция принадлежности с номером 1)». При этом вес данного правила равен 1. Если на месте некоторого терма расположена цифра 0, то это означает, что соответствующая входная переменная не используется в правиле нечеткого вывода с номером строки данной матрицы.

Таким образом, процесс разработки системы нечеткого вывода в командном режиме представляет собой последовательность функций, которые в фиксированном порядке определяют все элементы (поля и вложенные структуры) исходной структуры системы MATLAB. Ниже приводится последовательность команд, которые позволяют разработать систему нечеткого вывода для примера «*Чаевые в ресторане*». При вводе соответствующих функций в окно команд результаты их выполнения можно контролировать либо с помощью окна просмотра рабочей области, либо непосредственно в окне команд.

Особо отметим, что при разработке системы нечеткого вывода в режиме командной строки рекомендуется отказаться от символов кириллицы в записи имен переменных и их термов. В противном случае могут возникнуть проблемы с корректным функционированием системы MATLAB.

В качестве примера рассмотрим последовательность функций разработки системы нечеткого вывода для примера «Чаевые в ресторане», рассмотренного в предыдущем разделе. Для этого требуется выполнить набор команд, представленный ниже.

Иницилируем создание новой системы нечеткого вывода с именем 'mytip'.

```
a=newfis('mytip');
```

Задаем имя первой входной переменной 'service'.

```
a.input(1).name='service';
```

Задаем границы первой входной переменной

```
a.input(1).range=[0 10];
```

Задаем первую функцию принадлежности 'poor' первой входной переменной:

имя первой функции принадлежности первой входной переменной

```
a.input(1).mf(1).name='poor';
```

тип первой функции принадлежности первой входной переменной

```
a.input(1).mf(1).type='gaussmf';
```

параметры первой функции принадлежности первой входной переменной

```
a.input(1).mf(1).params=[1.5 0];
```

Аналогично задаем вторую и третью функции принадлежности первой входной переменной

```
a.input(1).mf(2).name='good';  
a.input(1).mf(2).type='gaussmf';  
a.input(1).mf(2).params=[1.5 5];  
a.input(1).mf(3).name='excellent';  
a.input(1).mf(3).type='gaussmf';  
a.input(1).mf(3).params=[1.5 10];
```

Задаем вторую входную переменную

```
a.input(2).name='food';  
a.input(2).range=[0 10];  
a.input(2).mf(1).name='rancid';  
a.input(2).mf(1).type='trapmf';  
a.input(2).mf(1).params=[-2 0 1 3];  
a.input(2).mf(2).name='delicious';  
a.input(2).mf(2).type='trapmf';  
a.input(2).mf(2).params=[7 9 10 12];
```

По аналогии задаем выходную переменную

```
a.output(1).name='tip';  
a.output(1).range=[0 30];  
a.output(1).mf(1).name='cheap';  
a.output(1).mf(1).type='trimf';  
a.output(1).mf(1).params=[0 5 10];  
a.output(1).mf(2).name='average';  
a.output(1).mf(2).type='trimf';  
a.output(1).mf(2).params=[10 15 20];  
a.output(1).mf(3).name='generous';  
a.output(1).mf(3).type='trimf';  
a.output(1).mf(3).params=[20 25 30];
```

Задаем базу нечетких продукционных правил.

Первое продукционное правило:

условия (номера функций принадлежности входных переменных)

```
a.rule(1).antecedent=[1 1];
```

заключение (номер функции принадлежности выходной переменной)

```
a.rule(1).consequent=[1];
```

вес правила

```
a.rule(1).weight=1;
```

логическая операция (1 — AND, 2 — OR)

```
a.rule(1).connection=2;
```

Аналогично задаем остальные нечеткие правила

```
a.rule(2).antecedent=[2 0];
```

```
a.rule(2).consequent=[2];
```

```
a.rule(2).weight=1;
```

```
a.rule(2).connection=1;
```

```
a.rule(3).antecedent=[3 2];
```

```
a.rule(3).consequent=[3];
```

```
a.rule(3).weight=1;
```

```
a.rule(3).connection=2;
```

Выводим описание построенной системы нечеткого вывода в командное окно

```
showfis(a)
```

В результате выполнения этой последовательности функций в окне команд отобразится полная информация о созданной структуре нечеткого вывода.

1.	Name	mytip
2.	Type	mamdani
3.	Inputs/Outputs	[2 1]
4.	NumInputMFs	[3 2]
5.	NumOutputMFs	3
6.	NumRules	3
7.	AndMethod	min
8.	OrMethod	max
9.	ImpMethod	min
10.	AggMethod	max
11.	DefuzzMethod	centroid
12.	InLabels	service
13.		food
14.	OutLabels	tip
15.	InRange	[0 10]
16.		[0 10]
17.	OutRange	[0 30]
18.	InMFLabels	poor
19.		good
20.		excellent
21.		rancid



```

22.          delicious
23. OutMFLabels  cheap
24.          average
25.          generous
26. InMFTypes    gaussmf
27.          gaussmf
28.          gaussmf
29.          trapmf
30.          trapmf
31. OutMFTypes   trimf
32.          trimf
33.          trimf
34. InMFParams   [1.5 0 0 0]
35.          [1.5 5 0 0]
36.          [1.5 10 0 0]
37.          [-2 0 1 3]
38.          [7 9 10 12]
39. OutMFParams  [0 5 10 0]
40.          [10 15 20 0]
41.          [20 25 30 0]
42. Rule Antecedent [1 1]
43.          [2 0]
44.          [3 2]
42. Rule Consequent 1
43.          2
44.          3
42. Rule Weigth    1
43.          1
44.          1
42. Rule Connection 2
43.          1
44.          2

```

Если сохранить построенную систему нечеткого вывода с именем `'mytyp'` в файл `mytyp.fis`, то полученный файл можно открыть при помощи текстового редактора, например, Notepad или WordPad. При этом содержание файла будет иметь следующий вид:

```

[System]
Name='tipper'
Type='mamdani'
NumInputs=2

```

```

NumOutputs=1
NumRules=3
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='service'
Range=[0 10]
NumMFs=3
MF1='poor':'gaussmf',[1.5 0]
MF2='good':'gaussmf',[1.5 5]
MF3='excellent':'gaussmf',[1.5 10]

[Input2]
Name='food'
Range=[0 10]
NumMFs=2
MF1='rancid':'trapmf',[0 0 1 3]
MF2='delicious':'trapmf',[7 9 10 10]

[Output1]
Name='tip'
Range=[0 30]
NumMFs=3
MF1='cheap':'trimf',[0 5 10]
MF2='average':'trimf',[10 15 20]
MF3='generous':'trimf',[20 25 30]

[Rules]
1 1, 1 (1) : 2
2 0, 2 (1) : 1
3 2, 3 (1) : 2

```

Открытый файл можно редактировать, но при этом следует очень внимательно отслеживать связь между элементами системы нечеткого вывода. Например, если удалить одну из функций принадлежности какой-нибудь переменной, не удалив ее из базы правил, полученная система не сможет быть открытой при помощи редактора

FIS, что приведет к невозможности ее дальнейшего использования. При этом будет выдано сообщение о том, что открываемая структура неполна или ошибочна:

```
??? Error: Incomplete or misformed expression or statement.
```

```
Error in ==> readfis at 141
    MFName=eval(MFStr((nameStart+1):(nameEnd-1)));
```

```
Error in ==> fisgui at 117
    [fis,errorStr]=readfis;
```

```
??? Error while evaluating uimenu Callback.
```

# 4

## Описание функций пакета Fuzzy Logic Toolbox for MATLAB

### **addmf**

#### *Назначение*

Добавление новой функции принадлежности к системе нечеткого вывода.

#### *Синтаксис*

```
a=addmf(a, 'varType', varIndex, 'mfName', 'mfType', ...  
mfParams)
```

#### *Описание*

Функция принадлежности может быть добавлена только к существующей переменной FIS для текущей рабочей области MATLAB. При этом номера или индексы функциям принадлежности назначаются в том порядке, в котором они добавляются. Таким образом, первая добавленная к переменной функция принадлежности будет всегда являться для этой переменной функцией принадлежности с номером 1. Если в FIS определена только одна входная переменная, то добавить функцию принадлежности для любой другой входной переменной с номером 2 нельзя.

Функция использует шесть аргументов (входных параметров) в следующем порядке:

- *a* — имя переменной структуры FIS в рабочей области MATLAB.
- *'varType'* — строка, представляющая *тип* переменной, к которой добавляется функция принадлежности. Может принимать одно из двух значений: *'input'* или *'output'*.
- *varIndex* — индекс переменной, к которой добавляется функция принадлежности.
- *'mfName'* — строка, представляющая *имя* новой функции принадлежности.

- 'mfType' — строка, представляющая *тип* новой функции принадлежности.
- mfParams — вектор параметров, которые специфицируют добавляемую функцию принадлежности.

Пример добавления трех функций принадлежности для одной входной переменной:

```
a=newfis('tipper');
a=addvar(a,'input','service',[0 10]);
a=addmf(a,'input',1,'poor','gaussmf',[1.5 0]);
a=addmf(a,'input',1,'good','gaussmf',[1.5 5]);
a=addmf(a,'input',1,'excellent','gaussmf',[1.5 10]);
plotmf(a,'input',1);
```

Результат этого примера изображен на рис. 4.1.

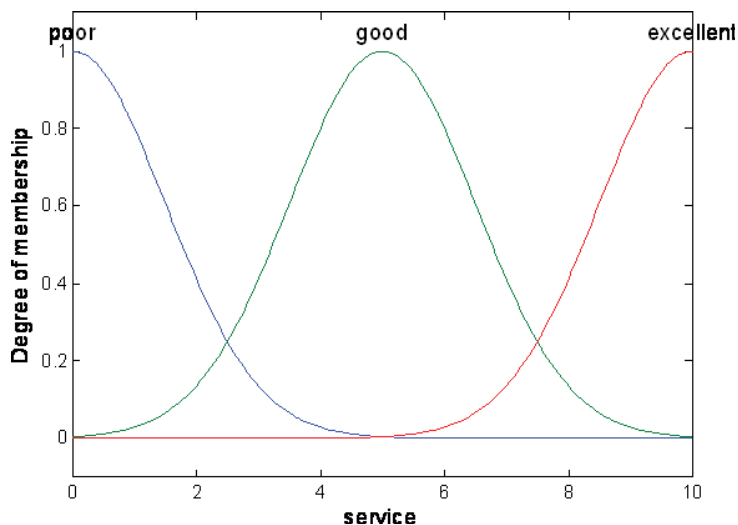


Рис. 4.1. Результат добавления трех функций принадлежности для одной входной переменной системы нечеткого вывода

См. также: `addrule`, `addvar`, `plotmf`, `rmmf`, `rmvar`.

## **addrule**

### *Назначение*

Добавление нового правила к системе нечеткого вывода FIS.

### *Синтаксис*

```
a=addrule(a, ruleList)
```

### *Описание*

Функция `addrule` имеет два аргумента. Первый аргумент — имя переменной FIS в рабочей области MATLAB. Второй аргумент `ruleList` - матрица, содержащая одну или несколько строк, каждая из которых представляет данное правило. Следует помнить, что для представления этой матрицы используется специальный формат. Если система вывода имеет  $m$  входных переменных и  $n$  выходных переменных, то соответствующая матрица `ruleList` должна иметь в точности  $m+n+2$  столбцов.

Первые  $m$  столбцов относятся к входным переменным системы. При этом номер столбца должен соответствовать индексу функции принадлежности для конкретной входной переменной. Следующие  $n$  столбцов относятся к выходным переменным системы вывода. При этом каждый столбец также имеет номер, который должен соответствовать индексу функции принадлежности для выходной переменной.

Столбец с номером  $m+n+1$  содержит вес, с которым применяется данное правило. Вес может принимать любое значение между нулем и единицей. В общем случае целесообразно задавать вес равным 1.

Столбец с номером  $m+n+2$  содержит число 1, если для подзаклучений данного правила используется нечеткий оператор AND (нечеткое И). Этот столбец содержит число 2, если для подусловий данного правила используется нечеткий оператор OR (нечеткое ИЛИ).

Пример для системы вывода с двумя входными переменными и одной выходной переменной

```
ruleList=[1 1 1 1 1  
          1 2 2 1 1];  
a=addrule(a, ruleList);
```

Для данного примера добавляемое правило может быть интерпретировано следующим образом: «Если входная переменная 1 имеет функцию принадлежности 1 и входная переменная 2 имеет функцию принадлежности 1, то выходная переменная 1 имеет функцию принадлежности 1».

*См. также:* `addmf`, `addvar`, `rmmf`, `rmvar`, `parsrule`, `showrule`.

## **addvar**

### *Назначение*

Добавление новой переменной к системе нечеткого вывода.

### *Синтаксис*

```
a=addvar(a,'varType','varName',varBounds)
```

### *Описание*

Функция использует четыре аргумента в следующем порядке:

- `a` — имя структуры FIS в рабочей области MATLAB.
- `'varType'` — строка, представляющая *тип* добавляемой переменной. Может принимать одно из двух значений: `'input'` или `'output'`.
- `'varName'` — строка, представляющая *имя* добавляемой переменной.
- `varBounds` — вектор, задающий границы области определения добавляемой переменной.

Номера или индексы переменным назначаются в том порядке, в котором они добавляются. Таким образом, первая добавленная переменная будет всегда являться переменной с номером 1. Нумерация входных и выходных переменных производится независимо друг от друга.

Пример добавления одной входной переменной с именем `'service'`

```
a=newfis('tipper');  
a=addvar(a,'input','service',[0 10]);  
getfis(a,'input',1)
```

В результате выполнения этой последовательности функций в окне команд появится следующий результат:

```
Name=                service  
NumMFs=              0  
MFLabels=  
Range=               [0 10]
```

*См. также:* `addmf`, `addrule`, `rmmf`, `rmvar`.

## **anfis**

### *Назначение*

Программа обучения для системы нейро-нечеткого вывода типа ANFIS.

### *Синтаксис*

```
[fismat, error1, stepsize]=anfis(trnData)
[fismat, error1, stepsize]=anfis{trnData, fismat}
[fismat1, error1, stepsize]=...
    anfis(trnData, fismat, trnOpt, dispOpt)
[fismat1, error1, stepsize, fismat2, error2]=...
    anfis(trnData, trnOpt, dispOpt, chkData)
[fismat1, error1, stepsize, fismat2, error2]=...
    anfis(trnData, trnOpt, dispOpt, chkData, optMethod)
```

### *Описание*

Это основная программа обучения для адаптивных систем нейро-нечеткого вывода ANFIS. Функция `anfis` использует алгоритм гибридного обучения для определения параметров систем нечеткого вывода типа Саджено. Этот алгоритм представляет собой комбинацию метода наименьших квадратов и метода убывания обратного градиента для обучения параметров функции принадлежности FIS с целью воспроизведения заданного множества обучающей выборки.

Функция `anfis` может также вызываться с необязательным аргументом с целью проверки правильности нечеткой модели. Типом проверки модели, который применяется в этом случае, является проверка совпадения данных модели, а аргумент представляет собой некоторое множество данных, которое называется множеством проверочных данных.

Функция `anfis` может быть использована со следующими аргументами:

- `trnData` — имя обучающей выборки. Представляет собой матрицу, все столбцы которой, кроме последнего, содержат входные данные. Последний столбец матрицы представляет собой обычный вектор выходных данных;
- `fismat` — имя системы нечеткого вывода FIS, используемой для того, чтобы предоставить функции `anfis` некоторое исходное множество функций принадлежности для обучения. Без этого аргумента функция `anfis` будет использовать по умолчанию функцию `genfis1` для задания некоторой системы нечеткого вывода FIS для выполнения последующего обучения.



Эта система нечеткого вывода FIS, задаваемая по умолчанию, будет иметь две функции принадлежности типа функции Гаусса, в то время как функция `anfis` вызывается только с одним аргументом. Если `fismat` представляет собой некоторое число (или вектор), то этот аргумент принимается равным номеру функции принадлежности (или вектору, компоненты которого представляют номера функций принадлежности соответствующих входных переменных). В этом случае оба эти аргумента функции `anfis` используются функцией `genfis1` для задания некоторой «правильной» структуры FIS до начала процесса обучения;

- `trnOpt` — вектор параметров обучения. Если любой из параметров обучения вводится как `NaN`, то будут использованы соответствующие параметры по умолчанию. Это следующие параметры обучения:
  - `trnOpt(1)` — количество циклов (эпох) обучения (по умолчанию `trnOpt(1)=10`);
  - `trnOpt(2)` — целевой уровень ошибки обучения (по умолчанию `trnOpt(2)=0`);
  - `trnOpt(3)` — величина исходного шага алгоритма обучения (по умолчанию `trnOpt(3)=0.01`);
  - `trnOpt(4)` — коэффициент уменьшения величины шага (по умолчанию `trnOpt(4)=0.9`);
  - `trnOpt(5)` — коэффициент увеличения величины шага (по умолчанию `trnOpt(5)=1.1`);
- `dispOpt` — вектор параметров отображения информации, которые определяют характер сообщений, выводимых в окно команд системы MATLAB в ходе процесса обучения. По умолчанию значение любого из этих параметров равно единице, что означает отображение соответствующей информации. Если значение некоторого параметра равно нулю, то соответствующая информация не отображается. Когда какой-либо из параметров отображения принимает значение `NaN`, то для этого параметра используется значение по умолчанию. Эти параметры отображения следующие:
  - `dispOpt(1)` — ANFIS-информация, такая как число входных и выходных функций принадлежности и т. д. (по умолчанию `dispOpt(1)=1`);
  - `dispOpt(2)` — ошибка обучения (по умолчанию `dispOpt(2)=1`);

- `dispOpt(3)` — величина шага обновления по каждому из параметров (по умолчанию `dispOpt(3)=1`);
- `dispOpt(4)` — конечные результаты (по умолчанию `dispOpt(4)=1`);
- `chkData` — имя необязательного множества проверочных данных для проверки правильности модели. Представляет собой матрицу, формат которой совпадает с форматом матрицы обучающей выборки;
- `optMethod` — применение необязательного метода оптимизации для обучения параметров функций принадлежности. Этот параметр может принимать одно из следующих значений: 1 — гибридный алгоритм, 0 — алгоритм обратного распространения ошибки. Гибридный алгоритм представляет собой комбинацию метода наименьших квадратов и метода обратного распространения ошибки. Гибридный алгоритм используется по умолчанию и вызывается всякий раз, когда соответствующий параметр принимает любое отличное от нуля значение.

Процесс обучения сети заканчивается, когда выполнено заданное число циклов обучения или достигнут целевой уровень ошибки обучения.

Если функция `anfis` вызывается с двумя или большим числом аргументов, а любой из необязательных аргументов задан в форме NaN или равен пустой матрице, то этот параметр принимает значение по умолчанию. Значения параметров по умолчанию могут быть изменены непосредственной модификацией файла `anfis.m`. Если пользователь не желает специфицировать некоторые из параметров, но желает специфицировать следующие за ними параметры, то на месте первых из них необходимо указать значение NaN или пустую матрицу.

Функция `anfis` имеет следующие выходные параметры:

- `fismat1` — имя структуры FIS, параметры которой представляют собой множество, полученное согласно критерию минимизации ошибки обучения;
- `error1` — массив (вектор) значений ошибок для обучающей выборки;
- `error2` — массив (вектор) значений ошибок для проверочной выборки;
- `stepsize` — массив значений величины шага в алгоритме обучения. Эти значения следуют в убывающем порядке, если мера ошибки подвергается двум чередующимся комбинациям

увеличения и уменьшения. Величины шага следуют в возрастающем порядке, если мера ошибки подвергается четырем последовательным уменьшениям;

- `fismat2` — имя структуры FIS, параметры которой представляют собой множество, полученное согласно критерию минимизации ошибки для проверочной выборки.

Пример использования функции `anfis` с тремя аргументами:

```
x=(0:0.1:10)';  
y=sin(2*x)./exp(x/5);  
trnData=[x y];  
numMFs=5;  
mfType='gbellmf';  
epoch_n=20;  
in_fismat=genfis1(trnData,numMFs,mfType);  
out_fismat=anfis(trnData,in_fismat,20);  
plot(x,y,x,evalfis(x,out_fismat));  
legend('Training Data','ANFIS Output');
```

Результат выполнения этой последовательности команд приведет к созданию и обучению гибридной сети ANFIS, графики обучающей выборки и выхода которой изображены на рис. 4.2.

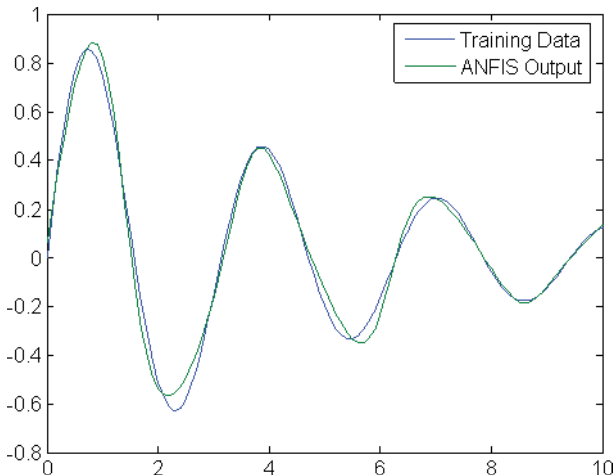


Рис. 4.2. Графики обучающей выборки и выхода гибридной сети с архитектурой ANFIS

См. также: `genfis1`, `anfisedit`.

## **anfisedit**

### *Назначение*

Открытие редактора гибридных нейронных сетей ANFIS.

### *Синтаксис*

`anfisedit('a')`

`anfisedit(a)`

`anfisedit`

### *Описание*

Функция `anfisedit` вызывает графический интерфейс редактора ANFIS, который позволяет пользователю загрузить некоторое множество данных и обучить гибридную сеть ANFIS. Эта функция в формате `anfisedit('a')` вызывает редактор с загруженной структурой FIS, которая хранится во внешнем файле с именем `a.fis`. Если функция используется в формате `anfisedit(a)`, то она вызывает редактор с загруженной структурой FIS, которая представляет собой переменную типа структуры с именем `a` в рабочей области MATLAB. Функция в формате `anfisedit` вызывает редактор без загруженной структуры FIS.

Редактор ANFIS имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д.

- Пункт меню **File** (Файл) редактора правил содержит такие же операции, что и соответствующий пункт меню редактора FIS.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** — отменяет выполнение последнего действия;
  - **FIS Properties...** — вызывает графический интерфейс редактора FIS;
  - **Membership Functions...** — вызывает графический интерфейс редактора функций принадлежности;
  - **Rules** — вызывает графический интерфейс редактора правил системы нечеткого вывода;
  - **Anfis...** — вызывает графический интерфейс редактора ANFIS.
- Пункт меню **View** (Вид) содержит следующие операции:
  - **Rules** — вызывает программу просмотра правил системы нечеткого вывода;
  - **Surface** — вызывает программу просмотра поверхности системы нечеткого вывода.

Пример вызова графического интерфейса редактора ANFIS с исходными данными из рабочей области:

```
anfisedit(out_fismat);
```

Результат вызова графического интерфейса редактора ANFIS с исходными данными из рабочей области изображен на рис. 4.3.

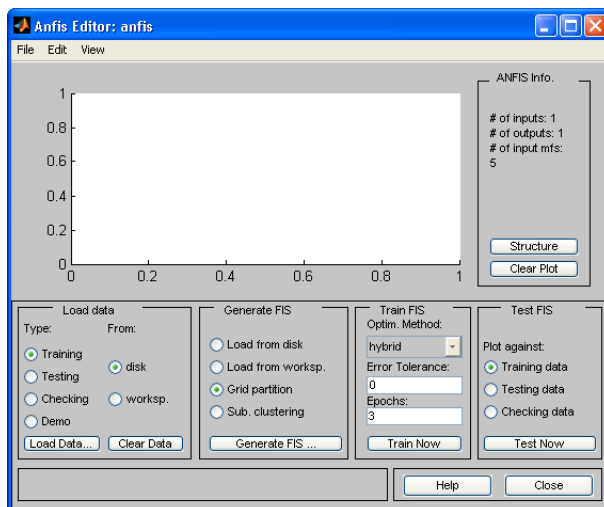


Рис. 4.3. Результат вызова графического интерфейса редактора ANFIS с загруженными исходными данными

*См. также:* fuzzy, mfedit, ruleedit, ruleview, surfview.

### **convertfis**

#### *Назначение*

Преобразует матрицу FIS Fuzzy Logic Toolbox версии 1.0 в структуру FIS версии 2.0.

#### *Синтаксис*

```
fis_new=convertfis(fis_old)
```

#### *Описание*

Функция `convertfis` использует в качестве аргумента матрицу FIS версии 1.0 и преобразует ее в структуру FIS версии 2.0.

### **defuzz**

#### *Назначение*

Выполнение дефазсификации функции принадлежности.

### Синтаксис

```
out=defuzz (x, mf, type)
```

### Описание

Функция `defuzz (x, mf, type)` позволяет получить число, которое является результатом дефаззификации функции принадлежности `mf` для соответствующей переменной `x`. При этом может быть использован один из методов дефаззификации, который определяется аргументом `type`. Этот аргумент может принимать одно из следующих значений:

`'centroid'` — метод центра тяжести для дискретного множества значений функции принадлежности

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)};$$

`'bisector'` — метод центра площади

$$\int_{Min}^y \mu(x) dx = \int_y^{Max} \mu(x) dx;$$

`'som'` — метод левого модального значения;

`'lom'` — метод правого модального значения;

`'mom'` — метод среднего максимума, определяемый как среднее арифметическое левого и правого модальных значений.

Метод среднего максимума выдает корректный результат только в случае унимодального нечеткого множества. В противном случае результат может оказаться даже не модальным значением соответствующего нечеткого множества.

Если для аргумента `type` не указано ни одно из перечисленных выше значений, то предполагается использование метода дефаззификации, который определен самим пользователем. Это можно сделать посредством добавления в `m`-файл этой функции `defuzz.m` соответствующих операторов, реализующих дополнительный метод дефаззификации.

Пример использования метода центра тяжести для дефаззификации функции принадлежности

```
x=-10:0.1:10;  
mf=trapmf(x, [-10 -8 -4 7]);  
xx=defuzz(x, mf, 'centroid');  
plot(x, mf)
```

```
line(xx,0,'linestyle','none','marker','*','...
'color','r');
```

В результате выполнения данной последовательности команд будет получено следующее изображение (рис. 4.4). На приведенном рисунке звездочкой отмечено значение  $xx=-3.2857$ , полученное при дефаззификации построенной функции принадлежности.

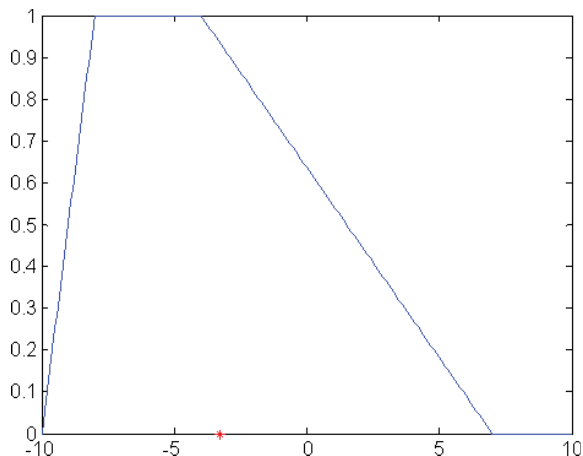


Рис. 4.4. График трапецевидной функции принадлежности и результат ее дефаззификации методом центраида

### **dsigmf**

#### *Назначение*

Встроенная П-образная функция принадлежности, которая является разностью двух сигмоидальных функций.

#### *Синтаксис*

```
y=dsigmf(x,[a1 c1 a2 c2])
```

#### *Описание*

Используемая в MATLAB сигмоидальная функция принадлежности зависит от двух параметров  $a$  и  $b$  и аналитически задается следующим выражением

$$f(x;a,b)=\frac{1}{1+e^{-a(x-b)}}.$$

Функция принадлежности `dsigmf` зависит от четырех параметров:  $a1, c1, a2, c2$  и представляет собой разность двух сигмоидальных функций:

$$f_1(x; a_1, c_1) - f_2(x; a_2, c_2).$$

Следует обратить внимание на правильную запись параметров, которые должны быть записаны сначала для первой функции, а потом для второй: [a1 c1 a2 c2].

Пример задания функции принадлежности dsignmf:

```
x=0:0.1:10;
y=dsigmf(x, [5 2 5 7]);
plot(x, y)
xlabel('dsigmf, P=[5 2 5 7]');
```

График соответствующей функции принадлежности изображен на рис. 4.5.

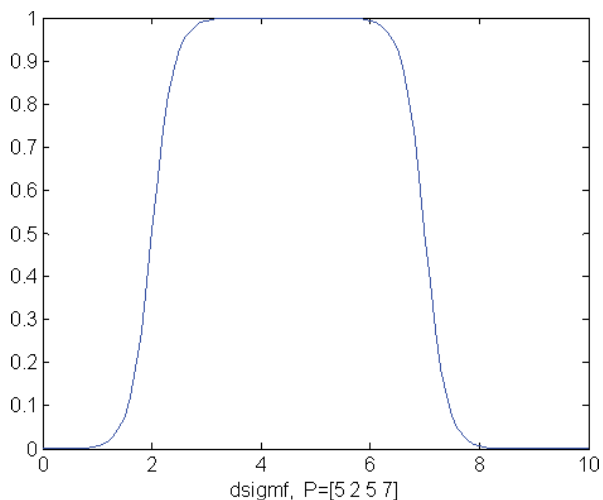


Рис. 4.5. График разности двух сигмоидальных функций

*См. также:* gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, pimf, psigmf, sigmf, smf, trapraf, trimf, zmf.

### **evalfis**

*Назначение*

Выполнение нечеткого вывода в системе FIS.

*Синтаксис*

```
output=evalfis(input, fismat)
```

```
output=evalfis(input, fismat, numPts)
```



```
[output, IRR, ORR, ARR]=evalfis (input, fismat)
[output, IRR, ORR, ARR]=evalfis (input, fismat, numPts)
```

### Описание

Функция `evalfis` имеет следующие аргументы:

- `input` — число или матрица, которая определяет входные переменные. Если `input` является матрицей порядка  $m \times n$ , где  $n$  — число входных переменных, то функция `evalfis` воспринимает каждую строку `input` как вектор, соответствующий отдельному набору значений входных переменных. Функция возвращает матрицу размерности  $m \times l$ , в которой каждая строка соответствует вектору значений выходных переменных, где  $l$  — число всех выходных переменных системы;
- `fismat` — имя структуры FIS, для которой выполняется нечеткий вывод;
- `numPts` — необязательный аргумент, задающий общее число выбранных точек, в которых оцениваются функции принадлежности в областях определения входных и выходных переменных. Если этот аргумент не указан, то по умолчанию используется значение `numPts=101`.

Функция `evalfis` возвращает значения, которые имеют следующий смысл:

- `output` — матрица выходов размерности  $m \times l$ , где  $m$  равно числу определенных ранее значений входных переменных, а  $l$  — числу выходных переменных системы нечеткого вывода FIS.

Следующие выходные значения функции являются необязательными и возвращаются функцией `evalfis` только в том случае, когда указан один набор значений входов ( $m = 1$ ):

- `IRR` — результат вычисления значений функций принадлежности для соответствующих входных переменных. Представляет собой матрицу размерности  $numRules \times n$ , где  $numRules$  — общее количество правил в системе FIS, а  $n$  — число входных переменных;
- `ORR` — результат вычисления значений функций принадлежности для соответствующих выходных переменных. Представляет собой матрицу размерности  $numPts \times numRules \times l$ , где  $numRules$  — общее количество правил в системе FIS, а  $l$  — число выходных переменных. Первые  $numRules$  столбцов этой матрицы соответствуют первой выходной переменной, следующие  $numRules$  столбцов этой матрицы соответствуют второй выходной переменной и т. д.;

- `ARR` — матрица размерности  $numPts \times l$  агрегированных значений в области задания каждой выходной переменной.

Если функция вызывается для одного набора значений входных переменных, то в качестве результата возвращается вектор значений выходных переменных. Если в структуре `FIS` определена одна выходная переменная, то возвращается скалярное значение этой выходной переменной после ее дефаззификации.

Пример использования функции `evalfis` для двух наборов значений входных переменных:

```
fismat=readfis('tipper');  
out=evalfis([2 1; 4 9],fismat)
```

Выполнение функции `evalfis` в этом случае приведет к получению следующего результата в окне команд:

```
out=  
    7.0169  
   19.6810
```

*См. также:* `ruleview`, `gensurf`.

## **evalmf**

### *Назначение*

Задание одной из встроенных функций принадлежности.

### *Синтаксис*

```
y=evalmf(x,mfParams,mfType)
```

### *Описание*

Функция `evalmf` позволяет задать любую из имеющихся в системе `MATLAB` функций принадлежности, тип которой устанавливается строкой `mfType`. При этом значение аргумента `x` устанавливает область определения задаваемой функции принадлежности (универсум), которая должна быть предварительно определена в виде *m*-файла, написанного пользователем, или предоставленного поставщиком. Строка `mfType` должна соответствовать имени *m*-файла, в котором определена задаваемая функция принадлежности. Второй аргумент `mfParams` представляет собой вектор параметров, определяющих задаваемую функцию принадлежности.

При необходимости пользователь может задать собственную функцию принадлежности. Для этого ее следует определить в отдельном *m*-файле как функцию и использовать имя этого файла в качестве третьего аргумента функции `evalmf`. Второй аргумент `mfParams` должен соответствовать параметрам этой функции принадлежности.

Пример использования функции `evalmf`:

```
x=0:0.1:10;  
y=evalmf(x, [4.5 1.5], 'tri2mf');  
plot(x, y)
```

Результат выполнения такой последовательности команд приведет к заданию треугольной функции принадлежности, график которой изображен на рис. 4.6.

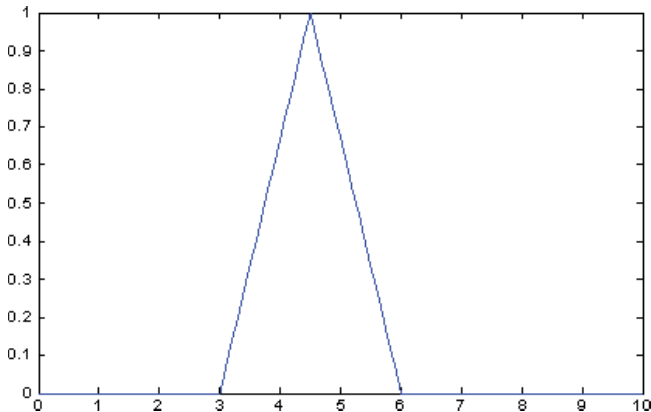


Рис. 4.6. График треугольной функции принадлежности, заданной с использованием функции `evalmf`

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `mf2mf`, `pimf`, `psigraf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

### **fcm**

#### *Назначение*

Кластеризация методом нечетких  $c$ -средних (алгоритм FCM).

#### *Синтаксис*

```
[center, U, obj_fcn]=fcm(data, cluster_n)  
[center, U, obj_fcn]=fcm(data, cluster_n, options)
```

#### *Описание*

Функция `fcm(data, cluster_n)` реализует нахождение нечеткой кластеризации заданного множества объектов алгоритмом FCM.

Входными аргументами данной функции являются:

`data` — матрица исходных данных кластеризации, каждая строка которой представляет информацию об одном объекте нечеткой кластеризации;

`cluster_n` — число искомым нечетких кластеров (должно быть больше единицы).

Выходными аргументами этой функции являются:

`center` — матрица центров искомым нечетких кластеров, каждая строка которой представляет координаты центра одного из нечетких кластеров;

`U` — матрица функций принадлежности искомого нечеткого разбиения;

`obj_fcn` — значения целевой функции на каждой из итераций работы алгоритма.

Функция `fcm(data, cluster_n, options)` использует переменную дополнительных аргументов `options` для управления процессом нечеткой кластеризации, для изменения критерия остановки работы алгоритма и/или отображения информации на экране монитора.

Дополнительные аргументы имеют следующее значение:

`options(1)` — экспоненциальный вес для расчета матрицы нечеткого разбиения (по умолчанию `options(1)=2`);

`options(2)` — максимальное число итераций (по умолчанию `options(2)=100`);

`options(3)` — параметр сходимости алгоритма (по умолчанию `options(3)=0.00001`);

`options(4)` — информация о текущей итерации, отображаемая на экране монитора (по умолчанию `options(4)=1`).

Если любое из значений дополнительных аргументов равно NaN, то для этого аргумента используется значение по умолчанию. Функция `fcm` заканчивает свою работу, когда алгоритм FCM выполнит максимальное количество итераций, или когда разность между значениями целевых функций на двух последовательных итерациях будет меньше заданного априори значения параметра сходимости алгоритма.

Пример решения задачи нечеткой кластеризации с помощью функции `fcm`:

```
data=rand(100,2);
[center,U,obj_fcn]=fcm(data,2);
maxU=max(U);
index1=find(U(1,:)==maxU);
index2=find(U(2,:)==maxU);
line(data(index1,1),data(index1,2),...
'linestyle','none','marker','*','color','g',...
'linewidth',2);
```

```

line(data(index2,1),data(index2,2),...
'linestyle','none','marker','*','color','r',...
'linewidth',2);

```

Выполнение этой последовательности команд приведет к графическому представлению результата нечеткой кластеризации, график которой изображен на рис. 4.7.

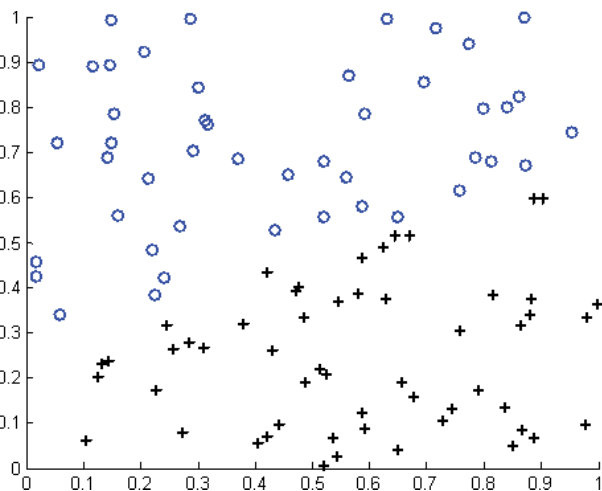


Рис. 4.7. Результат нечеткой кластеризации множества с использованием функции `fcm`

## **findcluster**

### *Назначение*

Вызов графического интерфейса для интерактивной кластеризации методами нечетких  $c$ -средних и субтрактивной кластеризации.

### *Синтаксис*

```

findcluster
findcluster('file.dat')

```

### *Описание*

Функция `findcluster` вызывает графический интерфейс GUI для выполнения нечеткой кластеризации алгоритмом FCM и/или субтрактивной (subtractive) кластеризации. Выбор метода нечеткой кластеризации осуществляется с помощью раскрывающегося списка **Methods**. Исходные данные загружаются в рабочую область из внешнего файла с помощью кнопки **Load Data**. Для каждого из методов

нечеткой кластеризации в соответствующих строках ввода установлены значения параметров алгоритмов по умолчанию. Данные значения могут быть изменены пользователем. Для этого необходимо установить курсор ввода в соответствующее поле и набрать нужные цифры с учетом допустимых значений параметров. Назначение этих параметров для алгоритма FCM описано при рассмотрении функции  $f_{cm}$ . Назначение этих параметров для алгоритма субтрактивной кластеризации описано при рассмотрении функции  $subclust$ .

Хотя графический интерфейс нечеткой кластеризации позволяет работать с многомерными исходными данными, он визуализирует данные измерений только по двум признакам. Выбор признаков осуществляется с помощью раскрывающихся списков **X-axis** и **Y-axis** в нижней части интерфейса. Например, для исходных данных из внешнего файла `clusterdemo.dat` можно изобразить два из трех признаков, выбрав в полях меню `data_1`, `data_2` или `data_3`. Нажатие на кнопку **Start** приводит к началу работы соответствующего алгоритма, после окончания его работы результаты отображаются на графике. Найденные центры кластеров можно сохранить во внешнем файле с целью их последующего анализа. Если функция используется в виде `findcluster('file.dat')`, то при вызове графического интерфейса в рабочую область автоматически загружаются данные

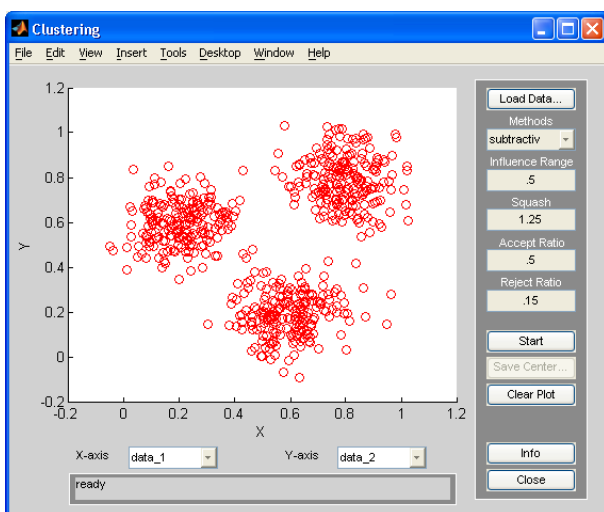


Рис. 4.8. Окно графического интерфейса функции нечеткой кластеризации `findcluster`

кластеризации из внешнего файла `file.dat`, а на графике отображаются значения матрицы данных для первых двух признаков.

Пример вызова графического интерфейса нечеткой кластеризации с исходными данными из внешнего файла `clusterdemo.dat`

```
findcluster('clusterdemo.dat');
```

Результат вызова графического интерфейса нечеткой кластеризации для этого случая изображен на рис. 4.8.

*См. также:* `fcm`, `subclust`.

## **fuzarith**

### *Назначение*

Выполнение операций нечеткой арифметики.

### *Синтаксис*

```
C=fuzarith(X,A,B,operator)
```

### *Описание*

Используя операции с нечеткими интервалами, функция `C=fuzarith(X,A,B,operator)` возвращает нечеткое множество `C` в качестве результата выполнения бинарной операции, определяемой строкой `operator`, над выпуклыми нечеткими множествами `A` и `B`. Нечеткие множества `A` и `B` предварительно должны быть заданы выпуклыми функциями принадлежности на универсуме `X`. Аргументами этой функции являются:

`A`, `B`, `X` — векторы одинаковой размерности;

`operator` — строка, принимающая одно из следующих значений: `'sum'`, `'sub'`, `'prod'`, `'div'`.

В качестве результата функция возвращает `C` — вектор-столбец той же размерности, что и вектор универсума `X`.

Нечеткое сложение может генерировать сообщение «деление на нуль» (`'divide by zero'`), но такое сообщение не влияет на правильность выполнения этой операции.

Пример использования функции `fuzarith`:

```
point_n=101;
min_x=-6;
max_x=9;
x=linspace(min_x,max_x,point_n)';
A=trapmf(x,[-2 -1 1 3]);
B=trimf(x,[1 2 3]);
C1=fuzarith(x,A,B,'sum');
subplot(4,1,1);
plot(x,A,'b',x,B,'m:',x,C1,'c');
```

```

title('нечеткое сложение C=A+B');
C2=fuzarith(x,A,B,'sub');
subplot(4,1,2);
plot(x,A,'b',x,B,'m:',x,C2,'c');
title('нечеткое вычитание C=A-B');
C3=fuzarith(x,A,B,'prod');
subplot(4,1,3);
plot(x,A,'b',x,B,'m:',x,C3,'c');
title('нечеткое умножение C=A*B');
C4=fuzarith(x,A,B,'div');
subplot(4,1,4);
plot(x,A,'b',x,B,'m:',x,C4,'c');
title('нечеткое деление C=A/B');

```

Результат выполнения данной последовательности команд изображен на рис. 4.9.

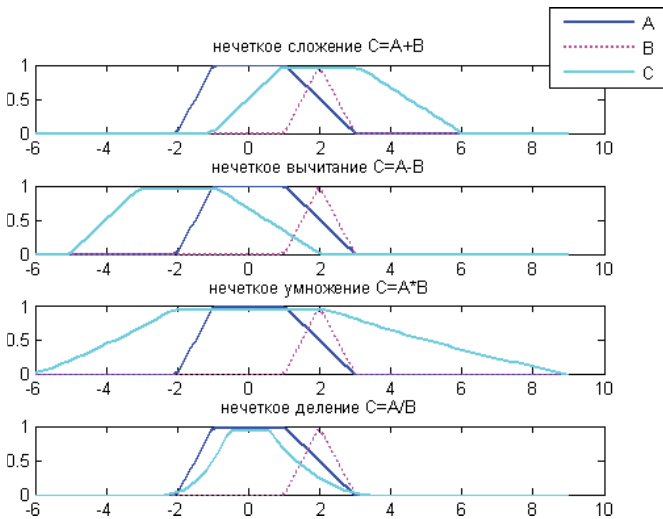


Рис. 4.9. Результаты выполнения операций с нечетким интервалом A и нечетким числом B

### **fuzblock**

#### *Назначение*

Блоки нечеткой логики из пакета Simulink.

#### *Синтаксис*

fuzblock



### Описание

Эта функция вызывает систему Simulink, в которой кроме демонстрационных примеров содержатся три специальных блока (рис. 4.10).

- Контроллер нечеткой логики (Fuzzy Logic Controller).
- Контроллер нечеткой логики с окном просмотра правил (Fuzzy Logic Controller with Rule Viewer).
- Блок функций принадлежности (Membership Function).

Последний из блоков позволяет просматривать правила в ходе моделирования, выполняемого в системе Simulink. После двойного щелчка левой кнопкой мыши на блоке контроллера нечеткой логики открывается диалоговое окно свойств этого блока. Окно содержит имя структуры FIS, которая находится в рабочей области системы MATLAB и которую предполагается моделировать в системе Simulink.

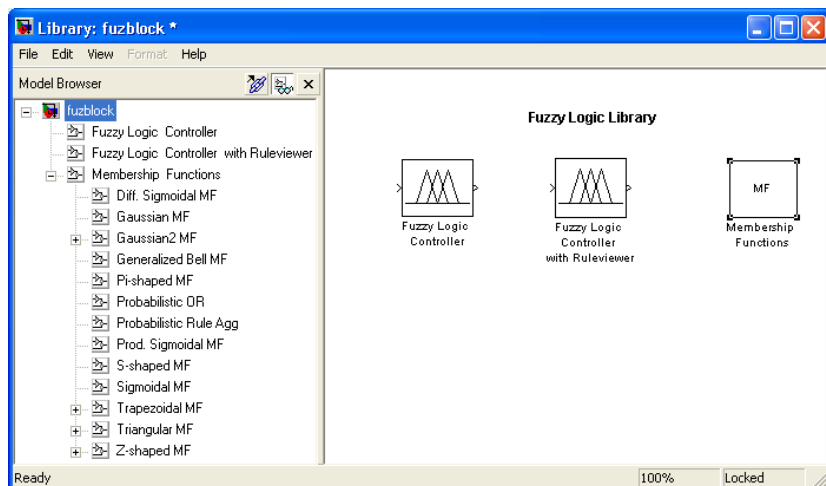


Рис. 4.10. Графический интерфейс блоков нечеткой логики системы Simulink

Чтобы открыть диалоговое окно свойств контроллера нечеткой логики с окном просмотра правил, необходимо:

1. Выполнить двойной щелчок левой кнопкой мыши на этом блоке, в результате чего откроется диалоговое окно свойств контроллера нечеткой логики.

2. Выполнить двойной щелчок левой кнопкой мыши на втором блоке контроллера нечеткой логики.

Если система нечеткого вывода имеет несколько входов, эти входы должны быть соединены вместе до размещения их в блоке контроллера нечеткой логики или в блоке контроллера нечеткой логики с окном просмотра правил. Аналогично, если система нечеткого вывода имеет несколько выходов, то выходы должны быть соединены вместе до размещения их в блоке контроллера нечеткой логики или в блоке контроллера нечеткой логики с окном просмотра правил.

*См. также:* `sffis`, `ruleview`.

## **fuzdemos**

### *Назначение*

Список всех демонстрационных примеров в пакете Fuzzy Logic Toolbox.

### *Синтаксис*

`fuzdemos`

*Описание.* Данная функция вызывает графический интерфейс, который позволяет выбрать для анализа любой из демонстрационных примеров, поставляемых с пакетом Fuzzy Logic Toolbox. С демонстрационными примерами можно также познакомиться, используя следующие функции командной строки:

`defuzzdm` — пример методов дефаззификации;

`fcmdemo` — пример кластеризации точек на плоскости методом нечетких  $c$ -средних FCM;

`gasdemo` — пример применения системы для эффективного управления топливом с использованием метода субтрактивной кластеризации;

`juggler` — пример просмотра правил для задачи жонглирования мяча;

`invkine` — пример решения обратной задачи кинематики захвата механической руки робота;

`irisfcm` — пример кластеризации точек в четырехмерном пространстве методом нечетких  $c$ -средних FCM;

`noisedm` — пример адаптивного устранения шума;

`slbb` — пример управления движением мяча на качающейся балке с помощью средств пакета Simulink;

`slcp` — пример управления инвертированным маятником с помощью средств пакета Simulink;

sltank — пример управления уровнем воды с помощью средств пакета Simulink;

sltankrule — пример управления уровнем воды с использованием программы просмотра правил с помощью средств пакета Simulink;

sltbu — пример нечеткого управления движением грузового автомобиля (только Simulink).

## **fuzzy**

### *Назначение*

Вызов редактор систем нечеткого вывода FIS.

### *Синтаксис*

fuzzy

fuzzy (fismat)

### *Описание*

Функция предоставляет пользователю возможность редактировать на высоком уровне свойства системы нечеткого вывода, такие, как число входных и выходных переменных, используемый метод дефаззификации и т. д.

Редактор систем нечеткого вывода FIS (или просто редактор FIS) обладает графическим интерфейсом и позволяет вызывать все другие редакторы и программы просмотра систем нечеткого вывода. Графический интерфейс редактора обладает максимальным удобством и гибкостью, необходимой для интерактивной работы с отдельными компонентами системы нечеткого вывода.

*См. также:* mfeedit, ruleedit, ruleview, surfview, anfisedit.

## **gauss2mf**

### *Назначение*

Встроенная П-образная функция принадлежности, которая является комбинацией двух функций Гаусса.

### *Синтаксис*

y=gauss2mf(x, [sig1 c1 sig2 c2])

### *Описание*

Функция Гаусса описывает плотность нормального распределения, определяется выражением

$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}},$$

где  $\sigma$  — среднее квадратическое отклонение,  $c$  — математическое ожидание. Функция gauss2mf является комбинацией двух таких

функций. Первая функция Гаусса задается параметрами `sig1`, `c1` и определяет форму кривой функции принадлежности слева от модального значения. Вторая функция Гаусса задается параметрами `sig2`, `c2` и определяет форму кривой функции принадлежности справа от модального значения.

Если выполняется неравенство  $c1 < c2$ , то функция `gauss2mf` достигает своего максимального значения, равного единице. В противном случае ее максимальное значение меньше единицы. Параметры должны быть указаны в следующем порядке: `[sig1 c1 sig2 c2]`.

Пример заданий функции принадлежности `gauss2mf`

```
x=0:0.1:10;  
y1=gauss2mf(x,[2 4 1 8]);  
y2=gauss2mf(x,[2 6 1 6]);  
y3=gauss2mf(x,[2 7 1 5]);  
y4=gauss2mf(x,[2 8 1 4]);  
plot(x,y1,x,y2,x,y3,x,y4)
```

Графики соответствующих функций принадлежности, полученные в результате выполнения указанной последовательности команд, изображены на рис. 4.11.

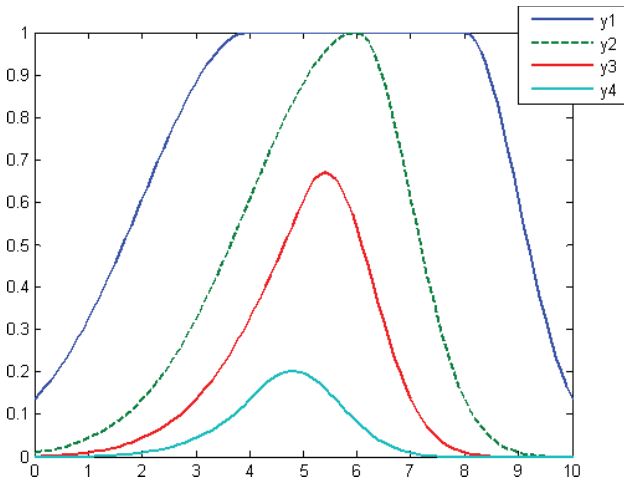


Рис. 4.11. Графики нескольких функций принадлежности, являющихся комбинацией двух функций Гаусса

*См. также:* `dsigmf`, `gaussmf`, `gbellmf`, `evalmf`, `mf2mf`, `pimf`, `psigmf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

## **gaussmf**

### *Назначение*

Встроенная П-образная функция принадлежности типа функции Гаусса.

### *Синтаксис*

```
y=gaussmf(x, [sig c])
```

### *Описание*

Симметричная функция Гаусса описывает плотность нормального распределения и определяется выражением

$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}},$$

где  $\sigma$  — среднее квадратическое отклонение,  $c$  — математическое ожидание. Эта функция имеет два параметра: `sig` и `c`, которые задаются в форме вектора `[sig c]`.

Пример задания функции принадлежности `gaussmf`:

```
x=0:0.1:10;  
y=gaussmf(x, [2 4]);  
plot(x, y);
```

График функции принадлежности изображен на рис. 4.12.

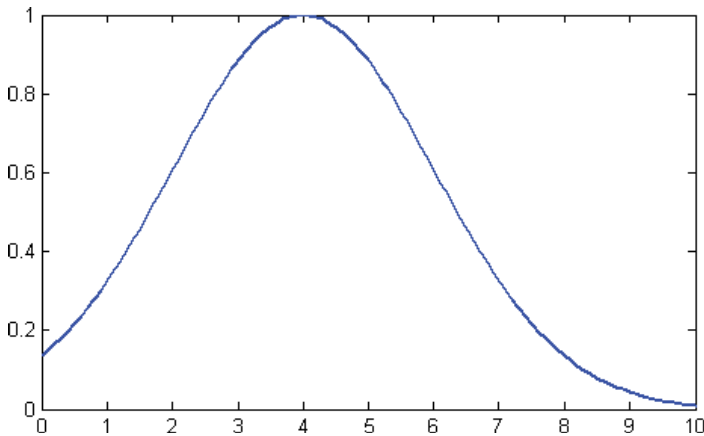


Рис. 4.12. График гауссовой функции принадлежности `gaussmf(x, [2 4])`

*См. также:* `dsigmf`, `gauss2mf`, `gbellmf`, `evalmf`, `mf2mf`, `pimf`, `psigmf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

## **gbellmf**

### *Назначение*

Встроенная П-образная функция принадлежности типа колоколообразной кривой.

### *Синтаксис*

```
y=gbellmf(x, params)
```

### *Описание*

Данная обобщенная функция принадлежности типа колоколообразной кривой (П — образная функция принадлежности) определяется выражением

$$f(x;a,b,c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

и зависит от трех упорядоченных параметров  $a < b < c$ , при этом параметр  $b$  должен быть обязательно положительным. Такие параметры являются вторым аргументом функции и задаются в форме вектора `params`, компоненты которого записываются в следующем порядке: `[a b c]`.

Пример задания функции принадлежности `gbellmf`

```
x=0:0.1:10;  
y=gbellmf(x, [2 3 6]);  
plot(x,y);
```

График функции принадлежности изображен на рис. 4.13.

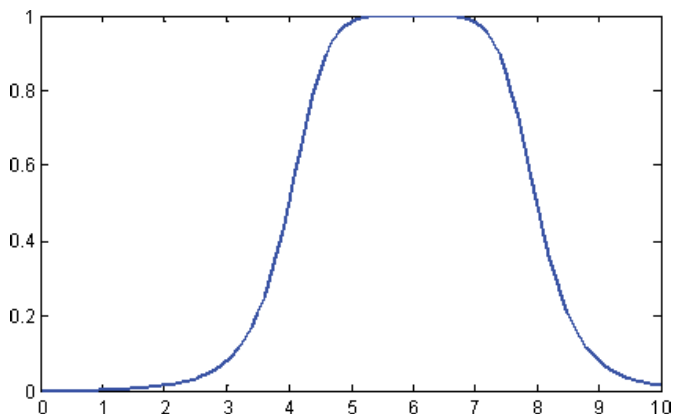


Рис. 4.13. График колоколообразной функции принадлежности `gbellmf(x, [2 3 6])`

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `evalmf`, `mf2mf`, `pimf`, `psigmf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

## **genfis1**

### *Назначение*

Генерирует структуру системы нечеткого вывода FIS на основе экспериментальных данных без кластеризации этих данных.

### *Синтаксис*

```
fismat=genfis1(data)  
fismat=genfis1(data, numMFs, inmfstype, outmfstype)
```

### *Описание*

Функция `genfis1` генерирует систему нечеткого вывода FIS типа Саджено, которая может быть использована в качестве исходной структуры для инициализации параметров функций принадлежности в процессе обучения гибридной сети `anfis`. Функция в формате `genfis1(data, numMFs, inmfstype, outmfstype)` генерирует структуру FIS на основе некоторой обучающей выборки с именем `data` с использованием некоторого сеточного разбиения этих данных без их кластеризации.

Аргументами функции являются:

`data` — матрица данных (обучающая выборка), последний столбец которой соответствует единственной выходной переменной, а остальные столбцы соответствуют входным переменным;

`numMFs` — вектор, компоненты которого определяют количество функций принадлежности у каждой из входных переменных. Если пользователю необходимо задать одинаковое количество функций принадлежности для всех входных переменных, то достаточно задать аргумент `numMFs` как обычное число (скаляр);

`inmfstype` — массив строковых значений, каждое из которых специфицирует тип функции принадлежности для соответствующей входной переменной. Аналогично, если пользователю необходимо определить один и тот же тип функции принадлежности для всех входных переменных, то достаточно задать этот аргумент как простую строку;

`outmfstype` — строка, которая специфицирует тип функции принадлежности выходной переменной. Поскольку используется система нечеткого вывода типа Саджено, то существует только одна выходная переменная. Тип функции принадлежности выходной переменной может быть или линейным, или константой.

Количество функций принадлежности для выходной переменной должно быть равно количеству правил, генерируемых функцией `genfis1`. По умолчанию значение аргумента `numMFs` равно двум, а тип функций принадлежности входных и выходной переменных равен `'gbellmf'`. Эти значения используются всякий раз, когда функция `genfis1` вызывается без последних трех аргументов.

Пример использования функции `genfis1`:

```
data=[rand(10,1) 10*rand(10,1)-5 rand(10,1)];
numMFs=[3 7];
mfType=str2mat('pimf','trimf');
fismat=genfis1(data, numMFs, mfType);
[x,mf]=plotmf(fismat,'input',1);
subplot(2,1,1),plot(x,mf);
xlabel('input1(pimf)');
[x,mf]=plotmf(fismat,'input',2);
subplot(2,1,2),plot(x,mf);
xlabel('input2(trimf)');
```

Результат выполнения функции `genfis1` изображен на рис. 4.14.

См. также: `anfis`.

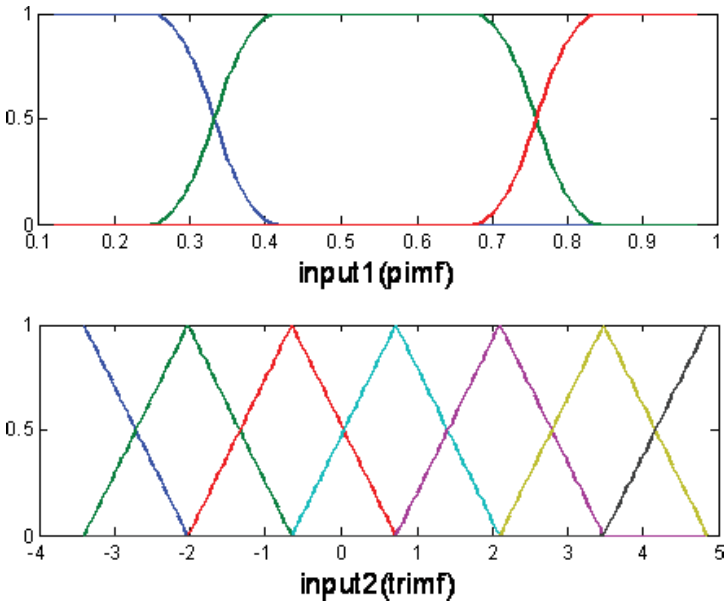


Рис. 4.14. Результат выполнения функции `genfis1`



## **genfis2**

### *Назначение*

Генерация структуры системы нечеткого вывода FIS с использованием метода субтрактивной кластеризации.

### *Синтаксис*

```
fismat=genfis2 (Xin, Xout, radii)
fismat=genfis2 (Xin, Xout, radii, xBounds)
fismat=genfis2 (Xin, Xout, radii, xBounds, options)
```

### *Описание*

Функция `genfis2` генерирует структуру системы нечеткого вывода FIS методом субтрактивной кластеризации на основе экспериментальных данных, заданных в виде двух отдельных множеств входных и выходных данных. Если задана только одна выходная переменная, функция `genfis2` может быть использована для генерации структуры адаптивной системы нейро-нечеткого вывода (FIS типа Саджжено) с целью ее последующего обучения на основе предварительной субтрактивной кластеризации экспериментальных данных.

Функция `genfis2` выполняет генерацию структуры посредством спецификации множества правил, которые соответствуют исходному множеству данных. При формировании правил нечетких продукций для определения количества правил и функций принадлежности их условий на первом этапе используется функция `subclust`. На втором этапе методом наименьших квадратов определяются заключения правил нечетких продукций. В итоге функция `genfis2` возвращает структуру системы нечеткого вывода FIS, база правил которой охватывает все пространство экспериментальных данных.

Аргументами этой функции являются:

- `Xin` — матрица, каждая строка которой представляет исходные значения входных переменных;
- `Xout` — матрица, каждая строка которой представляет исходные значения выходных переменных;
- `radii` — вектор, который задает интервал (радиус) влияния центров кластеров по каждой координате векторов исходных данных в предположении, что все данные находятся внутри некоторого единичного гиперкуба;

Например, если используются трехмерные исходные данные, т. е. матрица `Xin` имеет два столбца, а матрица `Xout` — один столбец, то значения `radii=[0.5 0.4 0.3]` задают относительные интервалы влияния в пределах рассматриваемого гиперкуба по каждой из трех переменных: по первой входной

переменной 0.5, по второй входной переменной — 0.4, и по выходной переменной — 0.3. Если параметр `radii` является скаляром, то соответствующее значение относится к каждой из переменных нечеткой модели, т. е. центр каждого кластера будет иметь сферическую окрестность влияния заданного радиуса;

- `xBounds` — необязательная матрица размерности  $2 \times n$ , специфицирующая диапазоны значений по каждой из входных и выходных переменных нечеткой модели (здесь  $n$  — общее количество входных и выходных переменных). Спецификация необходима для преобразования исходных значений в значения единичного гиперкуба. Первая строка матрицы `xBounds` задает минимальные значения диапазона значений по каждой из осей многомерных данных, а вторая строка дает максимальные значения диапазона значений по каждой их осей многомерных данных.

Например, значения матрицы `xBounds=[-10 0 -1; 10 50 1]` определяют, что исходные данные по первой оси нормализуются в интервале  $[-10, 10]$ , исходные данные по второй оси — в интервале  $[0, 1]$  и исходные данные по третьей оси нормализуются в интервале  $[-1, 1]$ . Если матрица `xBounds` не задана, то диапазоны нормализации значений определяются по умолчанию — как минимальные и максимальные значения каждой их переменных нечеткой модели, присутствующие в исходных данных;

- `options` — необязательный вектор, который специфицирует параметры алгоритма генерации структуры FIS аналогично параметрам функции `subclust`.

Пример использования функции `genfis2` с минимальным числом аргументов:

```
fismat=genfis2(Xin,Xout,0.5);
```

Пример использования функции `genfis2` с нормализацией исходных данных:

```
fismat=genfis2(Xin,Xout,0.5,[-10 -5 0; 10 5 20]);
```

В данном случае задается параметр нормализации для преобразования значений матриц `Xin` и `Xout` в значения единичного гиперкуба, т. е. в интервал  $[0, 1]$  по каждой из переменных. Поскольку для рассматриваемого примера матрица `Xin` имеет два столбца, а матрица `Xout` — один столбец, то выполняется шкалирование по каждой

из переменных. Значения первой входной переменной нормализуются в интервале  $[-10, 10]$ , второй входной переменной — в интервале  $[-5, 5]$ , а выходной переменной — в интервале  $[0, 20]$ .

*См. также:* subclust.

## **gensurf**

### *Назначение*

Генерация поверхности нечеткого вывода FIS.

### *Синтаксис*

```
gensurf (fis)
```

```
gensurf (fis, inputs, output)
```

```
gensurf (fis, inputs, output, grids)
```

```
gensurf (fis, inputs, output, grids, reinput)
```

### *Описание*

Функция `gensurf (fis)` позволяет получить изображение поверхности нечеткого вывода для одной из выходных переменных структуры системы нечеткого вывода FIS с именем `fis`. В этом формате для построения поверхности вывода используются первые две входные переменные и первая выходная переменная структуры `fis`.

Функция `gensurf (fis, inputs, output)` во втором виде позволяет получить изображение поверхности нечеткого вывода для одной или двух входных переменных, номера которых задаются вектором `inputs`, и одной выходной переменной с номером `output` для структуры системы нечеткого вывода FIS с именем `fis`.

Функция `gensurf (fis, inputs, output, grids)` в третьем виде позволяет получить изображение сетки на рисунке поверхности нечеткого вывода для первой горизонтальной переменной и второй вертикальной входной переменной. При этом номера переменных задаются вектором `grids`.

Функция `gensurf (fis, inputs, output, grids, reinput)` в четвертом виде позволяет получить изображение поверхности нечеткого вывода для более чем двух выходных переменных. В данном случае вектор `reinput` задает номера входных переменных, которые рассматриваются как неизменяющиеся.

Функция `gensurf` может быть также вызвана в формате `[x, y, z]=gensurf (...)`. В этом случае она возвращает значения переменных, которые определяют поверхность вывода и могут быть использованы для изображения этой поверхности средствами графики MATLAB.

Пример использования функции `gensurf`:

```
a=readfis('tipper');  
gensurf(a);
```

График полученной поверхности решений системы нечеткого вывода изображен на рис. 4.15.

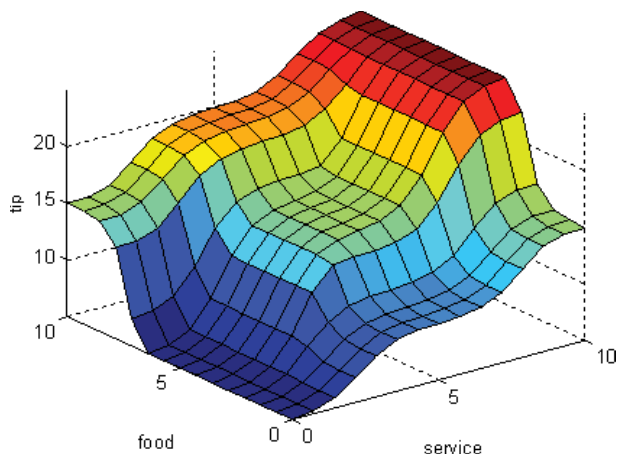


Рис. 4.15. График поверхности решений системы нечеткого вывода, полученный с помощью функции gensurf

*См. также:* evalfis, surfview.

### **getfis**

#### *Назначение*

Отображение различных свойств системы нечеткого вывода FIS.

#### *Синтаксис*

```
getfis(a)  
getfis(a,'fisprop')  
getfis(a,'vartype',varindex,'varprop')  
getfis(a,'vartype',varindex,'mf',mfindex)  
getfis(a,'vartype',varindex,'mf',mfindex,'mfprop')
```

#### *Описание*

Это основная функция, предназначенная для отображения в окне команд отдельных свойств структуры системы нечеткого вывода FIS. Функция getfis может использоваться в одном из пяти указанных форматов со следующими аргументами:

- a — имя структуры FIS в рабочей области MATLAB;

- 'vartype' — строка с указанием типа отображаемой переменной (input или output);
- varindex — целое число, соответствующее номеру отображаемой переменной (например, 1 для первой из входных переменных);
- 'mf' — строка, указывающая на отображение информации о функции принадлежности;
- mfindex — целое число, соответствующее номеру отображаемой функции принадлежности (например, 1 для первой из функций принадлежности).

Пример использования функции `getfis` с одним аргументом:

```
a=readfis('tipper');
getfis(a)
```

В этом случае в окно команд будет выведена следующая информация:

```
Name=          tipper
Type=          mamdani
NumInputs=    2
InLabels=
service
food
NumOutputs=   1
OutLabels =
          tip
NumRules=     3
AndMethod=    min
OrMethod=     max
ImpMethod=    min
AggMethod=    max
DefuzzMethod= centroid
ans=
          tipper
```

Пример использования функции `getfis` с двумя аргументами:

```
getfis(a,'type')
```

В этом случае в окно команд будет выведена следующая информация:

```
ans=
          mamdani
```

Пример использования функции `getfis` с тремя аргументами:

```
getfis(a,'input',1)
```

В этом случае в окно команд будет выведена следующая информация:

```
Name=      service
NumMFs=    3
MFLabels=
           poor
           good
           excellent
Range=     [0 10]
ans=
          [ ]
```

Пример использования функции `getfis` с четырьмя аргументами:  
`getfis(a, 'input', 1, 'name')`

В этом случае в окно команд будет выведена следующая информация:

```
ans=
      service
```

Пример использования функции `getfis` с пятью аргументами:  
`getfis(a, 'input', 1, 'mf', 2)`

В этом случае в окно команд будет выведена следующая информация:

```
Name = good
Type = gaussmf
Params = [1.5 5]
ans =
          [ ]
```

Пример использования функции `getfis` с шестью аргументами:  
`getfis(a, 'input', 1, 'mf', 2, 'name')`

В этом случае в окно команд будет выведена следующая информация:

```
ans =
      good
```

*См. также:* `setfis`, `showfis`.

## **mam2sug**

### *Назначение*

Преобразование системы нечеткого вывода FIS типа Мамдани в систему нечеткого вывода FIS типа Саджено.

### *Синтаксис*

```
sug_fis=mam2sug(mam_fis)
```

### Описание

Функция `mam2sug` преобразует структуру FIS типа Мамдани с именем `mam_fis` (не обязательно с единственным выходом) в структуру FIS типа Саджено с именем `sug_fis`. Получаемая в результате система нечеткого вывода типа Саджено имеет функции принадлежности, каждая из которых имеет некоторое постоянное значение. Константы определяются методом центра масс для соответствующих функций исходной системы нечеткого вывода типа Мамдани. Условия правил нечеткого вывода остаются без изменений.

Пример использования функции `mam2sug` для преобразования структуры FIS типа Мамдани из внешнего файла `tipper.fis` в структуру FIS типа Саджено:

```
mam_fismat=readfis('tipper.fis');
sug_fismat=mam2sug(mam_fismat);
subplot(2,2,1);
gensurf(mam_fismat,[1 2],1);
title('Система нечеткого вывода типа Мамдани');
subplot(2,2,2);
gensurf(sug_fismat,[1 2],1);
title('Система нечеткого вывода типа Саджено');
```

Результат выполнения данной последовательности команд изображен на рис. 4.16.

Система нечеткого вывода типа Мамдани Система нечеткого вывода типа Саджено

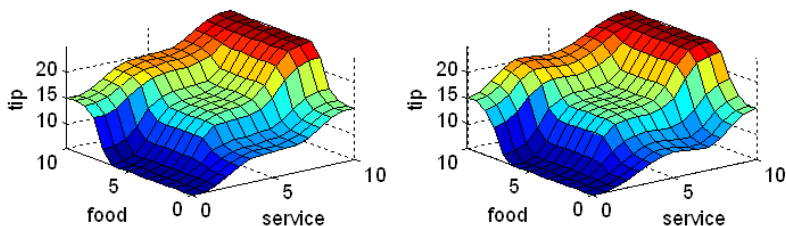


Рис. 4.16. Преобразование системы нечеткого вывода типа Мамдани в систему нечеткого вывода типа Саджено

### **mf2mf**

#### Назначение

Преобразование параметров двух функций принадлежности.

#### Синтаксис

```
outParams=mf2mf(inParams,inType,outType)
```

### Описание

Эта функция преобразует одну из встроенных функций принадлежности в любую другую встроенную функцию принадлежности. Подобное преобразование выполняется на уровне параметров соответствующих функций принадлежности. В принципе функция `mf2mf` сохраняет точки симметрии старой и новой функций принадлежности. Однако результат выполнения преобразования может привести к потере информации при выполнении обратного преобразования новой функции принадлежности в функцию принадлежности исходного типа. В этом случае преобразованная функция принадлежности может по своему виду отличаться от первоначальной функции принадлежности.

Функция `mf2mf` имеет следующие входные аргументы:

- `inParams` — параметры исходной функции принадлежности, которая подлежит преобразованию;
- `inType` — строка с именем типа исходной функции принадлежности, которая подлежит преобразованию;
- `outType` — строка с именем типа новой функции принадлежности, в который преобразуется исходная функция принадлежности.

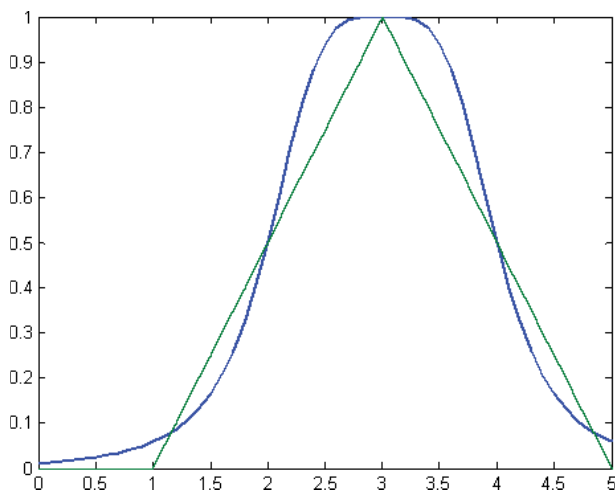


Рис. 4.17. Графики П-образной исходной функции принадлежности и треугольной функции принадлежности, полученной в результате преобразования с помощью функции `mf2mf`



Пример использования функции `mf2mf` для преобразования структуры FIS типа Мамдани из внешнего файла `tipper.fis` в структуру FIS типа Саджено

```
x=0:0.1:5;
mfp1=[1 2 3];
mfp2=mf2mf(mfp1,'gbellmf','trimf');
plot(x,gbellmf(x,mfp1),x,trimf(x,mfp2))
```

Результат выполнения этой последовательности команд изображен на рис. 4.17.

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `evalmf`, `pimf`, `psigmf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

### **mfedit**

#### *Назначение*

Вызов графического интерфейса редактора функций принадлежности системы нечеткого вывода FIS.

#### *Синтаксис*

```
mfedit('a')
mfedit(a)
mfedit
```

#### *Описание*

Эта функция, записанная в формате `mfedit('a')`, вызывает редактор функций принадлежности, который позволяет пользователю в графическом режиме анализировать и модифицировать все функции принадлежности некоторой структуры FIS, сохраненной во внешнем файле с именем `a.fis`. Функция в формате `mfedit(a)` работает с переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `a`.

Функция в формате `mfedit` просто вызывает редактор функций принадлежности без загрузки какой-либо структуры FIS. Для каждой функции принадлежности можно изменить ее имя, тип и параметры.

*См. также:* `fuzzy`, `ruleedit`, `ruleview`, `surfview`.

### **newfis**

#### *Назначение*

Создание новой системы нечеткого вывода FIS.

#### *Синтаксис*

```
a=newfis(fisName, fisType, andMethod, orMethod, ...
impMethod, aggMethod, defuzzMethod)
```

### *Описание*

Эта функция предназначена для создания новых структур систем нечеткого вывода FIS. Функция `newfis` имеет семь аргументов и возвращает единственное значение — структуру FIS с именем `a`. Аргументами функции являются:

- `fisName` — строка с именем создаваемой структуры FIS, при этом также создается файл с именем `fisName.fis`;
- `fisType` — тип системы нечеткого вывода FIS;
- `andMethod`, `orMethod`, `impMethod`, `aggMethod`, `defuzzMethod` — соответственно задают методы для выполнения нечетких логических операций AND, OR, импликации, агрегирования и дефаззификации. Если эти аргументы не указаны, то задаются соответствующие им значения по умолчанию.

Пример использования функции `newfis` с аргументами, принятыми по умолчанию

```
a=newfis('newsys');  
getfis(a)
```

В этом случае в командное окно будет выведена следующая информация:

```
Name=          newsys  
Type=          mamdani  
NumInputs=    0  
InLabels=       
NumOutputs=   0  
OutLabels=      
NumRules=     0  
AndMethod=    rain  
OrMethod=     max  
ImpMethod=    min  
AggMethod=    max  
DefuzzMethod= centroid  
ans=            
                newsys
```

*См. также:* `readfis`, `writefis`.

### **parsrule**

#### *Назначение*

Грамматический анализ нечетких правил.

#### *Синтаксис*

```
fis2=parsrule(fis,txtRuleList)
```

```
fis2=parsrule (fis,txtRuleList, ruleFormat)
fis2=parsrule (fis,txtRuleList, ruleFormat, lang)
```

### *Описание*

Эта функция выполняет грамматическую проверку текста в правилах нечетких продукций для структуры нечеткого вывода в рабочей области системы MATLAB. Функция может иметь следующие аргументы:

- `fis` — имя структуры системы нечеткого вывода в рабочей области;
- `txtRuleList` — строка текста, которая определяет некоторое правило нечеткой продукции;
- `ruleFormat` — строка, определяющая формат представления правил нечетких продукций, которая может принимать одно из следующих значений:
  - `'verbose'` — текстовый формат (по умолчанию);
  - `'symbolic'` — символический формат;
  - `'indexed'` — цифровой формат;
- `lang` — строка, определяющая язык записи правил нечетких продукций, которая может принимать одно из следующих значений:
  - `'english'` — английский язык;
  - `'francais'` — французский язык;
  - `'deutsch'` — немецкий язык.

Пример использования функции `parsrule` с тремя аргументами

```
a=readfis('tipper');
ruleTxt='if service is poor then tip is generous';
a2=parsrule(a,ruleTxt,'verbose');
showrule(a2)
```

В этом случае в окно команд будет выведена следующая информация:

```
ans =
1. If (service is poor) then (tip is generous) (1)
См. также: addrule, ruleedit, showrule.
```

### **pimf**

#### *Назначение*

Встроенная П-образная функция принадлежности.

#### *Синтаксис*

```
y=pimf(x,[a b c d])
```

### Описание

Эта функция принадлежности определяется как алгебраическое произведение двух сплайн-функций:

$$f_{\Pi}(x; a, b, c, d) = f_S(x; a, b) \cdot f_Z(x; c, d),$$

где  $S$  — и  $Z$  — образные функции задаются формулами

$$f_S(x; a, b) = \begin{cases} 0, & x \leq a, \\ 2\left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2}, \\ 1 - 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x \leq b, \\ 1, & x > b, \end{cases}$$
$$f_Z(x; c, d) = \begin{cases} 1, & x \leq c, \\ 1 - 2\left(\frac{x-c}{d-c}\right)^2, & c < x \leq \frac{c+d}{2}, \\ 2\left(\frac{d-x}{d-c}\right)^2, & \frac{c+d}{2} < x \leq d, \\ 0, & x > d. \end{cases}$$

Вторым аргументом функции является вектор параметров  $[a \ b \ c \ d]$ . Параметры  $a$  и  $b$  характеризуют  $Z$  — образную кривую, а параметры  $c$  и  $d$  характеризуют  $S$  — образную кривую.

Пример задания функции принадлежности `pimf`

```
x=0:0.1:10;  
y=pimf(x, [1 4 5 9]);  
plot(x, y)
```

График этой функции принадлежности изображен на рис. 4.18.

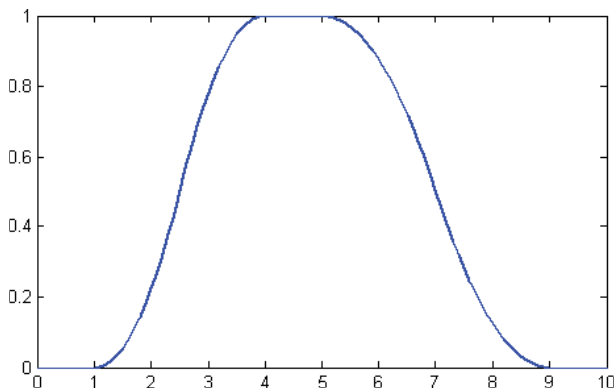


Рис. 4.18. График П-образной функции принадлежности `pimf(x, [1 4 5 9])`

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `evalmf`, `mf2mf`, `psigmf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

## **plotfis**

### *Назначение*

Изображение диаграммы системы нечеткого вывода FIS.

### *Синтаксис*

```
plotfis (fismat)
```

### *Описание*

Эта функция изображает диаграмму верхнего уровня системы нечеткого вывода FIS с именем `fismat`. Входные переменные и их функции принадлежности располагаются слева от структурных характеристик FIS, в то время как выходные переменные и их функции принадлежности располагаются справа.

Пример использования функции `plotfis`

```
a=readfis('tipper');
```

```
plotfis(a)
```

Результат вызова функции `plotfis` изображен на рис. 4.19.

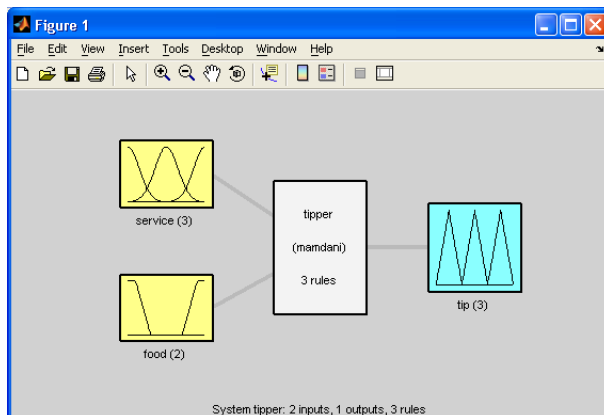


Рис. 4.19. Результат использования функции `plotfis` для изображения диаграммы системы нечеткого вывода FIS

*См. также:* `evalmf`, `plotmf`.

## **plotmf**

### *Назначение*

Изображение графиков всех функций принадлежности для заданной переменной.

### *Синтаксис*

```
plotmf (fismat, varType, varIndex)
```

### *Описание*

Эта функция изображает графики всех функций принадлежности для некоторой переменной из системы нечеткого вывода FIS с именем `fismat`, которое является первым аргументом данной функции. Вторым и третьим аргументами являются тип и номер соответствующей переменной: `varType` ('input' или 'output') и `varIndex`. Данная функция также может быть использована с функцией `subplot`.

Пример использования функции `plotmf`

```
a=readfis('tipper');  
plotmf(a, 'input', 1)
```

Результат вызова функции `plotmf` изображен на рис. 4.20.

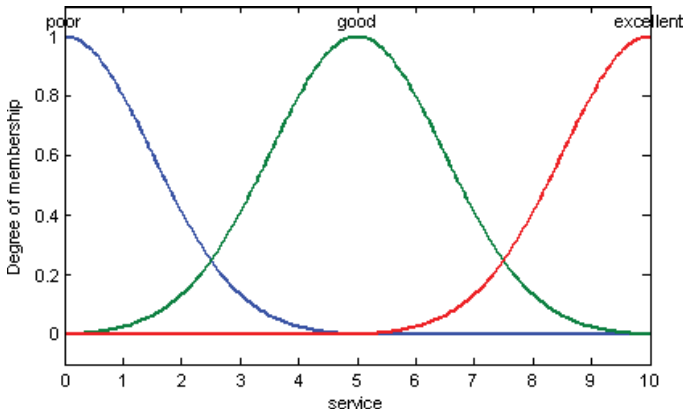


Рис. 4.20. Результат использования функции `plotmf` к первой входной переменной системы нечеткого вывода 'tipper'

*См. также:* `evalmf`, `plotfis`.

## **psigmf**

### *Назначение*

Встроенная П-образная функция принадлежности, являющаяся произведением двух сигмоидальных функций.

### Синтаксис

```
y=psigmf(x,[a b c d])
```

### Описание

Эта функция принадлежности определяется как алгебраическое произведение двух сигмоидальных функций

$$f_{\Pi}(x;a,b,c,d) = f_S(x;a,b) \cdot f_S(x;c,d),$$

каждая из которых задается как функция вида

$$f_S(x;a,b) = \frac{1}{1 + e^{-a(x-b)}}.$$

Вторым аргументом функции является вектор параметров [a b c d]. Параметры a и b характеризуют первую сигмоидальную функцию, а параметры c и d — вторую сигмоидальную функцию.

Пример задания функции принадлежности psigmf

```
x=0:0.1:10;  
y=psigmf(x,[2 4 -5 9]);  
plot(x,y)
```

График функции принадлежности изображен на рис. 4.21.

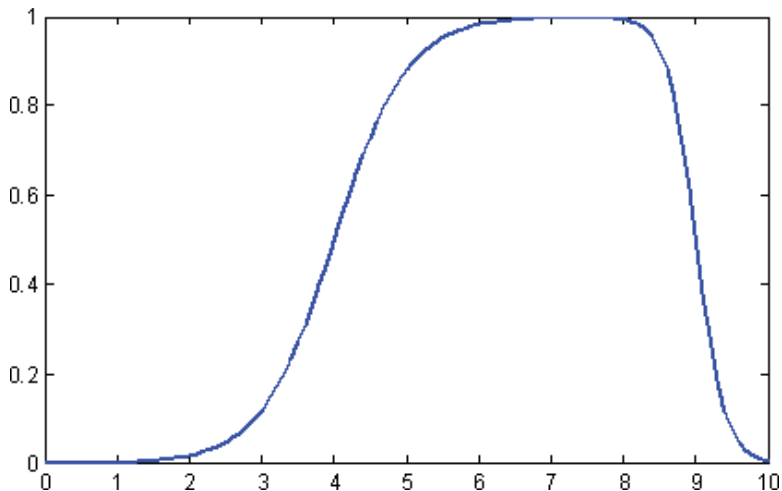


Рис. 4.21. График П-образной функции принадлежности  
`psigmf(x,[2 4 -5 9])`

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `evalmf`, `mf2mf`, `pimf`, `sigmf`, `smf`, `trapmf`, `trimf`, `zmf`.

## **readfis**

### *Назначение*

Загружает систему нечеткого вывода FIS из внешнего файла.

### *Синтаксис*

```
fismat=readfis('filename')
```

*Описание.* Эта функция считывает систему нечеткого вывода из внешнего файл; с именем filename.fis, и загружает его данные в рабочую область. Если данная функция вызывается без аргумента, то открывается стандартное диалоговое окно открытия файла.

Пример использования функции readfis

```
fismat=readfis('tipper');  
getfis(fismat)
```

В этом случае в командном окне будет выведена следующая информация:

```
Name=          tipper  
Type=          mamdani  
NumInputs=    2  
InLabels=     service  
              food  
NumOutputs=   1  
OutLabels=    tip  
NumRules=     3  
AndMethod=    min  
OrMethod=     max  
ImpMethod=    min  
AggMethod=    max  
DefuzzMethod= centroid  
  
ans=          tipper
```

*См. также:* writefis.

## **rmmf**

### *Назначение*

Удаление функции принадлежности из системы нечеткого вывода.

### *Синтаксис*

```
fis=rmmf(fis,'varType',varIndex,'mf',mfIndex)
```



### *Описание*

Функция удаляет функцию принадлежности с номером `mfIndex` для типа переменной `varType` ('input' или 'output') и номером переменной `varIndex` из системы нечеткого вывода с именем `fis`. Структура `fis` должна находиться в рабочей области системы MATLAB. При этом требуется еще один аргумент 'mf' — строка, указывающая тип удаляемой функции принадлежности.

Пример использования функции `rmmf`

```
a=newfis('mysys');  
a=addvar(a,'input','temperature',[0 100]);  
a=addmf(a,'input',1,'cold','trimf',[0 30 60]);  
getfis(a,'input',1)
```

В этом случае в командное окно будет выведена следующая информация:

```
Name=          temperature  
NumMFs=        1  
MFLabels=
```

```
          cold  
Range=         [0 100]
```

После выполнения команд

```
b=rmmf(a,'input',1,'mf',1);  
getfis(b,'input',1)
```

в командное окно будет выведена следующая информация:

```
Name=          temperature  
NumMFs=        0  
MFLabels=  
Range=         [0 100]
```

*См. также:* `addmf`, `addrule`, `addvar`, `plotmf`, `rmvar`.

### **rmvar**

#### *Назначение*

Удаляет переменные из системы нечеткого вывода FIS.

#### *Синтаксис*

```
[fis2,errorStr]=rmvar(fis,'varType',varIndex)  
fis2=rmvar(fis,'varType',varIndex)
```

#### *Описание*

Функция удаляет переменную типа 'varType' ('input' или 'output') с номером `varIndex` из системы нечеткого вывода с именем `fis`. Эта структура должна находиться в рабочей области

системы MATLAB. При этом происходит автоматическое изменение списка правил с целью согласования с новым числом переменных.

До удаления переменной из системы нечеткого вывода необходимо удалить все правила, в которых используется удаляемая переменная. Нельзя удалить переменную, которая присутствует в списке правил. При наличии ошибки функция возвращает сообщение об ошибке в строке с именем `errorStr`.

Пример использования функции `rmvar`

```
a=newfis('mysys');  
a=addvar(a,'input','temperature',[0 100]);  
getfis(a)
```

В этом случае в командное окно будет выведена следующая информация:

```
Name=          mysys  
Type=          mamdani  
NumInputs=     1  
InLabels=        
                temperature  
NumOutputs=    0  
OutLabels=       
NumRules=      0  
AndMethod=     min  
OrMethod=      max  
ImpMethod=     min  
AggMethod=     max  
DefuzzMethod=  centroid  
ans=  
                mysys
```

После выполнения команд

```
b=rmvar(a,'input',1);  
getfis(b)
```

в командное окно будет выведена следующая информация:

```
Name=          mysys  
Type=          mamdani  
NumInputs=     0  
InLabels=        
NumOutputs=    0  
OutLabels=       
NumRules=      0  
AndMethod=     min
```

```
OrMethod=      max
ImpMethod=     min
AggMethod=     max
DefuzzMethod=  centroid
ans=
```

```
mysys
```

*См. также:* addmf, addrule, addvar, rmmf.

## **ruleedit**

### *Назначение*

Вызов графического интерфейса редактора правил системы нечеткого вывода.

### *Синтаксис*

```
ruleedit('a')
ruleedit(a)
```

### *Описание*

Эта функция, записанная в формате `ruleedit('a')`, вызывает редактор правил, который позволяет пользователю в графическом режиме анализировать и модифицировать правила продукции системы нечеткого вывода FIS, сохраненной во внешнем файле с именем `a.fis`. Данная функция позволяет также выполнять грамматический анализ правил, которые используются в некоторой системе нечеткого вывода FIS.

Чтобы использовать данный редактор для создания правил, необходимо предварительно определить все входные и выходные переменные, для чего можно воспользоваться редактором системы нечеткого вывода FIS. Задать правила можно с помощью выбора соответствующих значений термов входных и выходных переменных.

Функция в формате `ruleedit(a)` вызывает редактор правил для переменной рабочего пространства MATLAB, которая соответствует структуре FIS с именем `a`.

*См. также:* addrule, fuzzy, mfeedit, parsrule, ruleview, showrule, surfview.

## **ruleview**

### *Назначение*

Вызов графический интерфейс программы просмотра правил системы нечеткого вывода FIS.

### *Синтаксис*

```
ruleview('a')
```

```
ruleview(a)
```

### *Описание*

Эта функция, записанная в формате `ruleview('a')`, вызывает программу просмотра правил, которая изображает диаграмму нечеткого вывода для структуры FIS, сохраненной во внешнем файле с именем `a.fis`. Функцию целесообразно использовать в том случае, когда необходимо визуально представить весь процесс нечеткого вывода от начала до конца. Пользователь имеет возможность оценить влияние каждого из правил на результат нечеткого вывода посредством изменения значений входных переменных.

Функция в формате `ruleview(a)` вызывает программу просмотра правил для переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `a`.

Программа просмотра правил имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д.

Пункт меню **File** (Файл) редактора правил содержит такие же операции, что и соответствующий пункт меню редактора FIS.

Пункт меню **Edit** (Редактирование) содержит следующие операции:

**Undo** – отменяет выполнение последнего действия;

**FIS Properties...** – вызывает редактор FIS;

**Membership Functions...** – вызывает редактор функций принадлежности;

**Rules** – вызывает программу редактирования правил.

Пункт меню **View** (Вид) содержит следующие операции:

**Surface** – вызывает программу просмотра поверхности вывода.

Пункт меню **Options** (Сервис) содержит следующие операции:

**Format** – позволяет выбрать формат записи правил системы нечеткого вывода: **Verbose** (в форме текста), **Symbolic** (в символической форме) или **Indexed** (в цифровой форме).

Если щелкнуть на номере правила в левой части диаграммы нечеткого вывода, то соответствующее правило появится в строке состояния в нижней части графического интерфейса программы просмотра правил в том формате, который выбран операцией **Options>Format**.

Пример использования функции `ruleview`

```
ruleview('tipper')
```

Результат выполнения функции `ruleview ('tipper')` изображен на рис. 4.22.

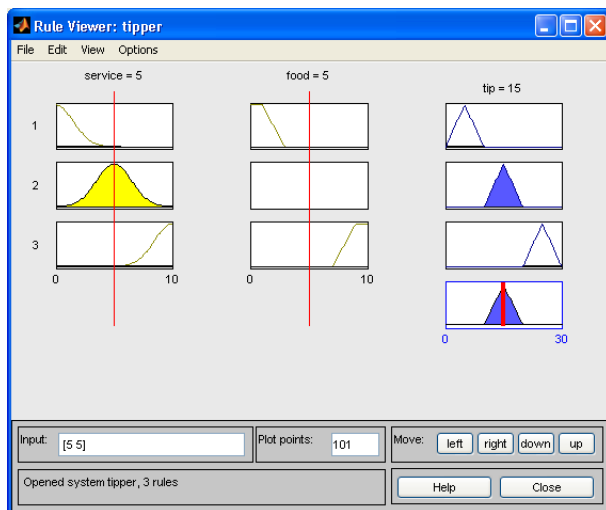


Рис. 4.22. Программа просмотра правил нечетких продукций, вызываемая функцией `ruleview('tipper')`

*См. также:* `fuzzy`, `mfedit`, `ruleedit`, `surfview`.

## **setfis**

### *Назначение*

Задаёт свойства системы нечеткого вывода FIS.

### *Синтаксис*

```
a=setfis(a, 'fispropname', 'newfisprop')
a=setfis(a, 'vartype', varindex, 'varpropname', ...
        'newvarprop')
a=setfis(a, 'vartype', varindex, 'mf', mfindex, ...
        'mfpropname', 'newmfprop')
```

### *Описание*

Функция `setfis` может быть вызвана с тремя, пятью или семью аргументами в зависимости от того, необходимо ли задать структуру FIS в целом, только переменную структуры FIS или отдельную функцию принадлежности для одной из переменных структуры FIS. Аргументы функции имеют следующие значения:

- a — имя структуры FIS в рабочей области MATLAB;
- 'vartype' — строка с указанием типа переменной (input или output);
- varindex — номер входной или выходной переменной;
- 'mf' — строка, необходимая для вызова функции с пятью или семью аргументами, с указанием на то, что функция setfis задает функцию принадлежности для некоторой переменной;
- mfindex — номер функции принадлежности для выбранной переменной;
- 'fispropname' — строка с указанием свойства поля структуры FIS, которая может принимать следующие значения: name, type, andmethod, ormethod, impmethod, aggmethoed, defuzzmethod;
- 'newfisprop' — строка с указанием имени свойства FIS или метода, который задается данной функцией;
- 'varpropname' — строка с указанием имени поля структуры FIS, которая задается данной функцией (name или range);
- 'newvarprop' — строка с указанием имени задаваемой переменной (для name) или диапазона изменения ее значений (для range);
- 'mfpropname' — строка с указанием имени поля задаваемой функции принадлежности (name, type или params);
- 'newmfprop' — строка с указанием имени или типа поля задаваемой функции принадлежности (для name или type) или вектор с указанием параметров задаваемой функции принадлежности (params).

Пример использования функции setfis с тремя аргументами

```
a=readfis('tipper');
a2=setfis(a,'name','eating');
getfis(a2,'name')
```

В этом случае в окне команд будет выведена следующая информация:

```
out =
      eating
ans =
      eating
```

Пример использования функции setfis с пятью аргументами

```
a2=setfis(a,'input',1,'name','help');
getfis(a2,'input',1,'name')
```

В этом случае в окне команд будет выведена следующая информация:

```
ans =  
      help
```

Пример использования функции `setfis` с семью аргументами  
`a2=setfis(a, 'input', 1, 'mf', 2, 'name', 'wretched');`  
`getfis(a2, 'input', 1, 'mf', 2, 'name')`

В этом случае в окне команд будет выведена следующая информация:

```
ans =  
      wretched
```

*См. также:* `getfis`

### **sffis**

*Назначение.*

*S* — функция нечеткого вывода из пакета Simulink.

*Синтаксис.*

```
output=sffis(t, x, u, flag, fismat)
```

*Описание.*

Для выполнения расчетов, выполняемых обычно функцией `evalfis`, в системе Simulink используется собственный MEX-файл. Такой файл специально оптимизирован для работы в среде Simulink. Это означает, кроме всего прочего, что функция `sffis` создает некоторую структуру данных в рабочей области в ходе выполнения фазы инициализации моделирования в системе Simulink и продолжает ее использовать до завершения процесса моделирования.

Аргументы `t`, `x` и `flag` являются стандартными параметрами *S*-функции пакета Simulink. Аргумент и является входом в структуру системы нечеткого вывода FIS с именем `fismat` в рабочей области MATLAB. Например, если структура `fismat` имеет две входных переменных, то и представляет собой вектор с двумя компонентами.

*См. также:* `evalfis`, `fuzblock`.

### **showfis**

*Назначение*

Отображает структуру указываемой системы нечеткого вывода FIS.

*Синтаксис*

```
showfis(fismat)
```

### *Описание*

Функция `showfis (fismat)` выводит в окне команд системы MATLAB значения всех полей структуры `fismat`, которая определяет систему нечеткого вывода FIS и должна находиться в рабочей области.

Пример вызова функции `showfis`

```
a=readfis('tipper');  
showfis(a)
```

В этом случае в окне команд будет выведена следующая информация:

1.	Name	tipper
2.	Type	Mamdani
3.	Inputs/Outputs	[2 1]
4.	NumInputMFs	[3 2]
5.	NumOutputMFs	3
6.	NumRules	3
7.	AndMethod	min
8.	OrMethod	max
9.	ImpMethod	min
10.	AggMethod	max
11.	DefuzzMethod	centroid
12.	InLabels	service
13.		food
14.	OutLabels	tip
15.	InRange	[0 10]
16.		[0 10]
17.	OutRange	[0 30]
18.	InMFLabels	poor
19.		good
20.		excellent
21.		rancid
22.		delicious
23.	OutMFLabels	cheap
24.		average
25.		generous
26.	InMFTypes	gaussmf
27.		gaussmf
28.		gaussmf
29.		trapmf
30.		trapmf
31.	OutMFTypes	trimf



```

32.                                trimf
33.                                trimf
34. InMFParams                    [1.5 0 0 0]
35.                                [1.5 5 0 0]
36.                                [1.5 10 0 0]
37.                                [0 0 1 3]
38.                                [7 9 10 10]
39. OutMFParams                   [0 5 10 0]
40.                                [10 15 20 0]
41.                                [20 25 30 0]
42. Rule Antecedent               [1 1]
43.                                [2 0]
44.                                [3 2]
42. Rule Consequent              1
43.                                2
44.                                3
42. Rule Weigth                  1
43.                                1
44.                                1
42. Rule Connection              2
43.                                1
44.                                2

```

*См. также:* getfis.

## **showrule**

### *Назначение*

Отображает правила системы нечеткого вывода FIS.

### *Синтаксис*

```

showrule(fis)
showrule(fis, indexList)
showrule(fis, indexList, format)
showrule(fis, indexList, format, Lang)

```

### *Описание*

Эта функция используется для визуализации правил системы нечеткого вывода FIS. Она может быть вызвана с различным числом аргументов: от одного до четырех.

- *fis* — обязательный аргумент, который должен соответствовать имени структуры системы нечеткого вывода FIS в рабочей области MATLAB;

- `indexList` — необязательный аргумент, который представляет собой вектор номеров правил, которые требуется отобразить в окне команд;
- `format` — необязательный аргумент, который представляет собой строку формата, в котором требуется отобразить правила. Эта строка может принимать одно из следующих значений: `'verbose'` (значение по умолчанию, которое соответствует отображению правил в форме текста на английском языке), `'symbolic'` (соответствует отображению правил в символической форме), `'indexed'` (соответствует отображению правил в цифровой форме, при этом указываются только номера правил);
- `Lang` — необязательный аргумент, который представляет собой строку с указанием языка отображения правил в окне команд в случае использования формата `verbose`. Эта строка может принимать одно из следующих значений: `'english'`, `'français'` или `'deutsch'`.

Пример использования функции `showrule` с двумя аргументами

```
a=readfis('tipper');
showrule(a,1)
```

В результате выполнения последовательности функций в окне команд появится следующий результат:

```
ans =
1.If (service is poor) or (food is rancid)
then (tip is cheap) (1)
```

Пример использования функции `showrule` с тремя аргументами

```
showrule(a,[3 1],'symbolic')
```

В результате выполнения функции в окне команд появится следующий результат:

```
ans =
3. (service==excellent) | (food==delicious) =>
(tip=generous) (1)
1. (service==poor) | (food==rancid) =>
(tip=cheap) (1)
```

Пример использования функции `showrule` с тремя аргументами (продолжение)

```
showrule(a,1:3,'indexed')
```

В результате выполнения этой функции в окне команд появится следующий результат:

```
ans =  
    1 1, 1 (1) : 2  
    2 0, 2 (1) : 1  
    3 2, 3 (1) : 2
```

См. также: parsrule, ruleedit, addrule.

### **sigmf**

#### *Назначение*

Встроенная сигмоидальная функция принадлежности.

#### *Синтаксис*

```
y=sigmf(x, [a b])
```

#### *Описание*

Сигмоидальная функция принадлежности определяется выражением

$$f(x;a,b)=\frac{1}{1+e^{-a(x-b)}}.$$

Вторым аргументом этой функции является вектор параметров [a b], которые принимают произвольные действительные значения и упорядочены отношением:  $a < b$ . В случае  $a > 0$  может быть получена S-образная функция принадлежности, а в случае  $a < 0$  — Z-образная функция принадлежности.

Пример задания функции принадлежности `sigmf`

```
x=0:0.1:10;  
y=sigmf(x, [3 6]);  
plot(x, y)
```

График функции принадлежности изображен на рис. 4.23.

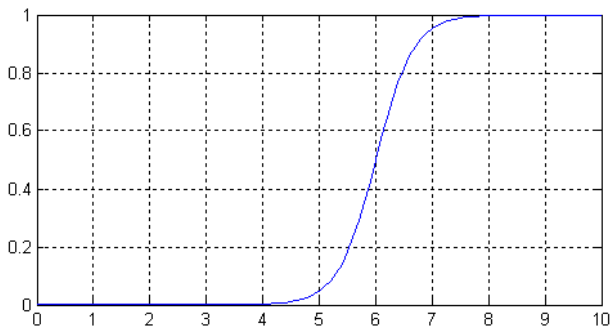


Рис. 23. График сигмоидальной функции `sigmf(x, [3 6])`

См. также: dsigmf, gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, pimf, psigmf, smf, trapmf, trimf, zmf.

### **smf**

#### *Назначение*

Встроенная S-образная функция принадлежности.

#### *Синтаксис*

$y = \text{smf}(x, [a \ b])$

#### *Описание*

Эта функция принадлежности является сплайн-функцией и определяется выражением

$$f_S(x; a, b) = \begin{cases} 0, & x \leq a, \\ 2\left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2}, \\ 1 - 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x < b, \\ 1, & x \geq b. \end{cases}$$

Вторым аргументом функции является вектор параметров  $[a \ b]$ , которые принимают произвольные действительные значения и упорядочены отношением  $a < b$ . Эти параметры характеризуют наклонную часть соответствующей кривой.

Пример задания функции принадлежности `smf`

`x=0:0.1:10;`

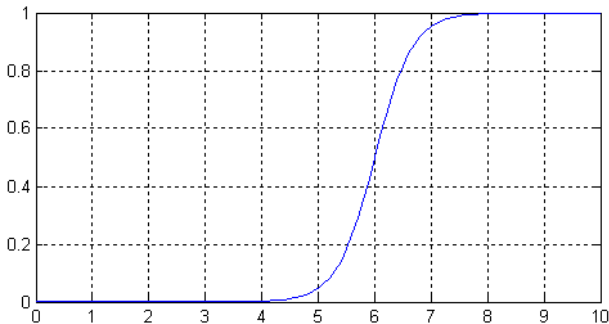


Рис. 24. График S-образной функции принадлежности  $\text{smf}(x, [3 \ 6])$

```
y=smf(x, [3 6]);  
plot(x, y)
```

График функции принадлежности изображен на рис. 4.24.

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `evalmf`, `mf2mf`, `pimf`, `psigmf`, `sigmf`, `trapmf`, `trimf`, `zmf`.

## **subclust**

### *Назначение*

Находит центры кластеров методом субтрактивной кластеризации.

### *Синтаксис*

```
[C, S]=subclust(X, radii, xBounds, options)
```

### *Описание*

Эта функция предназначена для нахождения центров нечетких кластеров в множестве данных методом субтрактивной кластеризации. Метод субтрактивной кластеризации предполагает, что каждая точка данных является центром потенциального кластера и рассчитывает меру подобия каждой точки данных представлять центр кластера. При этом мера подобия основана на оценке плотности точек данных вокруг соответствующего центра кластера.

Данный алгоритм, который является обобщением метода кластеризации Р. Ягера (R. Yager), позволяет:

- выбрать точку данных с максимальным потенциалом для представления центра первого кластера;
- удалить все точки данных в окрестности центра первого кластера, величина которой задается параметром `radii`, чтобы определить следующий нечеткий кластер и координаты его центра.

Эти две процедуры повторяются до тех пор, пока все точки данных не окажутся внутри окрестностей радиуса `radii` некоторого центра кластера.

Матрица `X` одержит данные кластеризации, каждая строка которой соответствует координатам отдельной точки данных. Параметр `radii` представляет собой вектор, компоненты которого принимают значения из интервала  $[0, 1]$  и задают диапазон расчета центров кластеров по каждому из признаков измерений. При этом делается предположение, что все данные содержатся в некотором единичном гиперкубе. В общем случае небольшие значения параметра `radii` приводят к нахождению малого числа точек кластеров. Наилучшие результаты получаются при значениях `radii` между 0.2 и 0.5.

Например, для двух признаков измерения (матрица  $X$  одержит два столбца) значения параметра `radii=[0.5 0.25]` задает диапазон расчета по первому признаку равным 0.5 ширины пространства данных, а по второму признаку равным 0.25 ширины пространства данных. Если параметр `radii` задан в форме скаляра, то это значение относится к каждому признаку, т. е. центр каждого из кластеров имеет сферическую окрестность указанного радиуса для расчета значений своих координат.

Аргумент `xBounds` представляет собой матрицу размерности  $2 \times q$ , которая определяет способ отображения матрицы данных  $X$  в некотором единичном гиперкубе. Здесь  $q$  – количество рассматриваемых признаков в множестве данных. Этот аргумент является необязательным, если матрица  $X$  уже нормализована. Первая строка матрицы `xBounds` содержит минимальные значения интервала измерения каждого из признаков, а вторая строка – максимальные значения измерения каждого из признаков. Например, `xBounds=[-10 -5; 10 5]` определяет диапазон измерения первого признака в интервале  $[-10 +10]$ , а диапазон измерения второго признака в интервале  $[-5 +5]$ . Если значения этой матрицы не указаны, то по умолчанию диапазон измерения признаков полагается равным интервалу от наименьшего значения до наибольшего значения по каждому из признаков в матрице данных.

Для изменения заданных по умолчанию параметров алгоритма кластеризации может быть использован дополнительный вектор `options`. Компоненты этого вектора могут принимать следующие значения:

- `options(1)=quashFactor` – параметр, используемый в качестве коэффициента для умножения значений `radii`, которые определяют окрестность центра кластера. Это осуществляется с целью уменьшения влияния потенциала граничных точек, рассматриваемых как часть нечеткого кластера (по умолчанию это значение равно 1.25);
- `options(2)=acceptRatio` – параметр, устанавливающий предельное значение потенциала точки, как долю потенциала центра первого кластера, выше которого другая точка данных может рассматриваться в качестве центра другого кластера (по умолчанию это значение равно 0.5);
- `options(3)=rejectRatio` – параметр, устанавливающий потенциал как часть потенциала центра первого кластера, ниже которого другая точка данных не может рассматриваться

в качестве центра другого кластера (по умолчанию это значение равно 0.15);

- `options(4)=verbose` – если значение этого параметра не равно нулю, то на экран монитора выводится информация о выполнении процесса кластеризации (по умолчанию это значение равно 0).

Функция `subclust` возвращает матрицу  $C$  значений координат центров нечетких кластеров. Каждая строка этой матрицы содержит координаты одного центра кластера. Функция также возвращает вектор  $S$ , компоненты которого представляют  $\sigma$ -значения, которые определяют диапазон влияния центра кластера по каждому из рассматриваемых признаков. Все центры кластеров обладают одинаковым множеством  $\sigma$ -значений.

Пример вызова функции `subclust` с минимальным числом аргументов:

```
[C, S]=subclust(X, 0.5)
```

В этом случае для всех признаков множества данных задается один диапазон влияния 0.5.

Пример вызова функции `subclust` с дополнительными значениями аргументов

```
[C, S]=subclust(X, [0.5 0.25 0.3], [], [2.0 0.8 0.7])
```

В этом случае для множества данных используется три признака (матрица  $X$  имеет три столбца) с радиусами окрестностей 0.5, 0.25, 0.3 по первому, второму и третьему признакам соответственно. Коэффициенты шкалирования для отображения множества данных в единичный гиперкуб получаются на основе минимального и максимального значений данных. Аргумент `squashFactor`, равный 2.0, указывает на то, что необходимо определить кластеры, далеко расположенные друг от друга. Аргумент `acceptRatio` равный 0.8, указывает на то, что для нахождения центров кластеров будет нужен очень высокий потенциал. Наконец, аргумент `rejectRatio` равен 0.7, что указывает на необходимость исключения из рассмотрения всех точек данных, не обладающих высоким потенциалом.

*См. также:* `genfis2`.

## **surfview**

### *Назначение*

Вызывает графический интерфейс программы просмотра поверхности системы нечеткого вывода.

### *Синтаксис*

```
surfview('a')
```

```
surfview(a)
```

### *Описание*

Эта функция, записанная в формате `surfview('a')`, вызывает программу просмотра поверхности, которая изображает поверхность нечеткого вывода для структуры FIS, сохраненной во внешнем файле с именем `a.fis`, для любой одной или двух из ее входных переменных. Программа не позволяет вносить изменения в систему нечеткого вывода и соответствующую ей структуру FIS. Используя главное меню программы, пользователь может выбрать входные переменные и соответствующие им горизонтальные оси системы координат (X и Y), а также выходную переменную, которой соответствует вертикальная ось системы координат (Z). Функция в формате `surfview(a)` вызывает программу просмотра поверхности вывода для переменной рабочего пространства MATLAB, соответствующей структуре FIS с именем `a`.

Щелкнув и задержав левую кнопку мыши, посредством последующего перемещения курсора мыши в том или ином направлении можно изменить угол просмотра поверхности вывода. Если рассматривается система нечеткого вывода с более чем двумя входными переменными, то для невидимых входных переменных следует задать некоторые постоянные значения (константы).

Программа просмотра поверхности вывода имеет главное меню, которое позволяет пользователю вызывать другие графические средства работы с системой нечеткого вывода FIS, загружать и сохранять структуру FIS во внешних файлах и т. д.

- Пункт меню **File** (Файл) редактора правил содержит такие же операции, что и соответствующий пункт меню редактора FIS.
- Пункт меню **Edit** (Редактирование) содержит следующие операции:
  - **Undo** – отменяет выполнение последнего действия;
  - **FIS Properties...** – вызывает редактор FIS;
  - **Membership Functions...** – вызывает редактор функций принадлежности;
  - **Rules** – вызывает программу редактирования правил.
- Пункт меню **View** (Вид) содержит следующие операции:
  - **Rules** – вызывает программу просмотра правил.
  - Пункт меню **Options** (Сервис) содержит следующие операции:



- **Plot** – позволяет выбрать один из восьми стилей изображения графика поверхности вывода;
- **Color Map** – позволяет выбрать одну из четырех цветовых схем изображения графика поверхности вывода;
- **Always evaluate** – пометка галочкой этого пункта вложенного меню приводит к автоматическому формированию новой поверхности вывода всякий раз, когда вносятся изменения в систему нечеткого вывода, влияющие на форму графика поверхности вывода (такие, как изменение количества точек сетки графика). Значение принято по умолчанию. Чтобы его отменить, необходимо снять галочку у этого пункта вложенного меню, щелкнув на этой позиции меню.

Пример использования функции `surfview`:

```
surfview('tipper')
```

Результат выполнения функции `surfview('tipper')` изображен на рис. 4.25.

*См. также:* `anfisedit`, `fuzzy`, `gensurf`, `mfedit`, `ruleedit`, `ruleview`.

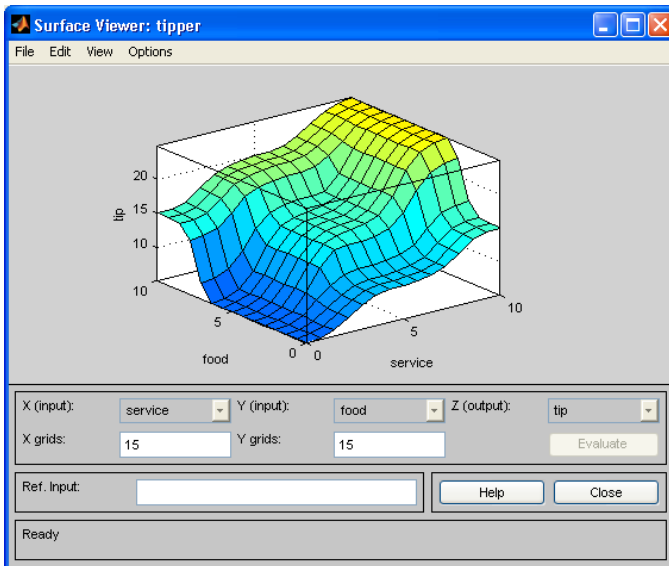


Рис. 4.25. Программа просмотра поверхности вывода, вызываемая функцией `surfview('tipper')`

### **trapmf**

#### *Назначение*

Встроенная трапециевидная функция принадлежности.

#### *Синтаксис*

```
y=trapmf(x,[a b c d])
```

#### *Описание*

Эта функция используется для задания трапециевидных функций принадлежности вида

$$f_T(x;a,b,c,d)=\begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ 1, & b < x \leq c, \\ \frac{d-x}{d-c}, & c < x \leq d, \\ 0, & x > d. \end{cases}$$

Функция trapmf использует четыре аргумента: a, b, c и d, в качестве некоторых числовых параметров, принимающих произвольные действительные значения и упорядоченные отношением:  $a < b < c < d$ . Значения a и d задают носитель соответствующего нечеткого множества, а значения b и c — ядро этого нечеткого множества.

Пример задания трапециевидной функция принадлежности trapmf

```
x=0:0.1:10;  
y=trapmf(x,[1 3 5 8]);  
plot(x,y)
```

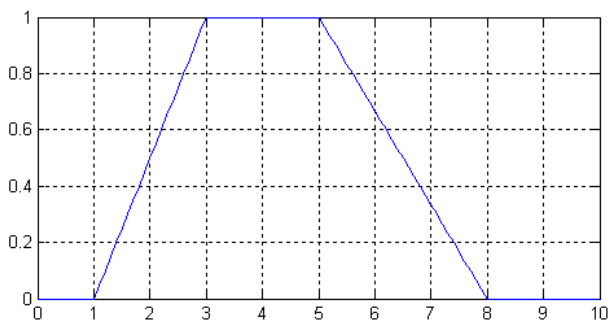


Рис. 4.26. График функции принадлежности trapmf(x,[1 3 5 8])

График полученной в этом случае функции принадлежности изображен на рис. 4.26.

*См. также:* dsigmf, gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, pimf, psigmf, sigmf, smf, trimf, zmf.

### **trimf**

#### *Назначение*

Встроенная треугольная функция принадлежности.

#### *Синтаксис*

```
y=trimf(x, [a b c])
```

#### *Описание*

Эта функция используется для задания треугольных функций принадлежности вида

$$f_{\Delta}(x; a, b, c) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ \frac{c-x}{c-b}, & b < x \leq c, \\ 0, & x > c. \end{cases}$$

Функция `trimf` использует три аргумента:  $a$ ,  $b$  и  $c$  – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением:  $a < b < c$ . Значения  $a$  и  $c$  определяют носитель соответствующего нечеткого множества, а значение  $b$  – модальное значение этого нечеткого множества.

Пример задания треугольной функции принадлежности `trimf`

```
x=0:0.1:10;  
y=trimf(x, [3 4 7]);  
plot(x, y)
```

График полученной в этом случае функции принадлежности изображен на рис. 4.27.

*См. также:* dsigmf, gaussmf, gauss2mf, gbellmf, evalmf, mf2mf, pimf, psigmf, sigmf, smf, trapmf, zmf.

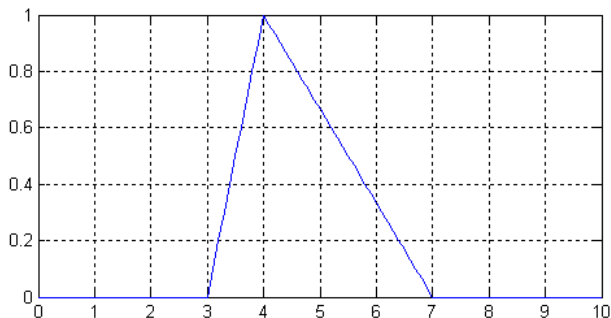


Рис. 4.27. График функции принадлежности `trimf(x, [3 4 7])`

### **writefis**

#### *Назначение*

Сохраняет систему нечеткого вывода FIS во внешнем файле.

#### *Синтаксис*

```
writefis (fismat)
writefis (fismat, 'filename')
writefis (fismat, 'filename', 'dialog')
```

#### *Описание*

Функция `writefis` сохраняет структуру системы нечеткого вывода FIS с именем `fismat`, находящуюся в рабочей области системы MATLAB, во внешнем файле с расширением `.fis`. Вызов функции в формате `writefis (fismat)` открывает стандартное диалоговое окно сохранения файла, использование функции в формате `writefis (fismat, 'filename')` записывает структуру системы нечеткого вывода FIS с именем `fismat`, находящуюся в рабочей области системы MATLAB, во внешнем файле с именем `filename.fis`. Диалоговое окно сохранения файла не открывается, а файл сохраняется в текущем каталоге на жестком или сменном диске.

Вызов функции в формате `writefis (fismat, 'filename', 'dialog')` открывает диалоговое окно сохранения файла, при этом запись будет производиться в файл с именем `filename.fis`. В том случае если расширение `fis` не содержится в имени файла, оно добавляется автоматически.

Пример использования функции `writefis`

```
a=newfis('tipper');
a=addvar(a,'input','service',[0 10]);
a=addmf(a,'input',1,'poor','gaussmf',[1.5 0]);
```

```

a=addmf(a,'input',1,'good','gaussmf',[1.5 5]);
a=addmf(a,'input',1,'excellent','gaussmf',[1.5
10]);
writefis(a,'my_file')
См. также: readfis.

```

### **zmf**

#### *Назначение*

$Z$  - образная функция принадлежности.

#### *Синтаксис*

```
y=zmf(x,[a b])
```

#### *Описание*

Эта функция используется для задания  $Z$  - образных функций принадлежности вида сплайн-функции

$$f_Z(x;a,b) = \begin{cases} 1, & x \leq a, \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2}, \\ 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x \leq b, \\ 0, & x > b. \end{cases}$$

Функция `zmf` использует два аргумента:  $a$  и  $b$  — некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением  $a < b$ .

Пример задания  $Z$ -образной функции принадлежности `zmf`

```

x=0:0.1:10;
y=zmf(x,[3 6]);
plot(x,y)

```

График полученной в этом случае функции принадлежности изображен на рис. 4.28.

*См. также:* `dsigmf`, `gaussmf`, `gauss2mf`, `gbellmf`, `evalmf`, `mf2mf`, `pimf`, `psigmf`, `sigmf`, `smf`, `trapmf`, `trimf`.

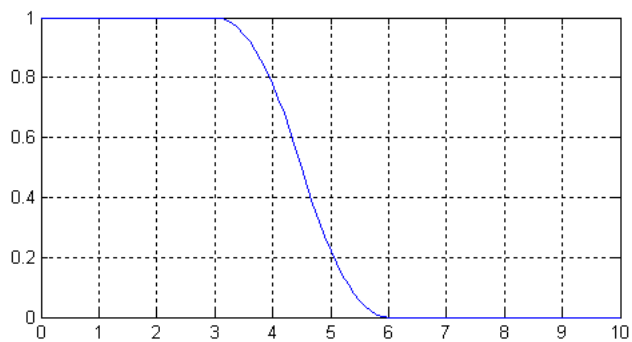


Рис. 4.28. График функции принадлежности  $zmf(x, [3 \ 6])$

# 5

## Алфавитный указатель функций пакета Fuzzy Logic Toolbox for MatLab

addmf .....	44	mfedit .....	81
addrule .....	46	newfis .....	81
addvar .....	47	parsrule .....	82
anfis .....	48	pimf .....	83
anfisedit .....	52	plotfis .....	85
convertfis .....	53	plotmf .....	86
defuzz .....	53	psigmf .....	86
dsigmf .....	55	readfis .....	88
evalfis .....	56	rmmf .....	88
evalmf .....	58	rmvar .....	89
fcm .....	59	ruleedit .....	91
findcluster .....	61	ruleview .....	91
fuzarith .....	63	setfis .....	93
fuzblock .....	64	sffis .....	95
fuzdemos .....	66	showfis .....	95
fuzzy .....	67	showrule .....	97
gauss2mf .....	67	sigmf .....	99
gaussmf .....	69	smf .....	100
gbellmf .....	70	subclust .....	101
genfis1 .....	71	surfview .....	103
genfis2 .....	73	trapmf .....	106
gensurf .....	75	trimf .....	107
getfis .....	76	writefis .....	108
mam2sug .....	78	zmf .....	109
mf2mf .....	79		

## Библиографический список

1. Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. — М. : Горячая линия – Телеком, 2007. — 284 с.
2. Дьяконов В. П., Круглов В. В. MATLAB 6.5SP1 / 7 / 7SP1 / 7SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики. — М. : СОЛОН-ПРЕСС, 2006. — 456 с.
3. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. — СПб. : БХВ-Петербург, 2005. — 736 с.
4. Fuzzy Logic Toolbox. User's Guide, Version 2. The MathWorks, Inc., 1999.