

Разработка систем нечеткого вывода в режиме командной строки с использованием пакета Fuzzy Logic Toolbox

Цель: Изучение особенностей и приобретение навыков разработки систем нечеткого вывода в режиме командной строки с использованием пакета расширения Fuzzy Logic Toolbox в вычислительной среде MatLab.

Задание: Ознакомиться с порядком разработки систем нечеткого вывода в режиме командной строки в среде MatLab по учебному пособию TarasianFuzzyLogicToolbox.pdf При помощи пакета расширения Fuzzy Logic Toolbox разработать в режиме командной строки систему нечеткого вывода типа Мамдани аналогичную той, которая была получена в интерактивном режиме в процессе выполнения лабораторной работы №1.

Листинг фрагмента кода М-файла, содержащего синтез системы нечеткого вывода в режиме командной строки:

```
fisK = newfis('firstMK_custom');

%-----
%   первая входная переменная
%-----

fisK.input(1).name = 'x1';
fisK.input(1).range = [-1.5 1.5];

fisK.input(1).mf(1).name = 'оч_низ';
fisK.input(1).mf(1).type = 'trimf';
fisK.input(1).mf(1).params = [-2.25 -1.5 -0.75];

fisK.input(1).mf(2).name = 'низ';
fisK.input(1).mf(2).type = 'trimf';
fisK.input(1).mf(2).params = [-1.5 -0.75 0];

fisK.input(1).mf(3).name = 'сред';
fisK.input(1).mf(3).type = 'trimf';
fisK.input(1).mf(3).params = [-0.75 1.388e-17 0.75];
```

```

fisK.input(1).mf(4).name = 'выс';
fisK.input(1).mf(4).type = 'trimf';
fisK.input(1).mf(4).params = [0 0.75 1.5];

fisK.input(1).mf(5).name = 'очвыс';
fisK.input(1).mf(5).type = 'trimf';
fisK.input(1).mf(5).params = [0.75 1.5 2.25];

%-----
%      вторая входная переменная
%-----

fisK.input(2).name = 'x2';
fisK.input(2).range = [-2.2 2.2];

fisK.input(2).mf(1).name = 'оч_низ';
fisK.input(2).mf(1).type = 'trimf';
fisK.input(2).mf(1).params = [-3.3 -2.2 -1.1];

fisK.input(2).mf(2).name = 'низ';
fisK.input(2).mf(2).type = 'trimf';
fisK.input(2).mf(2).params = [-2.2 -1.1 0];

fisK.input(2).mf(3).name = 'сред';
fisK.input(2).mf(3).type = 'trimf';
fisK.input(2).mf(3).params = [-1.1 0 1.1];

fisK.input(2).mf(4).name = 'выс';
fisK.input(2).mf(4).type = 'trimf';
fisK.input(2).mf(4).params = [0 1.1 2.2];

fisK.input(2).mf(5).name = 'оч_выс';
fisK.input(2).mf(5).type = 'trimf';
fisK.input(2).mf(5).params = [1.1 2.2 3.3];

%-----
%      выходная переменная
%-----

fisK.output(1).name = 'y';
fisK.output(1).range = [-1.5 12];

fisK.output(1).mf(1).name = 'низ';
fisK.output(1).mf(1).type = 'gaussmf';

```

```

fisK.output(1).mf(1).params = [0.8 -1.762];

fisK.output(1).mf(2).name = 'нул';
fisK.output(1).mf(2).type = 'gaussmf';
fisK.output(1).mf(2).params = [0.5 0];

fisK.output(1).mf(3).name = 'выш_нул';
fisK.output(1).mf(3).type = 'gaussmf';
fisK.output(1).mf(3).params = [0.8 2];

fisK.output(1).mf(4).name = 'оч_выс';
fisK.output(1).mf(4).type = 'gaussmf';
fisK.output(1).mf(4).params = [0.5612 12];

%-----
%           Задаем базу правил
%-----

% 1 правило
fisK.rule(1).antecedent = [1 1];
fisK.rule(1).connection = 1;
fisK.rule(1).consequent = 2;
fisK.rule(1).weight = 1;

% 2 правило
fisK.rule(2).antecedent = [1 2];
fisK.rule(2).connection = 1;
fisK.rule(2).consequent = 1;
fisK.rule(2).weight = 1;

% 3 правило
fisK.rule(3).antecedent = [1 3];
fisK.rule(3).connection = 1;
fisK.rule(3).consequent = 2;
fisK.rule(3).weight = 1;

% 4 правило
fisK.rule(4).antecedent = [1 4];
fisK.rule(4).connection = 1;
fisK.rule(4).consequent = 1;
fisK.rule(4).weight = 1;

% 5 правило
fisK.rule(5).antecedent = [1 5];

```

```

fisK.rule(5).connection = 1;
fisK.rule(5).consequent = 2;
fisK.rule(5).weight = 1;

% 6 правило
fisK.rule(6).antecedent = [2 1];
fisK.rule(6).connection = 1;
fisK.rule(6).consequent = 4;
fisK.rule(6).weight = 1;

% 7 правило
fisK.rule(7).antecedent = [2 2];
fisK.rule(7).connection = 1;
fisK.rule(7).consequent = 2;
fisK.rule(7).weight = 1;

% 8 правило
fisK.rule(8).antecedent = [2 3];
fisK.rule(8).connection = 1;
fisK.rule(8).consequent = 2;
fisK.rule(8).weight = 1;

% 9 правило
fisK.rule(9).antecedent = [2 4];
fisK.rule(9).connection = 1;
fisK.rule(9).consequent = 2;
fisK.rule(9).weight = 1;

% 10 правило
fisK.rule(10).antecedent = [2 5];
fisK.rule(10).connection = 1;
fisK.rule(10).consequent = 4;
fisK.rule(10).weight = 1;

% 11 правило
fisK.rule(11).antecedent = [3 1];
fisK.rule(11).connection = 1;
fisK.rule(11).consequent = 2;
fisK.rule(11).weight = 1;

% 12 правило
fisK.rule(12).antecedent = [3 2];
fisK.rule(12).connection = 1;
fisK.rule(12).consequent = 2;

```

```

fisK.rule(12).weight = 1;

% 13 правило
fisK.rule(13).antecedent = [3 3];
fisK.rule(13).connection = 1;
fisK.rule(13).consequent = 2;
fisK.rule(13).weight = 1;

% 14 правило
fisK.rule(14).antecedent = [3 4];
fisK.rule(14).connection = 1;
fisK.rule(14).consequent = 2;
fisK.rule(14).weight = 1;

% 15 правило
fisK.rule(15).antecedent = [3 5];
fisK.rule(15).connection = 1;
fisK.rule(15).consequent = 2;
fisK.rule(15).weight = 1;

% 16 правило
fisK.rule(16).antecedent = [4 1];
fisK.rule(16).connection = 1;
fisK.rule(16).consequent = 4;
fisK.rule(16).weight = 1;

% 17 правило
fisK.rule(17).antecedent = [4 2];
fisK.rule(17).connection = 1;
fisK.rule(17).consequent = 2;
fisK.rule(17).weight = 1;

% 18 правило
fisK.rule(18).antecedent = [4 3];
fisK.rule(18).connection = 1;
fisK.rule(18).consequent = 2;
fisK.rule(18).weight = 1;

% 19 правило
fisK.rule(19).antecedent = [4 4];
fisK.rule(19).connection = 1;
fisK.rule(19).consequent = 2;
fisK.rule(19).weight = 1;

```

```
% 20 правило
fisK.rule(20).antecedent = [4 5];
fisK.rule(20).connection = 1;
fisK.rule(20).consequent = 4;
fisK.rule(20).weight = 1;

% 21 правило
fisK.rule(21).antecedent = [5 1];
fisK.rule(21).connection = 1;
fisK.rule(21).consequent = 2;
fisK.rule(21).weight = 1;

% 22 правило
fisK.rule(22).antecedent = [5 2];
fisK.rule(22).connection = 1;
fisK.rule(22).consequent = 3;
fisK.rule(22).weight = 1;

% 23 правило
fisK.rule(23).antecedent = [5 3];
fisK.rule(23).connection = 1;
fisK.rule(23).consequent = 2;
fisK.rule(23).weight = 1;

% 24 правило
fisK.rule(24).antecedent = [5 4];
fisK.rule(24).connection = 1;
fisK.rule(24).consequent = 3;
fisK.rule(24).weight = 1;

% 25 правило
fisK.rule(25).antecedent = [5 5];
fisK.rule(25).connection = 1;
fisK.rule(25).consequent = 2;
fisK.rule(25).weight = 1;
```

Графики зависимости выходной переменной от входных для нечеткой системы, построенной в интерактивном режиме, а также в режиме командной строки представлены на рисунках 1 и 2 соответственно.

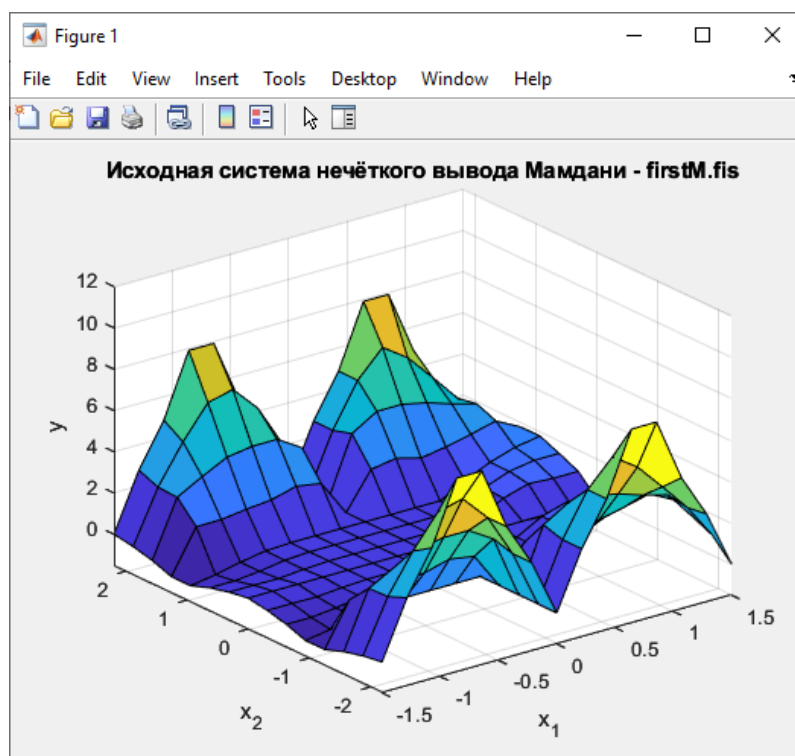


Рисунок 1 - нечеткая система в интерактивном режиме

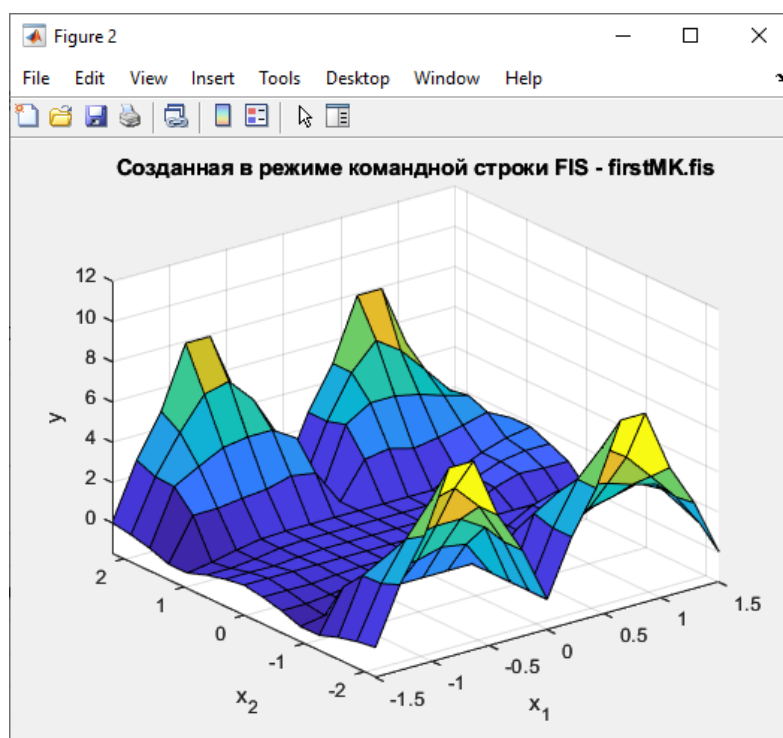


Рисунок 2 - нечеткая система в режиме командной строки

Величина максимального отклонения выходного сигнала разработанной системы от выхода исходной системы равна нулю.

Вывод

С помощью средств командной строки можно разработать систему нечеткого вывода, аналогичную той, которую мы можем получить с помощью взаимодействия с Fuzzy Logic Toolbox через UI.

По результатам работы можно выделить несколько недостатков метода командной строки:

- Субъективно неудобное и трудоемкое API, возможно эту проблему можно частично решить, написав свои обертки для имеющегося функционала;
- При отсутствии должного опыта в проектировании систем нечеткого вывода, отсутствие визуализации происходящего делает процесс проектирования не интуитивно понятным для разработчика.

Исходный код лабораторной работы на GitHub представлен по [ССЫЛКЕ](#).