

## Практическое занятие 6

**Адаптация системы нечёткого вывода Мамдани  
с использованием средств нелинейной оптимизации  
пакета *Optimization Toolbox***

**Цель:** Изучение особенностей построения адаптивных нечётких систем с использованием средств нелинейной оптимизации пакета *Optimization Toolbox* системы MatLab.

**Вводные замечания:**

При нечетком моделировании чаще других используют базу знаний в формате Мамдани, в которой антецеденты и консеквенты правил заданы нечеткими множествами типа «Низкий», «Средний», «Высокий» и т.п.

В противовес моделям типа «черный ящик», нечеткие модели Мамдани являются прозрачными – их структура содержательно интерпретируется в терминах, понятных не только разработчикам с высокой математической квалификацией, а и заказчикам. Прозрачность нечетких моделей Мамдани является одним из главных преимуществ, благодаря которому нечеткие технологии успешно конкурируют с другими методами. Это справедливо для тех прикладных задач, где возможность содержательной интерпретации важнее точности моделирования.

Для повышения точности нечеткую модель адаптируют, т.е. итерационно изменяют ее параметры с целью минимизации отклонения результатов логического вывода от экспериментальных данных. Настраивают как веса правил, так и функции принадлежности нечетких термов.

Адаптация нечеткой модели Мамдани представляет собой задачу нелинейной оптимизации, исследованию которой посвящено огромное количество теоретических и прикладных работ. В них основной акцент делается на достижении максимальной точности функционирования нечеткой модели. При этом настраиваемые параметры иногда изменяются настолько сильно, что возникают сложности содержательной интерпретации нечеткой модели.

Таким образом, «погоня за точностью» приводит к потере важного конкурентного преимущества – прозрачности нечеткой модели. Если прозрачность модели второстепенна, тогда при идентификации зависимостей разумнее использовать другие (не нечеткие) методы, адекватность которых обычно бывает лучше. Простейшими приёмами сохранения прозрачности модели является сокращение количества управляемых переменных и введение ограничений на их изменение в процессе адаптации.

**Исходные данные:**

При исследовании особенностей адаптации нечётких моделей будем использовать систему нечёткого вывода типа Мамдани с гауссовыми функциями принадлежности входных и выходных переменных (*firstMg.fis*), которая была разработана на практическом занятии №1 для аппроксимации нелинейной зависимости на основе её трёхмерного изображения согласно варианту задания.

Целью адаптации является уменьшение среднеквадратической ошибки аппроксимации.

Для обращения к конкретным параметрам системы нечёткого вывода используются приёмы, изученные на практическом занятии №5, которое было посвящено разработке систем нечёткого вывода в среде MatLab в режиме командной строки.

Для решения задачи параметрической оптимизации системы нечёткого вывода будем использовать возможности пакета *Toolbox Optimization* системы MatLab.

**Задание:**

1. Ознакомиться с особенностями использования функции **fmincon** пакета **Toolbox Optimization**, по справочному материалу *Spravka\_fmincon.pdf* или обратившись к справочной системе MatLab.

Функция **fmincon** позволяет находить минимум для скалярной функции нескольких аргументов при заданном начальном приближении и наличии линейных и нелинейных ограничений (задача нелинейного программирования).

2. Построить трёхмерные изображения исходной зависимости, согласно варианту задания практического занятия №1, а также результата её аппроксимации синтезированной системой нечёткого вывода.

**Примечание.** Процесс адаптации системы нечёткого вывода необходимо оформить в виде *m*-файла. Данный *m*-файл может выглядеть следующим образом (для системы нечёткого вывода, рассматривавшейся в лабораторной работе №1 в качестве примера).

```
% Нечёткая аппроксимация зависимости: y=x1^2*sin(x2-1)
% (LR_6_AdaptFIS.m)
%
% <<< Адаптация системы нечёткого вывода Мамдани >>>
% <<< с использованием функции нелинейной оптимизации fmincon >>>
% <<< пакета Optimization Toolbox >>>
%=====
clear all % очистка памяти (leaving the workspace empty)
clc % очистка командного окна (Clear Command Window)
%-----
% Построение графика функции y=x1^2*sin(x2-1)
% в области x1 = [-7, 3]; x2 = [-4.4, 1.7]
%-----

n = 15; % количество точек дискретизации
x1 = linspace(-7, 3, n);
x2 = linspace(-4.4, 1.7, n);

y = zeros(n, n);
for i = 1:n
    y(i,:) = x1.^2 * sin(x2(i)-1);
end

%-----
h1 = figure(1);
set(h1,'Position',[622 541 524 407])
% get(h1,'Position')
surf(x1, x2, y)
axis([-10 5 ...
      -6 2 ...
      -50 50]);
view(-40,30)
xlabel('x_1'); ylabel('x_2'); zlabel('y');
title('Исходная зависимость')

%-----
% Построение графика нечёткого отображения Мамдани
% с гауссовыми функциями принадл. термов вх.перем.
%-----

fisMg = readfis('firstMg'); % загрузка исходной нечёткой сист. с диска
% fuzzy(fisMg) % вызов FIS-Editor с исходной неч.сист.

yMg = zeros(n, n);
```

```

for i = 1:n
    yMg(i,:) = evalfis([x1; ones(size(x1))*x2(i)], fisMg)';
end

%-----

h2 = figure(2);
set(h2, 'Position', [1154 541 524 407])
colormap('default')
surf(x1, x2, yMg)
axis([-10 5 ...
      -6 2 ...
      -50 50]);
view(-40,30)
xlabel('x_1'); ylabel('x_2'); zlabel('y');
title('Система нечёткого вывода Мамдани до адаптации')

%-----

```

**3. Сформировать обучающий и тестовый массивы данных, которые будут использоваться в процессе адаптации системы нечёткого вывода. Элементами данных массивов являются точки, равномерно распределённые в области определения исходной нелинейной зависимости согласно варианту задания.**

Построить графическое изображение распределения точек обучающего и тестового массивов на изображении исходной зависимости.

```

%-----
% Формирование обучающего и тестового массивов данных
%-----

Nob = 60; % количество точек обучающего массива
Nts = 60; % количество точек тестового массива

RndStt = 3; % уст.начальн.сост.генер.случ.чисел
rand('state', RndStt);

% Входы:
xlob = min(x1) + (max(x1) - min(x1)) * rand(Nob, 1);
x2ob = min(x2) + (max(x2) - min(x2)) * rand(Nob, 1);
x1ts = min(x1) + (max(x1) - min(x1)) * rand(Nts, 1);
x2ts = min(x2) + (max(x2) - min(x2)) * rand(Nts, 1);

% Выходы:

yob = zeros(size(xlob));
for i = 1:Nob
    yob(i) = xlob(i)^2 * sin(x2ob(i)-1);
end

yts = zeros(size(x1ts));
for i = 1:Nts
    yts(i) = x1ts(i)^2 * sin(x2ts(i)-1);
end

%-----

h3 = figure(3);
set(h3, 'Position', [91 541 524 407])
clf
hold on

```

```

plot(xlob, x2ob, 'bs', 'MarkerSize',4)
plot(xlts, x2ts, 'ro', 'MarkerSize',4)
hold off
axis([min(x1)-0.01 max(x1)+0.01 ...
      min(x2)-0.01 max(x2)+0.01]);
xlabel('x1'); ylabel('x2');
title('Распределение данных обучающей (bs) и тестовой (or) выборок')

```

```
%-----
```

```
pause(2); % пауза 2,0 с
```

```

figure(2);
colormap('white') %
hold on
plot3(xlob, x2ob, yob, 'bs', 'MarkerSize',3, 'MarkerFaceColor','b')
plot3(xlts, x2ts, yts, 'ro', 'MarkerSize',3, 'MarkerFaceColor','r')
hold off

```

```
%-----
```

**4. Выбрать настраиваемые в процессе адаптации параметры системы нечёткого вывода (около 20 шт.).** Задать начальные приближения данных параметров, а также нижние и верхние границы их изменения. Сформировать векторы настраиваемых параметров, а также границ их изменения.

Для повышения эффективности работы алгоритма адаптации целесообразно ввести масштабирование настраиваемых параметров, которое приводит их значения к соизмеримым величинам.

```

%-----
% Процесс адаптации системы нечёткого вывода
%-----
% НАСТРАИВАЕМЫЕ ПАРАМЕТРЫ (19 шт.):
%   - весовые коэффициенты правил 5, 6, 7 (3 шт.);
%   - коэф. конц.термов переменных x1, x2, y (3+3+5=11 шт.);
%   - коорд.максимумв термов "средний" переменных x1, x2 (2 шт.);
%   - коорд.макс.терм."ниже сред.", "сред.", "выше сред." перемен. y (3 шт.)
%-----
% Весовые коэффициенты правил
w0 = [1 1 1]-0.001; % начальное приближение
wL = [0 0 0]; % нижняя граница
wU = [1 1 1]; % верхняя граница

% Коэффициенты концентрации термов переменных
x1_s10 = fisMg.input(1).mf(1).params(1); % исх. знач. для x1 "низкий"
x1_s20 = fisMg.input(1).mf(2).params(1); % -//- "средний"
x1_s30 = fisMg.input(1).mf(3).params(1); % -//- "высокий"

x2_s10 = fisMg.input(2).mf(1).params(1); % исх. знач. для x2 "низкий"
x2_s20 = fisMg.input(2).mf(2).params(1); % -//- "средний"
x2_s30 = fisMg.input(2).mf(3).params(1); % -//- "высокий"

y_s10 = fisMg.output(1).mf(1).params(1); % исх.знач.для y "низкий"
y_s20 = fisMg.output(1).mf(2).params(1); % -//- "ниже среднего"
y_s30 = fisMg.output(1).mf(3).params(1); % -//- "средний"
y_s40 = fisMg.output(1).mf(4).params(1); % -//- "выше среднего"
y_s50 = fisMg.output(1).mf(5).params(1); % -//- "высокий"

x12ys0 = [x1_s10 x1_s20 x1_s30 ... % объедин.цент.конц. в один вектор
          x2_s10 x2_s20 x2_s30 ...

```

```

        y_s10    y_s20    y_s30    y_s40    y_s50];
x12ysL = 0.7*x12ys0;      % нижняя граница = начальное приближение - 30%
x12ysU = 1.3*x12ys0;      % верхняя граница = начальное приближение + 30%

% коорд.максимумв некотор.термов входных и выходных переменных
x1_c20 = fisMg.input(1).mf(2).params(2);      % исх. знач. для x1 "средний"
x2_c20 = fisMg.input(2).mf(2).params(2);      % -//- x2 "средний"
y_c20 = fisMg.output(1).mf(2).params(2);      % -//- y "ниже среднего"
y_c30 = fisMg.output(1).mf(3).params(2);      % -//- y "средний"
y_c40 = fisMg.output(1).mf(4).params(2);      % -//- y "выше среднего"

x12yc0 = [x1_c20 x2_c20 y_c20 y_c30 y_c40];

dx1 = 0.3*(max(x1)-min(x1));
dx2 = 0.3*(max(x2)-min(x2));
dy = 0.2*(max(max(y))-min(min(y)));
x12ycL = x12yc0 - [dx1 dx2 dy dy dy];          % нижняя граница
x12ycU = x12yc0 + [dx1 dx2 dy dy dy];          % верхняя граница

% Объединение настраиваемых параметров в один вектор
ParamFis0 = [w0 x12ys0 x12yc0];                % начальное приближение
ParamFisL = [wL x12ysL x12ycL];                % нижняя граница
ParamFisU = [wU x12ysU x12ycU];                % верхняя граница

% Масштабир.настраив.параметров (демасштаб. в функц. F_changeFISmg)
Msht = [1 1 1 1 1 1 1 1 1 1 0.1 0.1 0.1 0.1 0.1 1 1 0.04 0.04 0.04];
% Msht = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]; % отсутствие масшт.парам.
ParamFis0 = ParamFis0 .* Msht;
ParamFisL = ParamFisL .* Msht;
ParamFisU = ParamFisU .* Msht;

%-----

```

## 5. Задать основные параметры оптимизации (см. справку по функции **fmincon** пакета **Toolbox Optimization**).

```

%-----
% ПАРАМЕТРЫ ОПТИМИЗАЦИИ

% optimset('fmincon')      % вывод структуры с инф.о парам. алгоритма оптим.

% options = [];
options = optimset('Display', 'iter');          % вывод инф. на каждой итерации
options.MaxIter = 25;                          % максимальное число итераций
options.DiffMinChange = 0.0001;
options.DiffMaxChange = 0.2;
options.LargeScale = 'off';

%-----

```

## 6. Разработать функцию обновления параметров системы нечёткого вывода, которые были выбраны в пункте 4 для настройки в процессе адаптации.

В том случае, если для повышения эффективности работы алгоритма адаптации использовалось масштабирование настраиваемых параметров, то в теле данной функции должно быть предусмотрено их демасштабирование.

Сохранить разработанную функцию в текущей директории под именем: «F\_changeFISmg.m».

```
function FISnew = F_changeFISmg(param, FISold, Msht)
% Установка новых параметров (param) нечеткой системы FISold

FISnew = FISold;

% Демасштабирование настраиваемых параметров
param = param ./ Msht;

% Весовые коэффициенты правил
FISnew.rule(5).weight = param(1);
FISnew.rule(6).weight = param(2);
FISnew.rule(7).weight = param(3);

% Коэффициенты концентрации термов входных переменных
FISnew.input(1).mf(1).params(1) = param(4);
FISnew.input(1).mf(2).params(1) = param(5);
FISnew.input(1).mf(3).params(1) = param(6);
FISnew.input(2).mf(1).params(1) = param(7);
FISnew.input(2).mf(2).params(1) = param(8);
FISnew.input(2).mf(3).params(1) = param(9);

% Коэффициенты концентрации термов выходных переменных
FISnew.output(1).mf(1).params(1) = param(10);
FISnew.output(1).mf(2).params(1) = param(11);
FISnew.output(1).mf(3).params(1) = param(12);
FISnew.output(1).mf(4).params(1) = param(13);
FISnew.output(1).mf(5).params(1) = param(14);

% коорд.максимумв некотор.термов входных переменных
FISnew.input(1).mf(2).params(2) = param(15);
FISnew.input(2).mf(2).params(2) = param(16);

% коорд.максимумв некотор.термов выходных переменных
FISnew.output(1).mf(2).params(2) = param(17);
FISnew.output(1).mf(3).params(2) = param(18);
FISnew.output(1).mf(4).params(2) = param(19);
```

7. Разработать функцию для определения ошибки аппроксимации и сохранить её в текущей директории под именем: «F\_errFISmg.m».

```
function errFISmg = F_errFISmg(param, FISiter, inputFIS, target, Msht)
% Расчет ошибки при новых параметрах (param) нечеткой системы FISiter

% Установка новых параметров нечеткой системы
FISiter = F_changeFISmg(param, FISiter, Msht);

% Нечеткий вывод:
outFIS = evalfis(inputFIS, FISiter);

% Расчет ошибки:
errFISmg = sqrt(sum((target-outFIS).^2)/numel(outFIS)); % среднекв.знач.ошибки
```

## 8. Запустить процесс адаптации системы нечёткого вывода

```
%-----
% ОПТИМИЗАЦИЯ

[ParamFis_opt, sqrtFis, flag] = fmincon(@F_errFISmg, ParamFis0, [],[],[],[], ...
    ParamFisL, ParamFisU, [], options, fisMg, [xlob, x2ob], yob, Msht);

fisMgOpt=F_changeFISmg(ParamFis_opt, fisMg, Msht); % сист.нечёт.выв.после оптимиз.

% fuzzy(fisMgOpt)
% showfis(fisMgOpt)

%-----
```

## 9. Построить трёхмерное изображение результата аппроксимации заданной зависимости системой нечёткого вывода после адаптации.

```
%-----
% ПОСТРОЕНИЕ ГРАФИКА НЕЧЁТКОГО ОТОБРАЖЕНИЯ МАМДАНИ ПОСЛЕ АДАПТАЦИИ

yMgOpt = zeros(n, n);
for i = 1:n
    yMgOpt(i,:) = evalfis([x1; ones(size(x1))*x2(i)], fisMgOpt)';
end

%-----

h4 = figure(4);
set(h4,'Position',[1154 52 524 407])
colormap('default')
surf(x1, x2, yMgOpt)
axis([-10 5 ...
      -6 2 ...
      -50 50]);
view(-40,30)
xlabel('x_1'); ylabel('x_2'); zlabel('y');
title('Система нечёткого вывода Мамдани после адаптации')

pause(2.5); % пауза 2,5 с

colormap('white') %
hold on
plot3(xlob, x2ob, yob, 'bs', 'MarkerSize',3, 'MarkerFaceColor','b')
plot3(xlts, x2ts, yts, 'ro', 'MarkerSize',3, 'MarkerFaceColor','r')
hold off

%-----
```

## 10. Определить среднеквадратические значения ошибок аппроксимации заданной нелинейной зависимости при помощи исходной системы нечёткого вывода и системы нечёткого вывода, полученной после выполнения параметрической оптимизации.

Полученные значения отобразить в поле соответствующих графических зависимостей.

```

%-----
% Вычисление среднеквадратических значений ошибки нечёткой аппроксимации
%-----

% FIS Мамдани до оптимизации
ytsMg = evalfis([xlts, x2ts], fisMg); % реакция FIS на тест.выборку
RMSE_Mg = sqrt(sum((yts-ytsMg).^2)/numel(yts)); % среднев.знач.ошибки

% FIS Мамдани после оптимизации
ytsMgOpt = evalfis([xlts, x2ts], fisMgOpt); % реакция FIS на тест.выборку
RMSE_MgOpt = sqrt(sum((yts-ytsMgOpt).^2)/numel(yts)); % среднев.знач.ошибки

disp(' ')
disp('Средние квадратические значения ошибки аппроксимации:')
disp([' - до адаптации: RMSE = ', num2str(RMSE_Mg)])
disp([' - после адаптации: RMSE = ', num2str(RMSE_MgOpt)])
disp(' ')

figure(2);
xlabel(['x_1 RMSE = ', num2str(RMSE_Mg)]);
figure(4);
xlabel(['x_1 RMSE = ', num2str(RMSE_MgOpt)]);

%-----

```

**11.** Повторить процесс адаптации системы нечёткого вывода (пункты 8, 9, 10) без использования масштабирования настраиваемых параметров (вектор `msht` в пункте 4). Сравнить полученные результаты.

**12.** Сделать выводы из проделанной работы, в которых отразить:

- область возможного применения адаптации систем нечёткого вывода для решения прикладных задач;
- особенности выбора оптимизируемых параметров системы нечёткого вывода;
- особенности использования масштабирования настраиваемых параметров в процессе адаптации.

**13.** Подготовить отчёт по лабораторной работе в виде pdf-файла с именем:

**LR6\_КалендарныйГод\_ФамилияИсполнителя.pdf**

(Фамилия исполнителя в имени файла отчёта приводится в латинской транскрипции).

### **Содержание отчёта:**

1. Фамилия, имя и отчество студента, выполнившего работу;
2. Номер учебной группы;
3. Дата выполнения работы;
4. Название работы;
5. Цель работы;

6. m-программа адаптации системы нечёткого вывода к нелинейной зависимости согласно варианту задания (вариант задания был получен при выполнении лабораторной работы №1).



7. m-программы функций обновления параметров системы нечёткого вывода и определения ошибки аппроксимации.

8. Графики:

- исходной зависимости согласно варианту задания;
- распределения точек обучающей и тестовой выборок данных в области определения аппроксимируемой зависимости;
- результат аппроксимации системой нечёткого вывода заданной зависимости до адаптации;
- результат аппроксимации системой нечёткого вывода заданной зависимости после адаптации.

9. Средние квадратические значения ошибки аппроксимации:

- до адаптации;
- после адаптации с использованием масштабирования настраиваемых параметров;
- после адаптации без использования масштабирования настраиваемых параметров.

10. Выводы по работе.