МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

Кафедра СТ

Отчёт

по лабораторной работе №1

по курсу: «Нечёткие регуляторы в системах автоматизированного управления»

|  |  |
| --- | --- |
| Выполнил:  ст. гр. КСУАм-16-1  Пивоваров С. Н. | Проверил:  доц. Коваленко А. И. |

Харьков 2017

1 РЕШЕНИЕ ТРАНСПОРТНОЙ ЗАДАЧИ МЕТОДАМИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

1.1 Цель работы

Изучить основные этапы построения нечеткого регулятора в интерактивной среде разработки MATLAB. Изучить методы фаззификации и дефаззификации, используемые при синтезе нечетких регуляторов систем автоматического управления (САУ). Изучить программные функции нечеткой логики, реализованные в среде MATLAB, и интерфейс их использования. Изучить на практике методы построения нечетких множеств с использованием различных типов функций принадлежности. Научится использовать операции над нечеткими множествами. Изучить функции пакета прикладных программ FUZZY LOGIC TOOLBOX среды MATLAB, используемых на этапе дефаззификации.

1.3 Исходные данные

Данные взяты для варианта №13 и приведены в таблице 1.1.

Таблица 1.1 – Исходные данные

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Experiment 1 | | | | | | | | | | | |
| a | b | c | d | σ1 | | c1 | σ2 | c2 | σ | m..n | Δ |
| 13 | 14 | 15 | 17 | 2 | | 13 | 1 | 15 | 1.5 | 10..20 | 0.1 |
| Experiment 2 | | | | | | | | | | | |
| a | b | c | d | σ1 | | c1 | σ2 | c2 | σ | SF1, SF2 | SF3 |
| 7 | 8 | 9 | 11 | 2 | | 7 | 1 | 9 | 0.9 | gaussmf | |
| 8 | 9 | 10 | 12 | 2 | | 8 | 1 | 10 | 1 |
| Experiment 3 | | | | | | | | | | | |
| k1=0.7 | | | | | k2=0.9 | | | | | k3=0.7 | |

1.4 Ход работы

Все функции в первом эксперименте были проведены на интервале х [10;20]. Результаты приведены в таблицах 1.2 – 1.12. Тело описания функции в MatLab приведено ниже:

x=10:0.1:20; % Базовое множество (min:шаг:max)

'функция принадлежности'; % Определяется треугольная ФП

plot(x,y);% Вывод графика функции

xlabel('подпись');% Подпись под осью абсцисс

ylabel('подпись')% Подпись под осью ординат

Таблица 1.2 – Результат фазификации при помощи треугольной функции принадлежности trimf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=trimf(x,[13 14 15]) |
| Носитель = (13;15) |
| Высота = [14] |
| Ядро = [14] |
| Нормальное |
| Граница = (13;14)U(14;15) |
| Точки перехода = 13.5; 14.5 |
|  |

Таблица 1.3 – Результат фазификации при помощи трапециевидной функции принадлежности trapmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=trapmf(x,[13 14 15 17]) |
| Носитель = (13;1) |
| Высота = [14;15] |
| Ядро = [14;15] |
| Нормальное |
| Граница = (13;14)U(15;17) |
| Точки перехода = 13.5; 16 |
|  |

Таблица 1.4 – Результат фазификации при помощи колокольной функции принадлежности gbellmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=gbellmf(x,[13 14 15]) |
| Носитель = [10;20] |
| Высота = [15] |
| Ядро = [15] |
| Нормальное |
| Граница = [10.2;15)U(15;19.8] |
| Точки перехода за пределом множества |
|  |

Таблица 1.5 – Результат фазификации при помощи гауссовской функции принадлежности gaussmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=gaussmf(x,[1.5 15]) |
| Носитель = [10;20] |
| Высота = [15] |
| Ядро = [15] |
| Нормальное |
| Граница = [10;15)U(15;20] |
| Точки перехода 13.2; 16.8 |
|  |

Таблица 1.6 – Результат фазификации при помощи двухсторонней гауссовской функции принадлежности gauss2mf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=gauss2mf(x,[2 13 1 15]) |
| Носитель = [10;20] |
| Высота = [13;15] |
| Ядро = [13;15] |
| Нормальное |
| Граница = [10;13)U(15;20] |
| Точки перехода 10.6; 16.2 |
|  |

Таблица 1.7 – Результат фазификации при помощи сигмоидальной функции принадлежности sigmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=sigmf(x,[13 15]) |
| Носитель = [10;20] |
| Высота = [16.7;20] |
| Ядро = [16.7;20] |
| Нормальное |
| Граница [10;20) |
| Точка перехода 15 |
|  |

Таблица 1.8 – Результат фазификации при помощи п-образной функции принадлежности dsigmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=dsigmf(x,[13 13 14 15]) |
| Носитель = [10;20] |
| Высота = [14] |
| Ядро = |
| Субнормальное |
| Граница [10;18) |
| Точки перехода 13; 15 |
|  |

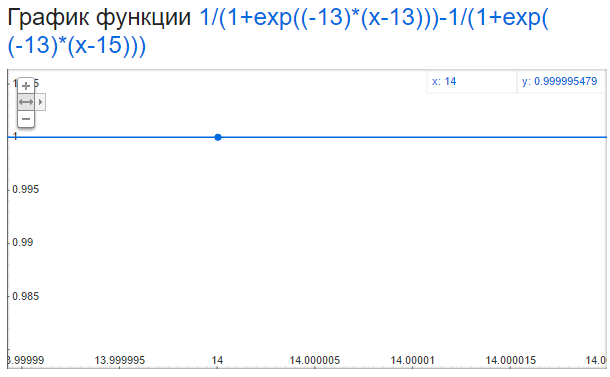


Рисунок 1.1 – Значение высоты для п-образной функции принадлежности

Таблица 1.9 – Результат фазификации при помощи произведения двух сигмоидальных функций принадлежности psigmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=psigmf(x,[13 14 15 17]) |
| Носитель = [10;20] |
| Высота = [18.4;20] |
| Ядро = [18.4;20] |
| Нормальное |
| Граница [10;18.4) |
| Точка перехода 17 |
|  |

Таблица 1.10 – Результат фазификации при помощи п-образной функции принадлежности pimf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=pimf(x,[13 14 15 17]) |
| Носитель = (13;17) |
| Высота = [14;15] |
| Ядро = [14;15] |
| Нормальное |
| Граница (13;14)U(15;17) |
| Точки перехода 13.5; 16 |
|  |

Таблица 1.11 – Результат фазификации при помощи S-функции принадлежности smf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=smf(x,[13 14]) |
| Носитель = (13;14) |
| Высота = [14;20] |
| Ядро = [14;20] |
| Нормальное |
| Граница (13;14) |
| Точка перехода 13.5 |
|  |

Таблица 1.12 – Результат фазификации при помощи Z-функции принадлежности zmf

|  |  |
| --- | --- |
| Команда в MatLab: |  |
| y=zmf(x,[13 14]) |
| Носитель = (13;14) |
| Высота = [10;13] |
| Ядро = [10;13] |
| Нормальное |
| Граница (13;14) |
| Точка перехода 13.5 |
|  |

Результаты второго эксперимента приведены в рисунках 1.2 – 1.8. Тела описания операций приведены ниже перед соответствующими рисунками.

Ввод данных для операции пересечения:

x=0:0.1:15;

plot(1,2,1); %график 1

y1=gaussmf(x,[2 7]); % ФП gaussmf

y2=gaussmf(x,[2 8]); % ФП gaussmf

y3 = min([y1; y2]); % пересечение Y3= Y1∩Y2 (min для минимаксной функции, prod для % функции алгебраического умножения)

plot(x,[y1;y2]); % вывод графика

legend('y1','y2')% вывод легенды

subplot(1,2,2); % график 2

plot(x,[y1;y2],':'); % вывод графиков, ':' – пунктир

hold on;% режим добавления в текущей график

plot(x,y3); % вывод графика

legend('y1','y2','y3')% вывод легенды

hold off; % выключение режима hold

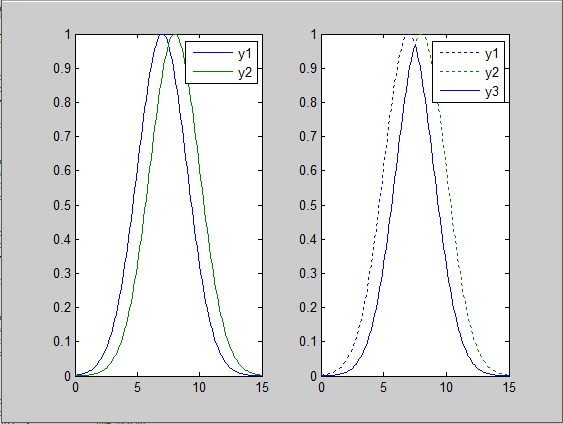


Рисунок 1.2 – Результат пересечения полученный при помощи минимаксной функции

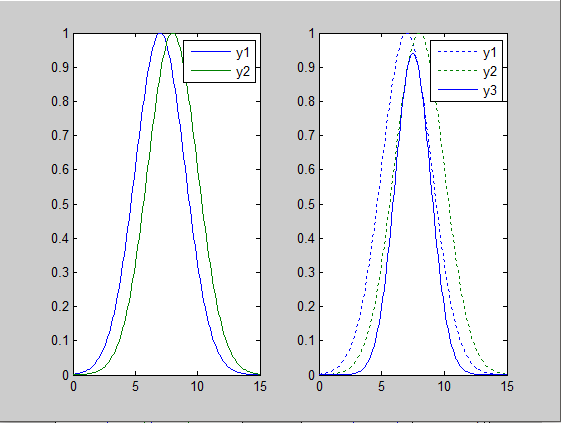


Рисунок 1.3 – Результат пересечения полученный при помощи функции алгебраического умножения

Ввод данных для операции объединения:

x=0:0.1:15; % базовое множество

sp1=subplot(1,2,1); % график 1

y1=gaussmf(x,[2 7]); % ФП gaussmf

y2=gaussmf(x,[2 8]); % ФП gaussmf

y3=max([y1; y2]); % объединение Y3= Y1UY2(max для минимаксной функции, probor для % функции алгебраического сложения)

plot(x,[y1;y2]); % вывод графика

grid(sp1,'on')% отображать сетку

legend('y1','y2')% вывод легенды

subplot(1,2,2); % график 2

plot(x,[y1;y2],':'); % вывод графиков, ':' – пунктир

hold on;%режим добавления в текущей график

plot(x,y3); % вывод графика

legend('y1','y2','y3')% вывод легенды

hold off; % выключение режима hold

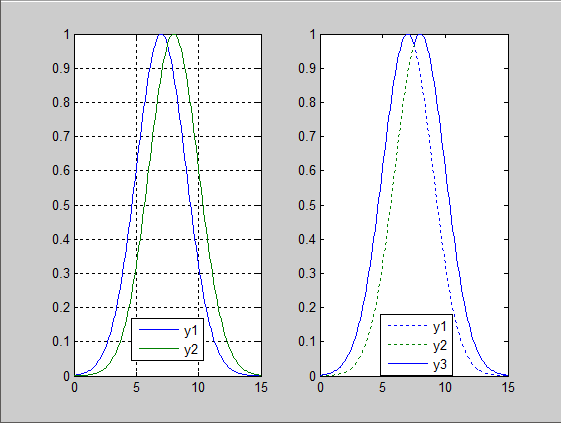


Рисунок 1.4 – Результат пересечения полученный при помощи минимаксной функции

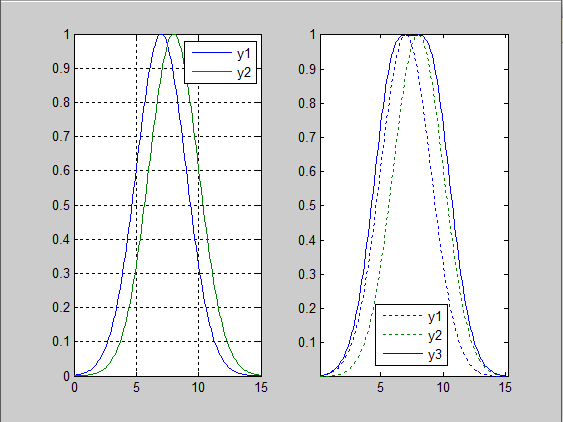


Рисунок 1.5 – Результат пересечения полученный при помощи функции алгебраического сложения

Ввод данных для операции дополнения:

x=0:0.1:15; % базовое множество

y1=gaussmf(x,[2 7]); % ФП gaussmf

y2=1- y1; % Y2 дополнение Y1

plot(x,[y1;y2]); % вывод графика

ylim ([0 1.05])%задание границ оси Y

grid on% отображать сетку

legend('y1','y2')% вывод легенды

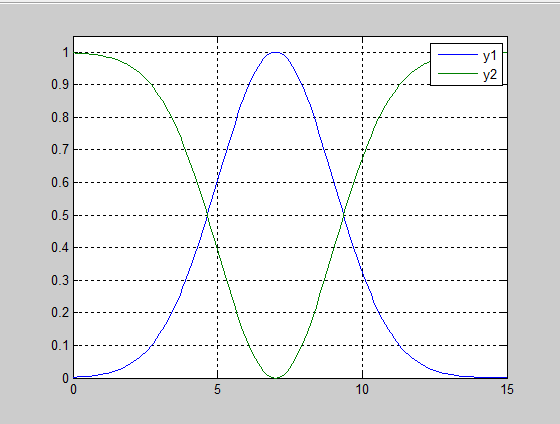


Рисунок 1.6 – Результат операции дополнения

Ввод данных для операции асимметричной разности:

x=0:0.1:15; % базовое множество

sp1=subplot(1,2,1); % график 1

y1=gaussmf(x,[2 7]); % ФП gaussmf

y2=gaussmf(x,[2 8]); % ФП gaussmf

y4=0\*x;

y3=max([y4;y1-y2]);

plot(x,[y1;y2;y3]); % вывод графика

ylim ([0 1.05])%задание границ оси Y

grid(sp1,'on')% отображать сетку

legend('y1','y2','y3=y1-y2')% вывод легенды

sp2=subplot(1,2,2); % график 2

y3=max([y4;y2-y1]); % Y3= max{0,Y2-Y1}

plot(x,[y1;y2;y3]);% вывод графиков

grid(sp2,'on')% отображать сетку

legend('y1','y2','y3=y2-y1')% вывод легенды

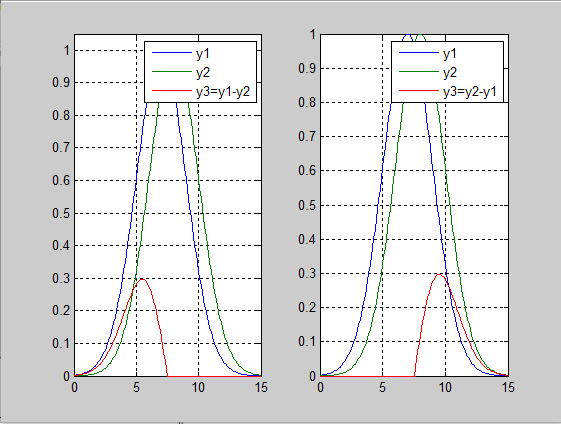


Рисунок 1.7 – Результат выполнения операции ассиметричной разности

Ввод данных для операции симметричной разности:

sp1=subplot(1,2,1); % график 1

y1=gaussmf(x,[2 7]);

y2=gaussmf(x,[2 8]);

y3=abs(y1-y2);

plot(x,[y1;y2;y3]);

ylim ([0 1.05

grid(sp1,'on')% отображать сетку

legend('y1','y2','y3=|y1-y2|')% вывод легенды

sp2=subplot(1,2,2); % график 2

y3=abs(y2-y1);

plot(x,[y1;y2;y3]);

grid(sp2,'on'

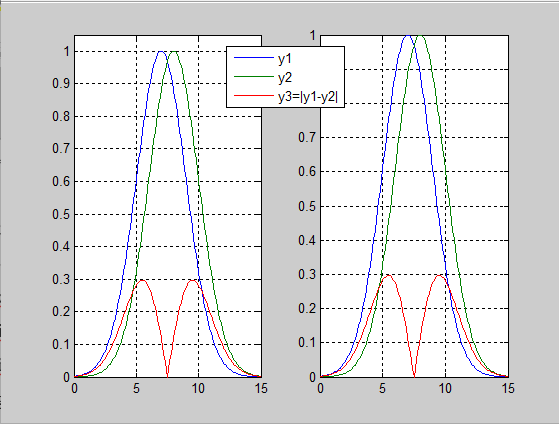


Рисунок 1.8 – Результат выполнения операции симметричной разности

Результаты третьего эксперимента приведены ниже в рисунках 1.9 – 1.10. Тела описания операций приведены ниже перед соответствующими рисунками.

Задание данных варианта:

k1=0.7

k2=0.9;

k3=0.7;

x = -10:0.1:10; % базовое множество

mf1 = trapmf(x,[-10 -8 -2 2]); % функция 1

mf2 = trapmf(x,[-5 -3 2 4]); % функция 2

mf3 = trapmf(x,[2 3 8 9]); % функция 3

mf1 = max(k1\*mf2,max(k2\*mf1,k3\*mf3)); % функция региона

Для дефазификации методами центра тяжести и методом центра площади необходимо использовать следующий набор команд:

P1=plot(x,mf1); % вывод графика региона

set(P1, 'LineWidth', 4, 'Color', 'b')% параметры линии

ylim([0 1]); %задание границ оси Y

x1 = defuzz(x,mf1,'centroid'); % нахождение ЦТ centroid

x2 = defuzz(x,mf1,'bisector'); % нахождение ЦП bisector

L1 = line([x1 x1],[-0.2 1.2])% вывод линии ЦП

L2 = line([x2 x2],[-0.2 1.2])% вывод линии ЦП

set(L1,'LineWidth', 3, 'Color','r')% параметры линии L1

set(L2,'LineWidth', 3, 'Color','g')% параметры линии L2

T1 = text(x1,0.2,'centroid')% текст возле линии L1

T2 = text(x1-8,0.2,'bisector')% текст возле линии L2

set(T1,'FontSize', 14,'FontWeight','bold','Color','r')

set(T2,'FontSize', 14,'FontWeight','bold','Color','g')

legend('mf1', 'centroid','bisector') % вывод легенды

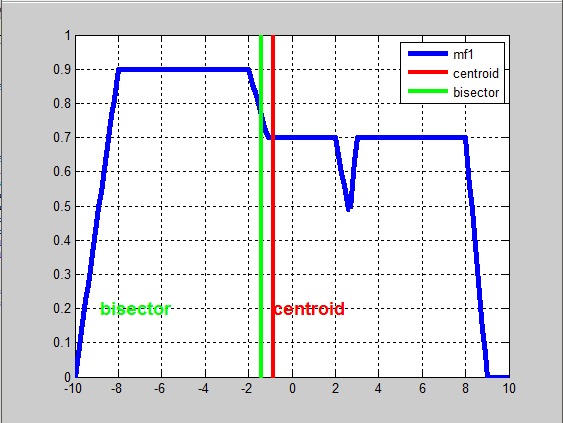


Рисунок 1.9 – Результат дефазификации методами центра тяжести (centroid) и центра площади (bisector)

Для дефазификации методами левого, правого и среднего модального значения необходимо выполнить следующий ряд команд:

P1=plot(x,mf1); % вывод графика региона

set(P1,'LineWidth',4,'Color','k')% параметры линии

ylim([0 1]); % задание границ оси Y

x3 = defuzz(x,mf1,'mom')% нахождение MOM

L3 = line([x3 x3],[-0.2 1.2])% вывод линии

set(L3,'LineWidth',3,'Color','b')% параметры линии

x4 = defuzz(x,mf1,'som') % нахождение SOM

L4 = line([x4 x4],[-0.2 1.2])% вывод линии

set(L4,'LineWidth',3,'Color','r')% параметры линии

x5 = defuzz(x,mf1,'lom') % нахождение LOM

L5 = line([x5 x5],[-0.2 1.2])% вывод линии

set(L5,'LineWidth',3,'Color','m')% параметры линии

T3 = text(x3+0.2,0.2,'mom')% текст возле линии

set(T3,'FontSize',14,'FontWeight','bold','Color','b')

T4 = text(x4+0.2,0.2,'som')% текст возле линии

set(T4,'FontSize',14,'FontWeight','bold','Color','r')

T5 = text(x5+0.2,0.2,'lom')% текст возле линии

set(T5,'FontSize',14,'FontWeight','bold','Color','m')

legend('mf1','mom','som','lom') % вывод легенды

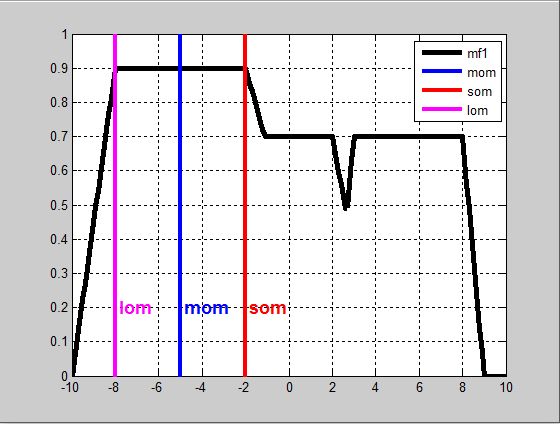


Рисунок 1.10 – Результат дефазификации методами левого, правого и среднего модального значения

ВЫВОДЫ

В первой части лабораторной работы были исследованы различные функции принадлежности, на основе которых можно строить нечёткие регуляторы. Были определены их характеристики, изучено влияние различных коэффициентов на её значение и построены графики, показывающие зависимость выхода от входа. Эта операция называется фазификацией и заключается в создании нечёткости при помощи функций принадлежности.

Для построения нечёткого регулятора одной функции принадлежности может быть недостаточно. Для использования 2 и более в одном регуляторе необходимо использовать операции для согласования множества выходов. Различные способы операций разности, дополнения, пересечения и объединения были продемонстрированы во второй части лабораторной работы.

В третьем этапе была проведена операция дефазификации, которая заключается в определении конкретного значения нечёткой переменной для входного значения.