

УДК 004

Метод управления трехзвенным манипулятором с коррекцией ошибки на основе использования нейро-нечеткой сети

***Шанов Д. Н., студент**
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

*Научный руководитель: **Рогозин О.В., к.т.н., доцент** кафедры «Программное обеспечение
ЭВМ и информационные технологии»
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru*

Введение

Одним из важнейших направлений развития современной робототехники является разработка интеллектуальных систем управления. Долгое время популярностью пользовались математические методы, но они требуют создания сложных программно-алгоритмических комплексов способных работать только в predetermined условиях. Но развитие теории искусственного интеллекта позволяет создавать гибкие системы управления. Уже существует множество подходов к созданию таких систем, примерами могут служить: экспертные системы, нейронные сети, нечеткий вывод и другие. В зависимости от назначения объекта управления и его окружения в систему управления должен быть вложен критерий принятия решения. В большинстве случаев такими критериями служат минимизация ошибки выполнения работы и минимизация накладных расходов.

Цели и задачи:

Целью данной работы является создание нейро-нечёткого метода управления манипулятором.

Для достижения поставленной цели необходимо решить следующие задачи:

- Создание модели нейро-нечёткого управления манипулятора;
- Определение управляющих воздействий на манипулятор;
- Моделирование работы манипулятора.

Кинематическая задача

На рис. 1 представлена кинематическая схема трёхстепенного планарно-ангулярного манипулятора.

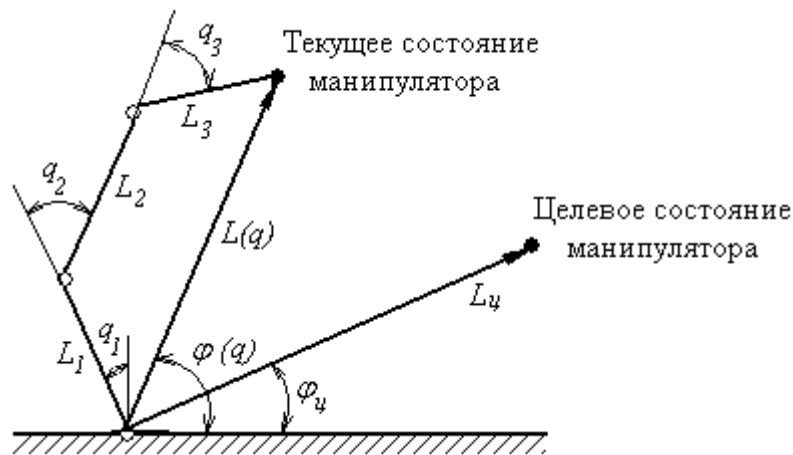


Рис. 1. Кинематическая схема трёхстепенного планарно - ангулярного манипулятора

Поскольку используется ангулярная система координат, то положение манипулятора определяется положением его звеньев. При этом относительное положение звеньев определяется вектором обобщённых координат, выступающих в роли управляющих параметров. Так исходя из значения полярных координат будет определяться величина и наклон соответствующего вектора положения кинематической цепи, а контролируемые рассогласования этих параметров для текущего и целевого состояний манипулятора - служить конструктивной основой при формировании соответствующего критерия управления:

$$\Delta L = L(q) - L_{\text{ц}} \leq \varepsilon_L,$$

$$\Delta \varphi = \varphi(q) - \varphi_{\text{ц}} \leq \varepsilon_{\varphi},$$

где $q = (q_1, q_2, q_3)$ - вектор обобщённых координат;

$L(q)$, $L_{\text{ц}}$ - величины векторов текущего и целевого положения манипулятора;

ΔL - рассогласование величин векторов текущего и целевого положений манипулятора, максимальное значение которого определяется:

$$\Delta L = L_1 + L_2 + L_3,$$

где L_1, L_2, L_3 - длины звеньев манипулятор;

ε_L - заданная точность минимизации рассогласования величин векторов текущего и целевого положений манипулятора;

$\varphi(q)$, $\varphi_{\text{ц}}$ - углы наклона векторов текущего и целевого положений манипулятора;

$\Delta\varphi$ - рассогласование углов наклона (направлений) векторов текущего и целевого положений манипулятора ($\Delta\varphi_{max} = \pi$);

ε_φ - заданная точность минимизации рассогласования направлений векторов текущего и целевого положений манипулятора.

Исходя из этих данных, можно сделать вывод:

- приращение первой обобщенной координаты обуславливает изменение только углового рассогласования векторов текущего и целевого положений манипулятора;
- приращение второй и третьей обобщенных координат обуславливает изменение рассогласований как величин, так и направлений векторов текущего и целевого положений манипулятора;
- постановка задачи управления движением манипулятора может быть сведена к минимизации рассогласований величин и направлений векторов его текущего и целевого положений путем независимого изменения отдельных обобщенных координат кинематической цепи.

Сравнение существующих подходов:

Математический метод – решение системы дифференциальных уравнений.

Плюсы данной системы:

1) Ещё до начала движения можно определить количество затраченных ресурсов

2) Предопределённость выполнения

Минусы:

1) Данный метод требует значительных вычислительных ресурсов на решение системы нелинейных уравнений

2) Не учитываются погрешности и ошибки (их можно определить, только после совершения поворотов)

3) Не происходит коррекция ошибки при каждом следующем вычислении

Нечеткий вывод – данный метод использует системы нечеткого вывода (примерами могут служить: метод Момдани и метод Суджено)

Плюсы:

1) Метод позволяет учитывать ошибки в ходе выполнения задачи

2) Не требует больших вычислительных ресурсов

3) Нет необходимости в построении полной математической модели

4) Возможность формирования правил поведения манипулятора на естественном языке

5) Легкость перенастройки правил поведения манипулятора в случае изменения параметров системы

Минусы:

1) Не происходит коррекция ошибки при каждом следующем определении действия

2) Метод производит расчеты по мере выполнения операций

Метод основанный на нейро-нечеткой логике.

Данный метод основан на том, что в него входят сразу два направления исследований искусственных интеллектов: нейронные сети и нечеткая логика. Это приводит к тому, что данный метод имеет часть недостатков и преимуществ своих составляющих.

Плюсы:

1) Метод позволяет учитывать ошибки в ходе выполнения задачи

2) Не требует больших вычислительных ресурсов

3) Нет необходимости в построении полной математической модели

4) Возможность формирования правил поведения манипулятора на естественном языке

5) Легкость перенастройки правил поведения манипулятора в случае изменения параметров системы

Минусы:

1) Метод производит расчеты по мере выполнения операций

2) Необходимость обучения нейронной сети

Нейро-нечеткая модель NEFCLASS

Целью модели NEFCLASS (NEuro Fuzzy CLASSifier) является получение нечетких правил из множества данных, которые можно разделить на различные классы. Нечеткие правила описывают данные в форме:

R: если x_1 имеет функцию принадлежности μ_1 , $x_2 - \mu_2, \dots, x_n - \mu_n$,

то образец $(x_1, x_2, \dots, x_n)$ принадлежит классу i , где $\mu_1, \dots, \mu_n$ - нечеткие множества.

Задача NEFCLASS состоит в том, чтобы определить принадлежность к классу $C \subset R^n$ входного образца $x = (x_1, \dots, x_n) \in R^n$. Подразумевается, что пересечение

двух различных множеств пусто. Система NEFCLASS имеет 3-слойную последовательную архитектуру. Первый слой U_1 содержит входные нейроны, в которых представляются входные образцы. Активация a_x нейрона $x \in U_1$ обычно не изменяет входное значение. Скрытый слой U_2 содержит нечеткие правила, и третий слой U_3 состоит из выходных нейронов каждого класса. Активация для нейронов правил $R \in U_2$ и для нейронов выходного слоя $c \in U_3$ с образцом p вычисляется так:

$$a_R^{(p)} = \min_{x \in U_1} \{W(x, R)(a_x^{(p)})\},$$

$$a_c^{(p)} = \sum_{R \in U_2} W(c, R) \cdot a_R^{(p)},$$

или альтернативно

$$a_c^{(p)} = \max_{R \in U_2} \{a_R^{(p)}\},$$

где $W(x, R)$ – нечеткий вес соединения входного нейрона x с нейроном правила R , а $W(R, c)$ – нечеткий вес соединения нейрона правила R с нейроном выходного слоя c . Вместо применения операций взятия максимума и минимума можно использовать другие функции t-нормы и t-конормы соответственно.

База правил представляет собой аппроксимацию неизвестной функции $\varphi: R^n \rightarrow \{0,1\}^m$ и описывает классификационную задачу, где $\varphi(x) = (c_1, \dots, c_m)$ такая, что $c_i = 1, c_j = 0$ ($j \in \{1, \dots, m\}, j \neq i$) и x принадлежит классу C_i .

Нечеткие множества и лингвистические правила представляют аппроксимацию и определяют результат системы NEFCLASS. Они получаются из множества выборок путем обучения. Обязательно должно выполняться правило, что для каждого лингвистического значения может существовать только одно представление нечеткого множества. На рисунке 2 представлена система NEFCLASS в виде 3-слойной нейронной сети с двумя входными нейронами.

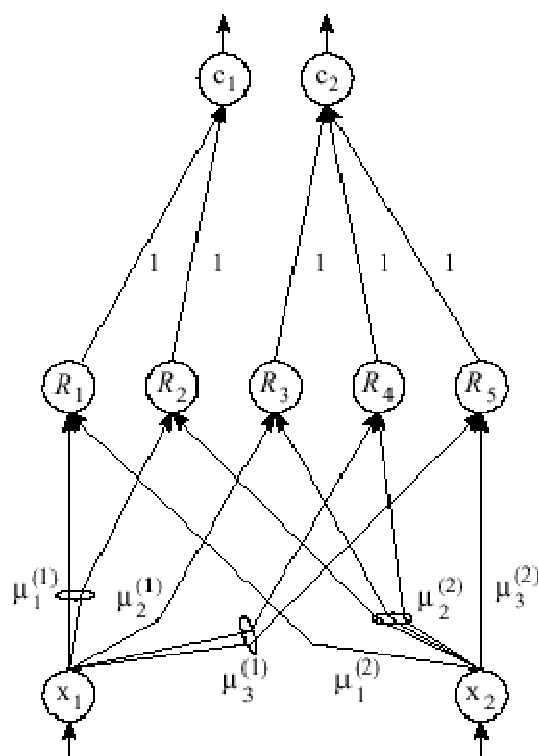


Рис. 2. Система NEFCLASS в виде 3-слойной нейронной сети
с двумя входными нейронами

Обучение в системе NEFCLASS

Система NEFCLASS может быть построена по частичным знаниям об образцах. Пользователь должен определить количество начальных нечетких множеств и задать значение k – максимальное число узлов правил, которые могут быть созданы в скрытом слое. Для обучения мы будем использовать определенную ранее треугольную функцию принадлежности.

Рассмотрим систему NEFCLASS с n входными нейронами $x_1, \dots, x_n$, $k \leq k_{\max}$ нейронами правил и m выходными нейронами $c_1, \dots, c_m$. Также задано обучающее множество образцов $L = \{(p_1, t_1), \dots, (p_s, t_s)\}$, каждый из которых состоит из входного образца $p \in \mathbb{R}^n$ и желаемого образца $t \in \{0, 1\}^m$. Обучающий алгоритм, цель которого создать k нейронов правил системы NEFCLASS, состоит из следующих этапов:

1. Выбираем следующий образец (p, t) из L .
2. Для каждого входного нейрона $x_i \in U_1$ находим такую функцию принадлежности $\mu_{j_1}^{(p)}$, что

$$\mu_{j_1}^{(p)} p_{(i)} = \max_{j \in 1, \dots, q_1} \left\{ \mu_{j_1}^{(p)} p_{(i)} \right\}$$

3. Если по-прежнему узлов правил осталось $< k_{\max}$ и не существует узла правила R с

$$W(x_1, R) = \mu_{j_1}^{(1)}, \dots, W(x_n, R) = \mu_{j_n}^{(n)},$$

то создаем такой узел и соединяем его с выходным узлом c_1 , если $t_1 = 1$.

4. Если еще остались необработанные образцы в L и $k \leq k_{\max}$, то едем на шаг (i), а иначе стоп.

5. Определяем базу правил по одной из трех следующих процедур:

а) “Простое” обучение правил: оставляем только первые k правил (останавливаем создание правил, если было создано $k = k_{\max}$ правил).

б) “Лучшее” обучение правил: обрабатываем образцы в L и накапливаем активации каждого нейрона правил для каждого класса образцов, которые были распространены.

Если нейрон правила R показывает большее накопление активации для класса C_j , чем для класса C_R , который был специфицирован для следствия правила, тогда изменяем следствие R на C_j , то есть соединяем R с нейроном выхода c_j . Продолжаем обработку образцов в L дальше и вычисляем для каждого нейрона правил:

$$V_R = \sum_{p \in L} a_R^{(p)} \cdot e_p,$$

$$e_p = \begin{cases} 1, & \text{если } p \text{ классифицировано правильно} \\ -1, & \text{иначе} \end{cases}$$

Оставляем k нейронов правил с наивысшими значениями V_R и удаляем другие нейроны правил из системы NEFCLASS

в) “Лучший для каждого класса” алгоритм обучения: действуем также как и в предыдущем случае, но оставляем для каждого класса C_j те лучшие $\left\lceil \frac{k}{m} \right\rceil$ правил, следствия которых представляют класс C_j (где $[x]$ – целая часть от x).

Алгоритм обучения нечетких множеств

Алгоритм обучения с учителем системы NEFCLASS должен адаптировать его нечеткие множества, пробегать циклически через все обучающее множество L, продолжая ниже перечисленные шаги, пока не выполнится один из критериев останова.

Шаги:

1. Выбираем следующий образец (p,t) из L и распространяем его через систему NEFCLASS и определяем выходной вектор c .

2. Для каждого выходного нейрона c_i определяем значение δ_{c_i}

$$\delta_{c_i} = t_i - a_{c_i}$$

3. Для каждого нейрона правил R с $a_R > 0$:

а) Определяем значение δ_R , равное

$$\delta_R = a_R \cdot (1 - a_R) \cdot \sum_{c \in U_3} W(R, c) \delta_c$$

б) Находим такое x' , что

$$W(x', R)(a_{x'}) = \min_{x \in U_1} \{W(x, R)(a_x)\}$$

в) Для нечетких множеств $W(x', R)$ определяем $\delta_a, \delta_b, \delta_c$, используя скорость обучения $\sigma > 0$:

$$\delta_b = \sigma \cdot \delta_R \cdot (c - a) \cdot \text{sgn}(a_{x'} - b)$$

$$\delta_a = -\sigma \cdot \delta_R \cdot (c - a) \cdot \delta_b$$

$$\delta_c = \sigma \cdot \delta_R \cdot (c - a) \cdot \delta_b$$

и применяем изменения к $W(x', R)$

г) Вычисляем ошибку правила:

$$E = a_R \cdot (1 - a_R) \cdot \sum_{c \in U_3} (2 \cdot W(R, c) - 1) \delta_c$$

В качестве критерия останова можно взять, например, такие:

1. Ошибка на протяжении n итераций не уменьшается.
2. Прекратить обучение по достижении ошибкой определенного (желательно близкого к нулю) значения.

Нейро-нечеткая сеть ANFIS

Настройка параметров функций принадлежности сети ANFIS.

Настройка (параметрическая идентификация модели) заключается в настройке параметров функций принадлежности на основе обучающего множества с помощью нейронных сетей.

Система вывода ANFIS реализуется в виде нейроподобной структуры, состоящей из пяти слоев (рис. 3).

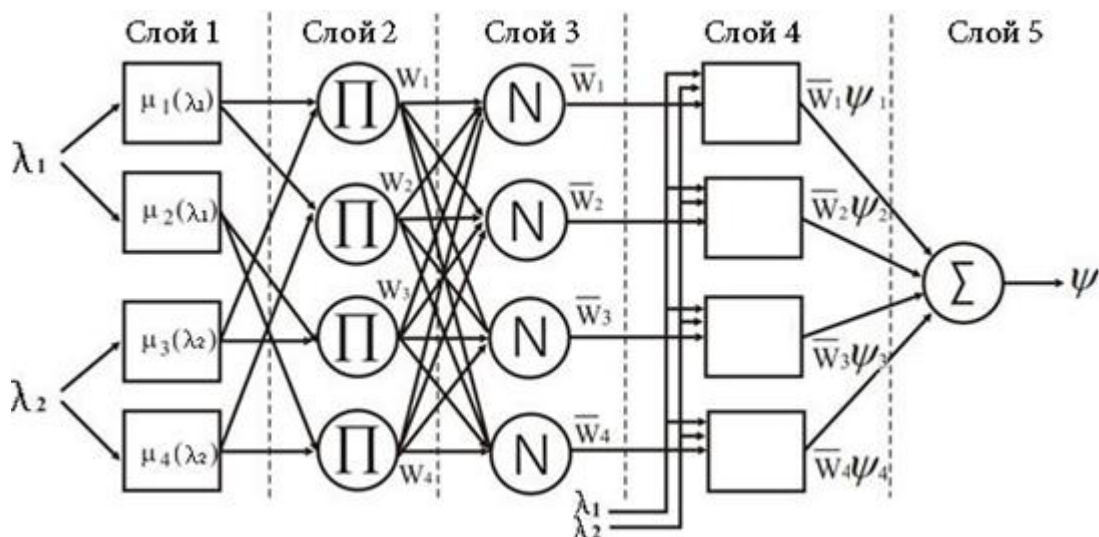


Рис. 3. Структура нейро-нечеткой сети ANFIS

Нейроны первого слоя служат для представления термов входных переменных. При конкретных (заданных) значениях входов на выходе слоя формируются значения функций принадлежности.

Нейроны второго слоя исполняют роль антецедентов (посылок) нечетких правил. Выходами этого слоя являются степени истинности предпосылок каждого правила базы знаний системы.

Нейроны третьего слоя служат для нормализации степеней выполнения правил. Выход i -го нейрона этого слоя представляет собой отношение степени истинности предпосылки i -го правила к сумме степеней предпосылок всех правил.

Четвертый слой предназначен для вычисления заключений правил. Здесь

$$\psi_i = p_i \lambda_1 + q_i \lambda_2 + r_i,$$

где $\{p_i, q_i, r_i\}$ – набор параметров данного слоя (так называемые параметры вывода).

Пятый слой выполняет агрегирование результата, полученного по различным правилам. Единственный нейрон этого слоя вычисляет выходное значение сети.

Для обучения сети ANFIS в работе используется гибридный градиентный метод.

Для оценки погрешности нечеткой нейронной сети строится функция ошибки

$$E(a, b, p, q, r) = \frac{1}{n} \sum_i \left(\psi(\Lambda_i) - \bar{\psi}(\Lambda_i) \right)^2 \rightarrow \min_{a, b, p, q, r}, \quad i \in [1: N],$$

представляющая собой среднеквадратическое отклонение между фактическими значениями выходной переменной $\psi(\Lambda_i)$ и точечной оценкой $\bar{\psi}(\Lambda_i)$, полученной на

основе нечеткого вывода. Здесь a, b – параметры функций принадлежности первого слоя нечеткой нейронной сети; p, q, r – параметры четвертого слоя нечеткой нейронной сети.

Генерация нечетких правил.

В процессе решения задачи система вывода ANFIS имеет возможность приобретать новые знания, поэтому база знаний не остается фиксированной, а модернизируется по мере поступления экспериментальных данных. Поэтому топология нечеткой нейронной сети также меняется по мере приобретения новых знаний. Для модификации топологии нечеткой нейронной сети используется процедура, предложенная в работе.

Пусть исследуемый объект имеет на входе 3-мерную координату, на выходе – число y – угол поворота манипулятора. Пусть кроме того «истинное» описание объекта имеет вид

$$y = f(x) + e,$$

где $f(x)$ – неизвестная функция, e – случайная аддитивная помеха. Предположим, что может быть реализован эксперимент, заключающийся в регистрации N наборов значений $\langle x_i, y_i \rangle$, $i=1,2$. При этом компоненты вектора x , измеряются без ошибок, значение величины N при необходимости допускает модификацию.

Алгоритм построения системы вывода ANFIS имеет следующий вид.

ШАГ 1. Из m , ($m < n$) произвольных наборов значений $\langle x_i, y_i \rangle$ составляется начальная база знаний модели.

$$P_i: \text{ЕСЛИ } x_1 \text{ есть } x_{1i} \text{ и } x_2 \text{ есть } x_{2i} \text{ и } \dots \text{ и } x_n \text{ есть } x_{ni}, \text{ ТО } y = y_i, i = 1, \dots, m$$

ШАГ 2. Для каждого нового экспериментального набора $\langle x_i, y_i \rangle$ определяется прогнозируемое значение $\hat{y}$ величины y .

ШАГ 3. Проверяется неравенство

$$|y - \hat{y}| > \varepsilon,$$

где ε – заданная константа, определяющая погрешность аппроксимации. При выполнении неравенства база знаний системы пополняется.

ШАГ 4. Проверяется **выполнение** правила останова. Если это правило выполнено, то алгоритм завершается; если правило не выполнено, алгоритм переходит к шагу 2.

Исследование работы различных методов и подходов

Количество шагов в зависимости от количества термов

Рассмотрим манипулятор из трёх звеньев общей длины 9.

Точность равна $\Delta L = 0.05$, $\Delta \phi = 0.01$ и, соответствующее им, $\Delta R = 0.06$

Рассмотрим алгоритмы для одного терма, двух термов, трёх термов и четырех термов.

На рисунке 4 представлены виды термов, которые использовались для исследования.

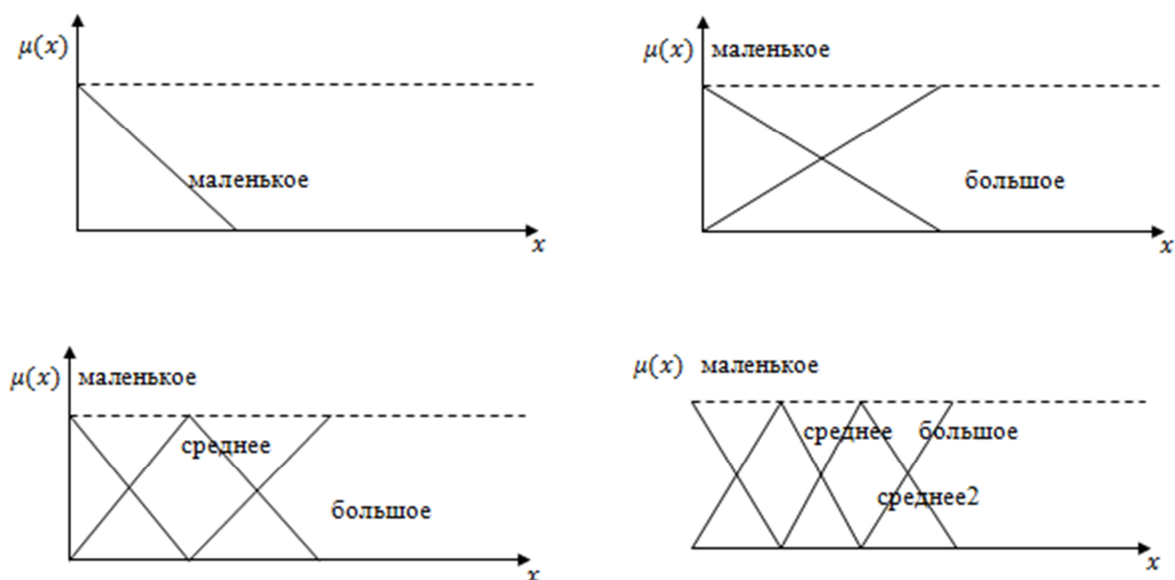


Рис. 4. Виды термов

В таблице 1 представлены результаты исследования зависимости количества шагов от количества термов.

Таблица 1

Количество шагов от количества термов

	Mamdani	ANFIS	NEFCLASS
1 терм	110	96	105
2 терма	131	89	126
3 терма	124	87	114
4 терма	185	74	124

Максимальная точность в зависимости от функций принадлежности выходных параметров

Рассмотрим манипулятор из трёх звеньев общей длины 9.

Точность равна $\Delta L = 0.05$, $\Delta \varphi = 0.01$ и, соответствующее им, $\Delta R = 0.06$

Количество термов - 2.

Для выходных параметров будем использовать функции принадлежности, представленные на рисунке 5.

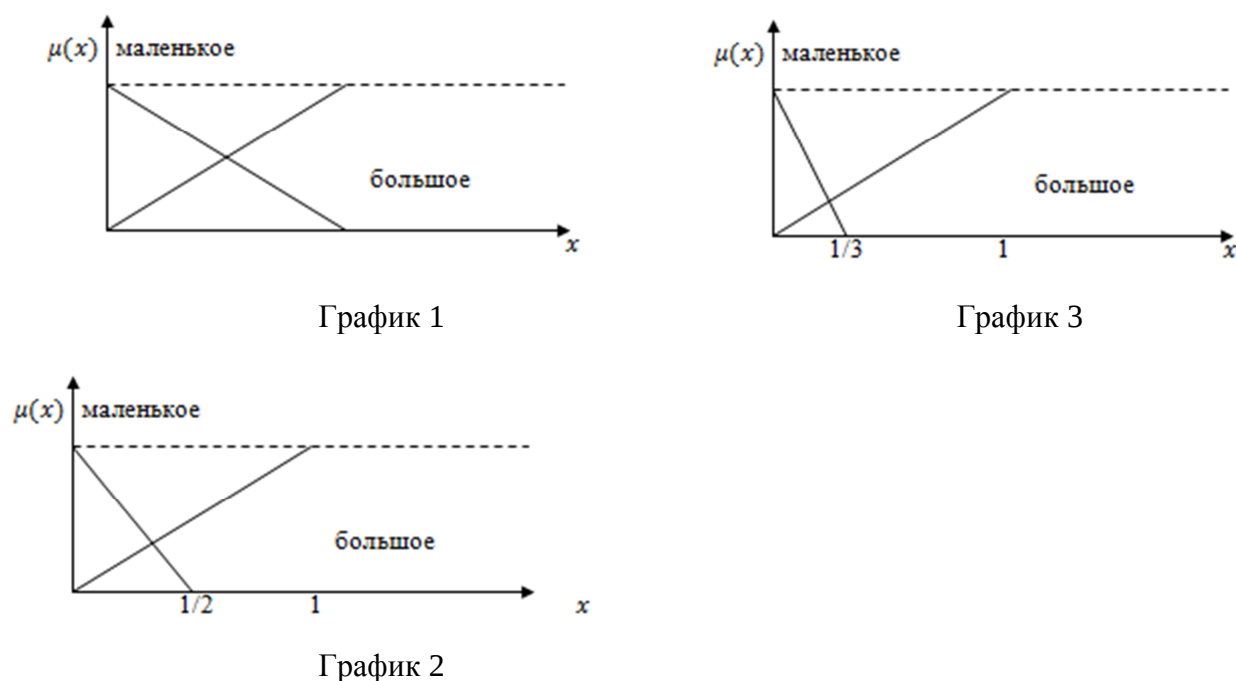


Рис. 5. Используемые термы

В таблице 2 показана предельная точность различных алгоритмов и их зависимость от выходных термов.

Таблица 2

Предельная точность от выходных термов

	Mamdani	ANFIS	NEFCLASS
Граф ик 1	0.01534	0.000047	0.0184
	0.00311	0.000008	0.0082
	0.00052	0.000007	0.0001
Граф ик 2 (1/2)	0.02232	0.000091	0.01212
	0.00523	0.000016	0.00223
	0.00009	0.000013	0.00003

Граф ик 3 (1/3)	0.01132	0.000143	0.01423
	0.00082	0.000025	0.00241
	0.000007	0.000012	0.00012

Первое число - ΔR , второе - ΔL , третье - $\Delta \varphi$.

Как видно из приведенных таблиц, наилучшие результаты показала нейронная сеть ANFIS, чем больше термов в нечеткой логике сети, тем меньше совершалось действий. Точность результатов у данной сети выше.

Список литературы

1. Юревич Е.И. Интеллектуальные роботы. М.: Наука, 2007. 360с.
2. Щебек. Ф. Г. Вариации на тему одной планеты. Будапешт: Корвина, 1972. 272с.
3. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. Режим доступа: <http://matlab.exponenta.ru/fuzzylogic/book1/index.php> (дата обращения 30.03.2015).
4. Синюк В. Г. Гибкие нейро-нечеткие системы вывода и программная реализация для решения задач аппроксимации. М.: Молодой ученый, 2014. 967с.
5. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия - Телеком, 2004. 452 с.
6. Лю Б. Теория и практика неопределенного программирования. М.: БИНОМ. Лаборатория знаний, 2013. 416 с.
7. Rutkowski L. Flexible neuro-fuzzy systems. Boston: Kluwer Academic Publishers, 2004. 279 p.