**Лабораторная работа №7**

**Экспертные системы на основе нечетной логики.**

**Цель работы:** Изучить основные алгоритмы реализации экспертной системы с использованием метода нечетких множеств.

**Постановка задачи**

Построить экспертную(советующую) систему на основе нечеткой логики в выбранной предметной области.

Для этого необходимо: В режиме формирования

1. Сформулировать базу правил (не менее 3) антецеденты которых, включают не менее двух входных лингвистистических переменных со связками “И” , “ИЛИ”.
2. Построить структуру системы, представив её в виде иерархии, опираясь на теорему о декомпозиции.
3. Определить компоненты всех лингвистических переменных.
4. Формализовать правила на основе выбранной логики.

В режиме применения.

Для выбранных четких входных значений последовательно провести преобразование информации с уровня фазификации до последнего уровня т.е. до проведения процедуры дефазификации. Все вычисления запротоколировать.

**Контрольные вопросы**

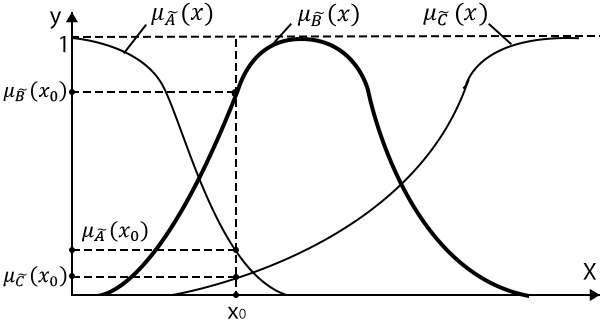
1. Задача фазификации и дефазификации.
2. Формализация нечетких продукций для n входов.
3. Проблема сложности вычисления КПВ и теорема о декомпозиции.
4. Обобщенное правило «modus ponens» для n входов.
5. Особенности нечетких моделей управления.
6. Структура АСУТП с использованием систем вычислительного интеллекта.
7. Парадигма вычислительного интеллекта.

**Теоретические сведения**

**Задача фазификации**

Фазификацию еще называют введением нечеткости. В задачах управления чаще всего применяется фазификация типа синглетона.

Обозначим четкое значение поступающее от датчиков, либо от пользователя. Это значение можно представить в виде функции принадлежности:



Термы лингвистической переменной, которые формируются нечеткими множествами

1 вариант. На выходе получаем синглетон. Он определяется как максимальная степень принадлежности, которые получаются при пересечении со всеми функциями принадлежности лингвистической переменной

*,* где

;

;

2 вариант. На выходе – функция принадлежности того терма, который имеет максимальную степень принадлежности (например ), т.е. осуществляется переход от непрерывной числовой переменной к дискретной шкале термов.

**Задача дефазифиикации**.

Рассмотрим следующий пример. Советующая системе управления запасами сформировала решение «Закупить небольшое количество деталей». Нечеткое понятие «Небольшое количество» задано дискретным нечетким числом

<НЕБОЛЬШОЕ КОЛИЧЕСТВО, N, >, где

Поскольку закупить можно лишь четкое целое число деталей, необходимо произвести выбор четкого значения лингвистической переменной «Кол. ДЕТАЛЕЙ» при наличии ее нечеткого значения НЕБОЛЬШОЕ.

Пусть дано некоторое понятие естественного языка и формулизующая его нечеткая переменная <>. Построить процедуру выбора конкретного элемента по исходным значениям

Имеется несколько методов:

* + - 1. Метод весов. Выбор осуществляется согласно соотношения: Не учитывает форму и используется для унимодальных функций (т. е. выбирает тот икс, которому соответствует максимальное значение)
      2. Метод центра площади. Выбор такого элемента , который делит площадь под графиком пополам, т.е. (тут уже честный интеграл, без всяких там):
      3. Метод центра тяжести, т.е.:

*,* где S = support( или в дискретном варианте:

**Формализация нечетких продукций для n-входов**

Рассмотрим эту задачу для 2-ух входов, а затем обобщим для n-входов.

Если есть и есть , то есть , где , , – лингвистические переменные, , , – термы этих ЛП, семантика которых характеризуется нечеткими переменными

Рассмотрим анцедент <есть и есть > ( и рассматриваются как значение векторной лингвистической переменой

Т.е. () есть ( и )

не может быть характеристикой этого значения, т.к. операция определяется на одном базовом множестве.

Для того, чтобы определить нечеткое множество которое соответствует векторному терму ( и ) выполним следующе:

* Каждое множество задано на X и Y и порождают на XY нечеткие множества, которые называются цилиндрическими продолжением (цилиндрические множества)

Теперь базовое множество одно и тоже и векторный терм можно охарактеризовать (согласно свойству цилиндрического множества):

Эти правила преобразования справедливы при условии не взаимодействия переменных X и Y, т.е. их элементы не связаны функционально.

Теперь по аналогии с системой с одним входом:

Если есть и есть

Получим определение для векторной лингвистической переменой по аналогии:

Если () есть ( и ) то есть

Т.е размерность матрицы нечеткой отношений возрастает (уже не бинарное а тернарное.

Для случая с n входами, т.е.:

Если , то .

Таким образом при формализации получаем (n+1)-арное нечеткое отношение, которое характеризуется функцией принадлежности , т.е. и и … и

Сложность этой функции O( (т. е.. экспоненциальный рост)

**Теорема о декомпозиции**

Если функция принадлежности бинарного отношения , полученное при формализации импликации , с использованием той или иной логической системы не возрастает по аргументу , то функция принадлежности (n+1) одного нечетного отношения, которая характеризует импликацию

Этим условием удовлетворяет из рассмотренной логики Лукашевича логики

В том случае, когда импликация имеет вид:

Если или или … или то

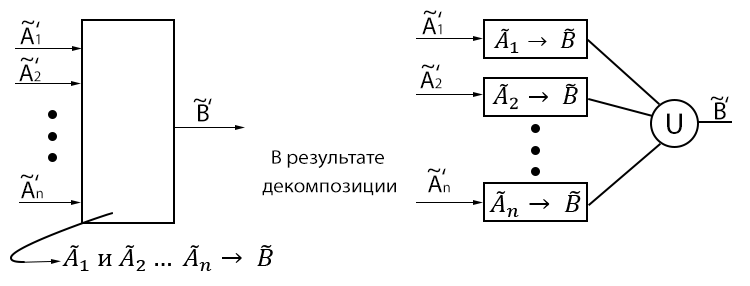
**Обобщенное правило modus ponens для n-входов**

**Определение.** Пусть нечеткие унарные отношения , а также (n+1)-арное нечеткое отношение представляют собой ограничения на и () соответственно и являются нечеткими множествами на базовых множествах и .

Пусть заданы нечеткие множества на базовых множествах и (n+1)-арное отношение, которое формализует импликацию . При этом функция принадлежности бинарного отношения соответствующая импликации удовлетворяет теореме(1). Тогда композиционное правило утверждает, что решением системы уравнений назначений:

Предпосылки

Вывод относительно y имеет вид:

Структурно это можно показать следующим образом:

Вычислительная сложность (полиномиальная; странно, что тут не |X|^2)

Совокупность импликации для выбранного выходного параметра представляются в виде:

: Если

: Если

: Если

m-количество термов выходной лингвистической переменной.

Тогда общий результат по данному выходу определяется:

Пример. Построить экспертная (советующая) системы: сколько дать на «чай» официанту за обслуживание в ресторане.

Основываясь на устоявшихся обычаях и интуитивных представлениях принимаем, что база правил может быть описана следующими предпочтениями.

Если обслуживание плохое() или еда подгоревшая(), то чаевые-малые ()

Если обслуживание хорошее(), то чаевые средние ()

Если обслуживание отличное() и еда превосходная(), то чаевые щедрые ()

<чаевые, T, [5..25]>

T={“малые()”, ”средние ()”, ”щедрые()”}

<обслуживание, T, [0..10]>

T={“плохое() ”, ”хорошее()”, ”отличное() ”}

<еда, T, [0..10]>

T={“подгорелая”, ”превосходная”}

0 – наихудшие оценки

10 – наилучшие оценки

Плохие чаевые: около 0% от стоимости заказанных блюд

Средние чаевые: около 15%

Щедрые чаевые: около 25%

А1->B1

A4->B1

F\_k

A3->B1

A5->B2

F\_e

A2->B2

**На фото были уровни:**

1. Фаззификация — введение нечёткости
2. Активизация — процедура нахождения функций принадлежности каждого из подусловий, аналогично обобщённому modus ponens
3. Активизация для каждого правила
4. Аккумуляция — процедура нахождения функции принадлежности для выходной лингвистической переменной
5. Дефуззификация — приведение к чёткости

**Активизация** – в системе n-вывода представляет собой процедуру определение реакции по каждому входу (на основе FMP)

**Нечеткие модели управления. Обоснование использования**

Условно можно выделить два класса объектов, с которыми приходиться сталкиваться специалистам в области автоматизации управления. Их называют «простые» и «сложные».

«Простыми» являются объекты точные математические модели которых (например в виде системы алгебраических уравнений или модели линейного программирования) при учете всех необходимых количественных факторов, влияющих на поведение объекта, пригодны для реализации на компьютере и адекватны объекты.

«Сложные» объекты управления имеют следующие отличительные особенности:

1. Не все цели выбора управляющих решений и условия, влияющие на этот выбор, могут быть выражены в виде количественных соотношений.
2. Отсутствует либо является неприемлемо сложным формализованное описание.

Создание математической модели для реального объекта может потребовать годы работы и большинство математических моделей требуют сильных упрощений и линеаризации.

Построение точных математических моделей сложных объектов, пригодных для реализации и эксплуатации на современных компьютерах, либо затруднительно, либо вообще невозможно, Здесь специалист оказывается перед необходимостью выбора одной из альтернатив. Первая альтернатива – при построении модели сложного объекта, как и в случае «простого» объекта постараться учесть все возможные факторы, влияющие на поведение объекта. К сожалению, это попытка «объять необъятное» если и можно построить такую модель, используя традиционные математические методы, то она получается чрезвычайно громоздкой и неприемлемой для практического использования, т.к. время реакции разрабатываемой системы на изменение ситуации на объекте оказывается недопустимо большим. Вторая альтернатива – абстрагирование от некоторых параметров объекта в целях получения модели более простой и удобной для реализации, но в этом случаем модель получается неадекватной. В общем случае возможных поправлений поиска может быть два.

Первое – попытаться применить нетрадиционный математический аппарат для построения модели, учитывающей все особенности объекта и пригодной для реализации.

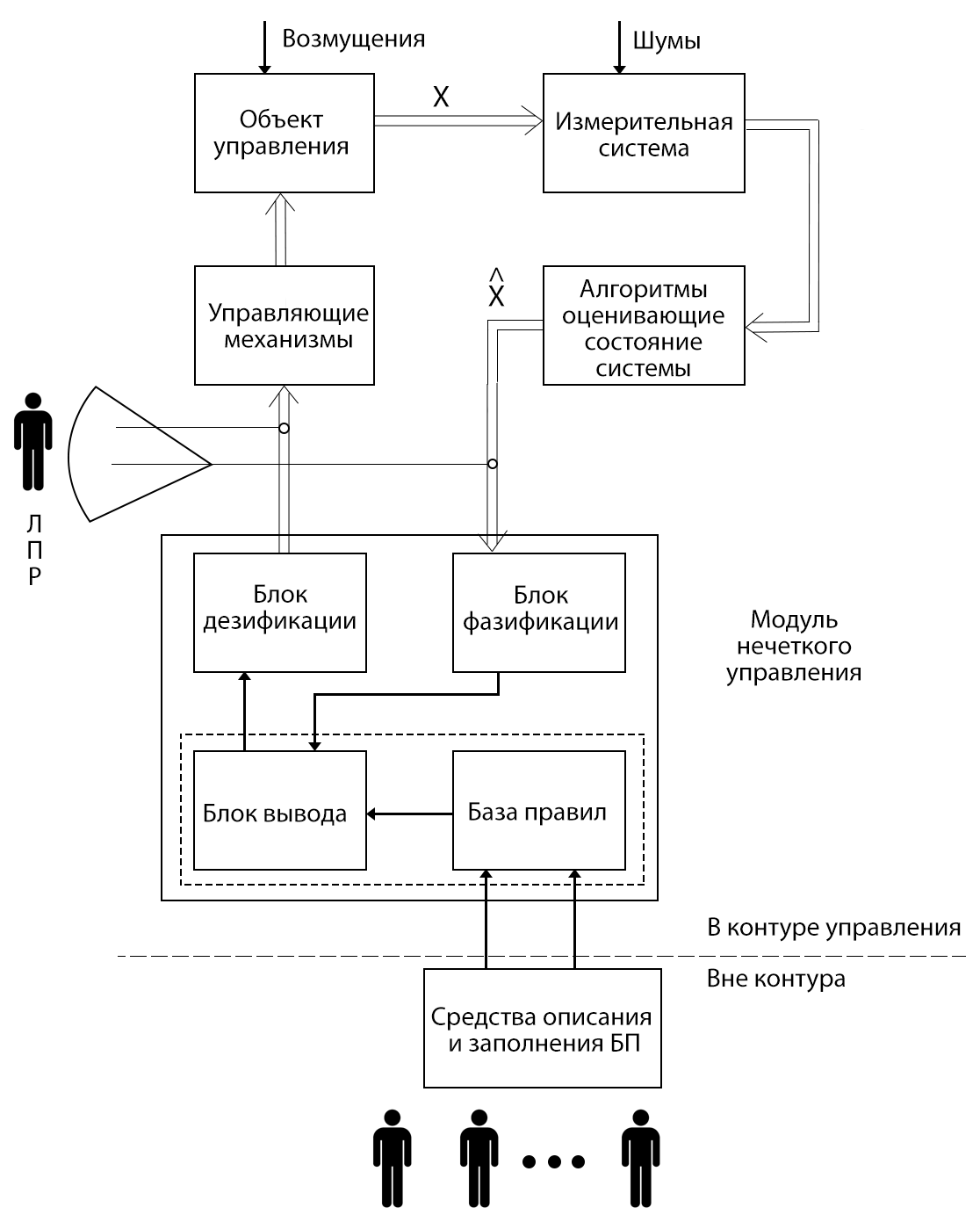
Второе направление состоит в попытке построения не модели объекта, а модели управления объектом. Иными словами, моделирование осуществляется не самого объекта, а человека-оператора в процессе управления объектом.

Естественно, что моделировать стоит только квалифицированного оператора, хорошо знакомого со всеми особенностями управления данным объектом и успешно случавшегося с управлением им «вручную».

Если есть опытный оператор, то модель уравнения объекта уже создана. Она существует либо в виде набора инструкции по управлению, либо в памяти оператора. Остается только эту модель представить в форме удобной для реализации для компьютера. В процессе своей управленческой деятельности человек получает и осознает огромное количество информации. Однако ограниченные возможности человеческого мозга заставляет его осуществлять вербальное перекодирование исходной информации, используя при этом уникальные возможности человеческого языка. Едва ли не все рассуждения человека по своей природе являются приближенными, нечеткими и осуществляются на уровне качественных описаний.

Оператор использует качественные нечеткие оценки типа «много», «мало», «довольно высокий», «далеко», «очень близко», «быстро».

**Структура АСУТП с использованием систем вычислительного интеллекта (СВИ)**

Вместо блоков связанных с блоками идентификации и оптимизации (как это в класс АСУТП) появились новые блоки. База правил правило выбора это модуль нечетного управления. Чем выше уровень АСУ, тем в большей степени необходимо использовать СВИ. Среднее время АСУ программное обеспечение должно разрабатываться в сочетании «жестких» и «нечетких» алгоритмов. Нижний уровень АСУ т.е. на уровнях допускающих строгую формализацию процессов структура ПО на основе четких «жестких» алгоритмов, реализованных в традиционных системах.

**Нечеткий регулятор. Однофакторное уравнение**

Цели контроллера – свести ошибку к нулю.

Имеем ЗЛП

<”отклонение”,T,E,G,M>

<”Скорость изменения отклонения”,T,E,G,M>

<”Управление”,T,U,G,M>

T = {NB, NM, NS, Z, PS, PM, PB}

Где Z – около нуля (нулевое отклонение)

NB – отрицательно большое; NM – отрицательно среднее; NS – отрицательно малое; PS – положительное малое; PM – положительное среднее; PB – положительное большое.

Выходные переменные OY X(T) сравниваются с разным значением y и ошибка согласования поступает в масштабный блок 1 с координатой , а также на дифференциатор 3. Элементы 2 и 5 – фазификаторы преобразующие текущее значение отклонение и его производную в нечеткие множества, которые поступают в главный элемент Н.Р. Основу блока составляет лингвистические правила.

I : Если отклонение = PB и «скорость» = PB, то «управление» = PM.

Правила могут быть заданы в виде таблицы.

Каждое правило формализуется в н.о.

Воспользуемся определением Мондели

Обозначим

Тернарное отношение и есть закон управления.

Далее определяется по текущим значениям и управляющее значение с помощью композиционного правила

После чего применяется фазификация и получаем четкое значение величины управления И. (блок 7)

С помощью этого результат используется лучшие переходные характеристики.

Дает улучшение производительности, где обычные контроллеры не справляются с нелинейностями управляемого процесса.

**Проектирование систем основанной на нечеткой логики**

Л. Заде создавал нечеткую логику как математическую дисциплину (первая статья появилась в 1965 г.)

Основы нового подхода к анализу сложных систем и процессов принятия решений (математика сегодня М. Знание 1974г.)

Первыми, кто применил нечеткую логику при разработке реальных систем были:

- В области управления Э. Мондана (1974 г. Англия)

- В области принятия решения задач бизнеса Г. Зиннерман (1920 г. ФРГ)

В настоящий момент нечетка логика стала часто используемой технологией и как и методология разработки разбилась в большое количество. Сама методология стала частью расширенного стандарта Международной Электротехнической Комиссии (IEC 1137-3), обеспечивающего разработчиков документацией на протяжении всех этапов разработки проекта.

Одной из самых популярных в настоящее время CAD – система является fuzzyTECH, которая может быть использована как для разработки нечеткой системы управления, а также для систем поддержки принятия решений многочисленных задач в бизнесе.

Дистрибутив программы в форме самораспаковывающегося архивного файла занимает 16 Мгбайт. Основными ограничениями демонстрации версии программы fuzzeTECH является отсутствие возможности сохранения проекта и генерации программного кода на том или ином языке программирования.

Использование нечеткой логики для управления парогенератором. Решение этой задачи обычными методами было сопряжено с рядом трудностей вычислительного характера. Предложенный Э. Мондана алгоритм, основанный на нечеткой логики вывода, позволил избежать чрезмерно большого объема вычислений. В этот же период нечеткие модели были применены при управлении …. Для обжига цемента.

Япония – лидер в области промышленных применений, запатентовано более чем 30000.

Преимущество:

* Разработка быстрого прототипа (с дальнейшими усложнением)
* Нечеткая логика более проста для понимания, чем математические модели на основе дифференциальных уравнений.
* Более просты для своей ….. реализации.

Датчики определяют цвет и вид …, степень физификации. Нечеткий контролер выбирает наиболее подходящую программу стирки из 600 доступных комбинаций территории выбора, количества стирки из России и времени процедуры, цикла быстрого или медленного времени и промывании.

Процесс разработки проекта нечеткой системы управления на fuzzzyTECH разбивается на четыре основных этапа

**Первый Этап. Описание системы**

На этом этапе при помощи средств, доступных в fuzzyTECH, задача формализуется. Процесс создания систем полностью визуализирован. Здесь необходимо описать ЛП, их функции, принадлежности; описать стратегию (концепцию) управления посредством нечетких правил, которые можно объединить в единую базу правил или знаний о системе.

В целой CASE – технология по ошибке, которой построен пакет, позволяет все эти действия выполнить только посредством общения эти действия выполнить только посредством общения с экраном компьютера, не заглядывая в программный код. Поэтому начальный этап проектирования воспринимается с легкостью, несмотря на кажущуюся сложность. Можно установить разрядность машинного кода, генерируемого пакетом

8 – битов целочисл.

16 – битов целочисл.

32 – битов целочисл.

Пример.Контроллер для стиральной машины

: Если <количество большое> = <много> и <температура воды> = <высокая>, то <длительность> = <высокая>

: Если <количество белья> = <много> и <температура воды> = <высокая> и <загрязненность> = <низкая>, то <длительность> = <низкая>

: Если <количество белья> = <мало> и <температура воды> = <низкая> и < загрязненность > = <низкая>, то <длительность> = <низкая>

**Второй этап. Offline – оптимизация**

На этом этапе следует проверить работоспособность созданной системы, посредством всех средств fuzzyTECH. Можно использовать заранее созданный программный симулятор объекта управления. Для связи системы управления с моделью, используется специально разработанный протокол связи fTlink, в основу которого положена концепция обмена сообщениями Windows. Все необходимые средства для установления связи с моделью находятся в исходных текстах программ связи, поставленных с пакетом.

**Третий этап. Online – оптимизация**

На этом шаге разрабатываемая система управления и реальный объект управления соединяются физической линией связи. Такой вид отладки линией связи позволяет наблюдать поведение системы в реальных условиях и при необходимости вносить изменения в систему управления.

**Четвертый этап. Реализация**

На этом этапе необходимо получить окончательный вариант для конкретного микроконтроллера и если нужно, связать его с основной программной.

Основу программного кода генерируемого пакетной fuzzyTECH составляет аппаратного ориентированный на конкретный тип ядра процессора. Например, поставляемое с пакетом fuzzyTECH программного ядра МСИ -96 совместим, но с такими контроллерами как: 8096BH, 8096 – 90, 80196KD, 80196MC. Об оптимальности создаваемого fuzzyTECH кода можно судить по данным таблицы.

Структура генерируемого кода состоит как правило из частей:

* Код библиотечных функций
* Сегмент базы проверен и функциональные принадлежности и функции нечеткой системы

В качестве примера бизнес – приложений это, например систем поддержки ПР по оценки банком кредитоспособности клиента (т.е. оценка диска банковского кредита).

Второй пример. Нечеткая СППР, которая определяет начальную цену аукциона. Слишком высокая отправная цена может препятствовать процессу предположения цены, но слишком низкая цена может принести убытки продавцом.

Входных переменных 30.

**Парадигма вычислительного интеллекта**

Все традиционные системы ИИ реализованы на технологии Hard Computing, т.е. с использованием компьютерной технологии, определяющее программу как жесткую последовательность операторов, не подвергающейся модификациям под действием процедуры обучения направленной на достижение необходимого функционирования. Также традиционный искусственный интеллект не приемлет численные методы (так важные для учета неопределенности и неточности), т.к. они были ориентированы на манипулирование символами и логике предикатов. По указанными выше обстоятельствам уровень машинного интеллекта MIQ традиционных систем ИИ оказались низкими. Поэтому стояла задача повышения MIQ ИС. При этом ключевой методологией выступает Soft Computing (SC) предусматривающая совместное (не автономное) использование таких численных новых подходов как нечеткая логика (FL), нейровычисление (NC), генетические вычисление (GC) позволяющая решать многие важных проблем реального мира, решения которых традиционными методами ИИ не возможно.

Каждая из составляющих методологии имеет много возможностей для ее использования в рамках мягких вычислений.

Нечеткая логика лежит в основе методов работы с неточность, зернистой структурой (гранулирование) информации, приближенных рассуждений и вычислений со словами.

Нейровычисления отражают способность к обучению и адаптации.

В случае генетических речь идет о возможности настройки (FL) и достигает оптимального значения характеристик, например параметров функций принадлежности.

Не пройдет и 10 лет ,как отметил Л. Заде , что ИИ будет основываться на SoftComputing, а не на традиционном ИС программе развития ИИ в XXI.

Лучшей моделью SoftComputing по его словам является человеческий мозг.

Для того, чтобы подчеркнуть значение вычислительного интеллекта, выделим специфичные задачи:

* Разумный робот, способный выполнить домашние дела или вести автомобиль по городским улицам с интенсивным движением.
* Переводчик с одного естественного языка на другой сохраняющие смысл текста и речи.
* Обработчик текстов, выполняющегося качественные реферирования или аннотирование текстов на естественном языке.
* Робот, распознающий образы сложных пространственных объектов, в том числе по плоским зашумленным, искаженным образам.
* Поиск в мультимедийных базах данных.
* Распознавание рукописей, понимание образов.

**Пример выполнения лабораторной работы**

**Лабораторная работа № 7**

*Экспертные системы на основе нечетной логики*

|  |  |  |
| --- | --- | --- |
| *Выполнил:* | *ст. гр.* | *ФИО* |
| *Принял:* | *проф.* | *Синюк В.Г.* |

**Цель работы:** Изучить основные алгоритмы реализации экспертной системы с использованием метода нечетких множеств.

**Постановка задачи**

Построить экспертную(советующую) систему на основе нечеткой логики в выбранной предметной области.

Для этого необходимо: В режиме формирования

1. Сформулировать базу правил (не менее 3) антецеденты которых, включают не менее двух входных лингвистистических переменных со связками “И” , “ИЛИ”.
2. Построить структуру системы, представив её в виде иерархии, опираясь на теорему о декомпозиции.
3. Определить компоненты всех лингвистических переменных.
4. Формализовать правила на основе выбранной логики.

В режиме применения.

Для выбранных четких входных значений последовательно провести преобразование информации с уровня фазификации до последнего уровня т.е. до проведения процедуры дефазификации. Все вычисления запротоколировать.

**Пример выполнения:**

*В режиме формирования:*

Управление выдержкой в процессе фотографирования в зависимости от положения диафрагмы и освещенности сцены с целью получения оптимальной экспозиции снимка.

Правила:

1. Если уровень освещенности – низкий и диафрагма – закрыта, то выдержка - длинная

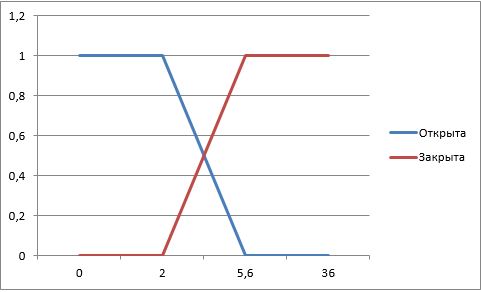
2. Если уровень освещенности – средний и диафрагма - закрыта, то выдержка – средняя

3. Если уровень освещенности – высокий и диафрагма открыта, то выдержка – короткая

**Положение диафрагмы:**

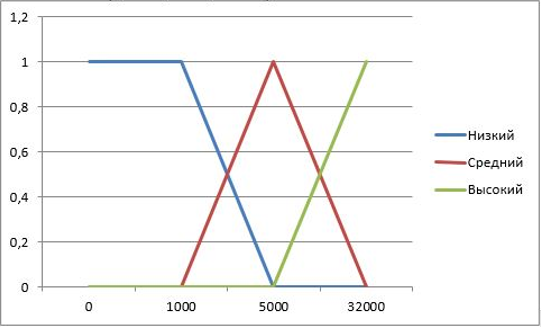
Т1 = {“ОТКРЫТА”(A1) , “ЗАКРЫТА”(A2)}

Базовое множество X1 = {0, 2, 5.6, 36}



**Уровень освещенности:**

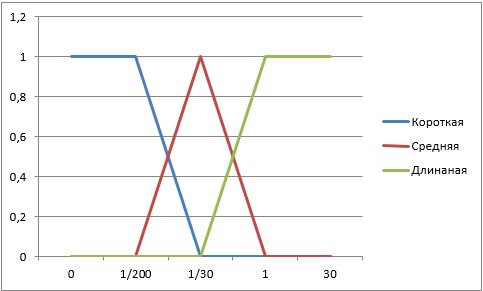
Т2 = {“НИЗКИЙ”(A3) , “СРЕДНИЙ”(A4), “ВЫСОКИЙ ”(A5)}

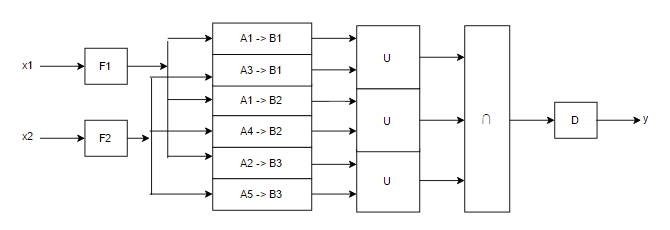
Базовое множество X1 = {0, 1000, 5000, 32000}

**Выдержка**:

Т3 = {“КОРОТКАЯ”(B1) , “СРЕДНЯЯ”(B2), “ДЛИННАЯ”(B3)}

Базовое множество X1 = {0, 1/200, 1/30, 1, 30}



Структура разработанной базы знаний:

Формализация правил:

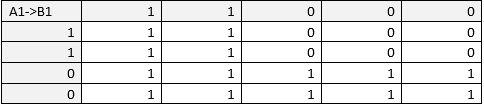
D1: A1 и A3 -> B1

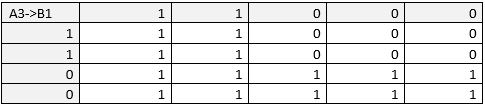
D2: A1 и A4 -> B2

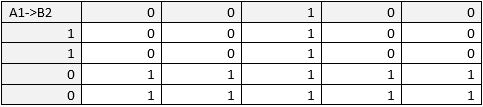
D3: A2 и A5 -> B3

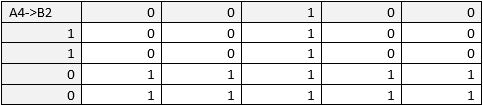
Вывод:

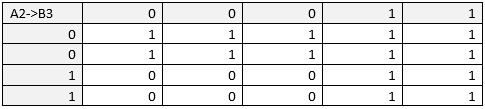
Формализуем импликацию по Лукасевичу:

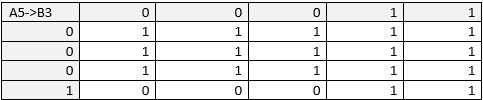








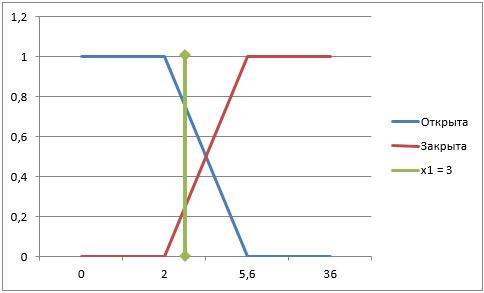




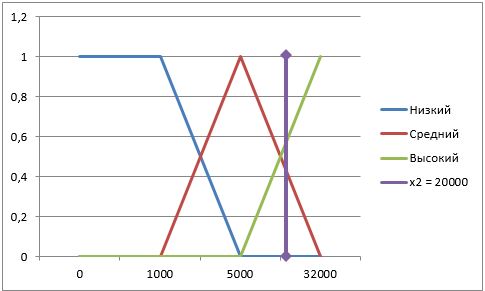
*В режиме применения:*

Входные данные: x1 = 3, x2 = 20000

x̃1={A1, A2} ={ 0.722, 0.277 }

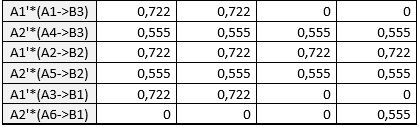


x̃2={A3, A4, A5} ={ 0, 0.444, 0.555}

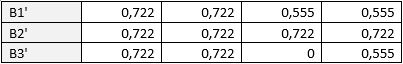


D:\Институт\3 курс\2 семестр\Практика\Лаб7\часть 2\12.JPG

Производим композицию:



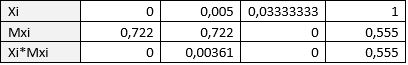
Активируем:



Агрегируем:

D:\Институт\3 курс\2 семестр\Практика\Лаб7\часть 2\15.JPG

Этап дефазификации:



Результат: 0,27944472

Таким образом, система советует выбрать выдержку ~1/4 секунды, что соответствует терму «средняя выдержка».

Вывод: в ходе выполнения данной лабораторной работы был изучен алгоритм реализации экспертных систем с использованием метода нечеткой логики. Также была реализована нечеткая продукционная система определения выдержки снимка в зависимости от положения диафрагмы и уровня освещенности сцены.