

ЛАБОРАТОРНАЯ РАБОТА № 19. ИЗУЧЕНИЕ ОСНОВНЫХ ФУНКЦИЙ ПРОГРАММНОГО ПРОДУКТА FUZZYTECH

Цель работы: изучить функциональные возможности программного пакета *fuzzyTech*; освоить графический редактор проекта и графических редакторов правил системы нечёткого логического вывода.

Краткие теоретические сведения

Программа *fuzzyTECH* предназначена для решения различных задач нечёткого моделирования и разработки приложений на базе нечёткой логики, разработанная и постоянно обновляемая компанией INFORM GmbH (Inform Software Corporation, Германия). Основными шагами разработки программ на *fuzzyTECH* являются:

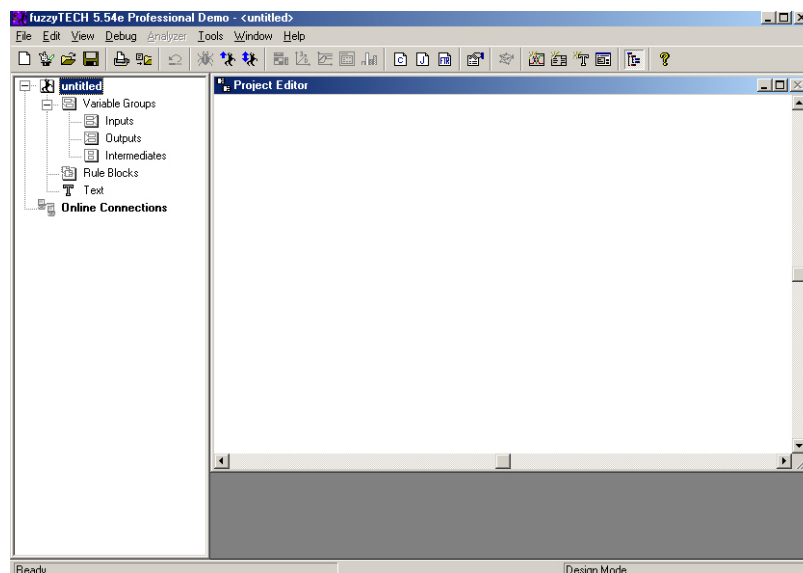
- 1) формализация поставленной задачи - определение входных и выходных лингвистических переменных, сопоставление термов с конкретными физическими значениями;
- 2) определение логических операций на основе *t*-нормальных функций;
- 3) задание функций принадлежности для каждого терма;
- 4) разработка базы правил;
- 5) задание метода дефаззификации выходных данных;
- 6) анализ и отладка.

Каждая система нечеткого вывода задается в форме проекта (*Project*). Все операции по созданию, редактированию, отладке и анализу проектов выполняются в интерактивном режиме. Проекты сохраняются в отдельных файлах формата *FTL* (*Fuzzy Technology Language*) с расширением *ftl*. Указанные файлы представляют собой текстовые файлы, в которых хранится система нечеткого вывода в форме структурированного текста.

Процесс нечеткого моделирования в среде *fuzzyTech* осуществляется в интерактивном режиме с помощью специальных графических средств, предназначенных для редактирования и визуализации компонентов системы нечеткого вывода.

Для редактирования и анализа систем нечёткого вывода используются следующие средства:

- графический редактор проекта системы нечёткого вывода (*Project Editor*, рис. 1);
- графический редактор лингвистических переменных и функций принадлежности их термов (*Variable Editor*);
- графические редакторы правил системы нечёткого вывода (*Spreadsheet Rule Editor*, *Matrix Rule Editor*);
- графические средства анализа результатов нечёткого вывода (*Watch Window*, *Rule Analyzer*);
- графические средства просмотра поверхности системы нечёткого вывода (*Transfer Plot Window*, *3D Plot Window*, *Time Plot Window*).

Рисунок 1 – Главное окно программы *fuzzyTech*

В верхней части графического интерфейса программы fuzzyTECH расположена строка заголовка, в которой указывается название и версия программы, а также имя разрабатываемого или загруженного проекта. В правой части строки заголовка находятся стандартные кнопки управления графическим окном программы. Под строкой заголовка находится строка главного меню программы fuzzyTECH и строка панели инструментов. Вид строки главного меню программы fuzzyTECH и строки панели инструментов показан на рис. 2.



Рисунок 2 – Вид строки главного меню программы fuzzyTECH и строки панели инструментов

Панель инструментов содержит кнопки, позволяющие осуществить доступ к наиболее часто используемым командам и операциям.

Редактор проекта позволяет визуализировать структуру всего проекта и графически представить отношения между компонентами проекта. Двойное нажатие на изображении того или иного прямоугольника позволяет открыть окно редактирования соответствующего компонента системы нечеткого вывода. Слева от окна редактора проекта расположено окно просмотра структуры проекта (*Treeview*). Указанное окно содержит перечень всех компонентов проекта. Нажатие на изображение символа «+» позволяет раскрыть соответствующую вложенную структуру группы. Двойное нажатие на имени выбранного компонента проекта позволяет открыть окно редактирования свойств данного компонента системы нечёткого вывода. Редакторы правил предназначены для редактирования правил продукций и представлены одной из следующих форм:

- табличный редактор (*Spreadsheet Rule Editor*);
- матричный редактор (*Matrix Rule Editor*).

Графическое окно табличного редактора имеет панель инструментов и секцию правил продукций. Причем правила в секции представлены в форме таблицы. Каждому правилу соответствует отдельная строка, которая состоит из двух частей: IF- и THEN-

части. В IF-части указаны имена термов входных переменных, в THEN-части – имена термов выходных переменных.

Ход работы:

1. Рассмотрим применение нечёткой логики на примере разработки нечёткого контроллера погрузки с помощью крана (рис. 3):



Рисунок 3 – Фото в процессе погрузки бетонных модулей с помощью крана

Поскольку в процессе погрузки высокогабаритные бетонные модули могут быть повреждены, необходимо свести к минимуму степень раскачивания груза краном с использованием опыта оператора (машиниста крана). Предлагается модель нечёткого PID-регулятора (Proportional-integral-differential controller) для решения задачи погрузки с помощью крана.

2. Запустить демонстрационный пример «Container crane» и нажать на кнопку «Fuzzy» (рис. 4).

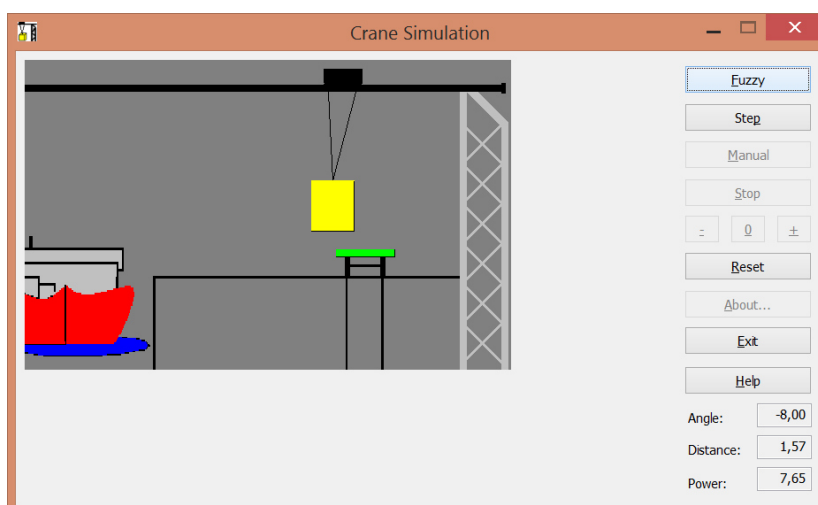


Рисунок 4 – Моделирование системы нечёткого управления контейнерным краном

Цель управления процессом погрузки с применением нечёткого контроллера заключается в том, чтобы как можно быстрее доставить контейнер (желтый) с корабля (красный) до платформы (зеленый). Когда груз достигнет цели, раскачивание должно

быть компенсировано перед погрузкой контейнера на платформу. По этим причинам большинство контейнерных кранов по-прежнему используют операторов-машинистов для контроля скорости двигателя. Оператор одновременно компенсирует влияние и обеспечивает быстрое достижение целевой позиции. Эта задача непростая, но опытный оператор способен достичь хороших результатов.

Когда головка крана перемещается по горизонтальной дорожке, контейнер начинает колебаться. Контейнер может быть опущен на платформу только, когда перестанет колебаться. Существуют два тривиальных решения этой проблемы. Один из них заключается в том, чтобы расположить головку крана точно над целевым положением (платформой) и подождать, пока колебание не снизится до приемлемого уровня. В безветренный и ясный день это так и происходит, но это занимает слишком много времени. Данное решение не оптимально, поскольку контейнерное судно должно быть загружено и выгружено за минимальное время.

Альтернативой является перемещение контейнера настолько медленно, что не происходит никакого колебания. Опять же, эта техника работает в безветренный и ясный день и занимает слишком много времени. Другое решение заключается в создании контейнерных кранов, где дополнительные кабели фиксируют положение контейнера во время работы, но эта альтернатива очень дорога.

На рис. 5 представлена структурная схема управления контейнерным краном с помощью нечёткого контроллера.

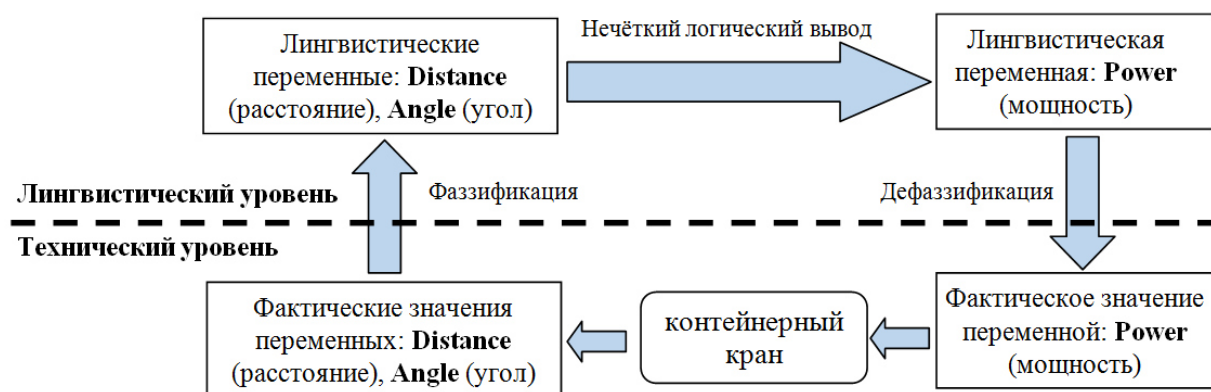


Рисунок 5 – Управление контейнерным краном с помощью нечёткого контроллера

Все сигналы с датчиков преобразуются в лингвистические переменные. Например, измеренное расстояние равное 12 ярдам преобразуется в лингвистическую переменную **«still medium, just slightly far»** («ещё среднее, но немного далеко»). Этот этап называется фаззификацией – преобразованием фактического (измеренного) значения переменной в лингвистическое значение переменной. На этапе нечёткого логического вывода все лингвистические переменные используются в качестве термов в продукционных нечётких правилах (IF, THEN), которые определяют поведение системы. На этапе дефаззификации происходит преобразование лингвистической оценки в фактическое значение переменной **Power** (мощность), которая представляет собой настройку мощности двигателя в киловаттах (кВт).

3. После нажатия на кнопку «Stop» открыть редактор проекта «Project Editor» для просмотра каркаса проекта «Container Crane Controller» (рис. 6)

Container Crane Controller

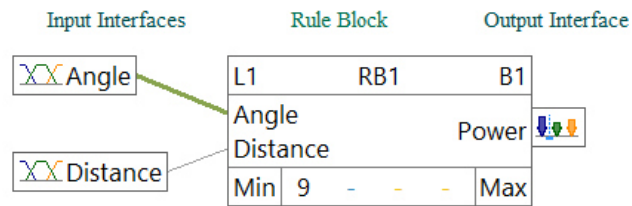


Рисунок 6 – Отображение информации о проекте «Container crane controller»

В редакторе проекта отображается структура системы нечёткого вывода в форме прямоугольников переменных и блоков правил. Редактор проекта является основным средством, которое используется для визуализации общей структуры и быстрого доступа к различным инструментам редактирования свойств систем нечёткого вывода в графическом режиме. Редактор проекта программы fuzzyTECH позволяет визуализировать структуру всего проекта и графически представить отношения между компонентами проекта.

Из рис. 6 видно, что входными переменными являются: Angle (угол) и Distance (расстояние). Двойной щелчок на изображении входной переменной позволяет открыть окно редактирования свойств соответствующего компонента системы нечёткого вывода, что показано на рис. 7-9.

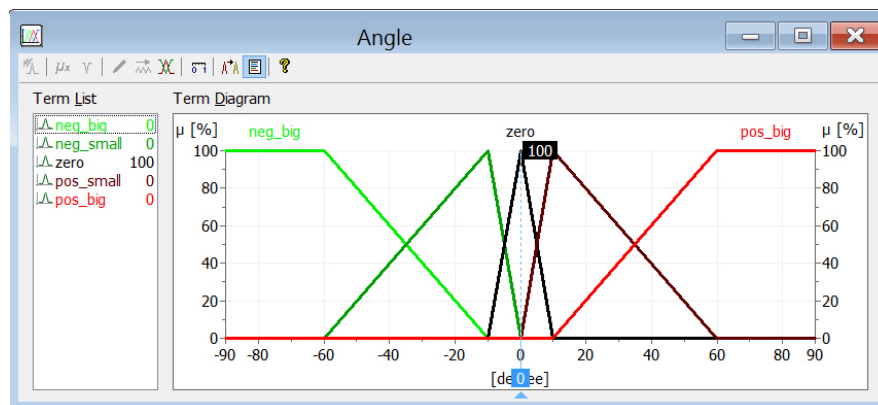


Рисунок 7 – Описание лингвистической переменной «Angle» (угол)

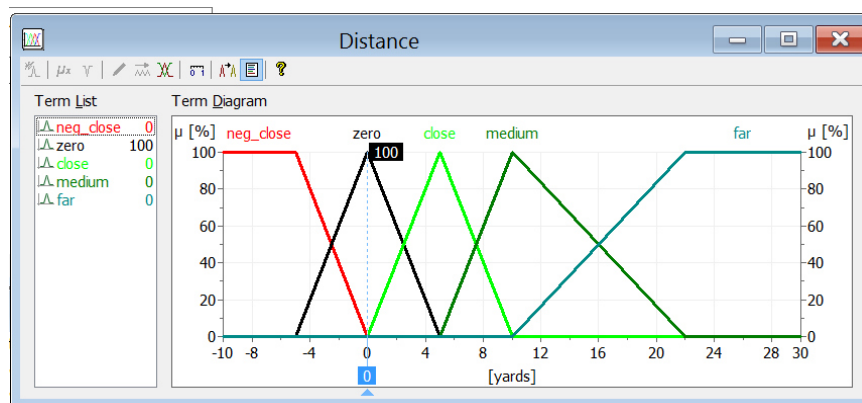


Рисунок 8 – Описание лингвистической переменной «Distance» (расстояние)

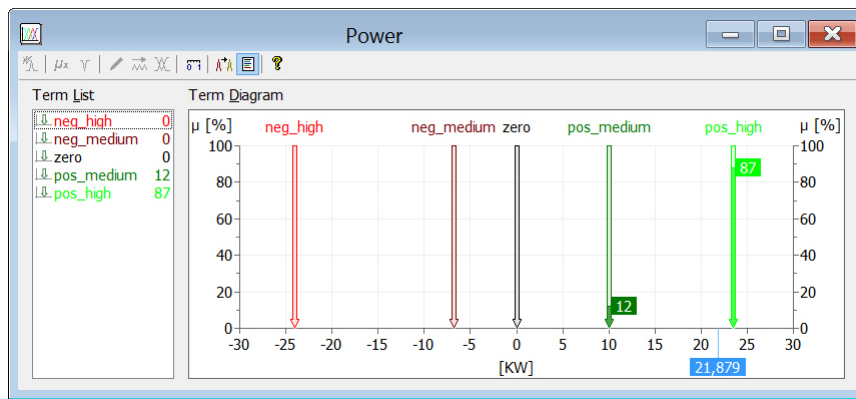


Рисунок 9 – Диаграмма переменной «Power» (мощность, сила)

Двойным нажатием на блок «Rule block» открыть окно табличного редактора правил системы нечёткого управления контейнерным краном (рис. 10):

Rule Editor 1

Rule Blocks

RB1

	Name	If	And	Operators	Then	With	Comment	Audit	GUID
►	B1	RB1	1	2	Min / Max	1			F6235
	B1.G1	*x Angle: -28	*x Distance: 2,59		Power: 10	DoS [%]			21F37
	B1.G1.R1	Angle_pos_small	Distance_zero	=>	Power_neg_medium	100			7D3C:
	B1.G1.R2	Angle_zero	Distance_zero	=>	Power_zero	100			66D9f
	B1.G1.R3	Angle_pos_small	Distance_close	=>	Power_neg_medium	100			7916A
	B1.G1.R4	Angle_zero	Distance_close	=>	Power_zero	100			0A6D
	B1.G1.R5	Angle_neg_small	Distance_close	=>	Power_pos_medium	100			E1578
	B1.G1.R6	Angle_neg_small	Distance_medium	=>	Power_pos_high	100			74914
	B1.G1.R7	Angle_neg_big	Distance_medium	=>	Power_pos_medium	100			0AD5I
	B1.G1.R8	Angle_zero	Distance_far	=>	Power_pos_medium	100			8B3C7
	B1.G1.R9	Angle_neg_small	Distance_far	=>	Power_pos_high	100			0D01f
*									

Highlight all: ☐ Match case

Рисунок 10 – Окно табличного редактора правил системы нечёткого управления контейнерным краном

Каждая строка в табличном редакторе соответствует нечёткому продукционному правилу «**IF**» (ЕСЛИ).... «**AND**» (И).... «**THEN**» (ТО) ... **WITH** (C). В столбце **WITH** значение DoS означает «Degree of Support» (степень поддержки) и содержит индивидуальный вес каждого правила.

Для выполнения анализа системы нечёткого вывода для задачи управления перейдем в интерактивный режим отладки. Для перехода в режим отладки необходимо в процессе моделирования нажать на кнопку «Stop». Внешний вид рабочего интерфейса программы fuzzyTECH в режиме отладки проекта изображен на рис. 11.

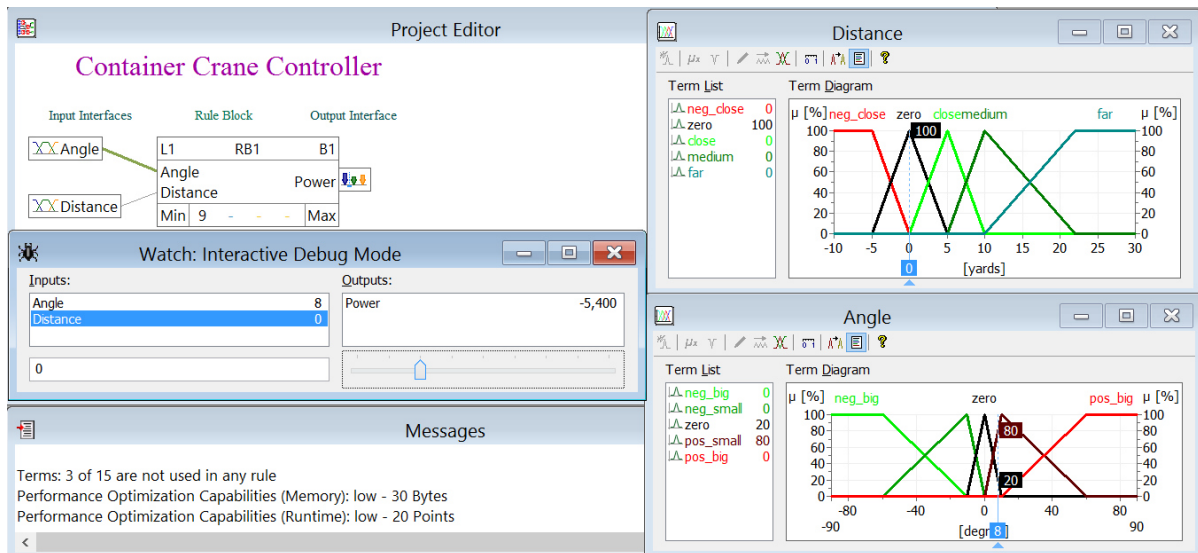


Рисунок 11 – Внешний вид рабочего интерфейса программы fuzzyTECH в режиме отладки проекта

Для общего анализа рассматриваемой нечёткой модели можно воспользоваться графическим окном просмотра поверхности нечёткого вывода на плоскости и графическим окном просмотра трёхмерной поверхности нечёткого вывода. Для получения поверхности необходимо правой кнопкой выделить «3D Plots» и выбрать «New 3D Plot...» (рис. 12):

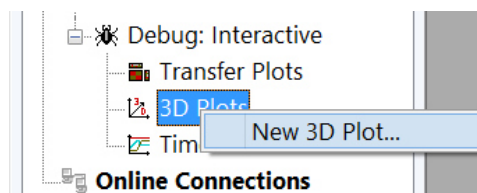


Рисунок 12 – Получение трёхмерной поверхности

На рис. 13 представлена полученная поверхность, которая сформирована тремя переменными: Angle (угол), Distance (расстояние) и Power (мощность, сила).

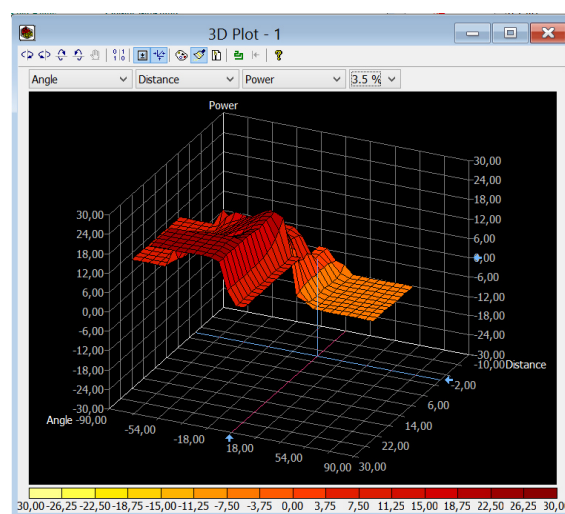


Рисунок 13 – Графическое окно просмотра трёхмерной поверхности нечёткого вывода на плоскости

Данная поверхность нечёткого вывода позволяет установить зависимость значений выходной переменной от значений входных переменных нечёткой модели систе-

мы управления краном. Эта зависимость может послужить основой для программирования контроллера или аппаратной реализации соответствующего нечёткого алгоритма управления в форме соответствующей таблицы решений.

Задание для самостоятельной работы

Выполнить отчёт по результату работы в среде fuzzyTech для следующих примеров: Container Crane, Formula 1, Steam Generation Drum, Hyperinference position control. Отчёт должен быть выполнен в редакторе Microsoft Word и содержать следующие обязательные пункты: титульный лист, моделируемую нелинейную функцию, каркас проекта, описание переменных, описание правил.

ЛАБОРАТОРНАЯ РАБОТА № 20. ФОРМИРОВАНИЕ НЕЧЁТКОЙ БАЗЫ ЗНАНИЙ В СРЕДЕ FUZZYTECH

Цель работы: разработать нечёткую продукционную базу знаний в среде *fuzzyTech*.

Ход работы:

- 1) На основании данных лабораторной работы №18 определите количество правил, число входных и выходных переменных в нечёткой системе.
- 2) Запустите *fuzzyTech*. После запуска на экране появится главное окно программы с пустым проектом. Панель инструментов содержит кнопки, позволяющие осуществить доступ к наиболее часто используемым командам и операциям. В центре главного окна расположено окно редактора проекта (*Project Editor*), в котором отображается структура системы нечеткого вывода в форме прямоугольников переменных и блоков правил.
- 3) Выберите команду «*New*» в меню «*File*». В открывшемся окне (рис. 14) задайте количество входных и выходных переменных, а также количество термов для входных и выходных переменных.

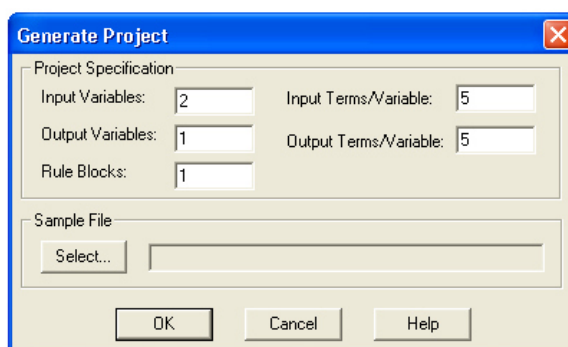


Рисунок 14 – Окно спецификации проекта

- 4) После нажатия кнопки «Ok» в окне «Project Editor» появится каркас проекта (рис. 15).

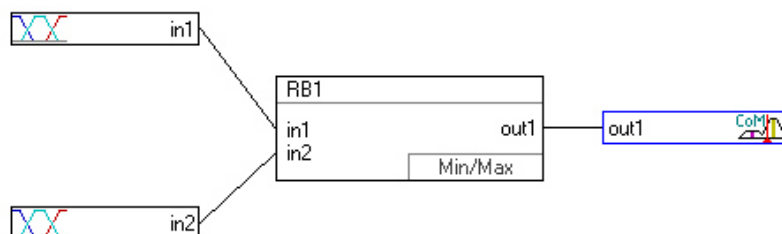


Рисунок 15 – Каркас проекта

- 5) Сохранить проект в файле (расширение *ftl* автоматически добавляется к выбранному имени файла).

3. Работа с редактором переменных

Нажав правую кнопку мыши на выбранном прямоугольнике, обозначающем переменную, вызовите меню редактора переменных (рис. 16). Переменные определены в лабораторной работе №18.

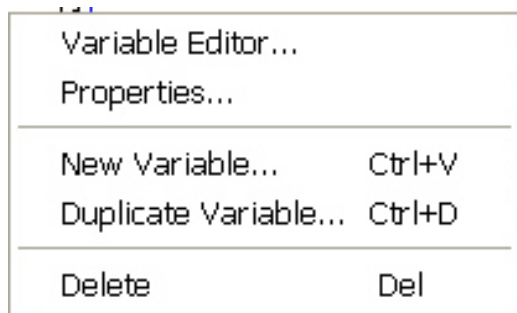


Рисунок 16 – Меню редактора переменных

Выбрав пункт *Properties*, в закладке *General* поля *Name* задайте имя переменной; в закладке *Base Variable* задайте минимальное и максимальное значение, а также значение переменной по умолчанию.

Выбрав пункт *Variable Editor*, задайте имена термов и тип функции принадлежности, щелкнув два раза левой клавишей мыши на имени терма (тип можно оставить по умолчанию).

Функции принадлежности в среде *fuzzyTech* определяются графически. Для этого необходимо кликнуть один раз на терме, затем на графике задать расположение точек. Для добавления точек изгиба необходимо дважды кликнуть на графике в нужном месте, для удаления необходимо выделить точку сгиба одним нажатием мыши, затем нажать на клавиатуре клавишу «delete».

4. Работа с редактором правил

Для входа в редактор нажмите правую кнопку мыши на выбранном прямоугольнике, обозначающем блок правил.

Выбрав пункт *Properties*, в закладке *General* поля *Rule Block Name* задайте имя базы правил. В поле *Configuration* определяем входные и выходные переменные.

Выбрав пункт *Spreadsheet Rule Editor*, задайте собственно нечеткие правила, воспользовавшись редактором правил (рис. 17).

Сформируйте несколько баз знаний, поместив в базу правила, полученные вами в лабораторной работе №18.

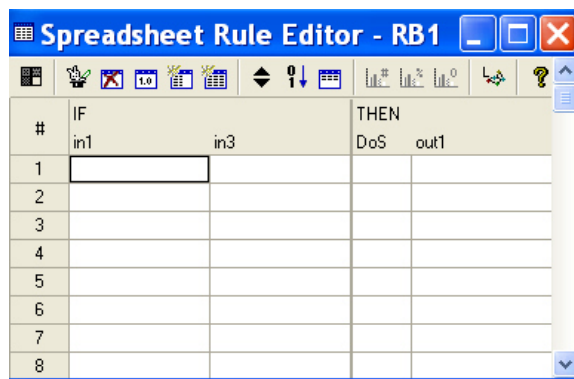


Рисунок 17 – Меню табличного редактора правил

Для этого установите курсор мыши на нужную клетку и нажмите правую кнопку мыши. В появившемся подменю выберите необходимый терм. Столбец DoS заполните значением 1.0.

Сформировать базу правил можно, воспользовавшись матричным редактором *Matrix Rule Editor* (рис. 18), вызвать который можно, нажав правую кнопку мыши на выбранном прямоугольнике, обозначающем блок правил.

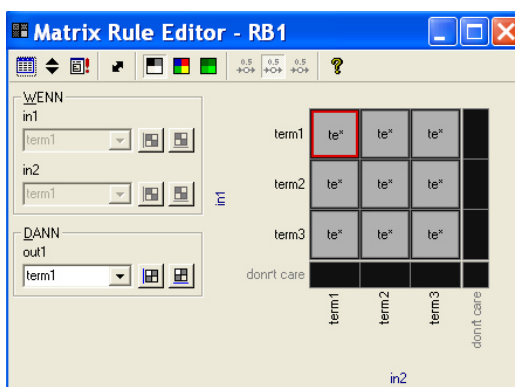


Рисунок 18 – Меню матричного редактора правил

Сохранить сформированные базы знаний с помощью меню File → Save (Save as).

5. Документирование

Оформление документации в среде *fuzzyTech* выполняется с помощью меню File → Project Information File → Documentation.

ЛАБОРАТОРНАЯ РАБОТА № 21.

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ И КОЛИЧЕСТВА ТЕРМОВ ЛИНГВИСТИЧЕСКИХ ПЕРЕМЕННЫХ НА КАЧЕСТВО ВЫВОДА В СРЕДЕ FUZZYTECH

Цель работы.

Знакомство с графическим редактором переменных и с графическими средствами просмотра поверхности нечеткого вывода пакета *fuzzyTech*. Исследование влияния количества термов лингвистических переменных на качество вывода в среде *fuzzyTech*.

Ход работы:

1. Изучение способов задания основных функции принадлежности

Существует множество типов и форм функций принадлежности термов лингвистических переменных. Рассмотрим четыре из них наиболее часто используемые в практических и теоретических работах: треугольные, трапециевидные, гауссовы.

Треугольная функция принадлежности $\mu_{\tilde{a}}(x)$ аналитически описывается следующим образом:

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-a_1}{a_0-a_1}, & a_1 \leq x \leq a_0, \\ \frac{x-a_2}{a_0-a_2}, & a_0 \leq x \leq a_2, \\ 0, & \text{в других случаях.} \end{cases}$$

Трапециевидная функция принадлежности $\mu_{\tilde{u}}(x)$ аналитически описывается следующим образом:

$$\mu_{\tilde{u}}(x) = \begin{cases} \frac{x-u_1}{u_{01}-u_1}, & u_1 \leq x \leq u_{01}, \\ 1, & u_{01} \leq x \leq u_{02}, \\ \frac{x-u_2}{u_{02}-u_2}, & u_{02} \leq x \leq u_2, \\ 0, & \text{в других случаях.} \end{cases}$$

Гауссова функция принадлежности $\mu_{\tilde{n}}(x)$ описывается следующим образом:

$$\mu_{\tilde{n}}(x) = \exp\left(-\left(\frac{x-m_0}{\sigma_0}\right)^2\right)$$

В проектах *fuzzyTech* могут использоваться различные типы и формы функций принадлежности термов лингвистических переменных. Чаще всего используется стандартный графический (не аналитический) вариант формы функции принадлежности, основанный на использовании четырех точек для задания соответствующей формы функции принадлежности, которые могут иметь одну из следующих форм:

- линейную (*L-shape*), предполагающую представление функции принадлежности в форме треугольной, трапециевидной или их комбинации;
- S-образную, которая предполагает задание функции принадлежности в форме нелинейного представления.

Графическим средством задания функций принадлежности в среде *fuzzyTech* является редактор переменных, предназначенный для спецификации термов лингвисти-

ческой переменной проекта и редактирования ее функции принадлежности в графическом режиме.

Графическое окно редактора переменных может быть открыто двойным щелчком мыши на изображении прямоугольника соответствующей переменной в окне редактора проекта (*Project Editor*) или двойным щелчком мыши на имени соответствующей переменной в окне просмотра структуры проекта. Результат вызова редактора переменной in1, которая имеет пять термов с именами term1, term2, term3, term4, term5, представлен на рис. 19.

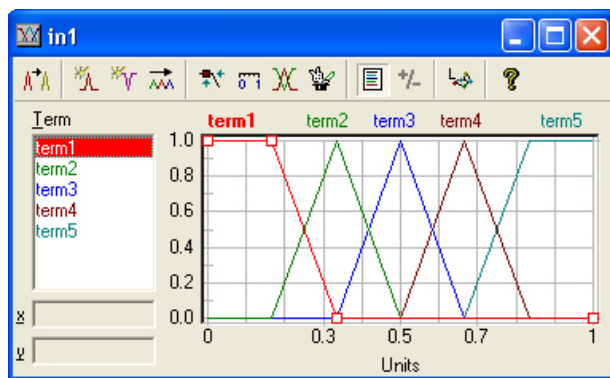









Рисунок 19 – Окно редактора переменных

В секции графиков функций принадлежности изображены две оси, на горизонтальной представлен весь диапазон значений редактируемой переменной, на вертикальной – интервал значений функции принадлежности $[0,1]$.

Редактирование ведется для одного выделенного терма, которому в окне функции принадлежности соответствует график с характерными малыми квадратами или маркерами в местах пересечения кривой с горизонтальными линиями, соответствующими значениям функции принадлежности 0 и 1. Маркеры могут быть использованы для изменения вида и значения функции принадлежности. Для этого следует позиционировать курсор мыши и, не отпуская ее, переместить маркер в нужном направлении.

Выделить нужный терм в секции имен можно последовательным нажатием клавиш «↑» или «↓» на клавиатуре или щелчком левой кнопки мыши на имени терма в данной секции. Выделенный терм можно удалить нажатием клавиши «Delete» на клавиатуре или воспользоваться операцией контекстного меню, которое может быть вызвано нажатием правой кнопки мыши при позиционировании в окне редактора. Для добавления новых  термов следует воспользоваться кнопкой панели инструментов редактора или операцией контекстного меню, либо нажатием клавиш Ctrl+T.

2. Нечёткий логический вывод

Для проведения нечеткого вывода необходимо нажать  или зайти в главное меню Debug и выбрать Interactive. При этом активизируются дополнительные кнопки      (Transfer Plot, 3D Plot, Time Plot, Rule Analyzer, Statistics). При нажатии первых двух выводиться двумерное и трехмерное изображение результатов работы, соответственно.

3. Анализ результатов нечеткого вывода

Анализ необходимо провести путем сравнения поверхностей моделируемой нелинейной функции и поверхности, полученной в результате нечеткого вывода.

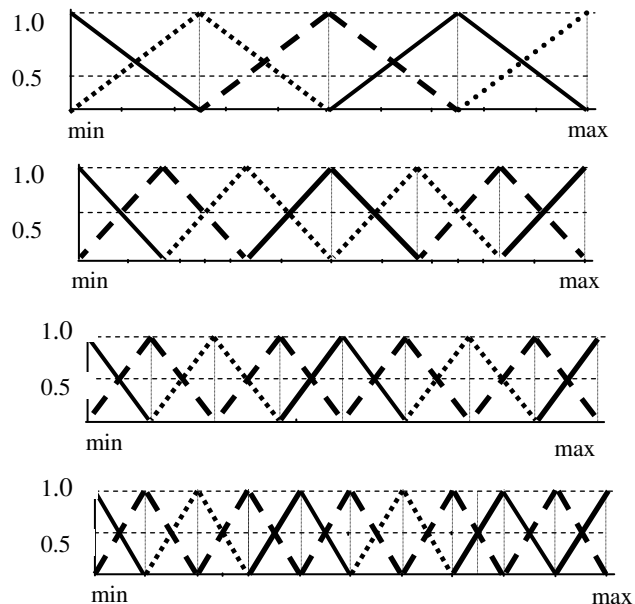


Рисунок 20 - Функции принадлежности

Задание для самостоятельной работы

Выполнить отчёт по лабораторной работе в программе *fuzzyTech* средствами редактора Microsoft Word. В качестве задачи использовать построение нечёткой системы полива цветов в зависимости от температуры ($^{\circ}\text{C}$) в помещении и ширины листьев растения (см) из лабораторной работы № 18. Отчёт должен содержать следующие обязательные пункты: титульный лист, моделируемую нелинейную функцию, каркас проекта, описание переменных, описание всех баз правил.