

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт электронного обучения

Специальность 220201 «Управление и информатика в технических системах»

Кафедра Автоматики и компьютерных систем

ДИПЛОМНАЯ РАБОТА

Тема работы	
Программно-методическое обеспечение для изучения цифровых регуляторов, построенных на нечеткой логике	

УДК 004.41:658.515

Студент

Группа	ФИО	Подпись	Дата
3-8101	Воронов Владимир Александрович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры АиКС	Скороспешкин В. Н.	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры менеджмента	Конотопский В. Ю.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ЭБЖ	Невский Е. С.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Заведующий кафедрой АиКС	Фадеев А. С.	к.т.н.		

Томск – 2016 г.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт Электронного обучения

Специальность Управление и информатика в технических системах

Кафедра Автоматики и компьютерных систем

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

дипломной работы

Студенту:

Группа	ФИО
3-8101	Воронову Владимиру Александровичу

Тема работы:

Программно-методическое обеспечение для изучения цифровых регуляторов, построенных на нечеткой логике	
Утверждена приказом директора (дата, номер)	15.04.2016 г. №2917/с

Срок сдачи студентом выполненной работы:	06.06.2016
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	1. Промышленный микропроцессорный контроллер «Кросс» 2. Программный пакет Isagraf 3. Методы фазификации 4. Дефазификация методом центра тяжести и максимума 5. Нечеткие П- и ПИ-регуляторы
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Основы теории нечетких множеств, нечеткой логики и структуры систем автоматического регулирования с нечеткими регуляторами 2. Архитектура, технические характеристики и функциональные возможности контроллера КРОСС 3 Структурная схема лабораторного комплекса, схема внешних соединений 4. SCADA-система Master SCADA 5. Программно-методическое обеспечение 6. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.

	7. Социальная ответственность.
Перечень графического материала	Презентация в формате *.ppt, 15 слайдов Слайд 1. Титульный лист Слайд 2. Цели и задачи Слайд 3. Одноконтурная система управления Слайд 4. Структура нечеткого регулятора Слайд 5. Этап фазификации Слайд 6. Фазификация Слайд 7. Методы дефазификации Слайд 8. Этап дефазификации Слайд 9. Дефазификация Слайд 10. Методические указания по выполнению лабораторной работы «Цифровые регуляторы, построенные на нечеткой логике» Слайд 11. Методические указания по выполнению лабораторной работы «Нечеткий П-регулятор» Слайд 12. Структурная схема лабораторного стенда Слайд 13. Техничко-экономическое обоснование НИР Слайд 14. Социальная ответственность Слайд 15. Заключение

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
«Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	Конотопский В.Ю., Доцент кафедры менеджмента, к.э.н.
«Социальная ответственность»	Невский Е.С. , ассистент кафедры ЭБЖ

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Заключение

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	18.01.2016
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры АиКС	Скороспешкин В.Н.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8101	Воронов В.А.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
3 - 8101	Воронов Владимир Александрович

Институт	Институт электронного обучения	Кафедра	Автоматики и компьютерных систем
Уровень образования	Специалист	Направление/специальность	Управление и информатика в технических системах

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	...
2. Нормы и нормативы расходования ресурсов	...
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	...

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала инженерных решений (ИР)	...
2. Формирование плана и графика разработки и внедрения ИР	...
3. Обоснование необходимых инвестиций для разработки и внедрения ИР	...
4. Составление бюджета инженерного проекта (ИП)	...
5. Оценка ресурсной, финансовой, социальной, бюджетной эффективности ИР и потенциальных рисков	...

Перечень графического материала (с точным указанием обязательных чертежей)

1. «Портрет» потребителя
2. Оценка конкурентоспособности ИР
3. Матрица SWOT
4. Модель Кано
5. ФСА диаграмма
6. Оценка перспективности нового продукта
7. График разработки и внедрения ИР
8. Инвестиционный план. Бюджет ИП
9. Риски ИП

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры менеджмента	Конотопский В. Ю.	К.Э.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3 - 8101	Воронов Владимир Александрович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
3-8101	Воронов В.А.

Институт	Электронного обучения	Кафедра	АиКС
Уровень образования		Направление/специальность	Управление и информатика в технических системах

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Разработка программно-методического обеспечения для изучения цифровых регуляторов, построенных на нечеткой логике с использованием промышленного контроллера КРОСС-500.
--	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность 1.1. Анализ выявленных вредных и опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности: – Опасность поражения электрическим током – повышенный уровень электромагнитных излучений	Электрический ток, проходя через тело человека, может вызвать поражение внутренних или внешних органов. При поражении внутренних органов может наступить паралич органов дыхания или фибриляция, что часто влечёт за собой смертельный исход. Электромагнитные поля оказывают негативное влияние на работу другого оборудования в качестве помех и здоровье человека, защитные ресурсы организма начинают истощаться быстрее.
2. Экологическая безопасность: – Сократит расходы электроэнергии; – Увеличит эффективность водоснабжения	Организовав работу насосов в экономичном режиме и при условии обеспечения достаточного расхода воды, эффективность работы насосного отделения и всей водопроводной станции будет значительно увеличена.
3. Безопасность в чрезвычайных ситуациях: – Гидроудар – Взрыв гидроаккумулятора	Неправильно запрограммированный контроллер может привести к значительному увеличению давления в системе. Это в свою очередь может привести к гидроудару, а иногда и взрыву гидроаккумулятора.
4. Правовые и организационные вопросы обеспечения безопасности: – правовые нормы трудового законодательства; – санитарные правила и нормы.	СанПиН 2.2.4.1329-03 - Требования по защите персонала от воздействия импульсных электромагнитных полей; СН 2.2.4/2.1.8.562-96 - Шум на рабочих местах, в помещениях жилых, общественных зданий; ГОСТ Р 12.1.019-2009 ССБТ. – Электробезопасность; СНиП 2.01.02-85* - Противопожарные нормы; СанПиН 2.1.2.568-96 - Гигиенические требования к устройству, эксплуатации и качеству воды; СанПиН 2.2.2.542-96

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ЭБЖ	Невский Е.С.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8101	Воронов В.А.		

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**



Институт кибернетики

Кафедра автоматизации и компьютерных систем

Направление 220200-«Управление и информатика в технических системах»

Уровень образования – специалист

Период выполнения – осенний/весенний семестр 2015/2016 учебного года

Форма представления работы:

Дипломная работа

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	06.06.2016 г.
--	---------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
27.05.2016 г.	Основная часть	75
20.05.2016 г.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
28.05.2016 г.	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель доцент кафедры АИКС	Скороспешкин В.Н.	к.т.н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Заведующий кафедрой АиКС	Фадеев А.С.	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа содержит 113 страницы, 30 рисунков, 15 таблиц, 22 источника, 15 слайдов, 3 приложения.

Объектом исследования являются цифровые регуляторы, построенные на нечеткой логике.

Цель работы – создание программно методического обеспечения для изучения цифровых регуляторов, построенных на нечеткой логике.

В процессе работы были изучены основы нечеткой логики, принципы построения нечетких регуляторов, состав языка программирования FBD в системе ISaGRAF, проведено ознакомление со SCADA-системой Master SCADA. В результате разработано программно-методическое обеспечение по выполнению двух лабораторных работ, целью выполнения которых является получение знаний по разработке нечетких регуляторов и получение практических навыков их программной реализации.

Методическое обеспечение предназначено для выполнения лабораторных работ по учебному курсу “Автоматизированное управление в технических системах” студентами, обучающимися по направлению 27.04.04-“Управление в технических системах” кафедры автоматики и компьютерных систем Томского политехнического университета.

СОДЕРЖАНИЕ

Введение.....	11
1 Основы теории нечетких множеств, нечеткой логики и структуры систем автоматического регулирования с нечеткими регуляторами	13
1.1 Понятие нечеткого множества его характеристики и его основные свойства	13
1.2 Лингвистические переменные и нечеткие числа.....	21
1.3. Нечеткая логика и ее основные свойства	29
1.4 Методы фаификации (приведения к нечеткости) и дефаификации (приведения к четкости)	38
1.5 Структуры систем автоматического регулирования с нечеткими регуляторами	46
2 Архитектура, технические характеристики и функциональные возможности контроллера КРОСС	54
2.1 Назначение, состав и архитектура контроллера	54
2.2 Технические характеристики контроллера	57
2.3 Функциональные возможности контроллера.....	61
2.4 Система программирования контроллера	64
3 Структурная схема лабораторного комплекса, схема внешних соединений. 67	
3.1 Структурная схема комплекса	67
3.2 Краткое описание аналогового вычислительного комплекса АВК-6 ...	69
3.2.1 Назначение АВК-6	69
3.2.2 Общие характеристики и состав комплекса.....	69
3.2.3 Конструкция комплекса	69
3.3.Схема внешних соединений.....	70
4 SCADA-система Master SCADA.....	73
4.1 Назначение SCADA-систем, виды SCADA-систем	73
4.2 Отличительные особенности пакета программ MasterScada.....	75
5 ПРОГРАММНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	79
5.1 Цифровые регуляторы, построенные на нечеткой логике.....	79

5.1.1 Структура нечеткого регулятора.....	79
5.1.2 Нечеткая логика и ее основные свойства	79
5.1.3 Методы фазификации и дефазификации.....	79
5.1.4 Структура лабораторного комплекса.....	79
5.1.5 Система программирования контроллера	80
5.1.6 Scada-система MASTER SCADA	80
5.1.7 OPC-Сервер	80
5.1.8 Описание программы	81
5.1.9 Задание	81
5.1.10 Содержание отчета	82
5.1.11 Контрольные вопросы	82
5.2 Нечеткий П-регулятор	83
5.2.1 Нечеткая логика и ее основные свойства	83
5.2.2 Структура САР с нечетким П-регулятором	83
5.2.3 Программа реализации нечеткого П-регулятора в системе программирования контроллеров Isograf	83
5.2.4 Задание	83
5.2.5 Содержание отчета	84
5.2.6 Контрольные вопросы	84
5.2.7 Методические указания по установке связи с контроллером и настройке прожектора	84
6 Финансовый менеджмент, ресурсоэффективность	86
и ресурсосбережение	86
6.1 Организация и планирование работ	86
6.1.1 Продолжительность этапов работ	87
6.1.2 Расчет накопления готовности проекта.....	92
6.2 Расчет сметы затрат на выполнение проекта	93
6.2.1 Расчет затрат на материалы	94
6.2.2 Расчет заработной платы.....	94
6.2.3 Расчет затрат на социальный налог	96

6.2.4 Расчет затрат на электроэнергию	96
6.2.5 Расчет амортизационных расходов	97
6.2.6 Расчет прочих расходов	98
6.2.7 Расчет общей себестоимости разработки	99
6.2.8 Расчет прибыли	99
6.2.9 Расчет НДС	100
6.2.10 Цена разработки НИР	100
6.3 Оценка экономической эффективности проекта	100
7 Социальная ответственность	101
7.1 Производственная безопасность	101
7.2 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения	101
7.3 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	103
7.4 Экологическая безопасность.....	105
7.5 Безопасность в чрезвычайных ситуациях.....	107
7.6 Правовые и организационные вопросы обеспечения безопасности. ..	108
ЗАКЛЮЧЕНИЕ	110
CONCLUSION	111
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	112
Приложение А «Методические указания к выполнению лабораторной работы «Цифровые регуляторы, построенные на нечеткой логике».....	114
Приложение Б «Программа реализации нечеткого ПИ-регулятора»	151
Приложение В «Методические указания к выполнению лабораторной работы «Нечеткий П-регулятор»	155

Введение

Одной из основных задач при проектировании систем автоматического регулирования (САР) является выбор структурной схемы, параметров отдельных устройств и способов их технической реализации, обеспечивающих высокие требования в отношении качества.

В системах автоматического регулирования объектами управления, у которых свойства и параметры могут изменяться в течении времени, необходимо обеспечивать показатели качества регулирования, такие как: затухание, перерегулирование, интегральный квадратичный критерий качества. В теории автоматического управления имеется несколько подходов, которые позволяют решить задачу качественного управления объектами, имеющими переменные и неопределенные параметры. Основной подход основывается на применении обычных ПИД – регуляторов.

Но, так как классический ПИД – регулятор обеспечивает управление при небольшом диапазоне изменения параметров объекта управления, при которых он сохраняет работоспособность и обеспечивает необходимый уровень качества регулирования, то необходимо применение других методов для обеспечения качественного регулирования системы.

В данной работе рассматривается альтернативный подход к решению вопроса обеспечения требуемого качества системы автоматического регулирования с нестационарными объектами, который заключается в использовании нечетких регуляторов.

Основной идеей данных регуляторов является использование аппарата нечеткой логики для подстройки параметров ПИД-регулятора.

Настоящий проект выполнен на кафедре Автоматики и компьютерных систем Томского политехнического университета и посвящен разработке программно-методического обеспечения для изучения цифровых регуляторов, построенных на нечеткой логике.

Целью использования программно-методического обеспечения является получение знаний по разработке нечетких регуляторов и получение практических навыков их программной реализации.

1 Основы теории нечетких множеств, нечеткой логики и структуры систем автоматического регулирования с нечеткими регуляторами

1.1 Понятие нечеткого множества его характеристики и его основные свойства

Под понятием множество понимают совокупность элементов, обладающих некоторым общим свойством. При этом любой элемент заранее аксиоматически либо принадлежит данному множеству, либо не принадлежит. Однако, как показывает практика прикладных исследований, подобный «булев» принцип в подавляющем большинстве случаев не отвечает процессам, протекающим в реальных сложных системах, то есть приводит к неоправданной идеализации математического описания таких систем. Иными словами, язык обычных множеств оказывается недостаточно гибким для формализации элементов неопределенности, свойственных реальным системам.

Понятие нечеткого множества строится на гипотезе, что любой элемент лишь частично принадлежит данному множеству, поэтому одним из основных способов математического описания нечеткого множества является определение степени такой принадлежности некоторым числом, например из интервала $[0,1]$. При этом границы интервала, то есть 1 и 0, означают, соответственно, «принадлежит» и «не принадлежит» [1,20].

Определение 1.1.1 Пусть дано универсальное множество E с элементом x и некоторым свойством R . Обычное четкое подмножество A множества E с элементами, удовлетворяющими свойству R , определяется как множество упорядоченных пар:

$$A = \{ \mu_A(x) / x \} ,$$

где $\mu_A(x)$ - характеристическая функция, которая принимает значение 1 или 0 в зависимости от степени принадлежности элемента x к свойству R .

Отличие нечеткого подмножества от обычного заключается в отсутствии однозначности для элементов x относительно свойства R ,

вследствие подмножество A множества E определяется множеством упорядоченных пар:

$$A = \{\mu_A(x) / x\},$$

где $\mu_A(x)$ - функция принадлежности, которая принимает в упорядоченном множестве M .

Функция принадлежности характеризует уровень принадлежности элемента x подмножеству A . Множество M является множеством принадлежностей.

Пример записи нечеткого множества:

Запишем

$$E = \{x_1, x_2, x_3, x_4, x_5\}, \quad M = [0, 1];$$

Где A – нечеткое множество

$$\mu_A(x_1) = 0,3; \quad \mu_A(x_2) = 0; \quad \mu_A(x_3) = 1; \quad \mu_A(x_4) = 0,5; \quad \mu_A(x_5) = 0,9.$$

Тогда A можно записать следующим образом:

$$A = \{0,3 / x_1, 0 / x_2, 1 / x_3, 0,5 / x_4, 0,9x_5\},$$

Рассмотрим основные характеристики нечетких множеств. Пусть $M = \{0,1\}$ и A – нечеткое множество, которой принадлежат элементы множества E и M – множество принадлежностей.

Определение 1.1.2 Величина $\sup_{x \in E} \mu_A(x)$ называется высотой нечеткого множества A . Нечеткое множества можно считать нормальным, если выполняется условие:

$$\sup_{x \in E} \mu_A(x) = 1,$$

и субнормальным, если

$$\sup_{x \in E} \mu_A(x) < 1.$$

Определение 1.1.3 Нечеткое множество можно охарактеризовать пустым, если

$$\forall x \in E, \mu_A(x) = 0$$

Непустое субнормальное множество нормализуется следующей формулой:

$$\mu_A(x) = \frac{\mu_A(x)}{\sup_{x \in E} \mu_A(x)}.$$

Определение 1.1.4 Нечеткое множество унимодально, если $\mu_A(x) = 1$ только на одном x из E .

Определение 1.1.5 Носителем нечеткого множества A следует считать обычное подмножество, удовлетворяющее свойству

$$\mu_A(x) > 0,$$

отсюда

$$A = \{x / x \in E, \mu_A(x) > 0\}.$$

Определение 1.1.6 Элементы x , принадлежащие E , для которых

$$\mu_A(x) = 0,5 \text{ являются точками перехода } A.$$

Примеры нечетких множеств:

1. Пусть $E = \{0, 1, 2, \dots, 10\}$, $M = \{0, 1\}$. Определение нечеткого множества «Несколько» выглядит следующим образом:

$$\text{«Несколько»} = 0,5/3 + 0,8/4 + 1/5 + 1/6 + 0,8/7 + 0,5/8;$$

Характеристики нечеткого множества «Несколько»:

$$\text{высота} = 1, \text{ носитель} = \{3, 4, 5, 6, 7, 8\}, \text{ точки перехода} - \{3, 8\}.$$

2. Имеем универсальное множество $E = \{\text{Иванов}, \text{Петров}, \text{Сидоров}\}$

Пусть $E = \{0, 1, 2, \dots, 100\}$ соответствует возрасту. Тогда нечеткое множество «Молодой» определим по формуле:

$$\mu_{\text{молодой}}(x) = \begin{cases} 1, & x \in [1, 25] \\ \frac{1}{1 + ((x - 25)/5)^2}, & x > 25 \end{cases}.$$

Зададим при помощи функции принадлежности $\mu_{\text{молодой}}(x)$ нечеткое множество «Молодой» на $E = \{0, 1, 2, \dots, 100\}$. при этом:

$$\mu_{\text{молодой}}(\text{Сидоров}) = \mu_{\text{молодой}}(x),$$

За x принимаем возраст Сидорова.

3. Пусть $E = \{\text{Запорожец}, \text{Жигули}, \text{Мерседес}, \dots\}$ — марки автомобилей, а $E = \{0, \infty\}$ универсальное множество «Скорость», тогда на E определяем

нечеткие множества «Тихоходные», «Средние», «Скоростные», функции принадлежности которых приведены на рисунке 1.1.1.

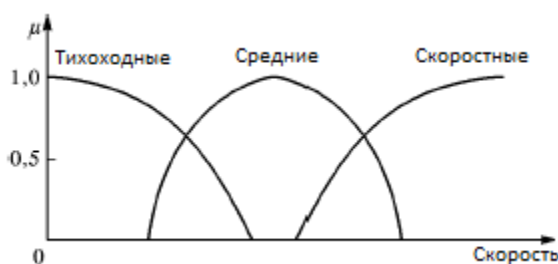


Рисунок 1.1.1 – Примеры функций принадлежности

Зная скорость автомобилей, можем определить для этих функций принадлежности нечеткие множества. К примеру заданное на множестве E нечеткое множество «Тихоходные» изображено на рисунке 1.1.2



Рисунок 1.1.2 – Пример задания нечеткого множества

Рассмотрим основные логические операции над нечеткими множествами и их свойства.

Определение 1.1.7 Операция включения. Пусть A и B – нечеткие множества на множестве E . $A \subset B$, если $\forall x \in E \quad \mu_A(x) \leq \mu_B(x)$.

Определение 1.1.8 Равенство. Пусть A и B – нечеткие множества на множестве E . $A = B$, если $\forall x \in E \quad \mu_A(x) = \mu_B(x)$.

Определение 1.1.9 Дополнение. Пусть $M = [0,1]$, A и B – нечеткие множества на множестве E . $B = \bar{A}$ или $\bar{A} = B$, если $\forall x \in E \quad \mu_A(x) = 1 \Leftrightarrow \mu_B(x) = 0$.

Замечание: Очевидно, что $\bar{\bar{A}} = A$. Дополнение определено для $M = [0,1]$, но можно определить для $\forall$ упорядоченного M .

Определение 1.1.10 Пересечение. $A \cap B$ - наибольшее нечеткое подмножество, которое одновременно содержится и в A и в B

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)).$$

Определение 1.1.11 Объединение. $A \cup B$ - наименьшее нечеткое подмножество, которое включает как A , так и B

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

Определение 1.1.12 Разность. $A \ominus B = A \cap \bar{B}$

$$\mu_{A \ominus B}(x) = \mu_{A \cap \bar{B}} = \min(\mu_A(x), 1 \ominus \mu_B(x)).$$

Определение 1.1.13 Дизъюнктивная сумма.

$A \oplus B = (A \ominus B) \cup (B \ominus A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$, функция принадлежности которой имеет вид:

$$\mu_{A \oplus B}(x) = \max(\min(\mu_A(x), 1 \ominus \mu_B(x)); \min(1 \ominus \mu_A(x), \mu_B(x))).$$

Примеры: Пусть

$$A = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4;$$

$$B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4;$$

$$C = 0,1/x_1 + 1/x_2 + 0,2/x_3 + 0,9/x_4.$$

1) $A \subset B$, другими словами A содержится в B (B доминирует над A);
 C несравнимо с A и B ;

2) $A \neq B \neq C$;

$$3) \bar{A} = 0,6/x_1 + 0,8/x_2 + 1/x_3 + 0/x_4;$$

$$\bar{B} = 0,3/x_1 + 0,1/x_2 + 0,9/x_3 + 0/x_4;$$

$$4) A \cap B = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4;$$

$$5) A \cup B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4;$$

$$6) A \ominus B = A \cap \bar{B} = 0,3/x_1 + 0,1/x_2 + 0/x_3 + 0/x_4;$$

$$B \ominus A = \bar{A} \cap B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4;$$

$$7) \quad A \oplus B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4.$$

Свойства пересечения и объединения:

1. Коммутативность.

$$A \cap B = B \cap A,$$

$$A \cup B = B \cup A.$$

2. Ассоциативность.

$$(A \cap B) \cap C = (B \cap A) \cap C,$$

$$(A \cup B) \cup C = (B \cup A) \cup C.$$

3. Идемпотентность.

$$A \cap A = A,$$

$$A \cup A = A.$$

4. Дистрибутивность.

$$(A \cap (B \cup C)) = (A \cap B) \cup (A \cap C),$$

$$(A \cup (B \cap C)) = (A \cup B) \cap (A \cup C).$$

5. Пусть $\otimes$ есть пустое множество ($\mu_{\otimes}(x) = 0 \quad \forall x \in E$). Тогда

$$A \cup \otimes = A,$$

$$A \cap \otimes = \otimes.$$

6. Пусть E есть универсальное множество. Тогда

$$A \cap E = A,$$

$$A \cup E = E.$$

7. Теоремы де Моргана

$$\overline{A \cap B} = \bar{A} \cup \bar{B},$$

$$\overline{A \cup B} = \bar{A} \cap \bar{B}.$$

Для нечетких множеств обобщенно:

$$A \cap \bar{A} \neq \otimes, \quad A \cup \bar{A} \neq E.$$

Замечание: Операции над нечеткими множествами, приведенные выше, выведены используя операции *MIN* и *MAX*.

Один из подходов к операторам пересечения и объединения состоит в их определении в классе треугольных норм и конорм.

Треугольной нормой (t-нормой) называется двуместная действительная функция $T : [0,1] \times [0,1] \rightarrow [0,1]$, которая удовлетворяет условиям:

1. ограниченность

$$T(0,0) = 0; \quad T(\mu_A, 1) = \mu_A; \quad T(1, \mu_A) = \mu_A;$$

2. монотонность

$$T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D), \text{ при условии } \mu_A \leq \mu_C, \mu_B \leq \mu_D;$$

3. коммутативность

$$T(\mu_A, \mu_B) = T(\mu_B, \mu_A);$$

4. ассоциативность

$$T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C).$$

Примеры t – норм:

$$\min(\mu_A, \mu_B);$$

$$\text{произведение } \mu_A \cdot \mu_B;$$

$$\max(0, \mu_A + \mu_B - 1).$$

Треугольной конормой (t-конормой) называется двуместная действительная функция $S : [0,1] \times [0,1] \rightarrow [0,1]$, которая удовлетворяет условиям:

1. ограниченность

$$S(1,1) = 1; \quad T(\mu_A, 0) = \mu_A; \quad T(0, \mu_A) = \mu_A;$$

2. монотонность

$$S(\mu_A, \mu_B) \geq S(\mu_C, \mu_D), \text{ при условии } \mu_A \geq \mu_C, \mu_B \geq \mu_D;$$

3. коммутативность

$$S(\mu_A, \mu_B) = S(\mu_B, \mu_A);$$

4. ассоциативность

$$S(\mu_A, S(\mu_B, \mu_C)) = S(S(\mu_A, \mu_B), \mu_C).$$

Примеры t – конорм:

$$\max(\mu_A, \mu_B);$$

$$\mu_A + \mu_B - \mu_A \cdot \mu_B;$$

$$\min(1, \mu_A + \mu_B).$$

Далее перейдем к рассмотрению алгебраических операций над нечеткими множествами, а также их свойств.

Определение 1.1.14 Алгебраическое произведение $A * B$

$$\forall x \in E, \mu_{A \times B}(x) = \mu_A(x) \times \mu_B(x).$$

Определение 1.1.15 Алгебраическая сумма $A + B$

$$\forall x \in E, \mu_{A+B}(x) = \mu_A(x) + \mu_B(x) \Leftrightarrow \mu_A(x) \times \mu_B(x).$$

Свойства операций произведения и суммы

1. Коммутативность.

$$A \times B = B \times A,$$

$$A + B = B + A.$$

2. Ассоциативность.

$$(A \times B) \times C = A \times (B \times C),$$

$$(A + B) + C = A + (B + C).$$

3. $A \times \emptyset = \emptyset, A + \emptyset = A, A \times E = A, A + E = E.$

4. Теоремы де Моргана:

$$\overline{A \times B} = \bar{A} + \bar{B},$$

$$\overline{A + B} = \bar{A} \times \bar{B}.$$

Не выполняются:

1. Идемпотентность.

$$A \times A = A,$$

$$A + A = A.$$

2. Дистрибутивность.

$$A \times (B + C) = (A \times B) + (A \times C),$$

$$A + (B \times C) = (A + B) \times (A + C).$$

3. $A \times \bar{A} = \emptyset, A + \bar{A} = E.$

Свойства, которые выполняются при одновременном комбинировании вышеописанных операций:

$$1) A \times (B \cup C) = (A \times B) \cup (A \times C);$$

$$2) A \times (B \cap C) = (A \times B) \cap (A \times C);$$

$$3) A + (B \cup C) = (A + B) \cup (A + C);$$

$$4) A + (B \cap C) = (A + B) \cap (A + C).$$

Основываясь на операции алгебраического произведения найдем возведение в степень α ($\alpha > 0$). Нечеткое множество A^α можно определить как

$$\mu_A^\alpha = \mu_A^\alpha(x).$$

Частный случай:

1. операция концентрирования

$$CON(A) = A^2;$$

2. операция растяжения

$$DIL(A) = A^{0,5}$$

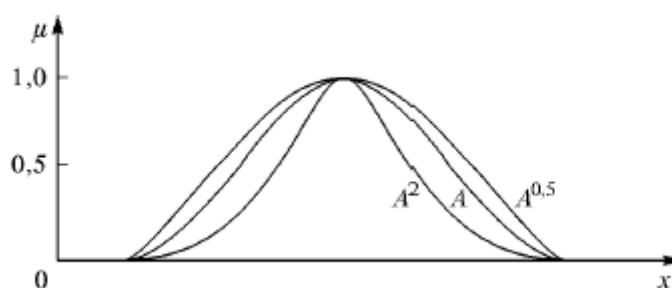


Рисунок 1.1.3 –концентрирование (уплотнение) и растяжение

1.2 Лингвистические переменные и нечеткие числа

Практические исследования последних лет показали, что обычные методы анализа систем и моделирования на ЭВМ, основанные на точной обработке численных данных, фактически не способны охватить огромную сложность реальных технологических процессов. Это обстоятельство приводит к тому, что для получения существенных выводов о поведении последних приходится отказаться от традиционных требований к точности измерений, которые были необходимы при математическом анализе четко определенных механических систем.

Необходимость жертвовать точностью и детерминированностью диктуется также появлением определенных классов задач управления, связанных с принятием решений оператором в связке «человек – ЭВМ».

Выполнение диалога в таком контуре невозможно, не используя языки, близкие к естественному, способные к описанию нечетких категорий, приближенных к понятиям и представлениям, свойственным человеку. В связи с этим разумно использовать понятия лингвистической переменной, введенной впервые Л. Заде. Они адекватно отразить приближенное описание предметов и явлений в словесной форме тогда, когда точное определенное описание отсутствует. Вместе с тем необходимо учесть, что многие нечеткие категории, описанные лингвистически, очень часто не менее информативны, чем точное описание [2, 20].

Определение 1.2.1 Лингвистической переменной (ЛП) - это набор (β, T, X, G, M) , где:

β - название лингвистической переменной;

T - множество значений β (терм-множество), представляющих собой наименования нечетких переменных, областью определения каждой из которых является множество X . Множество T называется базовым терм-множеством лингвистической переменной;

G - синтаксическая процедура, позволяющая оперировать элементами терм-множества T , в частности, генерировать новые термы (значения). Множество $T \cup G(T)$, где $G(T)$ - множество сгенерированных термов, называется расширенным терм-множеством лингвистической переменной;

M – семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G , в нечеткую переменную, то есть сформировать соответствующее нечеткое множество.

Замечание: для избежания большого количества символов:

1. Символ β используют как для названия переменной, так и для значений этой переменной.

2. Используют один и тот же символ для обозначения нечеткого множества и его названия. В качестве примера рассмотрим терм «Молодой»,

являющийся значением лингвистической переменной β = «возраст», который в тоже время означает и нечеткое множество M («Молодой»).

Присвоение нескольких значений символам предполагает, что контекст позволяет разрешить возможные неопределенности.

Пример. Предположим, эксперт определяет расход драгоценных металлов при помощи «Малый расход», «Средний расход» и «Большой расход», при этом минимальный расход равен 10 гр, а максимальный - 80 гр.

Формализация такого описания может быть проведена при помощи следующей лингвистической переменной (β ; T ; X ; G ; M), где

β - расход материала;

$T = \{\text{«Малый расход»}, \text{«Средний расход»}, \text{«Большой расход»}\};$

$X = [10; 80];$

G – процесс создания новых термов при помощи связок «и», «или» и модификаторов вида «очень», «не», «слегка» и т.д. Например: «Малый или средний расход», «Очень малый расход» и т.д.;

M - процедура задания на $X = [10; 80]$ нечетких подмножеств A_1 =«Малый расход», A_2 = «Средний расход», A_3 =«Большой расход», а также нечетких множеств для термов из $G(T)$ согласно правилам передачи нечетких связок и модификаторов и других операций над нечеткими множествами.

Замечание. Наряду с рассмотренными выше базовыми значениями лингвистической переменной «Расход» ($T = \{\text{«Малый расход»}, \text{«Средний расход»}, \text{«Большой расход»}\}$) допустимы значения, которые зависят от области определения X . Здесь значения лингвистической переменной «Расход» определяются как «около 20 гр», «около 50 гр», «около 70 гр», т. е. в виде нечетких чисел.

Терм-множество и расширенное терм-множество в условиях примера можно охарактеризовать функциями принадлежности, которые приведены на рисунках 1.2.1 и 1.2.2.

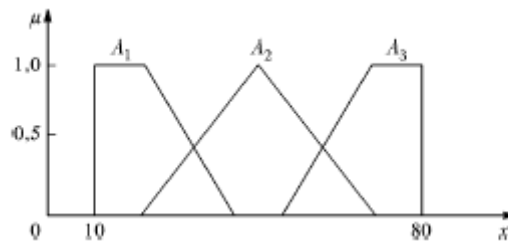


Рисунок 1.2.1 – Функции принадлежности нечетких множеств: A1=«Малый расход», A2=«Средний расход», A3=«Большой расход»

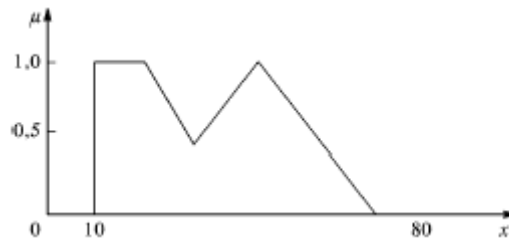


Рис. 1.2.2 – Функции принадлежности нечеткого множества

$$A_1 \cup A_2 = \text{«Малый или средний расход»}$$

Далее переходим к рассмотрению нечетких переменных и дадим основные определения.

Определение 1.2.2 Нечеткие числа – нечеткие переменные, определенные на числовой оси, то есть нечеткое число определяется как нечеткое множество A на множестве действительных чисел R с функцией принадлежности $\mu_A(x) \in [0,1]$, где x – действительное число, то есть $x \in R$.

Определение 1.2.3 Нечеткое число A – нормально, если $\max \mu_A(x) = 1$; выпуклое, если для любых $x \leq y \leq z$ выполняется:

$$\mu_A(x) \leq \mu_A(y) \leq \mu_A(z).$$

Множество α -уровня нечеткого числа A определяется как:

$$A_\alpha = \{x / \mu_A(x) \geq \alpha\}.$$

Подмножество $S_A \subset R$ называется носителем нечеткого числа A , если:

$$S_A = \{x / \mu_A(x) > 0\}.$$

Определение 1.2.4 Нечеткое число A – унимодально, если $\mu_A(x)=1$ справедливо только для одной точки действительной оси.

Определение 1.2.5 Выпуклое нечеткое число A называется нечетким нулем, если

$$\mu_A(0) \leq \sup(\mu_A(x)) .$$

Определение 1.2.6 Нечеткое число A положительно, если $\forall x \in S_A, x > 0$ и отрицательно, если $\forall x \in S_A, x < 0$.

После того, как мы дали основные определения, перейдем к обзору операций над нечеткими числами.

Расширенные бинарные арифметические операции (сложение, умножение и пр.) для нечетких чисел можно определить через операции, которые соответствуют четким числам с применением принципа обобщения следующим образом.

Пусть A и B – нечеткие числа, и $\tilde{\times}$ - нечеткая операция, которая соответствует произвольной алгебраической операции $\times$ над обычными числами. Тогда (используя здесь и в дальнейшем $\bigvee_x$ вместо $\max_x$ и $\bigwedge_x$ вместо $\min_x$) можно записать:

$$C = A \tilde{\times} B \Leftrightarrow \mu_C(z) = \bigvee_{Z=X \times Y} (\mu_A(x) \wedge \mu_B(y)).$$

Отсюда следует:

$$C = A \tilde{+} B \Leftrightarrow \mu_C(z) = \bigvee_{Z=X+Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = A \tilde{-} B \Leftrightarrow \mu_C(z) = \bigvee_{Z=X-Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = A \tilde{\cdot} B \Leftrightarrow \mu_C(z) = \bigvee_{Z=X \cdot Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = A \tilde{\div} B \Leftrightarrow \mu_C(z) = \bigvee_{Z=X \div Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = \tilde{\max}(A, B) \Leftrightarrow \mu_C(z) = \bigvee_{Z=\max(X,Y)} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = \tilde{\min}(A, B) \Leftrightarrow \mu_C(z) = \bigvee_{Z=\min(X,Y)} (\mu_A(x) \wedge \mu_B(y)).$$

Нечеткие числа (L-R)-типа представляют собой разновидность нечетких чисел специального вида, иными словами, они задаются по определенным правилам, целью которых является снижение количества вычислений при выполнении операций над ними.

Функции принадлежности нечетких чисел (L-R)-типа можно задать при помощи невозрастающих на множестве неотрицательных действительных чисел функций действительного переменного $L(x)$ и $R(x)$, которые удовлетворяют условиям:

1. $L(\Leftrightarrow x) = L(x)$, $R(\Leftrightarrow x) = R(x)$;
2. $L(0) = R(0)$.

Без сомнений можно утверждать, что к классу (L-R)-функций имеют отношения функции, изображенные на рисунке 1.2.3 в виде графиков.

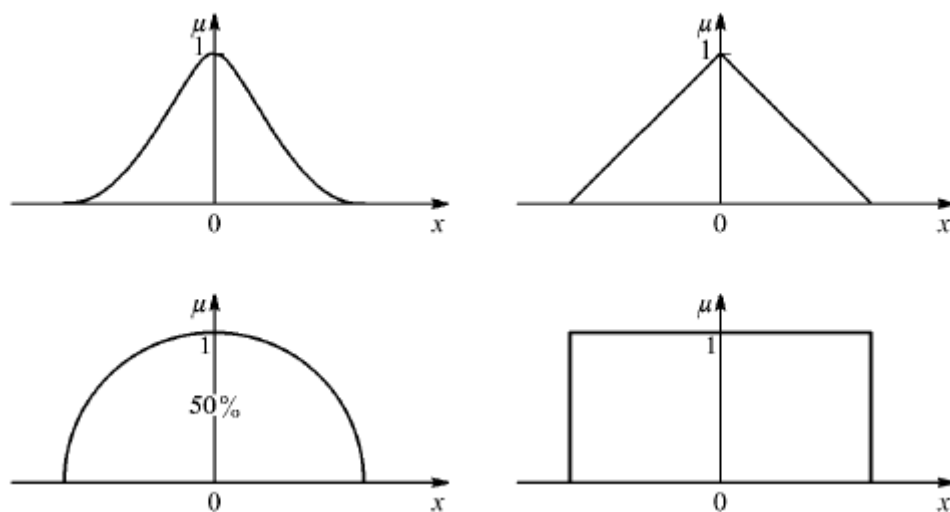


Рисунок 1.2.3 – Возможный вид (L-R)-функций

В качестве примера аналитического написания (L-R)-функций можно представить:

$$L(x) = e^{-|x|^p}, p \geq 0; \quad R(x) = \frac{1}{1+|x|^p}, p \geq 0,$$

и т.д.

Пусть $L(y)$ и $R(y)$ есть функции (L-R)-типа (конкретные).
Унимодальное нечеткое число A с модой a , то есть $\mu_A(a) = 1$, с помощью $L(y)$ и $R(y)$ можно задать способом, приведенным ниже:

$$\mu_A(x) = \begin{cases} L(\frac{a \Leftrightarrow x}{\alpha}) & \text{при } x \leq a, \\ R(\frac{x \Leftrightarrow a}{\beta}) & \text{при } x > a \end{cases},$$

где a - мода; $\alpha > 0, \beta > 0$ - левый и правый коэффициенты нечеткости.

Из этого следует, что при заданных $L(y)$ и $R(y)$ нечеткое унимодальное число можно задать следующим образом: $A = (a, \alpha, \beta)$.

Толерантное нечеткое число соответственно задается

$$A = (a_1, a_2, \alpha, \beta),$$

где a_1 и a_2 – границы толерантности, то есть на отрезке $[a_1, a_2]$ значение функции принадлежности равно 1.

В качестве примеров функций принадлежности нечетких чисел (L-R)-типа можно привести графики, изображенные на рисунке 1.2.4.

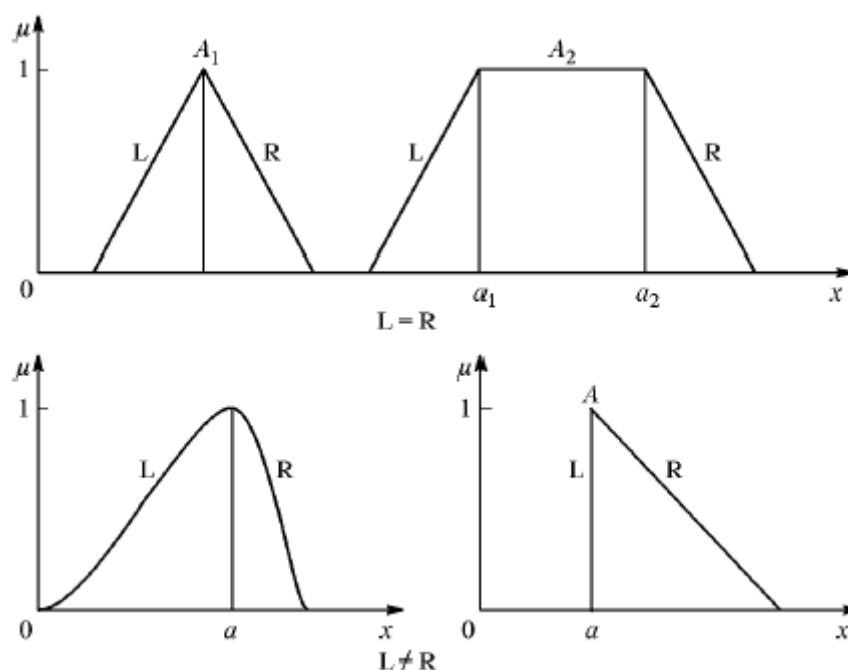


Рисунок 1.2.4 – Примеры графиков функций принадлежности нечетких чисел (L-R)-типа

Можно отметить, что в конкретных случаях функции $L(y)$ и $R(y)$, а также параметры α, β нечетких чисел (a, α, β) и $(a_1, a_2, \alpha, \beta)$ необходимо подбирать так, чтобы результат некоторой операции (сложение, вычитание, деление и т.п.) точно или приблизительно был равен нечеткому числу с теми же $L(y)$ и $R(y)$, а параметры α и β результата не покидали рамки ограничений на эти параметры для начальных нечетких чисел, тем более если результат в дальнейшем будет участвовать в операциях.

Замечание: Решение задач математического моделирования сложных систем, в которых применяется аппарат нечетких множеств, требует выполнения большого объема операций над различного рода лингвистическими и другими нечеткими переменными. Для удобства выполнения операций, а также для ввода-вывода и хранения данных, целесообразно работать с функциями принадлежности стандартного вида [2,21].

Нечеткие множества, которые применяются в большом количестве задач, являются, обычно, унимодальными и нормальными. Одним из возможных способов аппроксимации унимодальных нечетких множеств является аппроксимация с помощью функций (L-R)-типа. Примеры (L-R)-представлений представлены в таблице 1.2.1.

Таблица 1.2.1 - Возможное (L-R)-представление некоторых лингвистических переменных

Терм ЛП	(L-R)-представление	Графическое представление
Средний	$A = (a, \alpha, \beta)_{LR}$ $\alpha = \beta > 0$	$\alpha \quad \beta$
Малый	$A = (a, \infty, \beta)_{LR}$ $\alpha = \infty$	$\alpha = \infty \quad \beta$
Большой	$A = (a, \alpha, \infty)_{LR}$ $\beta = \infty$	$\alpha \quad \beta = \infty$
Приблизительно в диапазоне	$A = (a_1, a_2, \alpha, \beta)_{LR}$ $\alpha = \beta > 0$	$\alpha \quad \beta$ $a_1 \quad a_2$
Определенный	$A = (a, 0, 0)_{LR}$ $\alpha = \beta = 0$	$\alpha = 0 \quad \beta = 0$
Разнообразный: зона полной неопределенности	$A = (a, \infty, \infty)_{LR}$ $\alpha = \beta = \infty$	$\alpha = \beta = \infty$

1.3. Нечеткая логика и ее основные свойства

Опираясь на одно из самых распространенных определений, логика - это анализ методов рассуждений. В процессе изучения этих методов, логика интересуется в первую очередь формой, а не содержанием доводов в том или ином рассуждении. Ложность или истинность отдельных заключений или посылок не интересует логику. Ее интересует лишь, следует или нет достоверность заключения из достоверности посылок. Одной из основных задач логики является систематическая формализация и каталогизация верных методов рассуждений.

В логике из простых высказываний путем соединения их различными способами можно составить новые, более сложные высказывания.

Механизм нечетких выводов, который используется в различных экспертных и управляющих системах, в основе своей несет базу знаний, которая в свою очередь формируется специалистами и выглядит следующим образом:

П1: если x есть $A1$, тогда y есть $B1$,

П2: если x есть $A2$, тогда y есть $B2$,

.....

Пn: если x есть A_n , тогда y есть B_n ,

где x – переменная на входе (известные значения), y – переменная вывода (данные, которые будут вычисляться); A и B – функции принадлежности, которые определены на x и y .

Пример более подробного объяснения. Опыт эксперта $A \rightarrow B$ показывает нечеткое причинное отношение заключений. В следствии ранее сказанного назовем его нечетким отношением и обозначим через R :

$$R = A \rightarrow B,$$

где « $\rightarrow$ » называется нечеткой импликацией. $R = A \rightarrow B$ по другому можно назвать нечетким подмножеством прямого произведения $X * Y$ полного множества предпосылок X и заключений Y . Отсюда следует, что процедура нахождения нечеткого результата вывода B' , используя данное наблюдение A' и знания, $A \rightarrow B$ можно записать формулой:

$$B' = A' \circ R = A' \circ (A \rightarrow B),$$

где « $\circ$ » - описанная ранее операция свертки. Как процесс операции композиции, так и процесс операции импликации в алгебре нечетких множеств возможно реализовывать разными способами, при этом, естественно, будет отличаться и результат, но в итоге общий логический вывод будет осуществляться в четыре этапа [6,20].

1.Фазификация (или приведение к нечеткости). На данном этапе используют функции принадлежности для определения степени истинности каждого правила.

2. Логический вывод. Полученную степень истинности для каждого утверждения нужно применить к заключениям этих утверждений. В результате перейдем к нечеткому подмножеству, которое определяется для каждого правила каждой переменной вывода. В качестве правил логического вывода используют операции $\min$ (МИНИМУМ) или prod (УМНОЖЕНИЕ). В логическом выводе МИНИМУМА функция принадлежности вывода «отсекается» по высоте, которая соответствует вычисленной степени истинности предпосылки правила (нечеткая логика «И»). В логическом выводе УМНОЖЕНИЯ функция принадлежности вывода масштабируется при помощи вычисленной степени истинности предпосылки правила.

3. Композиция. Все нечеткие подмножества, которые были определены для каждой переменной вывода для всех правил, объединяются и формируют одно нечеткое подмножество для каждой переменной вывода. При использовании данного объединения обычно пользуются операцией $\max$ (МАКСИМУМ) или sum (СУММА). При композиции МАКСИМУМА комбинированный вывод нечеткого подмножества конструируется как поточечный максимум по всем нечетким подмножествам (нечеткая логика «ИЛИ»). При композиции СУММЫ комбинированный вывод нечеткого подмножества строится как поточечная сумма по всем нечетким подмножествам, назначенным переменной вывода правилами логического вывода.

4. Дефазификация (или приведение к четкости) используется в случае, если необходимо преобразовать нечеткий набор выводов в четкое число. Имеется множество способов приведения к четкости, несколько из них будут рассмотрены ниже.

Пример. Пусть определены нечеткие правила:

П1: если x есть A, тогда ϖ есть D,

П2: если y есть B, тогда ϖ есть E,

П3: если z есть C, тогда ϖ есть F,

где x , y и z – входные переменные, ϖ - переменная вывода, а A, B, C, D, E, F - заданные функции принадлежности (треугольной формы).

Процедура получения логического вывода показана на рисунке 1.3.1.

Можно предположить, что переменные на входе принимают некоторые конкретные (четкие) значения - x_0 , y_0 и z_0 .

Согласно приведенным этапам, на первом этапе находятся степени истинности $\alpha(x_0)$, $\alpha(y_0)$ и $\alpha(z_0)$ для всех приведенных правил (см. рисунок 1.3.1).

На втором этапе происходит «отсекание» функций принадлежности заключений на уровнях $\alpha(x_0)$, $\alpha(y_0)$ и $\alpha(z_0)$.

На третьем этапе оперируем уже усеченными функциями принадлежности, после чего они объединяются, используя операцию $\max$. В итоге получим комбинированное нечеткое подмножество, которое описывается ФП $\mu_c(\varpi)$. Это и будет логический вывод ϖ .

На четвертом этапе находим четкое значение выходной переменной, например, воспользовавшись любым из рассмотренных ранее методов дефазификации.

Рассмотрим наиболее часто используемые алгоритмы нечеткого вывода, применив базу знаний в виде двух правил::

П1: если x есть A1 и y есть B1, тогда z есть C1,

П2: если x есть A2 и y есть B2, тогда z есть C2,

где x и y – имена входных переменных, z – имя переменной вывода, A1, A2, B1, B2, C1, C2 – некоторые заданные функции принадлежности, при этом четкое значение z_0 необходимо определить основываясь приведенной информацией и четких значений x_0 и y_0 .

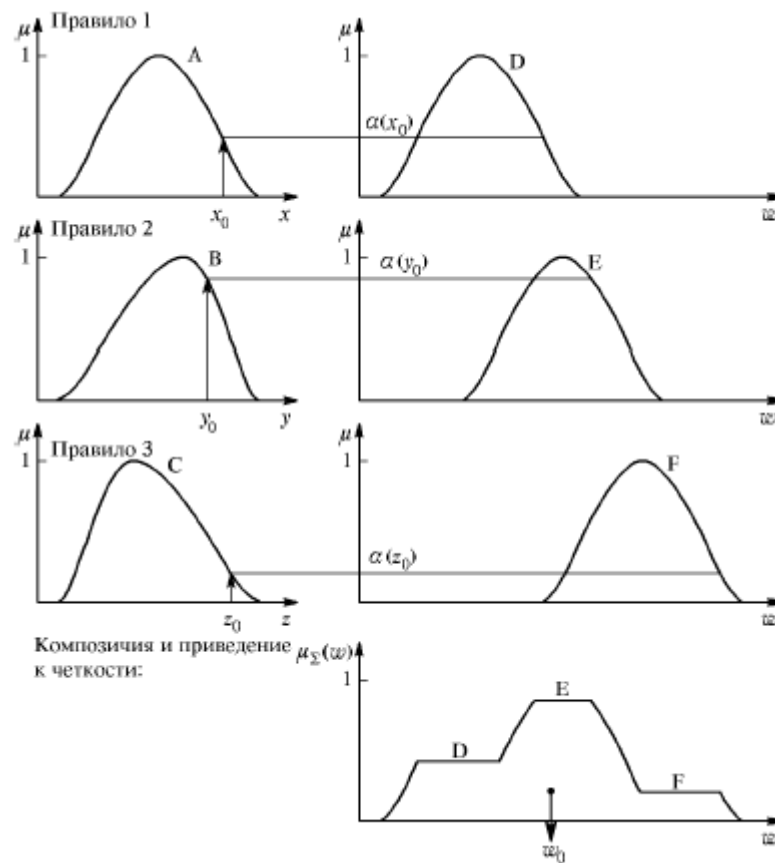


Рисунок 1.3.1 – Иллюстрация к процедуре логического вывода

Алгоритм Mamdani: Данный алгоритм соответствует рассмотренному примеру и рисунку 1.1.3. В ситуации, которую мы рассматриваем, запишем математически:

1. Определяем степени истинности для каждого правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(y_0)$, $B_2(y_0)$

2. Отсекаем функции принадлежности для каждого правила с при помощи операции МИНИМУМ

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

где через обозначение « $\wedge$ » показана операция логического минимума, далее определяются усеченные функции принадлежности:

$$C'_1(z) = (\alpha_1 \wedge C_1(z)),$$

$$C'_2(z) = (\alpha_2 \wedge C_2(z)).$$

3. Используя операцию МАКСИМУМ (обозначаемой « $\vee$ ») объединяем найденные выше усеченные функции. В итоге получили нечеткое подмножество для переменной выхода. Ее функция принадлежности выглядит следующим образом:

$$\mu_{\Sigma}(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

4. Нахождение четкого значения z_0 используя, к примеру, центроидным методом [6,19].

Алгоритм Tsukamoto. Исходные посылки соответствуют предыдущему алгоритму, но в конкретном примере строится предположение, что функции $C_1(z_1)$ и $C_2(z_2)$ являются монотонными.

1. Первый этап точно такой же, как в предыдущем алгоритме.

2. На этапе логического вывода сначала находятся уровни «отсечения» α_1 и α_2 , а затем – посредством решения уравнений:

$$\alpha_1 = C_1(z), \quad \alpha_2 = C_2(z),$$

находятся четкие значения (z_1 и z_2) для каждого из исходных правил.

3. затем находится четкое значение переменной вывода как взвешенное среднее значений z_1 и z_2 :

$$z_0 = \frac{\alpha_1 \cdot z_1 + \alpha_2 \cdot z_2}{\alpha_1 + \alpha_2}.$$

Общий случай (дискретный центроидный вывод).

$$z_0 = \frac{\sum_{i=1}^n \alpha_i \cdot z_i}{\sum_{i=1}^n \alpha_i}.$$

Пример: Пусть имеем $A_1(x_0)=0,7$, $A_2(x_0)=0,6$, $B_1(y_0)=0,3$, $B_2(y_0)=0,8$, соответствующие уровни отсечения

$$\alpha_1 = \min(A_1(x_0), B_1(y_0)) = \min(0,7; 0,3) = 0,3.$$

$$\alpha_2 = \min(A_2(x_0), B_2(y_0)) = \min(0,6; 0,8) = 0,6.$$

и значения $z_1=8$ и $z_2=4$, найденные в результате решения уравнений:

$$C_1(z_1) = 0,3, \quad C_2(z_2) = 0,6.$$

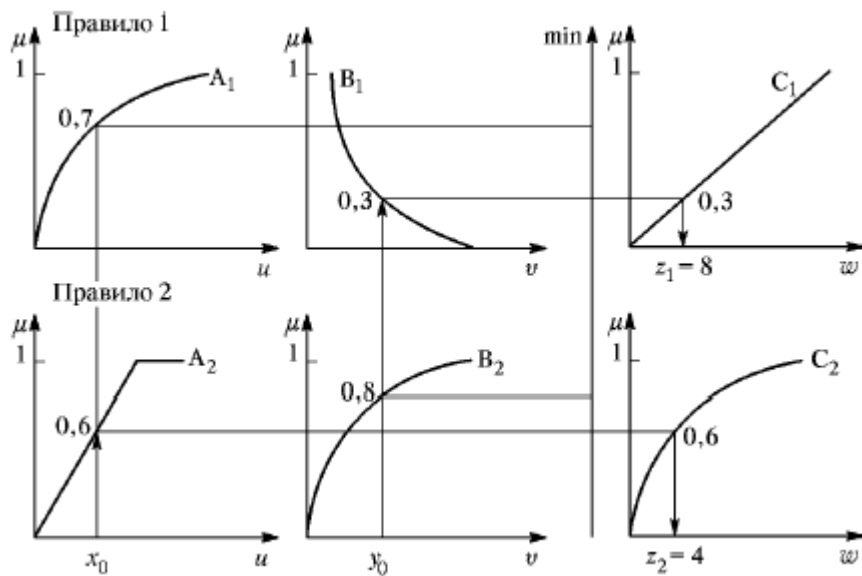


Рисунок 1.3.2 – Иллюстрации к алгоритму Tsukamoto

При этом четкое значение переменной вывода изображено на рисунке 1.3.2.

$$z_0(8 \cdot 0,3 + 4 \cdot 0,6) / (0,3 + 0,6) = 6.$$

Алгоритм Sugeno. Sugeno и Takagi применяли набор правил следующей формы:

П1: если x есть A_1 и y есть B_1 , тогда $z_1 = a_1 \cdot x + b_1 \cdot y$,

П2 если x есть A_2 и y есть B_2 , тогда $z_2 = a_2 \cdot x + b_2 \cdot y$.

Представление алгоритма:

1. Первый этап соответствует алгоритму Mamdani.

2. На втором этапе отыскиваются $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$

и частные выходы правил:

$$z_1^* = a_1 \cdot x_0 + b_1 \cdot y_0,$$

$$z_2^* = a_2 \cdot x_0 + b_2 \cdot y_0.$$

3. Далее находится четкое значение:

$$z_0 = \frac{\alpha_1 \cdot z_1^* + \alpha_2 \cdot z_2^*}{\alpha_1 + \alpha_2}.$$

На рисуне 1.3.3 показаны иллюстрации алгоритма Sugeno

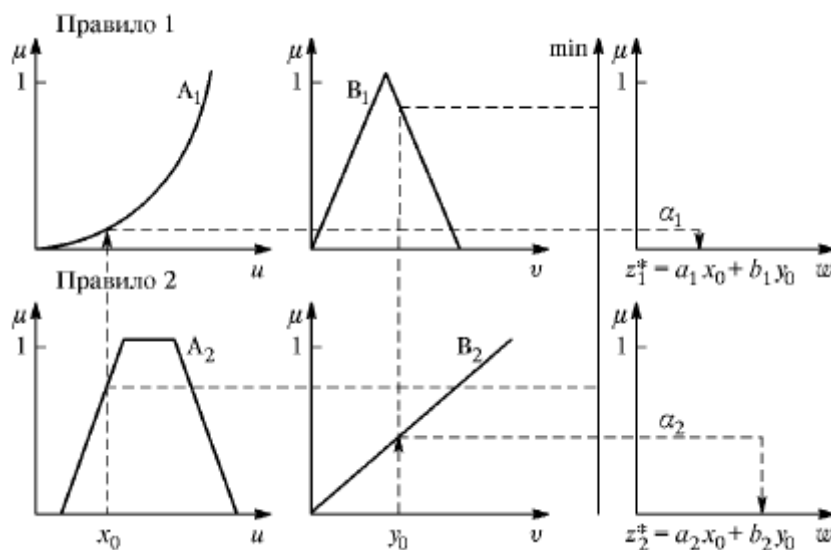


Рисунок 1.3.3 –алгоритм Sugeno

Алгоритм Larsen. В алгоритме Larsen нечеткая операция моделируется с применением оператора умножения.

Описание алгоритма:

1. Первый этап точно такой же, как в первом рассматриваемом алгоритме.
2. На тапе логического вывода вначале находим значения

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

затем находятся частные нечеткие подмножества

$$\alpha_1 C_1(z), \quad \alpha_2 C_2(z).$$

3. Далее происходит нахождение итогового нечеткого подмножества с функцией принадлежности:

$$\mu_{\Sigma}(z) = C(z) = (\alpha_1 C_1(z)) \vee (\alpha_2 C_2(z)),$$

в общем случае n правил $\mu_{\Sigma}(z) = C(z) = \bigcup_{i=1}^n (\alpha_i C_i(z)).$

4. В случае, если необходимо, осуществляется приведение к четкости (точно так же, как в алгоритмах, рассмотренных ранее).

Алгоритм Larsen иллюстрируется рисунком 1.3.4.

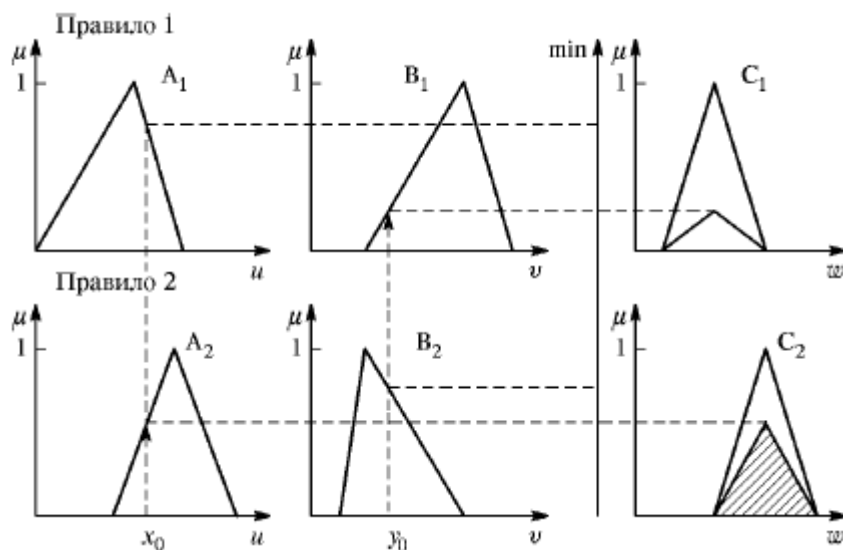


Рисунок 1.3.4 – Алгоритм Larsen

Упрощенный алгоритм нечеткого вывода. Правила в данном случае имеют вид:

П1: если x есть A_1 и y есть B_1 , тогда $z_1 = c_1$,

П2: если x есть A_2 и y есть B_2 , тогда $z_2 = c_2$,

где c_1 и c_2 - обычные (четкие) числа.

Описание алгоритма:

1. Первый этап точно такой же, как алгоритм Mamdani.

2. На втором этапе находятся числа:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0).$$

3. На третьем этапе осуществляется поиск четкого значения выходной переменной используя формулу:

$$z_0 = \frac{\alpha_1 \cdot c_1 + \alpha_2 \cdot c_2}{\alpha_1 + \alpha_2}.$$

Или, в общем случае наличия n правил, по формуле:

$$z_0 = \frac{\sum_{i=1}^n \alpha_i \cdot c_i}{\sum_{i=1}^n \alpha_i}.$$

Иллюстрация алгоритма приведена на рисунке 1.3.5

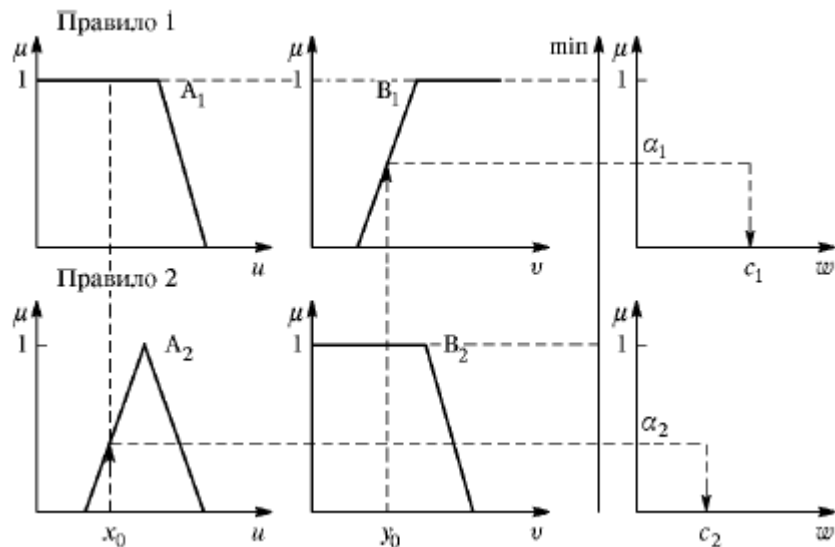


Рис. 1.3.5 – Иллюстрация упрощенного алгоритма нечеткого вывода

1.4 Методы фазификации и дефазификации

Упоминая нечеткую логику, зачастую подразумевают системы нечеткого вывода, которые в свою очередь имеют обширное применение в области управления техническими устройствами и процессами. Разработка и применение систем с нечетким выводом заключается в выполнении ряда этапов, реализация которых осуществляется при помощи основных положений нечеткой логики.

В качестве информации, поступающей на вход, являются измеренные различными способами входные переменные. Они соответствуют реальным переменным процесса управления. Формируемая информация на выходе нечеткого вывода соответствует выходным переменным, являющимися управляющими переменными процесса управления.

Фазификация. Под фазификацией принято понимать процесс поиска значений функций принадлежности (термов), основываясь на обычных

(четких) входных значениях. Фазификацию по другому можно назвать введением нечеткости.

Целью фазификации можно считать поиск соответствия между конкретным значением отдельной входной переменной системы нечеткого вывода и значением функции принадлежности соответствующего ей терма входной лингвистической переменной. В результате завершения этапа фазификации всем входным переменным должны соответствовать индивидуальные значения функций принадлежности по каждому из лингвистических термов, которые были использованы в подусловиях базы правил системы нечеткого вывода.

Формально процедура фазификации проходит в определенной последовательности. Перед тем, как начать этап фазификации, предполагается, что конкретные значения всех входных переменных известны, то есть $V' = \{a_1, a_2, \dots, a_n\}$. Обобщенно каждое $a_i \in X_i$, где X_i - универсум лингвистической переменной β_i . Далее отдельно анализируется каждое из подусловий вида " β_i есть a " правил системы нечеткого вывода, где a' - некоторый терм с функцией принадлежности $\mu(x)$. При этом значение a' используется как аргумент $\mu(x)$, тем самым находится количественное значение $\beta'_i = \mu(a_i)$. Данное значение и есть результат фазификации подусловия " β_i есть a ".

Этап фазификации можно считать законченным тогда, когда для каждого из подусловий всех правил значения $\beta'_i = \mu(a_i)$ будут найдены. Это множество значений можно обозначить через $B = \{\beta'_i\}$. Если некоторый терм a' лингвистической переменной β_i отсутствует во всех нечетких высказываниях, то значение функции принадлежности, которое ему соответствует, не находится в процессе фазификации.

Пример: В качестве иллюстрации рассмотрим пример процесса фазификации трех высказываний: «Расход цемента малый», «Расход цемента средний», «Расход цемента высокий» для входной лингвистической

переменной β_i - Расход цемента на кубический метр раствора. Им соответствуют нечеткие высказывания первого вида: " β_1 есть a_1 ", " β_1 есть a_2 ", " β_1 есть a_3 ". Предположим, на данный момент расход цемента равен 55 кг, то есть $a_1 = 55 \text{ кг}$.

Тогда фазификация первого нечеткого высказывания дает в результате число 0, означающее степень его истинности и получается в результате подстановки значения $a_1 = 55 \text{ кг}$ как аргумент функции принадлежности терма a_1 (рисунок 1.4.1 А). Фазификация второго высказывания даст в результате число приблизительно равное 0,67, которое означает степень его истинности и получается в результате подстановки значения $a_1 = 55 \text{ кг}$ как аргумент функции принадлежности терма a_2 (рисунок 1.4.1 Б). Фазификация третьего высказывания, так же как и в первом случае даст число 0, и будет означать его степень истинности (рисунок 1.4.1 В).

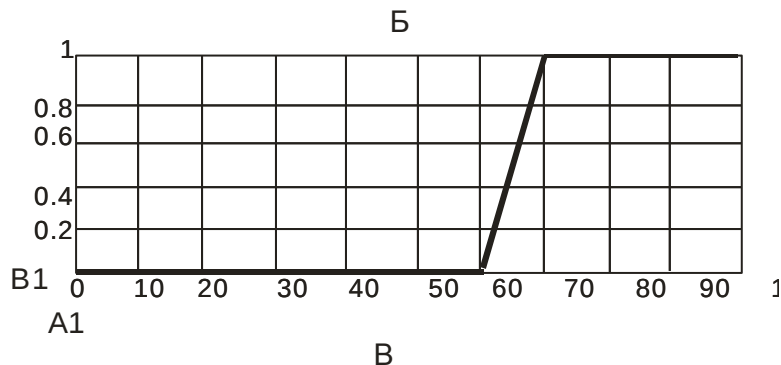
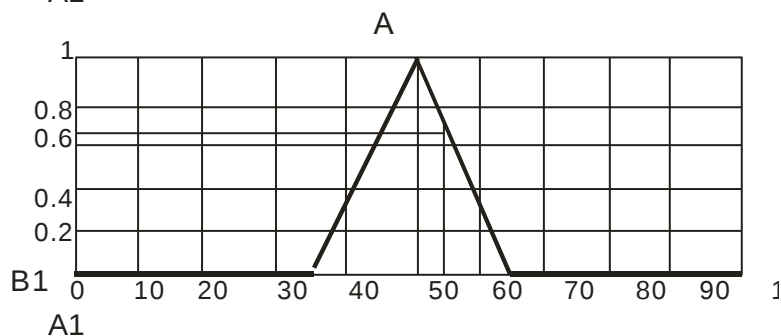
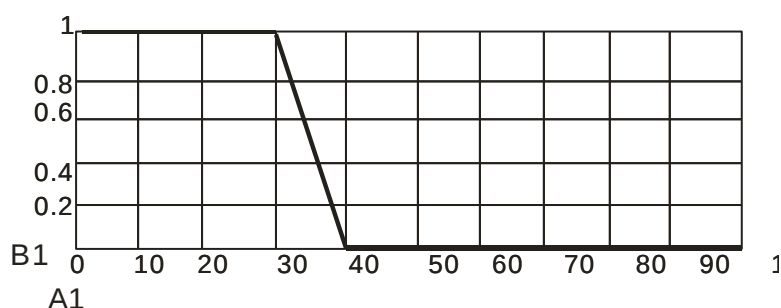


Рисунок 1.4.1 – Пример фазификации входной лингвистической переменной “Расход цемента” для трех нечетких высказываний

Дефазификация. Под дефазификацией понимается процедура поиска четкого значения для каждой из входных лингвистических переменных множества $W = \{w_1, w_2, \dots, w_s\}$.

Цель дефазификации заключается в следующем: используя результаты накопления всех выходных лингвистических переменных, необходимо найти четкое число для каждой из выходных переменных. По другому этап дефазификации можно называть приведением к четкости.

Процедура дефазификации. функции принадлежности всех выходных лингвистических переменных, до того как начать этап дефазификации, следует принять известными и записать в форме нечетких множеств:

$$C'_1, C'_2, \dots, C'_s,$$

где s – общее количество выходных лингвистических переменных согласно базы правил. Далее постепенно анализируем каждую выходную лингвистическую переменную $w_j \in W$, к которой относится нечеткое множество C'_j . Результком этапа дефазификации для w_j следует считать количественное значение $y_j \in R$, применив любой из методов.

Этап дефазификации можно считать законченным, когда каждой выходной лингвистической переменной найдено определенное значения в численном виде

$$y_1, y_2, \dots, y_s,$$

где s – количество лингвистических переменных согласно базы правил.

Для нахождения численных значений используют ниже описанные формулы (методы дефазификации).

Метод центра тяжести

Центр тяжести (CoG, COG, Center of Gravity) или центроид площади рассчитывается по формуле:

$$y = \frac{\int_{Min}^{Max} x \cdot \mu(x) dx}{\int_{Min}^{Max} \mu(x) dx} ,$$

где y является результатом дефазификации; x – соответствующее выходной лингвистической переменной w_s значение переменной; $\mu(x)$ – соответствующая выходной переменной w после этапа накопления функция принадлежности; Min и Max – левая и правая точки интервала носителя.

В данном методе обычное (четкое) значение выходной переменной равно абсциссе центра тяжести площади, ограниченной графиком кривой функции принадлежности, которая соответствует входной переменной.

Пример дефазификации ФП выходной лингвистической переменной «Расход цемента» выше описанным методом изображен на рисунке 1.4.2 В этом случае $y_1 = 40 \text{ кг}$ (приближенное значение).

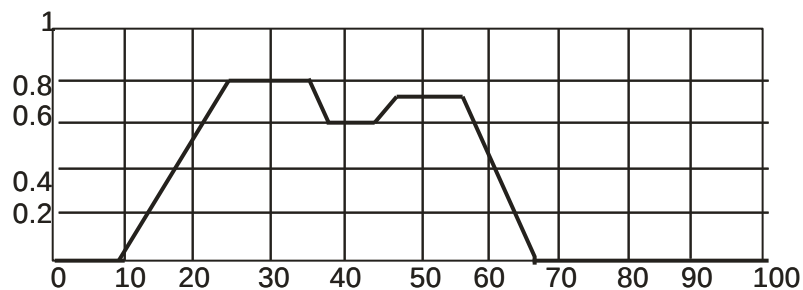


Рисунок 1.4.2 – Дефазификация методом центра тяжести

Метод центра тяжести для одноточечных множеств.

Центр тяжести для одноточечных множеств (COGS, Centre of Gravity for Singletons) можно рассчитать по формуле:

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} ,$$

где n – число однотоочечных (одноэлементных) нечетких множеств, каждое из которых характеризует единственное значение рассматриваемой выходной лингвистической переменной.

Пример дефазификации данным методом выходной лингвистической переменной «Расход цемента» изображен на рисунке 1.4.3. В этом случае $y_1=41$ кг (приближенное значение).

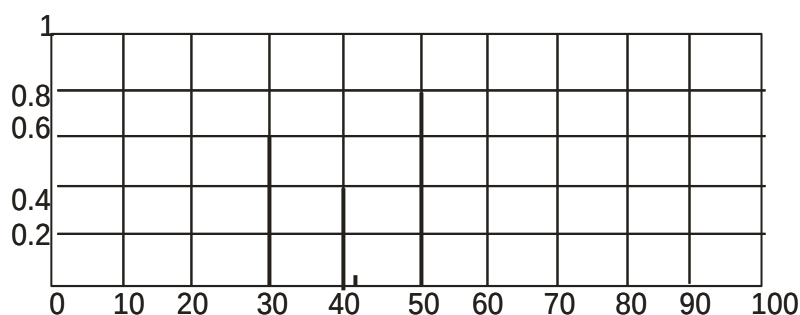


Рисунок 1.4.3 – Пример дефазификации выходной лингвистической переменной «Расход цемента» методом центра тяжести для однотоочечных множеств

Метод центра площади

Центр площади (CoA, COA, Centre of Area, Bisector of Area)

$y=u$, u определяется уравнением:

$$\int_{\min}^u \mu(x) dx = \int_u^{\max} \mu(x) dx$$

Пример дефазификации методом центра площади функции принадлежности выходной лингвистической переменной «Расход цемента» изображен на рисунке 1.4.4. В этом случае $y_1=35$ кг

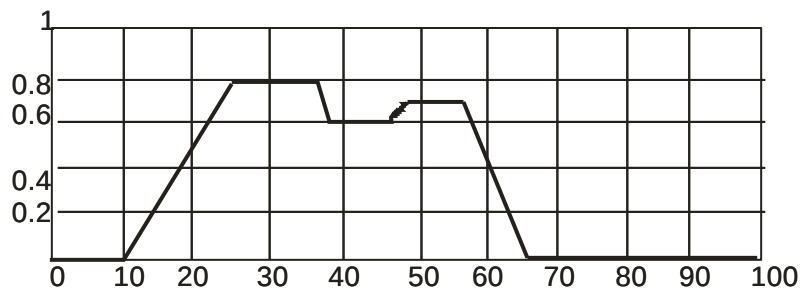


Рис. 1.4.4 – Дефазификация выходной лингвистической переменной «Расход цемента» методом центра площади

Иначе говоря центр площади равняется абсциссе, которая делит площадь, ограниченную графиком кривой функции принадлежности соответствующей выходной переменной, на две равные части.

Метод левого модального значения

Левое модальное значение (LM, Left Most Maximum) можно рассчитать по формуле:

$$y = \min\{x_m\},$$

где x_m - модальное значение (мода) нечеткого множества, соответствующего выходной переменной w после аккумуляции.

Другими словами, значение выходной переменной определяется как мода нечеткого множества для соответствующей выходной переменной или наименьшая из мод (самая левая), если нечеткое множество имеет несколько модальных значений.

Пример дефазификации методом левого модального значения функции принадлежности выходной лингвистической переменной «Расход цемента» изображен на рисунке 1.4.5. В этом случае $y_1=24$ кг.

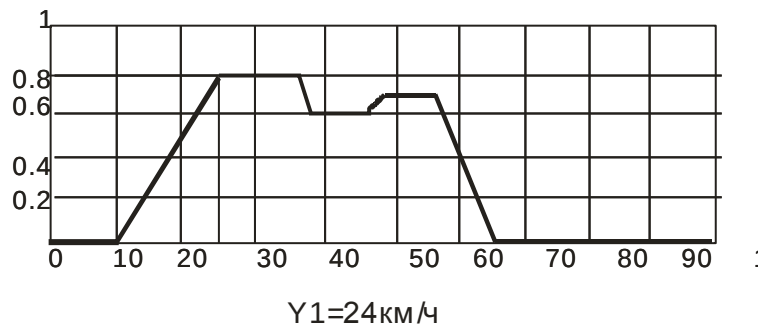


Рисунок 1.4.5 – Пример дефазификации выходной лингвистической переменной «Расход цемента» методом левого модального значения

Метод правого модального значения

Правое модальное значение (RM, Right Most Maximum) рассчитывается по формуле:

$$y = \max\{x_m\},$$

где x_m - модальное значение (мода) нечеткого множества, соответствующего выходной переменной w после аккумуляции.

В этом случае значение выходной переменной также определяется как мода нечеткого множества для соответствующей выходной переменной или наибольшая из мод (самая правая), если нечеткое множество имеет несколько модальных значений.

Пример дефазификации методом правого модального значения функции принадлежности выходной лингвистической переменной «Расход цемента» изображен на рисунке 1.4.6. В этом случае $y_1=54$ кг (приближенное значение).

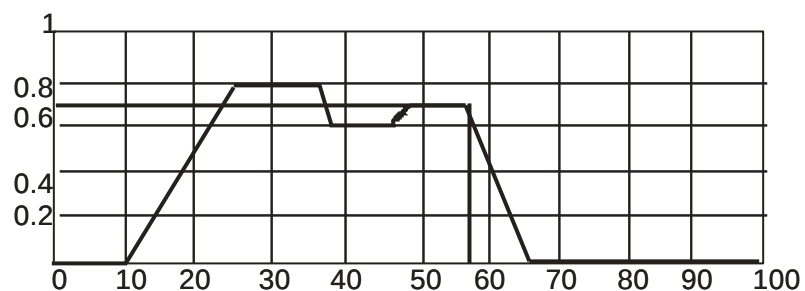


Рис. 1.4.6 – Пример дефазификации выходной лингвистической переменной «Расход цемента» методом правого модального значения

1.5 Структуры систем автоматического регулирования с нечеткими регуляторами

Нечеткий регулятор, функционирующий по законам нечеткой логики, извне не отличается от классического. Отсюда следует, что структуру нечетких систем управления так же строят на основе классического.

На рисунке 1.5.1 изображена самая простая одноконтурная системк управления

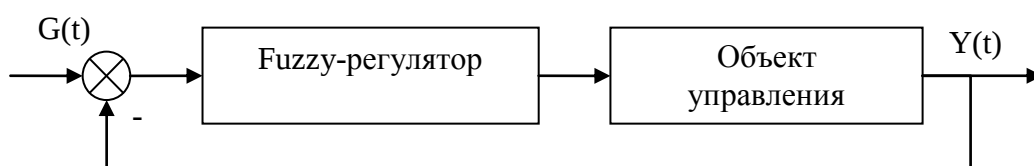


Рисунок 1.5.1– Одноконтурная система управления

Однако, так как сам по себе нечеткий регулятор не обладает внутренней динамикой, мы не можем, опираясь на данную структуру, обеспечить динамику нечеткого регулятора. Если поставить задачу обеспечить динамикой нечеткий регулятор (например, взяв за образец ПИД-регулятор), вопрос решается введением в структуру управления блока подготовки параметров контроля (рисунок 1.5.2).

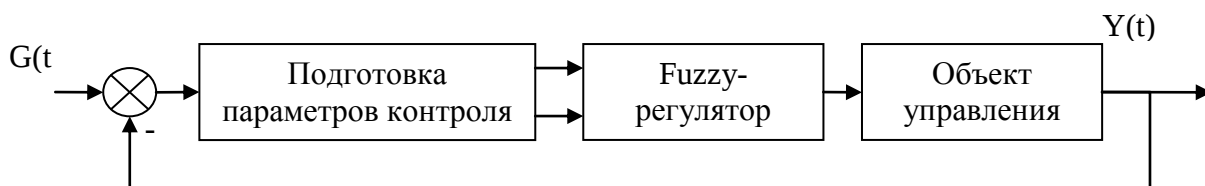


Рис. 1.5.2 – Одноконтурная система управления с блоком подготовки контролируемых параметров

Введенный блок необходим для расчета требуемых параметров на относительно ошибки регулирования и является самостоятельным модулем.

Но многоканальная структура нечеткого управления в большей степени влияет на динамику процесса управления. Такая система представлена на рисунке 1.5.3.

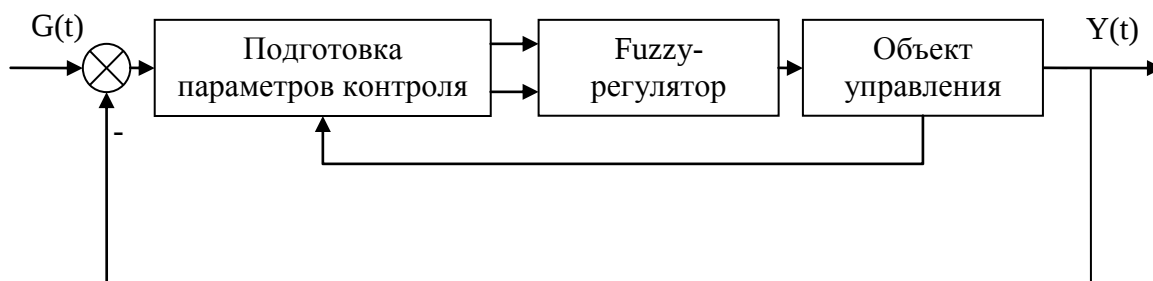


Рисунок 1.5.3 – Нечеткая система управления с многоканальной системой контроля состояния объекта управления

Сравнив с классическим регулятором состояния, можно увидеть, что регулятор получает информацию не только о регулируемой величине, но и переменных, которые в какой то мере характеризуют систему.

Широкие возможности применения нечеткого управления раскрываются при адаптивном регулировании. Наблюдаем улучшение качества регулирования в режиме on-line. Здесь создаются самоорганизующиеся нечеткие контроллеры, которые модифицируют параметры регулятора и оптимально настраиваются на процесс управления. Простейшая форма «умного» нечеткого контроллера изображена на рисунке 1.5.4

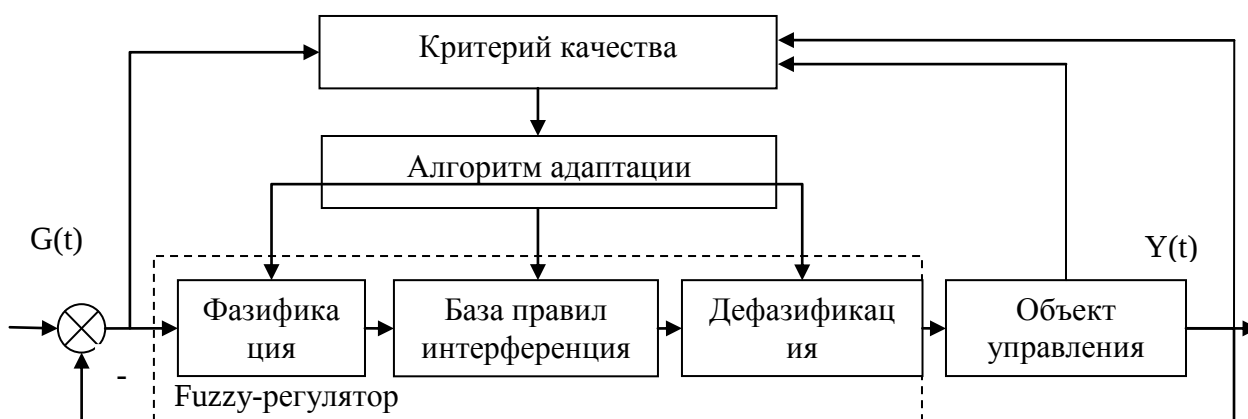


Рисунок 1.5.4 – Самоорганизующиеся fuzzy-контроллеры

В основном адаптация нечеткого контроллера происходит в ходе наблюдения за ошибкой объекта управления или его выходной величиной, и исходя из полученных данных изменяются показатели качества. Сам механизм

настройки параметров расположен в блоке алгоритма адаптации. В нем содержится «интеллект» адаптивного регулятора в виде команд уставок параметров, которые зависят от показателей качества.

Точками приложения алгоритма адаптации могут быть различные компоненты нечеткого регулятора. В простейшем случае дело может ограничиться изменением масштаба области значений входных величин для достижения переключения между грубой и точной настройкой (возможно с промежуточными ступенями). Более сложные алгоритмы могут вовлекать в процесс адаптации сами функции принадлежности или даже базу правил, модифицируя в зависимости от состояния объекта управления форму нечетких множеств, отдельные правила или переключаясь между разными множествами правил. Допустим выбор различных операторов логических связей, схем нечеткого логического вывода или методов дефазификации.

Рассмотренная схема является схемой косвенной адаптации через критерий качества.

Принципиально возможна также схема прямой адаптации посредством on-line оценки параметров объекта управления (рисунок 1.5.5). Однако здесь предпосылкой является наличие модели процесса в классическом варианте или в виде fuzzy-модели, то есть в форме правил типа ЕСЛИ ... ТО ... или реляционных матриц.

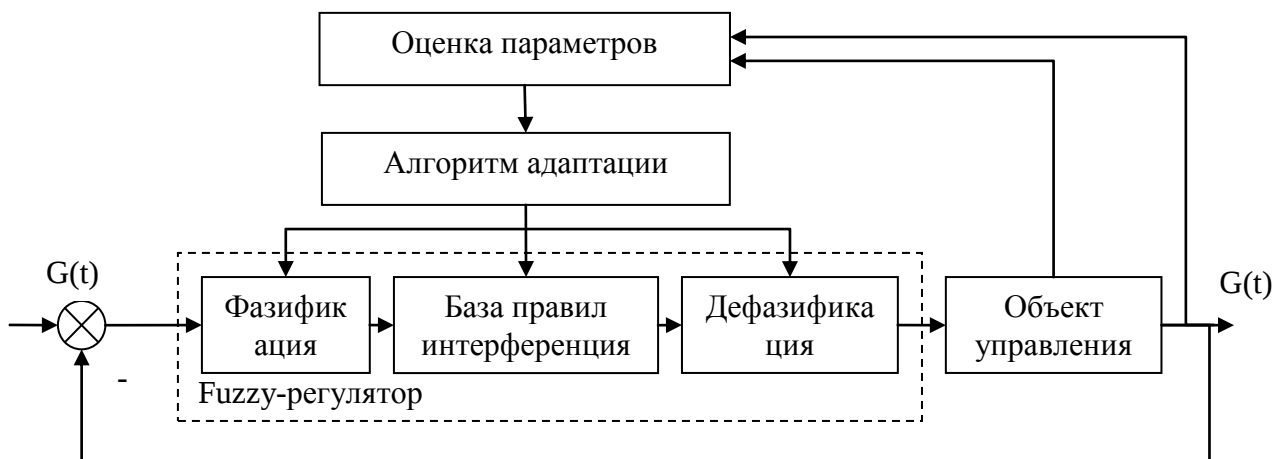


Рисунок 1.5.5 – Схема прямой адаптации посредством on-line оценки параметров объекта управления

Развитием этого подхода является использование fuzzy-концепции не только для управления, но и для идентификации, чтобы реализовать предикативную концепцию управления, основываясь на fuzzy-модели объекта управления (рисунок 1.5.6). Эта идея была использована в системе управления японским метро в Сендаи (Япония). При этом подходе простой нечеткий контроллер превращается в контроллер, базирующийся на модели, состоящей из трех компонентов:

- 1) Fuzzy-модели процесса (например, в форме реляционной матрицы), которая идентифицируется посредством измерения Входной и выходной величины в процессе работы одновременно с регулированием. Разговор идет о самоорганизующемся регуляторе, так как эта модель содержит базу правил контроллера.

- 2) Критерия/ев качества, конкретные fuzzy-значения которого/ых заданы функциями принадлежности. Если должны оптимизироваться несколько критериев качества, то соответствующие значения критериев качества можно привести к скалярному общему показателю качества с помощью min-оператора.

- 3) Выбора управляющих воздействий из конечного множества альтернатив. При этом выбираются те, которые, базируясь на оценке по модели процесса, обещают экстремальное значение критерия качества.

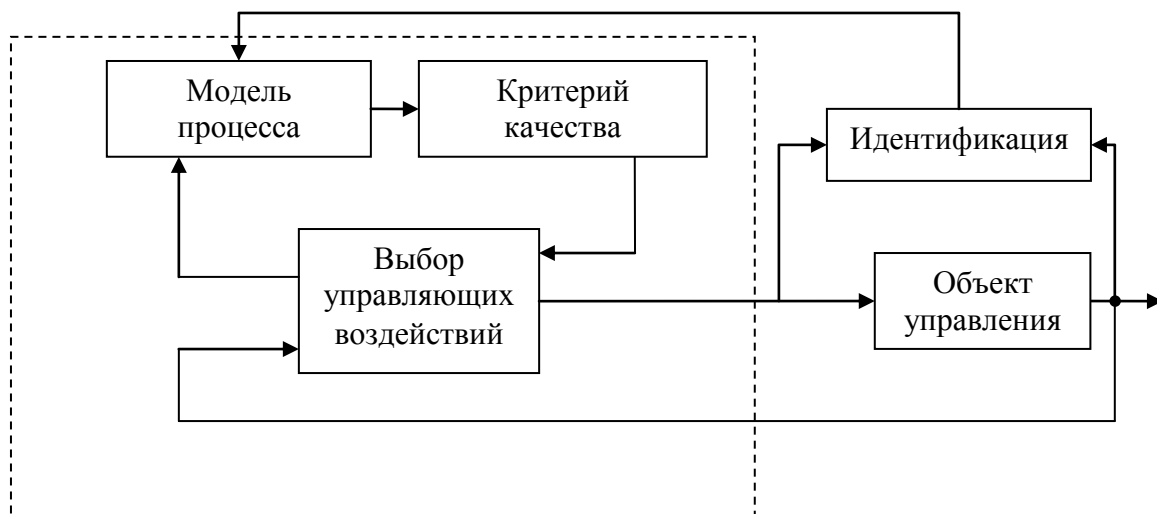


Рисунок 1.5.6 – Fuzzy-контроллер на базе модели

Более подробное рассмотрение такого процесса позволяет указать на ряд проблем:

1) В начале процесса регулирования не имеется никакой информации о модели процесса. Поэтому идентификация стартует с пустой реляционной матрицей. В качестве альтернативы возможен старт с начальной моделью, определенной в режиме on-line;

2) В начальной фазе процесса регулирования может возникнуть ситуация, при которой невозможна оценка последствий выбора альтернатив управляющего воздействия на основе неполной модели процесса. Для этого случая должен быть предусмотрен аварийный регулятор, который как минимум обеспечивает устойчивость контура управления, пока модель процесса не станет полной. В простейшем случае регулятор будет сохранять значения, определенные на предыдущем шаге.

Обобщенная структура нечеткого регулятора имеет вид, представленный на рисунке 1.5.7.

Создавая нечеткий регулятор предполагают, что эксперты умеет формировать базу правил вида:

ЕСЛИ <предпосылка>

ТО <вывод>

и базу данных с функциями принадлежности для $\mu(e)$ и выводов $\mu(u)$, то есть определить необходимые правила с лингвистическими переменными и термами.

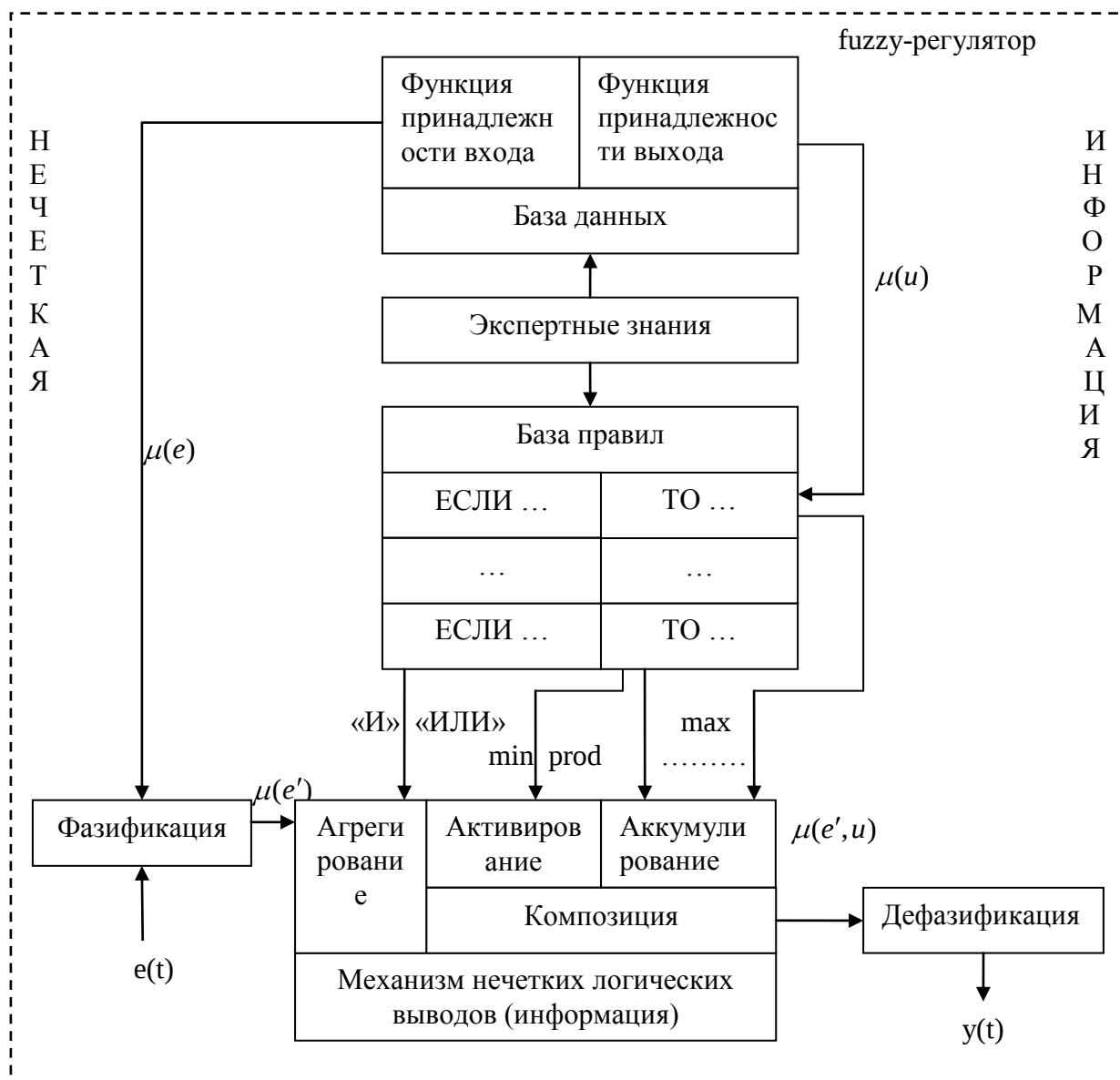


Рисунок 1.5.7 – Структура нечеткого регулятора

Для того, чтобы решить задачу принятия решения необходим механизм нечетких логических выводов.

1. Оценка предпосылок, то есть установление значения истинности ЕСЛИ-части каждого правила, представляющей собой совокупность связанных между собой нечеткими операторами «И», «ИЛИ» нечетких

логических высказываний. Определение значения истинности (степени выполнения) предпосылки называется агрегированием;

2. Оценка степени истинности ТО-части каждого правила на основе оценки предпосылки, то есть степени активирования каждого правила. Из анализа свойств композиции следует, что степень истинности заключения не может превосходить степень истинности предпосылки.

3. Обобщение частичных решений каждого активированного правила базы правил и формирование результирующего нечеткого логического вывода в форме функции принадлежности выходной лингвистической переменной, соответствующей текущему входному сигналу. Определение итоговой функции принадлежности выхода через полученные частные функции принадлежности ТО-частей активированных правил называется аккумулярованием.

Шаги 2 и 3 механизма нечетких логических выводов часто называют просто композицией.

Структура обработки нечеткой информации с вышеописанными составными частями уже является в принципе работоспособной (например, в нечетких экспертных системах). Однако в области автоматизации и управления для использования лингвистических правил в качестве модели принятия решений необходимы также модули преобразования как четкой (количественной) измерительной информации в нечеткую (качественную) (модуль фазификации), так и нечеткой (качественной) информации в четкую (количественную) (модуль дефазификации).

В данной работе использовался метод дефазификации с треугольными тегами, число которых равно трем в зависимости от величины ошибки: «Ошибка малая», «Ошибка средняя», «Ошибка большая», в соответствии с этим получалось значение лингвистической переменной u , с помощью которого в результате определяется значение коэффициентов K_p, K_d, T_i дефазификацией методом максимума. Этот метод был выбран так как он

наиболее прост в реализации, а, следовательно, и более доступен для понимания его студентами.

В данной работе программная реализация нечеткого регулятора осуществлена на базе промышленного микропроцессорного контроллера КРОСС. Ниже приведен состав и технические характеристики этого контроллера, а также приводится описание лабораторного стенда, на котором выполняются лабораторные работы.

2 Архитектура, технические характеристики и функциональные возможности контроллера КРОСС

2.1 Назначение, состав и архитектура контроллера

Основное назначение контроллера КРОСС 500 – построение эффективных систем автоматизации различных технологических объектов.

Контроллер КРОСС 500 является модернизацией и дальнейшим развитием контроллера для распределенных систем КРОСС.

КРОСС 500 предназначен для решения следующих типовых задач автоматизации:

- Сбор информации с датчиков различных типов и ее первичная обработка (фильтрация сигналов, линеаризация характеристик датчиков, «офизичивание» сигналов и т.п.).
- Выдача управляющих воздействий на исполнительные органы различных типов.
- Регулирование прямых и косвенных параметров по различным законам;
- Логическое, программно-логическое управление технологическими агрегатами, автоматический пуск и останов технологического оборудования.
- Математическая обработка информации по различным алгоритмам.
- Регистрация и архивация технологических параметров.
- Обмен данными с другими контроллерами в рамках контроллерной управляющей сети реального времени.
- Аварийная, предупредительная и рабочая сигнализация, индикация значений прямых и косвенных параметров.
- Обслуживание устройств верхнего уровня, прием и исполнение их команд; выдача значений параметров и различных сообщений на панель оператора.

- Обслуживание технического персонала при наладке, программировании, ремонте, проверке технического состояния контроллера.
- Самоконтроль и диагностика всех устройств контроллера в непрерывном и периодическом режимах, вывод информации о техническом состоянии контроллера обслуживающему персоналу.

Управление объектом осуществляется с помощью технологической программы пользователя, созданной пользователем на технологических языках программирования системы ISaGRAF.

Контроллер является проектно-компонуемым и программируемым изделием. Его состав и ряд параметров определяются потребителем и указываются в заказе.

Контроллер имеет сертификат типа средства измерений и может выполнять учет материальных и энергетических потоков (электроэнергия, тепло) на различных участках производства.

Контроллер КРОСС-500 является проектно-компонуемым изделием, состав которого определяет пользователь в зависимости от решаемых задач. Компоновка контроллера осуществляется блоками, модулями и другими изделиями, входящими в его состав.

В состав контроллера входят следующие устройства:

1. Блок центрального процессора БЦП.
2. Модули ввода-вывода аналоговых сигналов постоянного состава: AI1-8, AIO1-8/4, AIO1-8/0, AIO1-0/4, TC1-7, TR1-8.
3. Модули ввода-вывода дискретных сигналов постоянного состава: DI1-16, DIO1-8/8, DO1-16.
4. Модули ввода-вывода, имеющие проектно-компонуемый состав:
модуль ADIO1 для ввода-вывода аналоговых, частотных и дискретных сигналов с повышенным быстродействием;
модуль AIO2 для ввода-вывода аналоговых и частотных сигналов с повышенным быстродействием.
5. Микроконтроллер программируемый МК1.

6. Терминальные блоки:

- блок T1-AI для модуля AI1-8;
- блок T1-TC для модуля TC1-7;
- блок T1-TR для модуля TR1-8;
- блок T1-AIO для модулей AIO1-8/4, AIO1-8/4, AIO1-8/4;
- блок T2-A для модулей ADI O1, AI O2, MK1 (разъем «INOUT-A»);
- блок T1-DI для модуля DI1-16;
- блок T1-DIO для модуля DIO1-8/8 (разъем «INOUT»); ADI O1, MK1 (разъем «INOUT-D»);
- блок T1-DO для модуля DO1-16;
- - блоки T1-DI-8, T1-DI-8/220, T1-DI-8/110, T1-DI-8/24, T2-DI-8/220, T2-DI-8/110, T2-DI-8/24 для модулей DI1-16, DIO1-8/8 (разъем «INOUT»); ADI O1, MK1 (разъем «INOUT-D»);
- - блоки T1-DO-8, T1-DO-8S, T1-DO-8R, T1-DO-8P/220, T1-DO-8P/110, T1-DO-8P/24, для модулей D O1-16, DIO1-8/8 (разъем «INOUT»); ADI O1, MK1 (разъем «INOUT-D»).

7. Соединения гибкие:

- соединение гибкое C1-A для модулей AI1-8, AIO1-8/4, AIO1-8/4, AIO1-8/4; TC1-7, TR1-8;
- соединение гибкое C1-D для модулей DI1-16, DIO1-8/8, DO1-16, ADIO1, MK1;
- соединение гибкое C2-A для модулей ADI O1, AI O2, MK1
- соединение гибкое C2-D-8/8 для модулей DI1-16, DIO1-8/8, DO1-16, ADIO1, MK1;

8. Блок и модули питания:

- блок питания ЯЛБИ.426448.097 (~220 В/=24 В);
- модули питания AC220/5-15 (~220 В/=5 В); AC220/5R-15 (~220 В/=5 В; резервирование);

- модули питания DC220/5-15 (=24 В/=5 В); DC220/5R-15 (=24 В/=5 В; резервирование).

9. Блок переключения БПР-10.

10. Пульт настройки PN1.

11. Панель оператора.

Контроллер имеет функционально-децентрализованную архитектуру, которая обеспечивает его высокие технические характеристики и системные качества при их минимальной стоимости. Функционально-децентрализованная архитектура определяется:

- наличием встроенного бортового микропроцессора в модули ввода-вывода и МК1, выполняющего функции модуля независимо и асинхронно по отношению к БЦП;

- предварительной обработкой и преобразованием сигналов ввода-вывода (фильтрация, линеаризация, калибровка, широтно-импульсное модулирование и т.п.);

- децентрализованным обменом данными БЦП с модулями ввода-вывода и МК1 с помощью четырех полевых шин.

В состав контроллеров, кроме традиционных средств - блока БЦП и модулей ввода-вывода, введен МК1, который может выполнять технологические программы пользователя как совместно с БЦП, так и самостоятельно без БЦП, управляя другими модулями ввода-вывода. Отсутствие понятия «базовый состав» контроллера дает возможность оптимального применения средств контроллера для автоматизации объектов как высокой и средней сложности (структуры на базе БЦП с МК1 и модулями ввода-вывода), так и малой сложности (структуры на базе МК1, без БЦП).

2.2 Технические характеристики контроллера

Электрическое питание контроллера определяется по заказу и осуществляется по одному из вариантов:

- от сети переменного однофазного тока с напряжением от 85 до 264 В, частотой 50 Гц и коэффициентом высших гармоник до 5 %;
- от внешнего нестабилизированного источника постоянного тока напряжением от 18 до 36 В (24 В).

Максимальная мощность, потребляемая устройствами контроллера, указана в таблицах 2.2.1, 2.2.2.

Контроллер соответствует климатическому исполнению УХЛ 4.2 по ГОСТ 15150-69, но с условиями эксплуатации и транспортирования, приведенными в таблице 2.2.3.

Контроллер является изделием общего назначения по ГОСТ 18311, место установки контроллера - взрывопожаробезопасное помещение.

Таблица 2.2.1 – Максимальная мощность, потребляемая устройствами контроллера

Модуль	Вид и количество каналов на модуль	Входной/ выходной сигнал	Входное сопротивление (сопротивление нагрузки)	Максимальная потребляемая мощность по цепи 24 В, Вт, не более
ТС 1-7	7 каналов ввода (каналы 1-	Напряжение постоянного тока от минус 5 до 65 мВ от термопар	>100 кОм	1,25
	1 канал ввода (канал 8)	Сопротивление от термопреобразователя сопротивления ТСМ50	—	

TR1-8	8 каналов ввода	Сопротивление: (50-100), (100-200) Ом от термопреобразователя сопротивления	—	1,45
AI1-8	8 каналов ввода	Постоянный ток: (0-5) мА, (0-20), (4-20) мА; Напряжение: (0-10) В	(400±10) Ом (100±3) Ом >10 кОм	1,2
AI O1-8/0	8 каналов ввода			0,5
AI O1-8/4	8 каналов ввода			0,6
	4 канала вывода			
AI O1-0/4	4 канала вывода	Постоянный ток: (0-5) мА, (0-20), (4-20) мА	< 2 кОм <0,5 кОм	0,12
DI1-16	16 каналов ввода	Дискретный сигнал: напряжение постоянного тока: (0-7) В - логический «0», (24±6) В - логическая	не менее 2 кОм	0,12
DIO1-8/8	8 каналов ввода			Дискретный сигнал: бесконтактный ключ; коммутируемое постоянное напряжение до 40 В; максимальный ток не более:
	8 каналов вывода			
DO1-16	16 каналов вывода			

Примечание - Модуль ADI O1 и МК1 имеют submodule ввода-вывода, который содержит восемь дискретных каналов ввода и восемь дискретных каналов вывода, параметры которых соответствуют параметрам модуля DIO1-8/8.

Таблица 2.2.2 – Максимальная мощность, потребляемая устройствами контроллера

Наименование	Максимальная потребляемая мощность по цепи, Вт, не более		
	5 В	24 В	220 В
Блок центрального процессора БЦП	10,0		
Блок питания ЯЛБИ.426449.097			60,0
Модули питания AC220/5-15, AC220/5R-15			20,0
Модули питания DC24/5-15, DC24/5R-15		20,0	
Блок переключения БПР-10		1,2	
Панель оператора		3,5	
Модуль ADIO1, МК1		1,4	
Модуль AIO2		1,3	
Блок T1-DO-8R		0,6 *	
Блоки T1-DI-8/220, T1-DI-8/110, T1-DI-8/24; T2-DI-8/220, T2-DI-8/110, T2-DI-8/24; T1-DO-8S; T1-DO-8P/220, T1-DO-8P/110, T1-DO-8P/24		0,12 *	
* Максимальная потребляемая мощность на один канал (включено).			

Таблица 2.2.3 – Условия эксплуатации и транспортирования

Наименование воздействия	Диапазон воздействия		
	Нормальные условия	Рабочие условия	Условия транспортирования
Температура окружающей среды, °C	(20±5)	от плюс 5 до плюс 50	от минус 50 до плюс 50
Относительная влажность воздуха, %	до 80	до 95 35	до 95 35
Атмосферное давление,	84 - 106,7	84 - 106,7	84 - 106,7

Примечание - В нормальных условиях эксплуатации гарантируется основная погрешность контроллера как средства измерения, в рабочих условиях контроллер будет иметь дополнительные погрешности.

Габаритные размеры и масса устройств контроллера соответствуют значениям, указанным в таблице 2.2.4.

Таблица 2.2.4 – Габаритные размеры и масса устройств контроллера

Наименование	Габаритные	Масса, кг,
Блок центрального процессора БЦП	65x172x135	1,10
Блок переключения БПР-10	40x133x130	0,60
Блок питания ЯЛБИ.426449.097	40,5x90x115	0,40
Модули AI1-8, AI O1-8/0, AI O1-8/4, AI	45x132x115	0,30
Модули питания AC220/5-15, AC220/5R-	45x132x115	0,45
Модуль AIO2	45x132x115	0,40
Модули DI1-16, DIO1-8/8, DO1-16	30x132x115	0,20
Модули питания DC24/5-15, DC24/5R-15	30x132x115	0,30
Модуль ADI O1; МК1	60x132x115	0,45
Панель оператора	147x97x42	0,24
Пульт настройки PN1	100x180x 44	0,30
Блоки T1-AI, T1-DI-8, T1-DO-8	62,5x85x59	0,10
Блок T1-AIO	74,5x85x59	0,15
Блоки T1-DI, T1-DO, T1-DI O	110x85x59	0,20
Блок T1-TC	62,5x85x59	0,15
Блок T1-TR	62,5x85x75	0,15
Блоки T1-DI-8/220, T1-DI-8/110, T1-DI-8/24; T2-DI-8/220, T2-DI-8/110, T2-DI-8/24; T1-DO-8P/220, T1-DO-8P/110, T1-DO-8P/24, T1-DO-8S	122x85x59	0,20
Блок T1-DO-8R	122x85x59	0,25
Блок T2-A	107x85x59	0,18

2.3 Функциональные возможности контроллера

Все модули ввода-вывода контроллера имеют встроенный бортовой микро процессор, выполняющий независимо и асинхронно по отношению к блоку ц

центрального процессора БЦП различные функции ввода-вывода, аналого-цифрового и цифро-аналогового преобразования, предварительной обработке сигналов (фильтрация, линеаризация и т.п.), задаваемые программно, а также широтно-импульсное модулирование импульсных выходных сигналов, непрерывную диагностику (короткое замыкание, обрыв) входных аналоговых каналов.

Высокие динамические качества контроллера обеспечиваются следующими средствами:

- высокой производительностью БЦП, цикл обслуживания одного модуля ввода-вывода не более 20 мс;
- параллельным выполнением функций: ввода-вывода сигналов и их предварительной обработки модулями; обмена данными модулей ввода-вывода с БЦП по нескольким (до четырех) полевым шинам; автономным выполнением собственной ТПП программируемым микроконтроллером МК1 с меньшим, по сравнению с циклом БЦП, временем цикла;
- специальной организацией вычислительного процесса, обеспечивающей минимальное время инициализации контроллера при его рестарте (перезапуске) для безударности его переключений по различным причинам (просечки питания, переход на резерв и т.п.).

Время восстановления выходов контроллера (удар по объекту) составляет до 100 мс, время восстановления функционирования контроллера (нечувствительность объекта) – до 300 мс. Для МК1 время рестарта составляет всего 64 мс.

Контроллер обеспечивает высокую надежность управления технологическим процессом благодаря принятым мерам обеспечения собственной высокой надежности и живучести. Базовый уровень надежности обеспечивается следующими средствами:

- -использование элементной базы ведущих зарубежных фирм;
- резкое снижение числа межблочных контактных соединений и длины линий связи за счет использования последовательных полевых шин (2-4

сигнальных провода);

- использование высоконадежного программного обеспечения, имеющего сотни тысяч инсталляций (операционная система RTOS-32, исполнительная подсистема ISaGRAF, сетевое ПО);

- непрерывная диагностика модулей, наличие сторожевых таймеров во всех модулях;

- защита дискретных выходов модулей от короткого замыкания;

- SMD-монтаж, автоматизированная сборка и контроль модулей, современное производство, сертификат качества по ISO 9001:2000.

- Базовый уровень живучести обеспечивается следующими средствами:

- возможность автономного дублирования на блоках контроля и управления особо ответственных функций центрального процессора (защиты, блокировки, регулирование и т.п.);

- возможность «горячей» замены модулей.

Базовые уровни надежности и живучести могут быть повышены различными способами резервирования контроллеров или их составных частей, обеспечивающих оптимальное соотношение «надежность-стоимость» для различных применений.

Возможны следующие способы резервирования аппаратуры контроллеров:

- резервирование БЦП;
- резервирование контроллеров;
- резервирование аппаратуры ввода-вывода;

- резервирование полевых сетей.

Соответствие международным стандартам открытых систем обеспечивает системную и программную совместимость контроллеров с изделиями других фирм, поддерживающих данные стандарты, в рамках одной АСУ ТП.

Стандарты распространяются на следующие средства контроллеров:

- операционная система реального времени;
- технологические языки программирования системы;
- контроллерные промышленные сети;
- полевые сети;
- механизмы обмена со 8САОА-системами (OPC сервер).

Контроллеры КРОСС-500, выпускаемые ОАО «ЗЭиМ», являются семейством программно- и системно-совместимых приборов, обеспечивающих переносимость технологических программ пользователя между контроллерами семейства; возможность работы в одной контроллерной сети по единым сетевым протоколам.

2.4 Система программирования контроллера

Технологическое программирование контроллера осуществляется в системах ISaGRAF и UltraC/C++.

Рассмотрим систему программирования ISaGRAF, так как для обычного пользователя из промышленности, привыкшего к программированию традиционных программируемых логических контроллеров (PLC), операционная система OS-9 может оказаться не столь простой в изучении. А система ISaGRAF является достаточно доступной и простой.

ISaGRAF – наиболее известная в мире и удачная реализация стандарта МЭК 1131-3, выполненная фирмой CJ International.

Технология программирования традиционных PLC предполагает использование языков программирования, описываемых стандартом МЭК 1131-3:

- язык управляющих последовательностей (SFC);
- язык функциональных блоков (FBD);
- язык релейной логики (LD);
- структурированный текст (ST);
- язык инструкций (IL).

Привычный для многих потребителей язык функциональных блоков FBD в системе ISaGRAF построен на библиотеке элементарных арифметических, логических и динамических функций и может быть расширен библиотекой алгоритмов контроллера P-130.

Проект ISaGRAF разделен на несколько программных модулей, называемых программами. Программы проекта связаны в древовидную структуру. Команда **Соединение В/В** запускает редактор соединения переменных ISaGRAF. Этот инструмент используется для создания связей между объявленными в словаре переменными ввода/вывода и соответствующей аппаратурой ввода/вывода.

После написания программы, генерирования кода проекта, соединения ввода/вывода можно симулировать отладку. Команда **Симулировать** открывает отладчик в режиме симуляции. Эта программа очень полезна, когда целевая машина не доступна.

Команда **Установка связей** позволяет пользователю определить параметры связи, используемые для коммуникации между отладчиком на главном компьютере и целевой системой ISaGRAF.

Команда **Отладка** открывает главное окно отладчика при условии, что код сгенерирован и связь с системой установлена.

С помощью средства ISaGRAF «Прожектор» пользователь может определить списки наблюдения, которые могут показывать либо графические картинки, либо списки во время отладки. Графические символы должны быть связаны с переменными ISaGRAF. Картинки могут быть определены и анимированы во время работы.

3 Структурная схема лабораторного комплекса, схема внешних соединений

3.1 Структурная схема комплекса

Лабораторная работа будет выполняться на стенде, структурная схема которого приведена на рис.3.1.1 Данный лабораторный комплекс состоит из микропроцессорного контроллера КРОСС, аналогового вычислительного комплекса АВК-6, персонального компьютера с ОС Windows NT и автоматического выключателя (АВ). В свою очередь, контроллер КРОСС состоит из центрального блока, блока питания, трех модулей УСО (аналогового, дискретного и термопреобразовательного) и соответствующих им терминальных блоков. В состав центрального блока входит центральный процессор, модуль локальной сети ETHERNET (субмодуль SM2-ETH) и модуль питания. Центральный блок контроллера (ЦБ1) связан с модулями УСО шиной SPI-1. Внешние связи могут подключаться к модулям УСО через терминальные блоки, которые содержат клеммные колодки. Модуль УСО АЮ1-8/4 поддерживает связь с объектом управления (АВК-6) через терминальный блок Т1-АЮ.

Питание комплекса производится от сети переменного тока 220 В (50 Гц) через автоматический выключатель. Блок питания ЛОК 4601-2R/P выдает на контроллер постоянное напряжение 24 В, которое преобразуется модулем питания КР-DC24V1 в 5 В, питающего шину SPI-1.

Центральный процессор контроллера SM2-CPU-1,5 соединен с операторской станцией через субмодуль SM2-ETH, обеспечивающей соединение по локальной сети ETHERNET. Субмодуль и плата центрального процессора имеют общую лицевую панель и занимают в ЦБ1 одно посадочное место.

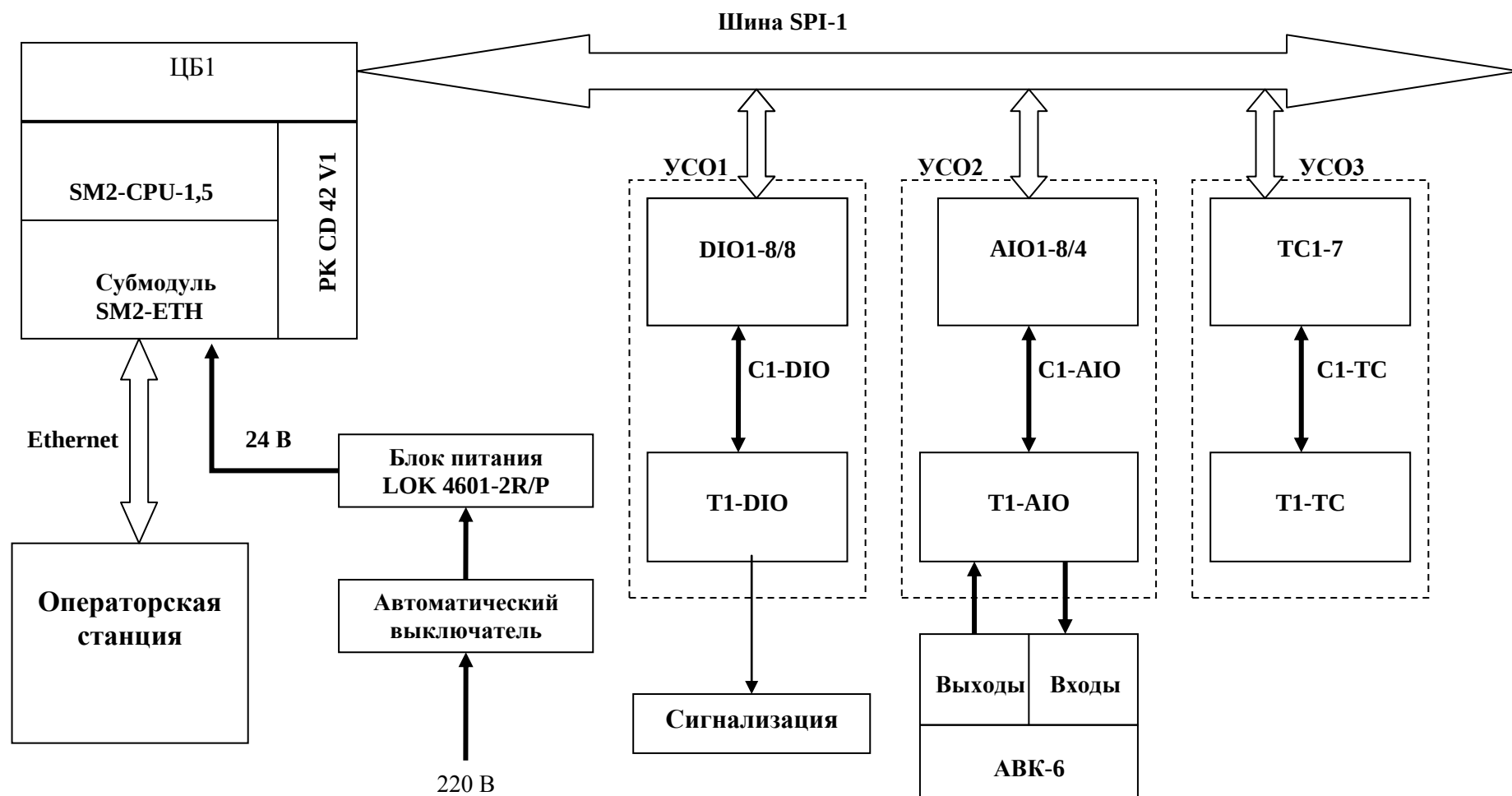


Рис 3.1.1 – Структурная схема лабораторного комплекса

3.2 Краткое описание аналогового вычислительного комплекса АВК-6

3.2.1 Назначение АВК-6

Комплекс аналоговый вычислительный АВК-6 предназначен для исследования путем математического моделирования динамических процессов и систем автоматического управления (САУ). АВК-6 применяется в инженерной практике, а также как техническое средство при изучении курсов физико-математических и общетехнических дисциплин.

3.2.2 Общие характеристики и состав комплекса

АВК-6 объединяет в одном приборе малую аналогово-вычислительную машину (МЭМ-6) и систему индикационно-измерительных средств (СИИС).

Организация МЭМ-6 обеспечивает возможность её программирования непосредственно по структурной математической модели изучаемого динамического процесса и не требует от исследователя владения специальными знаниями и навыками программирования, которые необходимы при использовании традиционных ЭВМ.

СИИС комплекса состоит из комплекта индикационных и измерительных устройств, позволяющих проводить полное исследование моделируемых на МЭМ-6 динамических процессов непосредственно по их графическим характеристикам.

Комплектность и организация СИИС освобождает пользователя от необходимости применения дополнительной аппаратуры в процессе изучения моделей динамических процессов.

3.2.3 Конструкция комплекса

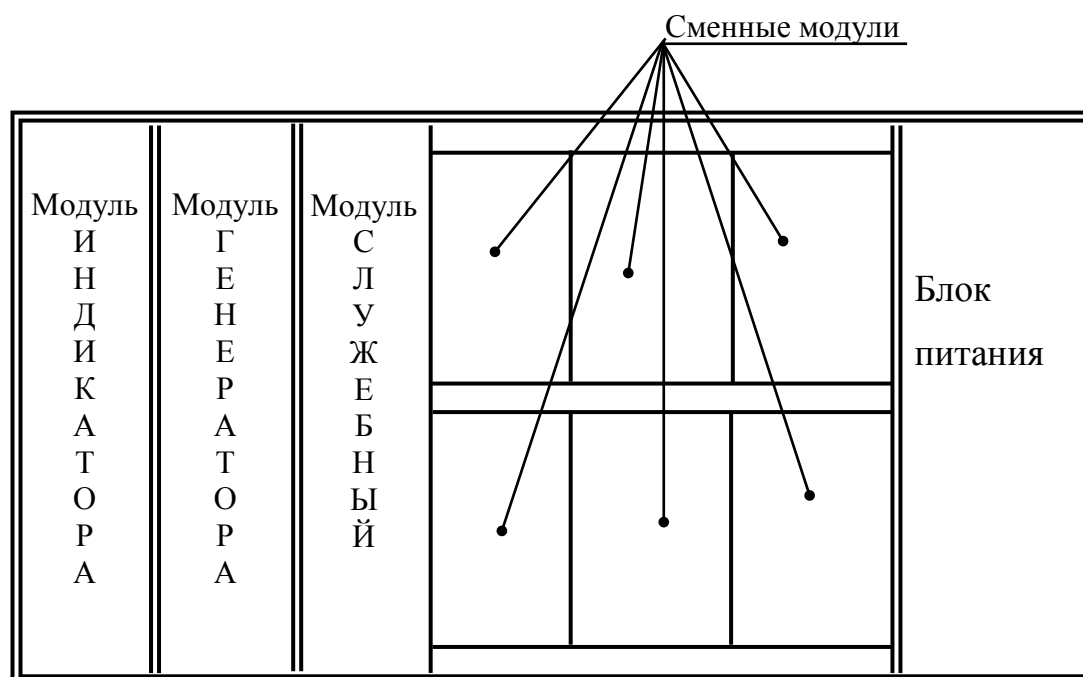
Конструктивно АВК-6 выполнен в виде отдельных функционально законченных блока и модулей, заключенных в единый корпус. Такая структура удобна при изготовлении и настройке, а также при ремонте комплекса.

Корпус АВК-6 представляет собой каркас с четырьмя отсеками. Каждый блок, модуль вставляется в свой отсек и винтами крепится к корпусу комплекса. Расположение блока, модулей на корпусе АВК-6 показано на рисунке 3.2.2.1.

Центральное место занимает аналоговая вычислительная машина МЭМ-6, в которую входят служебные модули, блок и монтажное поле со сменными решающими модулями. Монтажное поле состоит из распределительной платы с направляющими штырями и контактами. Справа от МЭМ-6 расположен блок питания, слева – генератор и коммутатор, а в крайнем левом отсеке – электронно-лучевой индикатор.

Соединения между модулями, блоком питания, платой распределительной выполнены проводами.

Снизу комплекс закрывается нижней крышкой, которая винтами соединяется с корпусом АВК-6.



3.3.Схема внешних соединений

Схема внешних соединений представлена на рисунке 3.3.1.

В данном лабораторном комплексе в качестве объекта управления была взята математическая модель САУ, набранная на аналоговом вычислительном комплексе АВК- 6.

Питание поступает от сети (переменное напряжение 220 В) через автомат на блок питания, который выдает постоянное напряжение 24 В на контакты 3 и 13 клеммы слота «Б». Это напряжение преобразуется в 5 В и запитывает шину SPI-1.

Терминальный блок Т1-АЮ соединяется с АВК- 6, причем входы АВК- 6 с выходами терминального блока и наоборот, один провод общий.

4 SCADA-система Master SCADA

4.1 Назначение SCADA-систем, виды SCADA-систем

SCADA (сокр. от англ. Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) – процесс сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами.

Применение SCADA-систем позволяет существенно сократить сроки разработки программного обеспечения, обеспечить высокое качество регулирования, при этом при создании программного обеспечения профессиональные программисты могут и не привлекаться.

Все современные SCADA-системы включают три основных структурных компонента, представленные на рисунке 4.1.1.

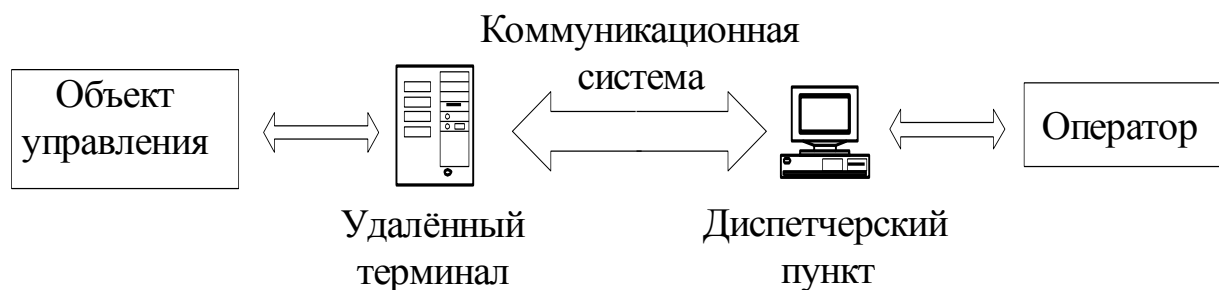


Рисунок 4.1.1 - Основные структурные компоненты SCADA системы

Удаленный терминал (Remote Terminal Unit - RTU) осуществляет обработку задачи (управление) в режиме реального времени.

Диспетчерский пункт управления (Master Terminal Unit - MTU) осуществляет обработку данных и управление высокого уровня, как правило, в режиме реального времени. Одна из его основных функций – обеспечение интерфейса между человеком-оператором и системой.

Коммуникационная система (каналы связи – Communication System (CS)) необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на диспетчерский пункт (или удаленный объект – в зависимости от конкретного исполнения системы).

К SCADA-системам предъявляются следующие основные требования:

- надежность системы (технологическая и функциональная);
- безопасность управления;
- точность обработки и представления данных;
- простота расширения системы.

В силу тех требований, которые предъявляются к системам SCADA, спектр их функциональных возможностей определен и реализован практически во всех пакетах. Перечислим основные возможности и средства, присущие всем системам и различающиеся только техническими особенностями реализации:

- автоматизированная разработка, дающая возможность создания программного обеспечения (ПО) системы автоматизации без реального программирования;
- средства сбора первичной информации от устройств нижнего уровня;
- средства управления и регистрации сигналов об аварийных ситуациях;
- средства хранения информации с возможностью ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);
- средства обработки первичной информации;
- средства визуализации представления информации в виде графиков, гистограмм и т.п.;
- возможность работы прикладной системы с наборами параметров, рассматриваемых как единое целое (setpoint , или установки).

На современном мировом рынке программного обеспечения существует большое количество фирм, выпускаемой продукцией которых являются SCADA-пакеты. Ниже перечислены только некоторые из популярных на западном и российском рынках SCADA-систем, имеющих некоторую поддержку в России (таблица 4.1.1).

Таблица 4.1.1 - Основные фирмы разработчики SCADA- пакетов

Фирма разработчик	Страна	Фирма распространитель в России	Название SCADA- пакета
AdAstra Res. Group	Россия	AdAstra Res. Group	TRACE MODE
Intellution	США	Индасофт	FIX
Ci Technologies	Австралия	РТСофт	Citect
Wonderware	США	РТСофт	InTouch
Iconics	США	ПроСофт	Genesis
Инсат	Россия	Инсат	VNS MasterSCADA

Перечисленные выше возможности систем SCADA в значительной мере определяют стоимость и сроки создания ПО, а также сроки ее окупаемости.

4.2 Отличительные особенности пакета программ MasterScada

MasterScada - это не просто один из современных SCADA-пакетов, это принципиально новый инструмент разработки АСУТП, в котором реализована совокупность средств и методов, обеспечивающих резкое сокращение трудозатрат и повышение надежности создаваемой системы. MasterScada является полнофункциональным SCADA-пакетом программ с расширяемой функциональностью. Пакет построен на клиент-серверной архитектуре с

возможностью функционирования, как в локальных сетях, так и в Интернете. Прием и передача данных и сообщений на основе стандартов OPC встроена в ядро пакета. Максимальная поддержка всех стандартов (XML, HTML, ODBC, OLE, COM/DCOM, ActiveX и др.) и открытые описания интерфейсов и форматов данных обеспечивают все необходимые возможности для стыковки с внешними программами и системами.

Основные преимущества пакета:

1. Единая среда разработки АСУТП.
2. Раздельное конфигурирование структуры АСУТП и логической структуры объекта.
3. Открытость и следование стандартам.
4. Интуитивная легкость освоения.
5. Удобство инструментария.
6. Удобство методики разработки.
7. Мощная трехмерная графика и мультимедиа.
8. Неограниченная гибкость вычислительных возможностей.
9. Объектный подход.

Пользовательский интерфейс MasterScada (рисунок 4.2.1) построен на идеологии «все в одном». Все модули расширения встроены в общую оболочку. Пользователь всегда работает с простым единым внешним видом программы, состоящим из древовидного проекта, палитры библиотечных элементов и окна редактирования документов и свойств. В зависимости от типа настраиваемого свойства или редактируемого документа в окне редактирования открывается страница настройки нужного свойства, либо необходимый встроенный или внешний редактор. Например, встроенный редактор мнемосхем или внешний редактор текстовых описаний.

Проект состоит из двух разделов: «Система» и «Объект». Раздел «Система» описывает техническую структуру реализуемой системы. Раздел «Объект» описывает иерархическую структуру контролируемого технологического объекта свойства и документы каждого объекта.

Окно проекта состоит из четырех основных частей (рисунок 4.2.1):

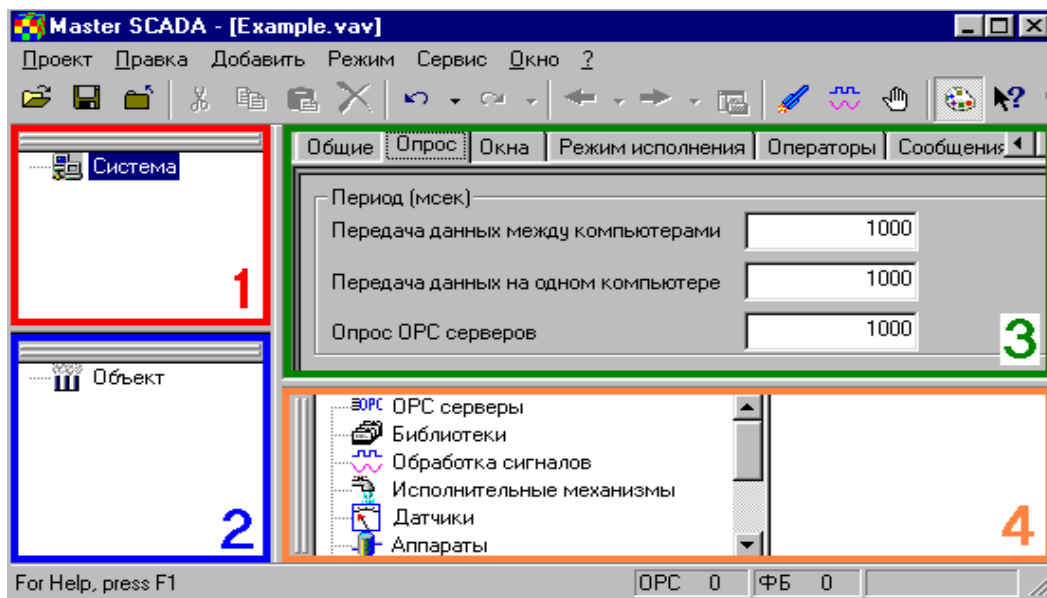


Рисунок 4.2.1 - Основные структурные компоненты SCADA системы

1. Дерево системы, в котором отображены элементы конфигурации такие, как компьютеры, OPC серверы, и т. д.
2. Дерево объектов, включающее в себя объекты, переменные, группы переменных, функциональные элементы.
3. Страницы свойств элементов, на которых производятся все необходимые настройки элементов.
4. Палитра элементов, из которой берутся библиотечные объекты, функциональные блоки и т. д.

Размеры дерева системы, дерева объектов и палитры можно изменять и они могут располагаться в любом месте экрана. Страницы свойств занимают все оставшееся пространство.

Встроенные программные средства позволяют:

- Вести обработку данных.
- Составлять сценарии наступления событий.
- Создавать схемы и мнемосхемы, тренды.
- Выдавать сообщения и составлять рапорты.
- Вести архивирование.
- Обмениваться данными по сети.
- Ограничивать права доступа и вести контроль действий оператора.
- Отлаживать созданные программы.
- И повышать надежность работы.
- Режимы работы.
- Существует три возможных варианта работы в режиме

исполнения.

Рабочий режим. Включается командой "Пуск" из меню "Режим", или кнопкой  на панели инструментов. Это основной режим исполнения. В Рабочем режиме должен быть осуществлен переход к нему на всех компьютерах системы. Программа производит реальное управление технологическим процессом.

Режим отладки. Включается командой "Отладка" из меню "Режим", или кнопкой  на панели инструментов. Этот режим предназначен для отладки проекта на одном компьютере. Независимо от того, сколько компьютеров находится в дереве системы, все объекты, функциональные блоки, ОРС серверы создаются на текущем компьютере и все действия производятся на нем.

Режим имитации. Включается командой "Имитация" из меню "Режим", или кнопкой  на панели инструментов. В этом режиме на все входы, не имеющие связей, вместо констант будет подаваться имитация в соответствии с настройками системы.

5 ПРОГРАММНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

В ходе выполнения данной ВКР было разработано программно-методическое обеспечение для выполнения двух лабораторных работ: «Цифровые регуляторы, построенные на нечеткой логике» и «Нечеткий П-регулятор». Методические указания содержат: цель работы, теоретическую часть, примеры реализации задание и контрольные вопросы. Полный вариант методического обеспечения содержится в приложениях А и В данной выпускной квалификационной работы.

5.1 Цифровые регуляторы, построенные на нечеткой логике

Данный разработанный программно-методический материал предназначен для изучения цифровых регуляторов, реализованных на нечеткой логике. В качестве примера взят нечеткий ПИ-регулятор, запрограммированный на языке FBD в среде программирования контроллеров Isograf. Структура методического обеспечения приведена ниже.

5.1.1 Структура нечеткого регулятора

В данном разделе приводится обобщенная структура нечеткого регулятора.

5.1.2 Нечеткая логика и ее основные свойства

В данном разделе приводятся понятия нечеткого множества, нечеткой логики и ее основные свойства, а также основные нечеткие выводы, такие как Mamdani, Tsukamoto, Sugeno, Laarsen.

5.1.3 Методы фазификации и дефазификации

В данном разделе рассматриваются понятия фазификации и дефазификации, методы их реализации, а так же функции принадлежности.

5.1.4 Структура лабораторного комплекса

В данном разделе приводится структурная схема лабораторного комплекса. Лабораторный стенд позволяет осуществлять проверку

работоспособности систем автоматического регулирования с регуляторами, реализованными программно на контроллере КРОСС. Модель объекта управления набирается на АВК-6. АВК-6 позволяет моделировать работу объекта до шестого порядка включительно. В процессе проверки работоспособности с помощью экранных форм, реализованных на SCADA, можно менять уставки регуляторов, коэффициенты настройки регуляторов и в графическом виде наблюдать за свойствами САР в режиме реального времени.

5.1.5 Система программирования контроллера

В данном разделе приводится описание системы программирования контроллера Isagraf. Приведены различные языки программирования, такие как:

- язык управляющих последовательностей (SFC);
- язык функциональных блоков (FBD);
- язык релейной логики (LD);
- структурированный текст (ST);
- язык инструкций (IL).

5.1.6 Scada-система MASTER SCADA

В данном разделе приводится описание Scada- системы MASTER SCADA, ее характеристики, функциональные возможности и отличительные особенности, а также описание панели инструментов и функций.

5.1.7 OPC-Сервер

CrossOPC сервер (далее OPC-сервер) предназначен для подсоединения и обмена данными между ISaGRAF-приложением контроллера Ремиконт Р-130ISa и SCADA-системами. OPC-сервер работает под управлением ОС Windows NT и предоставляет данные OPC-клиентам через интерфейс, определяемый спецификацией OPC Data Access 2.0).

С ISaGRAF-приложениями OPC-сервер соединяется по последовательному порту или по протоколу TCP/IP. Формат файла. Это текстовый файл в стиле стандартных ini -файлов windows, расположенный по адресу: C:\Program files\OPC Server\CROSSOPC. Каждая строка содержит либо комментарий, начинающийся с символа "точка с запятой", либо заголовок секции, либо набор параметров секции.

5.1.8 Описание программы

При выполнении данной лабораторной работы используется программа, реализующая нечеткое ПИ-регулирование. Программа предназначена для изучения цифровых регуляторов, построенных на нечеткой логике. В качестве простого примера возьмем нечеткий ПИ-регулятор. При изменении ошибки регулирования, меняются коэффициенты пропорциональной и интегральной составляющих регулятора. Этапы фазификации, дефазификации, их функции принадлежности и база правил, а так же полный листинг реализованной в среде программирования контроллеров Isograf приведены в приложении.

5.1.9 Задание

1. Набрать модель объекта управления 2 порядка и задать ее параметры (по заданию преподавателя) на АВК-6. Вход и выход модели объекта подключить к адаптеру, обеспечивающему связь с контроллером.

2. На ПК запустить приложение «ПРОЕКТЫ» программы ISaGRAF. Выбрать проект под названием «Fvva» и запустить программу «Fvva».

3. Установить связь с контроллером и запустить отладчик.

4. Задать уставку данной САУ от 0 В до 10 В (по заданию преподавателя).

5. Запустить редактор представления данных, в котором запустить программу «FVVAMS» в SCADA-системе MASTER SCADA.

6. Изменяя возмущающее воздействие путем изменения коэффициента обратной связи одного из звеньев, обеспечить реализацию переходного

процесса и исследовать работу системы с нечетким ПИИ-регулятором, то есть проследить в открытой программе, реализованной на Isograf, изменение коэффициентов в зависимости от величины возмущающего воздействия, сравнить работу классического и нечеткого ПИИ-регуляторов.

7. Используя графическую форму SCADA-системе MASTER SCADA повторить пункт 6.

8. Скопировать данную программу в новую директорию и на ее основе реализовать нечеткий ПИИ-регулятор. Для дефазификации использовать методом максимума, затем повторить пункты 6 и 7.

9. Сравнить результаты реализации данных программ.

5.1.10 Содержание отчета

1. Цель работы.
2. Обобщенная структура нечеткого регулятора.
3. Структурная схема комплекса.
4. Графики изменения управляющего воздействия нечеткого ПИД-регулятора.
5. Структурная схема созданной программы
6. Ответы на контрольные вопросы.
7. Краткие выводы о проделанной работе.

5.1.11 Контрольные вопросы

1. В каких системах технологического программирования работает контроллер.
2. Какие существуют методы дефазификации.
3. Что такое фазификация и дефазификация.
4. Для чего предназначено приложение «ПРОЖЕКТОР» в системе ISaGRAF.
5. Назовите состав контроллера.
6. Организация SCADA-системы MASTER SCADA.

7. Назначение редактора базы каналов, редактора представления данных.

5.2 Нечеткий П-регулятор

Данный разработанный программно-методический материал предназначен для самостоятельной разработке П-регулятора, реализованных на нечеткой логике в системе программирования контроллеров Isograf на языке функциональных блоков FBD. Структура методического обеспечения приведена ниже.

5.2.1 Нечеткая логика и ее основные свойства

В данном разделе приводятся понятия нечеткого множества, нечеткой логики и ее основные свойства, а также основные нечеткие выводы, такие как Mamdani, Tsukamoto, Sugeno, Laarsen.

5.2.2 Структура САР с нечетким П-регулятором

В данном разделе приведена структурная схема САР с нечетким П-регулятором, Определения и цели этапов фазификации и дефазификации с подробными функциями принадлежности, а так же база правил и алгоритм нечеткого вывода.

5.2.3 Программа реализации нечеткого П-регулятора в системе программирования контроллеров Isograf

В данном разделе приводится подробная поэтапная инструкция создания нового проекта, подробное описание используемых при программировании функциональных блоков, а так же пример реализации этапа фазификации по ошибке регулирования в виде листинга (скриншота).

5.2.4 Задание

1. С помощью системы программирования ISOGRAF составить программу реализации нечеткого П-регулятора.

2. Набрать модель объекта управления 2 порядка и задать ее параметры (по заданию преподавателя) на АВК-6. Вход и выход модели объекта подключить к адаптеру, обеспечивающему связь с контроллером.

3. Установить связь с контроллером и запустить отладчик.

4. Задать уставку данной САУ от 0 В до 10 В (по заданию преподавателя).

5. Запустить прожектор и создать два графика: ошибки регулирования и коэффициента пропорциональности.

6. Изменяя возмущающее воздействие путем изменения коэффициента обратной связи одного из звеньев, обеспечить реализацию переходного процесса и исследовать работу системы с нечетким П-регулятором.

7. Сделать выводы о проделанной работе.

5.2.5 Содержание отчета

1. Цель работы.

2. Структурная схема САУ с нечетким П-регулятором.

3. Графики изменения управляющего воздействия нечеткого П-регулятора.

4. Ответы на контрольные вопросы.

5. Краткие выводы о проделанной работе.

5.2.6 Контрольные вопросы

1. Дать определение фазификации, дефазификации.

2. Какие существуют методы дефазификации.

3. Что такое база правил. Как применяется.

4. Для чего предназначено приложение «ПРОЖЕКТОР» в системе ISaGRAF.

5.2.7 Методические указания по установке связи с контроллером и настройке прожектора

После того, как программа нечеткого П-регулятора готова, необходимо проверить ее на работоспособность. Для этого нужно сперва настроить Опции

компилятора и связь. В данном разделе приведена полная подробная инструкция по установке связи с контроллером при отладке. Так же приведен пример использования прожектора для визуализации изменения пропорциональной составляющей регулятора.

6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Цель раздела – комплексное описание и анализ финансово-экономических аспектов выполненной работы. Необходимо оценить полные денежные затраты на исследование (проект), а также дать хотя бы приближенную экономическую оценку результатов ее внедрения. Это в свою очередь позволит с помощью традиционных показателей эффективности инвестиций оценить экономическую целесообразность осуществления работы. Раздел должен быть завершен комплексной оценкой научно-технического уровня ВКР на основе экспертных данных.

6.1 Организация и планирование работ

При организации процесса реализации конкретного проекта необходимо рационально планировать занятость каждого из его участников и сроки проведения отдельных работ.

Наглядным результатом планирования работ является сетевой, либо линейный график реализации проекта. Так как исполнителей два человека, а именно научный руководитель проекта (НР) и инженера(И), предпочтительным является линейный график. Для его построения хронологически упорядоченные вышеуказанные сведения в таблицу:

Таблица 6.1 Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР, И	НР – 100% И – 10%
Разработка календарного плана	НР, И	НР – 100% И – 10%
Подбор и изучение материалов по тематике	НР, И	НР – 60%

		И – 100%
Обсуждение литературы	НР, И	НР – 60% И – 100%
Ознакомление с пакетом программирования контроллера Кросс – Isagraph	И	И – 100%
Ознакомление с пакетом Master Scada 3.6	И	И – 100%
Исследование и анализ структуры регуляторов, построенных на нечеткой логике	И	И – 100%
Разработка программного обеспечения	НР, И	НР – 30% И – 100%
Разработка методического обеспечения	НР, И	НР – 20% И – 100%
Технико-экономическое обоснование проекта	НР, И	НР – 40% И – 100%
Оценка безопасности и экологичности работы	НР, И	НР – 30% И – 100%
Составление и оформление пояснительной записки	И	И – 100%
Защита дипломного проекта	И	И – 100%

6.1.1 Продолжительность этапов работ

Расчет продолжительности этапов работ осуществляется экспертным способом. Для определения вероятных (ожидаемых) значений продолжительности работ $t_{ож}$ применяется формула:

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5},$$

где t_{min} – минимальная продолжительность работы, дн.;

t_{max} – максимальная продолжительность работы, дн.;

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях ($T_{РД}$) ведется по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д},$$

где $t_{ож}$ – продолжительность работы, дн.;

$K_{ВН}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $K_{ВН} = 1$;

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д} = 1-1,2$; в этих границах конкретное значение принимает сам исполнитель). Берем коэффициенты, равные: $K_{ВН} = 1$; $K_{Д} = 1,2$. Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К},$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}},$$

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ} = 366$);

$T_{ВД}$ – выходные дни ($T_{ВД} = 52$);

$T_{ПД}$ – праздничные дни ($T_{ПД} = 14$).

В нашем случае расчет ведем для шестидневной рабочей недели, $T_{К} = 1,22$

Таблица 6.1.1.1 Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.-дн.			
					Т _{РД}		Т _{КД}	
		t _{min}	t _{max}	t _{ож}	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка целей и задач, получение исходных данных	НР	1	2	1,4	1,68	0,00	2,05	0,00
Составление и утверждение ТЗ	НР, И	1	2	1,4	1,68	0,17	2,05	0,20
Разработка календарного плана	НР, И	2	3	2,4	2,88	0,29	3,51	0,35
Подбор и изучение материалов по тематике	НР, И	5	7	5,8	4,18	6,96	5,09	8,49
Обсуждение литературы	НР, И	1	2	1,4	1,01	1,68	1,23	2,05
Ознакомление с пакетом программирования контроллера Кросс – Isagraph	И	5	10	7	0,00	8,40	0,00	10,25
Ознакомление с пакетом Master Scada 3.6	И	5	10	7	0,00	8,40	0,00	10,25
Исследование и анализ структуры регуляторов, построенных на нечеткой логике	И	7	12	9	0,00	10,80	0,00	13,18
Разработка программного обеспечения	НР, И	10	15	12	4,32	14,40	5,27	17,57
Разработка методического обеспечения	НР, И	7	10	8,2	1,97	9,84	2,40	12,00
Технико-экономическое	НР, И	2	4	2,8	1,34	3,36	1,64	4,10

обоснование проекта								
Оценка безопасности и экологичности работы	НР, И	2	4	2,8	1,01	3,36	1,23	4,10
Составление и оформление пояснительной записки	И	7	12	9	0,00	10,80	0,00	13,18
Защита дипломного проекта	И	1	2	1,4	0,00	1,68	0,00	2,05
Итого:				71,60	20,06	80,14	24,48	97,77

Таблица 6.1.1.2 Линейный график работ

Этап	НР	И	Март			Апрель			Май			Июнь		
			10	20	30	40	50	60	70	80	90	100	110	120
1	2,05	0,00	■											
2	2,05	0,20	■											
3	3,51	0,35	■											
4	5,09	8,49		■										
5	1,23	2,05			■									
6	0,00	10,25			■									
7	0,00	10,25				■								
8	0,00	13,18					■							
9	5,27	17,57						■						
10	2,40	12,00							■					
11	1,64	4,10								■				
12	1,23	4,10									■			
13	0,00	13,18										■		
14	0,00	2,05											■	

НР – ; ■ И – ■

6.1.2 Расчет накопления готовности проекта

Цель данного пункта – оценка текущих состояний (результатов) работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего (i-го) этапа выполнен общий объем работ по проекту в целом.

ТР_{общ.} – общая трудоемкость проекта;

ТР_i (ТР_k) – трудоемкость i-го (k-го) этапа проекта, $i = \overline{1, I}$;

ТР_{iH} – накопленная трудоемкость i-го этапа проекта по его завершении;

ТР_{ij} (ТР_{kj}) – трудоемкость работ, выполняемых j-м участником на i-м этапе, здесь $j = \overline{1, m}$ – индекс исполнителя, в нашем примере m = 2.

Степень готовности определяется формулой (5.5)

$$СГ_i = \frac{ТР_i^H}{ТР_{общ.}} = \frac{\sum_{k=1}^i ТР_k}{ТР_{общ.}} = \frac{\sum_{k=1}^i \sum_{j=1}^m ТР_{km}}{\sum_{k=1}^I \sum_{j=1}^m ТР_{km}}.$$

Применительно к таблице (5.2) величины ТР_{ij} (ТР_{kj}) находятся в столбцах (6, j = 1) и (7, j = 2). ТР_{общ.} равна сумме чисел из итоговых клеток этих столбцов. Расчет ТР_i (%) и СГ_i (%) на основе этих данных содержится в таблице 6.1.2.

Таблица 6.1.2 Нарастание технической готовности работы и удельный вес каждого этапа

Этап	ТР _i , %	СГ _i , %
Постановка целей и задач, получение исходных данных	1,68	1,68
Составление и утверждение ТЗ	1,84	3,52
Разработка календарного плана	3,16	6,68
Подбор и изучение материалов по тематике	11,11	17,80
Обсуждение литературы	2,68	20,48

Ознакомление с пакетом программирования контроллера Кросс – Isagraph	8,38	28,86
Ознакомление с пакетом Master Scada 3.6	8,38	37,25
Исследование и анализ структуры регуляторов, построенных на нечеткой логике	10,78	48,02
Разработка программного обеспечения	18,68	66,71
Разработка методического обеспечения	11,78	78,49
Технико-экономическое обоснование проекта	4,69	83,19
Оценка безопасности и экологичности работы	4,36	87,54
Составление и оформление пояснительной записки	10,78	98,32
Защита дипломного проекта	1,68	100,00

6.2 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- командировочные расходы;
- оплата услуг связи;
- арендная плата за пользование имуществом;
- прочие услуги (сторонних организаций);

- прочие (накладные расходы) расходы.

6.2.1 Расчет затрат на материалы

К данной статье расходов относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ над объектом проектирования. Сюда же относятся специально приобретенное оборудование, инструменты и прочие объекты, относимые к основным средствам, стоимостью до 40 000 руб. включительно. Цена материальных ресурсов определяется по соответствующим ценникам или договорам поставки. Кроме того статья включает так называемые транспортно-заготовительные расходы, связанные с транспортировкой от поставщика к потребителю, хранением и прочими процессами, обеспечивающими движение (доставку) материальных ресурсов от поставщиков к потребителю. Сюда же включаются расходы на совершение сделки купли-продажи (т.н. транзакции). Приблизительно они оцениваются в процентах к отпускной цене закупаемых материалов, как правило, это 5÷20 %.

Таблица 6.2.1 Расчет затрат на материалы

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма, руб.
USB Flash Drive 4 Gb	500	1	500
Ручка	10	1	10
Карандаш	5	1	5
Тетрадь	12	1	12
Бумага для принтера формата А4	250	1	250
Картридж для принтера	1550	1	1550
Итого:			2327

Допустим, что ТЗР составляют 5 % от отпускной цены материалов, тогда расходы на материалы с учетом ТЗР равны $S_{\text{мат}} = 2327 * 1,05 = 2443.45$ руб.

6.2.2 Расчет заработной платы

Данная статья расходов включает заработную плату научного руководителя и инженера (в его роли выступает исполнитель проекта), а также премии, входящие в фонд заработной платы. Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Среднедневная тарифная заработная плата (ЗП_{дн-т}) рассчитывается по формуле:

$$ЗП_{дн-т} = МО/25,58,$$

учитывающей, что в году 300 рабочих дней и, следовательно, в месяце в среднем 25 рабочих дня (при шестидневной рабочей неделе).

Расчеты затрат на полную заработную плату приведены в таблице 6.2.1. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 6.1.1. Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов: КПП = 1,1; Кдоп.ЗП = 1,188; Кр = 1,3. Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент Ки = 1,1 * 1,188 * 1,3 = 1,699. Вышеуказанное значение Кдоп.ЗП применяется при шестидневной рабочей неделе.

Таблица 6.2.2 Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фонд з/платы, руб.
НР	23 264,86	930,59	20	1,699	31 722,79
И	14 874,45	594,98	80	1,699	81 006,89
Итого:					112 729,67

6.2.3 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН), включающий в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30 % от полной заработной платы по проекту, т.е. $C_{\text{соц.}} = C_{\text{зп}} \cdot 0,3$. Итак, в нашем случае $C_{\text{соц.}} = 112\,729,67 \cdot 0,3 = 33\,818,9$ руб.

6.2.4 Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot t_{\text{об}} \cdot C_{\text{э}} \quad ,$$

где $P_{\text{ОБ}}$ – мощность, потребляемая оборудованием, кВт;

$C_{\text{Э}}$ – тариф на 1 кВт·час;

$t_{\text{об}}$ – время работы оборудования, час.

Для ТПУ $C_{\text{Э}} = 5,257$ руб./кВт·час (с НДС).

Время работы оборудования вычисляется на основе итоговых данных таблицы 6.2.2 для инженера (ТРД) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{\text{об}} = T_{\text{РД}} \cdot K_t \cdot C_{\text{э}} \quad ,$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к ТРД, определяется исполнителем самостоятельно. В ряде случаев возможно определение $t_{\text{об}}$ путем прямого учета, особенно при ограниченном использовании соответствующего оборудования.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{ОБ}} = P_{\text{ном.}} \cdot K_{\text{с}},$$

где $P_{\text{ном.}}$ – номинальная мощность оборудования, кВт;

$KC \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $KC = 1$.

Расчет затраты на электроэнергию для технологических целей приведен в таблице 6.2.4.

Таблица 6.2.4 Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $\Delta_{об}$, руб.
Персональный компьютер	641,1	0,3	1 011,06
Принтер	5,0	0,2	5,26
Контроллер КРОСС	512,9	0,1	269,62
Итого:			1 285,93

6.2.5 Расчет амортизационных расходов

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта.

Используется формула

$$C_{ам} = \frac{N_A * C_{об} * t_{рф} * n}{F_D},$$

где N_A – годовая норма амортизации единицы оборудования;

$C_{об}$ – балансовая стоимость единицы оборудования с учетом ТЗР.

F_D – действительный годовой фонд времени работы соответствующего оборудования.

Для ПК (300 рабочих дней при шестидневной рабочей неделе) можно принять $F_D = 300 * 8 = 2400$ часа;

Для принтера $F_D = 500$ часов;

Для контроллера КРОСС $F_D = 2400$ часов;

trф – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта.

Для ПК trф = 641,1 часа.

Для принтера trф = 5 часов.

Для контроллера КРОСС ФД trф = 512,9 часа.

n – число задействованных однотипных единиц оборудования.

При использовании нескольких типов оборудования расчет по формуле делается соответствующее число раз, затем результаты суммируются.

Для определения НА следует обратиться к постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы». Оно позволяет получить рамочные значения сроков амортизации (полезного использования) оборудования $\equiv$ СА.

Для ПК возьмем СА = 2,5 года. Далее определяется НА как величина обратная СА, в данном случае это $1 : 2,5 = 0,4$.

Для принтера СА = 2 года, НА = 0,5.

Для контроллера КРОСС СА = 2,5 года НА = 0,4;

$САМ(ПК) = (0,4 * 45000 * 641,1 * 1) / 2400 = 4808,25$ руб;

$САМ(Пр) = (0,5 * 12000 * 5 * 1) / 500 = 60$ руб;

$САМ(К) = (0,4 * 25000 * 512,9 * 1) / 2400 = 2173,08$ руб.

Итого начислено амортизации 7 041,33 руб.

6.2.6 Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов, т.е.

$$C_{\text{проч.}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}} + C_{\text{нп}}) \cdot 0,1.$$

Для нашего примера это

$$C_{\text{проч.}} = (2\,443,45 + 112\,729,67 + 33\,818,9 + 1\,285,93 + 7\,041,33) \cdot 0,1 = 15\,487,58 \text{ руб.}$$

6.2.7 Расчет общей себестоимости разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта «Программно-методическое обеспечение для изучения цифровых регуляторов, построенных на нечеткой логике».

Таблица 6.2.7 Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	2 443,45
Основная заработная плата	$C_{\text{зп}}$	112729,67
Отчисления в социальные фонды	$C_{\text{соц}}$	33818,90
Расходы на электроэнергию	$C_{\text{эл.}}$	1285,93
Амортизационные отчисления	$C_{\text{ам}}$	7041,33
Непосредственно учитываемые расходы	$C_{\text{нр}}$	0,00
Прочие расходы	$C_{\text{проч}}$	15487,58
Итого:		170 363,41

Таким образом, затраты на разработку составили $C = 170\,363,41$ руб.

6.2.8 Расчет прибыли

Прибыль от реализации проекта в зависимости от конкретной ситуации (масштаб и характер получаемого результата, степень его определенности и коммерциализации, специфика целевого сегмента рынка и т.д.) может определяться различными способами. Если исполнитель работы не располагает данными для применения «сложных» методов, то прибыль следует принять в размере $5 \div 20 \%$ от полной себестоимости проекта. В нашем примере она составляет 34 072,68 руб. (20 %) от расходов на разработку проекта.

6.2.9 Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. В нашем случае это

$$(170\,363,41 + 34\,072,68) * 0,18 = 36\,798,5 \text{ руб.}$$

6.2.10 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае

$$\text{ЦНИР(КР)} = 170\,363,41 + 34\,072,68 + 36\,798,5 = 241\,234,59 \text{ руб.}$$

6.3 Оценка экономической эффективности проекта

На сегодняшний день изучению цифровых регуляторов, построенных на нечеткой логике, уделяют совсем мало учебного времени и затрагивают только теоретический курс. Поскольку нечеткое управление позволяет расширить сферу применения систем автоматизации за пределы применимости классической теории, разработанный проект позволяет продемонстрировать практическое применение нечетких систем управления, тем самым повысить уровень знаний студентов в достаточно актуальном на сегодняшний день направлении для изучения автоматизированных систем.

Таким образом данный проект не ориентирован на экономический результат, а направлен в сторону увеличения программно-методического обеспечения для обучения студентов, обучающихся по направлению 27.03.04 «Управление в технических системах» кафедры АиКС НИИ ТПУ.

7 Социальная ответственность

7.1 Производственная безопасность

В данной выпускной квалификационной работе производится разработка программно-методического обеспечения для изучения цифровых регуляторов, построенных на нечеткой логике с использованием промышленного контроллера КРОСС-500. При этом используются такие программные пакеты, как IsOGraf и Master scada 3.6. Таким образом, в дипломном проекте используется следующее оборудование: ПК и промышленный контроллер КРОСС-500.

Работа заключается в написании программного обеспечения, его тестировании и разработке методического пособия, которые будут использованы для обучения студентов.

7.2 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения

При реализации разработанного проекта может возникнуть ряд вредных факторов, которые могут повлиять на здоровье, моральное, психическое состояние человека, экологию окружающей среды.

К вредным факторам можно отнести

- повышенный уровень электромагнитных излучений;
- Повышенный уровень шума;

Проблема электромагнитного излучения, исходящего от персональных компьютеров, встает достаточно остро ввиду нескольких причин:

- Компьютер имеет сразу два источника излучения (монитор и системный блок).
- Пользователь ПК практически лишен возможности работать на расстоянии.
- Очень длительное время воздействия.

Кроме этого, существуют несколько вторичных факторов, которые усугубляют ситуацию, к ним можно отнести работу в тесном непроветриваемом помещении и концентрацию множества ПК в одном месте.

Наиболее сильным источником ЭМ излучения является монитор с электронно-лучевой трубкой, особенно его боковые и задние стенки, т.к. они не имеют специального защитного покрытия, которое есть у лицевой части экрана.

При работе с компьютером допустимые уровни электромагнитных полей (ЭМП) указаны в таблице 7.2

Таблица 7.2 Временные допустимые уровни ЭМП

Наименование параметров		Допустимый уровень ЭМП
Напряженность электрического поля	в диапазоне частот 5Гц - 2кГц	25 В/м
	в диапазоне частот 2кГц - 400кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5Гц-2кГц	250 нТл
	в диапазоне частот 2кГц - 400кГц	25 нТл
Напряженность электростатического поля		15 кВ/м
Электростатический потенциал экрана видеомонитора		500 В

Возможные способы защиты от электромагнитного поля:

Основной способ – увеличение расстояния от источника, для избежания последствий монитор должен находиться на расстоянии не менее 50 см от оператора.

Применение приэкранных фильтров, специальных экранов и других средств индивидуальной защиты, прошедших испытания в аккредитованных лабораториях и имеющих соответствующий гигиенический сертификат.

С целью предупреждения вредного действия шума необходимо, чтобы его уровень на рабочих местах с ПК не превышал допустимых значений (50

дБА), установленных ГОСТ 12.1.003–83 «Шум. Общие требования безопасности». Шум создается вентилятором, установленным в ПК и принтером. При выполнении работ с использованием ПК в производственных помещениях уровень вибрации не должен превышать допустимых значений вибрации для рабочих мест (категория 3, тип "в") в соответствии с действующими санитарно-эпидемиологическими нормативами. В соответствии с СН 2.2.4/2.1.8.566–96 «Производственная вибрация, вибрация в помещениях жилых и общественных зданий», значение виброускорения не более 83 Дб, а виброскорость не более 75 Дб.

Для уменьшения шума в помещении с ПК, как правило, применяют метод акустической обработки помещений, используя для облицовки ограждающих поверхностей звукопоглощающие материалы с максимальными коэффициентами звукопоглощения (α) в интервале частот 63 – 8000 Гц. С этой целью на потолках и стенах размещают перфорированные панели с звукопоглощающим наполнителем (минеральной ватой). Панели укрепляют непосредственно на поверхности ограждения или с отнесением от него на расстояние 20 см. В последнем случае применение звукопоглощающей облицовки более эффективно.

Дополнительным звукопоглощением могут служить однотонные занавеси из плотной ткани, гармонирующие с окраской стен и подвешенные в складку на расстоянии 15 – 20 см от оконного стекла. Ширина занавеси должна быть в 2 раза больше ширины окна. Снизить уровень шума можно также, используя для печати малошумные лазерные принтеры.

7.3 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения

В данном проекте разрабатывается программно-методическое обеспечение для обучения студентов программированию промышленных контроллеров. Соответственно студенту придется работать с ПК и

контроллером, и, при неправильной эксплуатации и несоблюдении мер безопасности, существует риск поражения электрическим током.

Опасность поражения электрическим током существует всегда, если имеется контакт с устройством, питаемым напряжением 36 В и выше, тем более от электрической сети 220 В. Это может произойти по оплошности в случае прикосновения к открытым токоведущим частям, но чаще всего из-за различных причин (перегрузки, не совсем качественная изоляция, механические повреждения и др.). В процессе эксплуатации может ухудшиться изоляция токоведущих частей, в том числе шнуров питания, в результате чего они могут оказаться под напряжением, и случайное прикосновение к ним чревато электротравмой, а в тяжелых случаях — и гибелью человека.

Зоной повышенной электроопасности являются места подключения электроприборов и установок. Часто случается перегрузка розеток по мощности, и, как следствие, происходит нарушение изоляции, приводящее к короткому замыканию.

Для сведения к минимуму потенциальной опасности электротравмирования необходимо придерживаться требований, установленных "Правилами эксплуатации электроустановок потребителей" и "Правилами техники безопасности при эксплуатации электроустановок потребителей" (ПЭ и ПТБ электроустановок потребителей), а также "Правилами устройства электроустановок (ПУЭ)".

Для предотвращения поражений электрическим током при работе с компьютером следует установить дополнительные оградительные устройства, обеспечивающие недоступность токоведущих частей для прикосновения; с целью уменьшения опасности можно использовать разделительный трансформатор для развязки с основной сетью, и обязательным во всех случаях является наличие защитного заземления или зануления (защитного отключения) электрооборудования. Для качественной работы компьютеров создается отдельный заземляющий контур.

Безопасность эксплуатации контроллера обеспечивается:

- изоляция электрических цепей друг от друга в контроллере;
- надежным креплением контроллера при монтаже на объекте;
- конструкцией - все составные части блока и модулей питания,

терминальных блоков, находящиеся под опасным напряжением, размещены внутри корпусов или закрыты защитными крышками, которые обеспечивают защиту персонала от случайного прикосновения к ним.

Во избежание поражения электрическим током внешнее соединение ввода вывода производить, сняв напряжение, подаваемое на контроллер и подключенную к нему схему.

Во время работы с электроустановками наряду с безусловным соблюдением определенных организационных мер, установленных ПЭ и ПТБ электроустановок потребителей, следует строго выполнять все технические мероприятия, обеспечивающие безопасность работ со снятием напряжения, а именно: отключение оборудования в помещении, выделенном для производства работ, и принятие мер против ошибочного или самопроизвольного включения; ограждение при необходимости рабочих мест и оставшихся под напряжением токоведущих частей; вывешивание предупредительных плакатов и знаков безопасности; проверка отсутствия напряжения; наложение заземления.

7.4 Экологическая безопасность

Студенты в процессе обучения проходят практику на различных предприятиях. Для оценки данного пункта в качестве примера возьмем технологический процесс управления насосными агрегатами водопроводной станции. Основная задача, поставленная перед студентом-практикантом - оптимизация расхода воды.

Большую часть электроэнергии, необходимой для функционирования водопроводной станции и, в частности, насосных отделений, потребляют насосные агрегаты. В связи с этим можно утверждать, что, организовав работу

насосов в экономичном режиме и при условии обеспечения достаточного расхода воды, эффективность работы насосного отделения и всей водопроводной станции будет значительно увеличена. При работе насосного отделения кроме потерь энергии, учитываемых с помощью КПД насоса и приводного двигателя, и потерь при энергетически неэффективных способах регулирования можно выделить потери, связанные с поддержанием избыточного для удовлетворения нужд потребителей давления. Учитывая все это, необходимо, чтобы система управления, принимая информацию о текущих расходе, развиваемом насосами напоре и его допустимых границах, потреблении ими электроэнергии, выбирала бы давление, необходимое для поддержания, с учетом обеспечения минимума потерь.

Для реализации поставленной задачи целесообразно применить реализованный цифровой нечеткий регулятор на базе промышленного контроллера. В результате внедрения получили следующие изменения:

- Общепроизводственные расходы возрастают за счет закупки и обслуживания на объекте нового оборудования.
- Расходы на ресурсы (электричество, отопление и т.д.) уменьшаются.
- Значительно сокращаются расходы на ремонт и эксплуатацию объекта.

Кроме того за счет внедрения контроллера с нечетким регулятором достигается эксплуатационный эффект.

- Снизилась средняя частота вращения насосного агрегата, что позволило повысить ресурс насоса и двигателя и исключить резонансные эффекты конструкции, соответственно и снизить затраты на ремонт и комплектующие.
- Трудоемкость работ по обслуживанию системы сведена к минимуму.

После модернизации можно заметить и положительное влияние на гидросферу. Система автоматически поддерживает оптимальное стабильное давление, за счет чего обеспечивается меньший расход воды и уменьшается гидравлическая нагрузка на трубопровод. Это намного уменьшает риск возникновения гидроударов, которые могут вызвать порывы в сети и, соответственно, нерациональный расход воды. На атмосферу и литосферу применение контроллера не влияет.

7.5 Безопасность в чрезвычайных ситуациях

Для создания в системе водоснабжения оптимального давления и качественного регулирования данного параметра, в системе водоснабжения используют ресиверы (гидроаккумуляторы). Результатом может быть значительное увеличенное давление в системе, что в свою очередь может привести к гидроудару, а иногда и взрыву гидроаккумулятора. Последствиями взрыва могут быть пожар, частичное или полное обрушение здания и привести к человеческим жертвам. Причиной возникновения ЧС следующие факторы:

- Неправильно запрограммированный контроллер студентом-практикантом, в частности: некорректно созданная модель синтеза объекта управления для поиска оптимальных параметров регулирования, неправильно выбранная методика построения системы управления с нечетким регулятором, плохо подобранная база правил;
- некачественное или не прошедшее вовремя поверку оборудование;
- отсутствие или некачественно проведенное техническое обслуживание оборудования;
- отсутствие контроля над исполнителем (студентом).

Для предупреждения не правильной работы контроллера и, как следствие, возникновения перечисленных ЧС необходимо предпринять ряд следующих мер:

- тщательно продумать алгоритм работы программы;
- тестирование и дальнейшая отладка программного обеспечения;
- проверка оборудования на работоспособность перед вводом в эксплуатацию объекта;
- Контроль студента-практиканта более опытным специалистом или преподавателем.

7.6 Правовые и организационные вопросы обеспечения безопасности.

Правовой основой законодательства в области обеспечения БЖД является Конституция – основной закон государства. Законы и иные правовые акты, принимаемые в РФ, не должны противоречить Конституции РФ. В состав этих основ входит:

Экологическая безопасность.

Федеральный закон «Об охране окружающей среды» в комплексе с мерами организационного, правового, экономического и воспитательного воздействия, который содержит свод правил охраны окружающей и регулирует природоохранные отношения в сфере всей природной среды, не выделяя ее отдельные объекты, охране которых посвящено специальное законодательство. Задачами этого законодательства являются: охрана природной среды, предупреждение вредного воздействия хозяйственной или иной деятельности, оздоровление окружающей природной среды, улучшение ее качества.

СанПиН 2.1.2.568-96 - Гигиенические требования к устройству, эксплуатации и качеству воды.

Охрана труда – это система обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающая правовые, социально-экономические, организационно-технические,

санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия.

СанПиН 2.2.4.1329-03 - Требования по защите персонала от воздействия импульсных электромагнитных полей.

СН 2.2.4/2.1.8.562-96 - Шум на рабочих местах, в помещениях жилых, общественных зданий.

ГОСТ Р 12.1.019-2009 ССБТ. - Электробезопасность.

Федеральный закон «О защите населения и территорий от ЧС природного и техногенного характера» определяет общие для РФ организационно-правовые нормы в области защиты населения, всего земельного, водного, воздушного пространства в пределах РФ, объектов производственного и социального назначения, а так же окружающей природной среды от ЧС природного и техногенного характера.

Основные цели закона: предупреждение возникновения и развития ЧС, снижение размеров ущерба и потерь от ЧС, ликвидация ЧС.

СНиП 2.04.02-84* - Водоснабжение, наружные сети и сооружения.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной выпускной квалификационной работы было создано программно методическое обеспечение для изучения цифровых регуляторов, построенных на нечеткой логике.

В процессе работы были изучены основы нечеткой логики, принципы построения нечетких регуляторов, состав языка программирования FBD в системе ISaGRAF, проведено ознакомление со SCADA-системой Master SCADA. В результате разработано программно-методическое обеспечение по выполнению двух лабораторных работ, целью выполнения которых является получение знаний по разработке нечетких регуляторов и получение практических навыков их программной реализации.

Методическое обеспечение предназначено для выполнения лабораторных работ по учебному курсу “Автоматизированное управление в технических системах” студентами, обучающимися по направлению 27.04.04-“Управление в технических системах” кафедры автоматики и компьютерных систем Томского политехнического университета.

CONCLUSION

As the result of this qualification work realization the methodical software was created for the research of the numerical regulators based on the fuzzy logic.

During the work the fuzzy logic basis, the modes of the careless regulators construction, the structure of the programming language FBD at the system ISaGRAF were researched, also the SCADA-system Master SCADA was acquainted. As the result the methodical software was created for two laboratorial works realization. The aim of these works is the receiving of the knowledge of the careless regulators creation and the receiving of the practical skills of the software realization.

The methodical software is for the laboratorial work realization according to the course “The automated management in the technical systems” by the students, studied on the direction 27.04.04 - “The management in the technical systems”, the department of the automation and computer system, the Tomsk Polytechnical University.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стальский Владимир Вильгельмович. Нечеткая логика и ее применение в автоматическом регулировании. — СПб: 1998. — 94 с.
2. Современная прикладная теория управления: Новые классы регуляторов технических систем/ Под ред. А.А. Колесникова. Таганрог: ТРТУ, 2000. —311 с.
3. Леоненков, Александр В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. — СПб. : БХВ-Петербург, 2003. — 719 с.
4. Усков, А. А. Алгоритм синтеза нечётких логических регуляторов на основе самоорганизации // Приборы и системы. Управление, контроль, диагностика : Журнал. — М. — 2004. — № 8. — С. 1-3.
5. Ротач, В. Я. Возможен ли синтез нечётких регуляторов с помощью теории нечётких множеств? // Промышленные АСУ и контроллеры : Ежемесячный научно-технический производственный журнал . — М. — 2004. — № 1. — С. 33-34.
6. Усков, А. А. Эмпирический принцип синтеза нечётких логических регуляторов // Приборы и системы. Управление, контроль, диагностика : Журнал. — М. — 2004. — № 1. — С. 16-18.
7. Ротач В.Я. О фазии-ПИД регуляторах//Теплоэнергетика. 1999. №8. С. 32-36.
8. Панько М.А. Особенности нечетких алгоритмов регулирования в сравнении с классическими// Теплоэнергетика.2001. №10. С. 39-42
9. Безопасность жизнедеятельности / Под ред. С.В. Белова. – М.: Высшая школа, 1999. – 448 с.
10. Охрана окружающей среды / Под ред. С.В. Белова. – М.: Высшая школа, 1991.
11. Расчёт искусственного освещения. Методические указания к выполнению индивидуальных заданий для студентов дневного и заочного обучения всех специальностей. – Томск: Изд. ТПУ, 2001. – 15 с.
12. СНиП 23-05-95. Естественное и искусственное освещение.

13. СанПиН 2.2.4/2.1.8.562-96. Шум на рабочих местах в помещениях жилых, общественных зданий и на территории жилой застройки.
14. СанПиН 2.2.2.542-96. Гигиенические требования к видеодисплейным терминалам, ПЭВМ и организация работы.
15. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
16. ОНТП 24-86. Определение категорий помещений и зданий по взрывопожарной и пожарной опасности.
17. ГОСТ 12.0.003-74. Опасные и вредные производственные факторы.
18. ГОСТ Р 50923-96. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения.
19. Mamdani E. H. Applications of fuzzy algorithms for simple dynamic plant. Proc. IEE., 1974.
20. Zade L. A. The concept of a linguistic variable and its application to approximate reasoning. Part 1, 2, 3 // Information Sciences, n. 8 pp.199-249, pp.301-357; n. 9 pp. 43-80.
21. F. Fujitec, FLEX-8800 series elevator group control system, Fujitec Co., Ltd., Osaka, Japan, 1988.
22. Прасов, М. Т. Алгоритм дефаззификации при синтезе нечеткого регулятора автоматизированных систем контроля и управления // Промышленные АСУ и контроллеры : Ежемесячный научно-технический производственный журнал . — М. — 2006. — № 6. — С. 41-42.

Приложение А

(Обязательное)

Методические указания к выполнению лабораторной работы «Цифровые регуляторы, построенные на нечеткой логике»

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Декан АВТФ

_____ Гайворонский С.А.

___ _____ 2016г.

ЦИФРОВЫЕ РЕГУЛЯТОРЫ, ПОСТРОЕННЫЕ НА НЕЧЕТКОЙ ЛОГИКЕ

Методические указания к выполнению лабораторных работ по курсу

“Автоматизированные информационно-управляющие системы”

для студентов

специальности 210200-”Управление и информатика в технических системах”

Томск 2016

Цель работы: Знакомство с системой программирования контроллера Кросс 500 и изучение цифровых регуляторов, построенных на нечеткой логике.

1 СТРУКТУРА НЕЧЕТКОГО РЕГУЛЯТОРА

Обобщенная структура нечеткого регулятора имеет вид, представленный на рисунке А.1.

При построении нечеткого регулятора исходят из предположения, что эксперты в состоянии сформировать базу правил в форме правил:

ЕСЛИ <предпосылка> ТО <вывод>

и базу данных с функциями принадлежности для предпосылок $\mu(e)$ и выводов $\mu(u)$, то есть определить все необходимые лингвистические правила с лингвистическими переменными и термами.

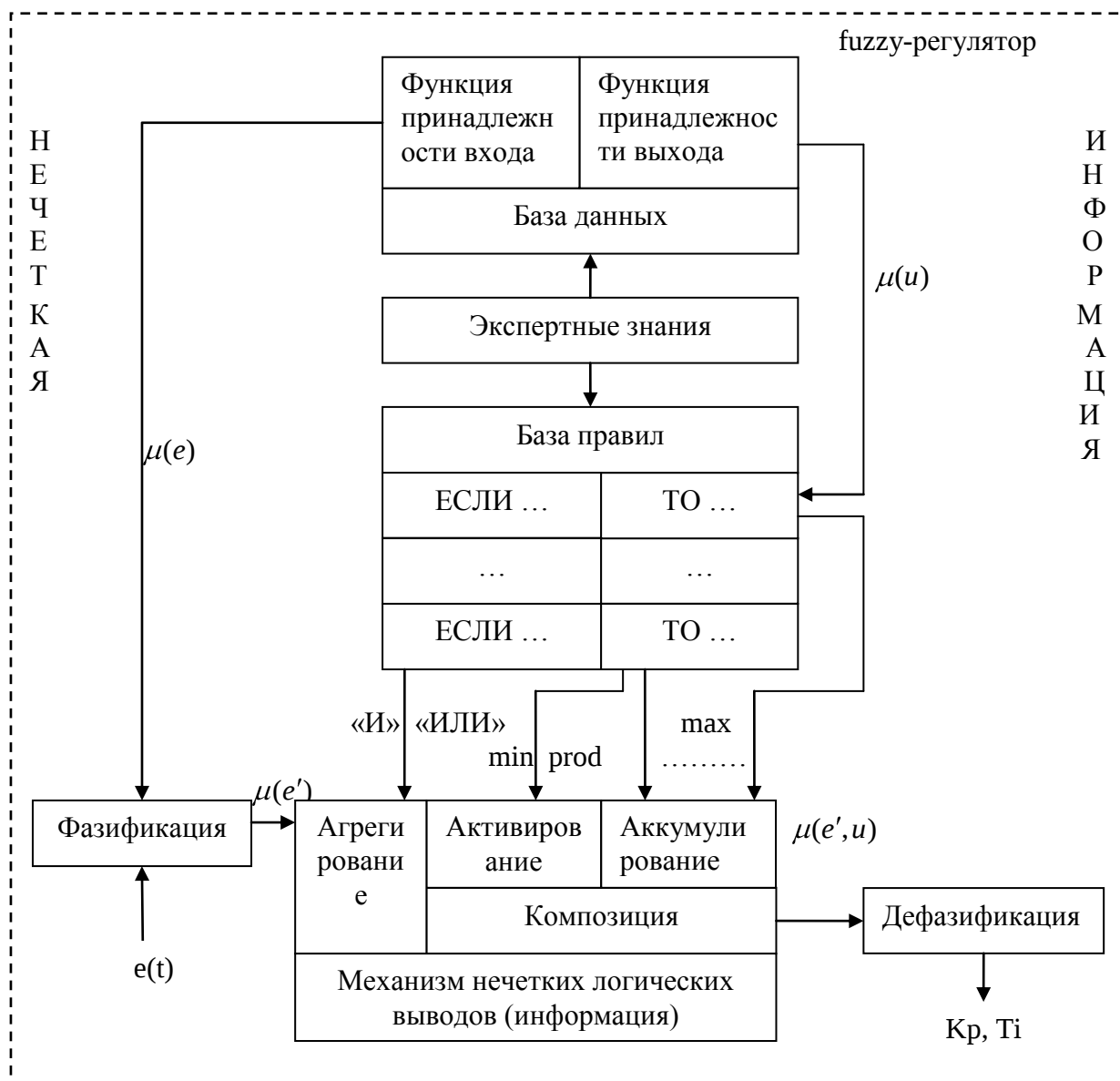


Рисунок А.1 – Обобщенная структура нечеткого ПИ-регулятора

2 Нечеткая логика, ее основные свойства и выводы

Опираясь на одно из самых распространенных определений, логика - это анализ методов рассуждений. В процессе изучения этих методов, логика интересуется в первую очередь формой, а не содержанием доводов в том или ином рассуждении. Ложность или истинность отдельных заключений или посылок не интересует логику. Ее интересует лишь, следует или нет достоверность заключения из достоверности посылок. Одной из основных задач логики является систематическая формализация и каталогизация верных методов рассуждений.

В логике из простых высказываний путем соединения их различными способами можно составить новые, более сложные высказывания.

Механизм нечетких выводов, который используется в различных экспертных и управляющих системах, в основе своей несет базу знаний, которая в свою очередь формируется специалистами и выглядит следующим образом:

П1: если x есть $A1$, тогда y есть $B1$,

П2: если x есть $A2$, тогда y есть $B2$,

.....

Пn: если x есть A_n , тогда y есть B_n ,

где x – переменная на входе (известные значения), y – переменная вывода (данные, которые будут вычисляться); A и B – функции принадлежности, которые определены на x и y .

Пример более подробного объяснения. Опыт эксперта $A \rightarrow B$ показывает нечеткое причинное отношение заключений. В следствии ранее сказанного назовем его нечетким отношением и обозначим через R :

$$R = A \rightarrow B,$$

где « $\rightarrow$ » называется нечеткой импликацией. $R = A \rightarrow B$ по другому можно назвать нечетким подмножеством прямого произведения $X * Y$ полного множества предпосылок X и заключений Y . Отсюда следует, что процедура нахождения нечеткого результата вывода B' , используя данное наблюдение A' и знания, $A \rightarrow B$ можно записать формулой:

$$B' = A' \circ R = A' \circ (A \rightarrow B),$$

где « $\circ$ » - описанная ранее операция свертки. Как процесс операции композиции, так и процесс операции импликации в алгебре нечетких множеств возможно реализовывать разными способами, при этом,

естественно, будет отличаться и результат, но в итоге общий логический вывод будет осуществляться в четыре этапа [6,20].

1. Фазификация (или приведение к нечеткости). На данном этапе используют функции принадлежности для определения степени истинности каждого правила.

2. Логический вывод. Полученную степень истинности для каждого утверждения нужно применить к заключениям этих утверждений. В результате перейдем к нечеткому подмножеству, которое определяется для каждого правила каждой переменной вывода. В качестве правил логического вывода используют операции $\min$ (МИНИМУМ) или prod (УМНОЖЕНИЕ). В логическом выводе МИНИМУМА функция принадлежности вывода «отсекается» по высоте, которая соответствует вычисленной степени истинности предпосылки правила (нечеткая логика «И»). В логическом выводе УМНОЖЕНИЯ функция принадлежности вывода масштабируется при помощи вычисленной степени истинности предпосылки правила.

3. Композиция. Все нечеткие подмножества, которые были определены для каждой переменной вывода для всех правил, объединяются и формируют одно нечеткое подмножество для каждой переменной вывода. При использовании данного объединения обычно пользуются операцией $\max$ (МАКСИМУМ) или sum (СУММА). При композиции МАКСИМУМА комбинированный вывод нечеткого подмножества конструируется как поточечный максимум по всем нечетким подмножествам (нечеткая логика «ИЛИ»). При композиции СУММЫ комбинированный вывод нечеткого подмножества строится как поточечная сумма по всем нечетким подмножествам, назначенным переменной вывода правилами логического вывода.

4. Дефазификация (или приведение к четкости) используется в случае, если необходимо преобразовать нечеткий набор выводов в четкое число. Имеется множество способов приведения к четкости, несколько из них будут рассмотрены ниже.

Пример. Пусть определены нечеткие правила:

П1: если x есть A , тогда w есть D ,

П2: если y есть B , тогда w есть E ,

П3: если z есть C , тогда w есть F ,

где x , y и z – входные переменные, w - переменная вывода, а A , B , C , D , E , F - заданные функции принадлежности (треугольной формы).

Процедура получения логического вывода показана на рисунке А.2.1.

Можно предположить, что переменные на входе принимают некоторые конкретные (четкие) значения - x_0 , y_0 и z_0 .

Согласно приведенным этапам, на первом этапе находятся степени истинности $\alpha(x_0)$, $\alpha(y_0)$ и $\alpha(z_0)$ для всех приведенных правил (см. рисунок 1.3.1).

На втором этапе происходит «отсекание» функций принадлежности заключений на уровнях $\alpha(x_0)$, $\alpha(y_0)$ и $\alpha(z_0)$.

На третьем этапе оперируем уже усеченными функциями принадлежности, после чего они объединяются, используя операцию $\max$. В итоге получим комбинированное нечеткое подмножество, которое описывается ФП $\mu_e(w)$. Это и будет логический вывод w .

На четвертом этапе находим четкое значение выходной переменной, например, воспользовавшись любым из рассмотренных ранее методов дефазификации.

Рассмотрим наиболее часто используемые алгоритмы нечеткого вывода, применив базу знаний в виде двух правил::

П1: если x есть $A1$ и y есть $B1$, тогда z есть $C1$,

П2: если x есть $A2$ и y есть $B2$, тогда z есть $C2$,

где x и y – имена входных переменных, z – имя переменной вывода, $A1, A2, B1, B2, C1, C2$ – некоторые заданные функции принадлежности, при этом

четкое значение z_0 необходимо определить основываясь приведенной информацией и четких значений x_0 и y_0 .

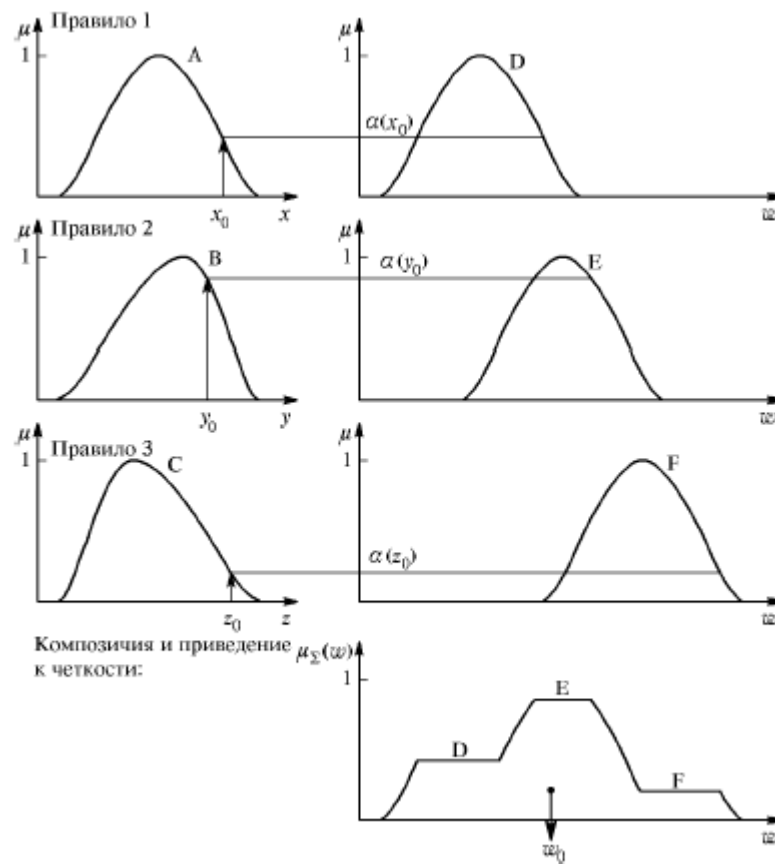


Рисунок А.2.1 – Иллюстрация к процедуре логического вывода

Алгоритм Mamdani: Данный алгоритм соответствует рассмотренному примеру и рисунку 1.1.3. В ситуации, которую мы рассматриваем, запишем математически:

1. Определяем степени истинности для каждого правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(y_0)$, $B_2(y_0)$
2. Отсекаем функции принадлежности для каждого правила с помощью операции МИНИМУМ

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

где через обозначение « $\wedge$ » показана операция логического минимума, далее определяются усеченные функции принадлежности:

$$C'_1(z) = (\alpha_1 \wedge C_1(z)),$$

$$C'_2(z) = (\alpha_2 \wedge C_2(z)).$$

3. Используя операцию МАКСИМУМ (обозначаемой « $\vee$ ») объединяем найденные выше усеченные функции. В итоге получили нечеткое подмножество для переменной выхода. Ее функция принадлежности выглядит следующим образом:

$$\mu_{\Sigma}(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

4. Нахождение четкого значения z_0 используя, к примеру, центроидным методом [6,19].

Алгоритм Tsukamoto. Исходные посыпки соответствуют предыдущему алгоритму, но в конкретном примере строится предположение, что функции $C_1(z_1)$ и $C_2(z_2)$ являются монотонными.

1. Первый этап точно такой же, как в предыдущем алгоритме.

2. На этапе логического вывода сначала находятся уровни «отсечения» α_1 и α_2 , а затем – посредством решения уравнений:

$$\alpha_1 = C_1(z), \quad \alpha_2 = C_2(z),$$

находятся четкие значения (z_1 и z_2) для каждого из исходных правил.

3. затем находится четкое значение переменной вывода как взвешенное среднее значений z_1 и z_2 :

$$z_0 = \frac{\alpha_1 \cdot z_1 + \alpha_2 \cdot z_2}{\alpha_1 + \alpha_2}.$$

Общий случай (дискретный центроидный вывод).

$$z_0 = \frac{\sum_{i=1}^n \alpha_i \cdot z_i}{\sum_{i=1}^n \alpha_i}.$$

Пример: Пусть имеем $A_1(x_0)=0,7$, $A_2(x_0)=0,6$, $B_1(y_0)=0,3$, $B_2(y_0)=0,8$, соответствующие уровни отсечения

$$\alpha_1 = \min(A_1(x_0), B_1(y_0)) = \min(0,7; 0,3) = 0,3.$$

$$\alpha_2 = \min(A_2(x_0), B_2(y_0)) = \min(0,6; 0,8) = 0,6.$$

и значения $z_1=8$ и $z_2=4$, найденные в результате решения уравнений:

$$C_1(z_1) = 0,3, \quad C_2(z_2) = 0,6.$$

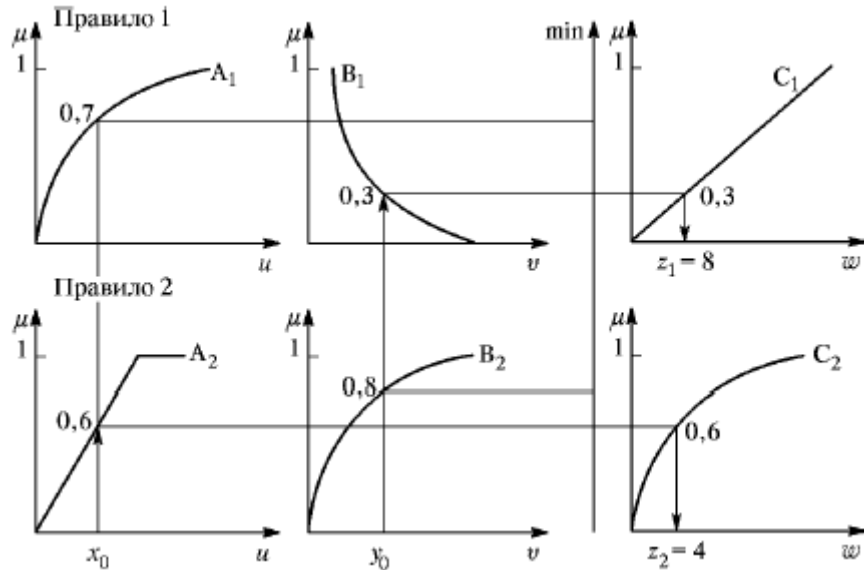


Рисунок А.2.2 – Иллюстрации к алгоритму Tsukamoto

При этом четкое значение переменной вывода изображено на рисунке А.2.2.

$$z_0(8 \cdot 0,3 + 4 \cdot 0,6) / (0,3 + 0,6) = 6.$$

Алгоритм Sugeno. Sugeno и Takagi применяли набор правил следующей формы:

П1: если x есть A_1 и y есть B_1 , тогда $z_1 = a_1 \cdot x + b_1 \cdot y$,

П2 если x есть A_2 и y есть B_2 , тогда $z_2 = a_2 \cdot x + b_2 \cdot y$.

Представление алгоритма:

1. Первый этап соответствует алгоритму Mamdani.

2. На втором этапе отыскиваются $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$

и частные выходы правил:

$$z_1^* = a_1 \cdot x_0 + b_1 \cdot y_0,$$

$$z_2^* = a_2 \cdot x_0 + b_2 \cdot y_0.$$

3. Далее находится четкое значение:

$$z_0 = \frac{\alpha_1 \cdot z_1^* + \alpha_2 \cdot z_2^*}{\alpha_1 + \alpha_2}.$$

На рисуне А.2.3 показаны иллюстрации алгоритма Sugeno

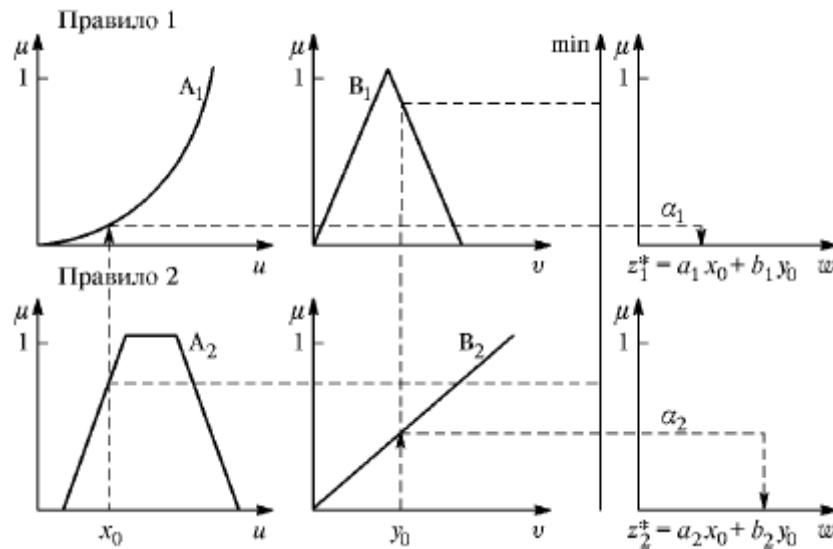


Рисунок А.2.3 –алгоритм Sugeno

Алгоритм Larsen. В алгоритме Larsen нечеткая операция моделируется с применением оператора умножения.

Описание алгоритма:

1. Первый этап точно такой же, как в первом рассматриваемом алгоритме.

2. На тапе логического вывода вначале находим значения

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

затем находятся частные нечеткие подмножества

$$\alpha_1 C_1(z), \quad \alpha_2 C_2(z).$$

3. Далее происходит нахождение итогового нечеткого подмножества с функцией принадлежности:

$$\mu_{\Sigma}(z) = C(z) = (\alpha_1 C_1(z)) \vee (\alpha_2 C_2(z)),$$

в общем случае n правил $\mu_{\Sigma}(z) = C(z) = \bigcup_{i=1}^n (\alpha_i C_i(z))$.

4. В случае, если необходимо, осуществляется приведение к четкости (точно так же, как в алгоритмах, рассмотренных ранее).

Алгоритм Larsen иллюстрируется рисунком А.2.4.

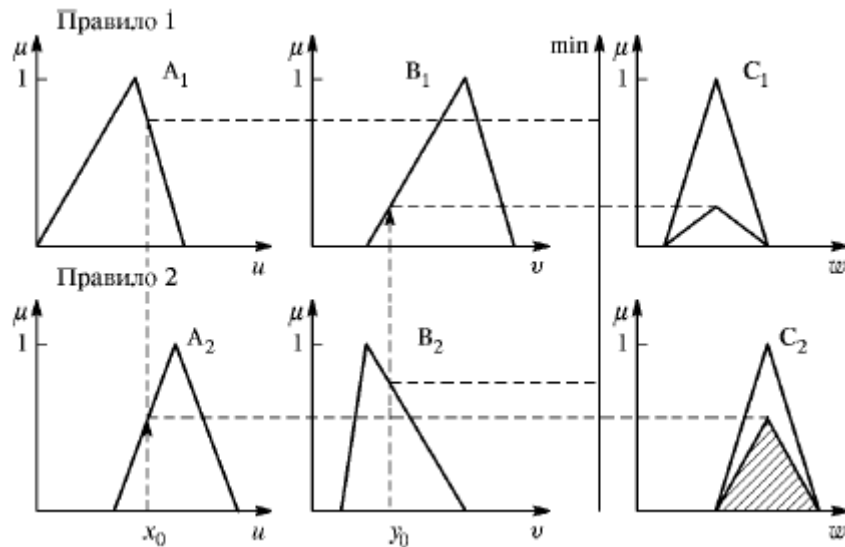


Рисунок А.2.4 – Алгоритм Larsen

Упрощенный алгоритм нечеткого вывода. Правила в данном случае имеют вид:

П1: если x есть A_1 и y есть B_1 , тогда $z_1 = c_1$,

П2: если x есть A_2 и y есть B_2 , тогда $z_2 = c_2$,

где c_1 и c_2 - обычные (четкие) числа.

Описание алгоритма:

1. Первый этап точно такой же, как алгоритм Mamdani.
2. На втором этапе находятся числа:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0).$$

3. На третьем этапе осуществляется поиск четкого значения выходной переменной используя формулу:

$$z_0 = \frac{\alpha_1 \cdot c_1 + \alpha_2 \cdot c_2}{\alpha_1 + \alpha_2}.$$

Или, в общем случае наличия n правил, по формуле:

$$z_0 = \frac{\sum_{i=1}^n \alpha_i \cdot c_i}{\sum_{i=1}^n \alpha_i}.$$

Иллюстрация алгоритма приведена на рисунке А.2.5

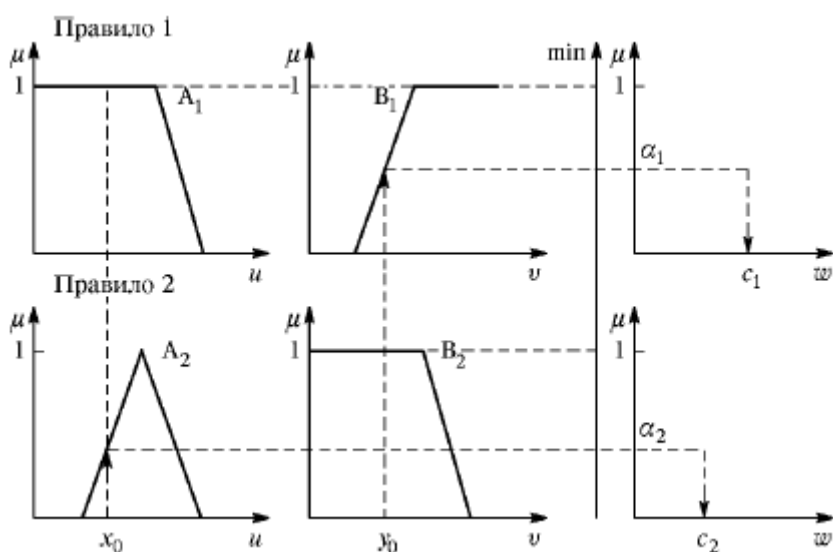


Рисунок А.2.5 – Иллюстрация упрощенного алгоритма нечеткого вывода

3 ФАЗИФИКАЦИЯ И ДЕФАЗИФИКАЦИЯ

Фазификация. Под фазификацией принято понимать процесс поиска значений функций принадлежности (термов), основываясь на обычных (четких) входных значениях. Фазификацию по другому можно назвать введением нечеткости.

Целью фазификации можно считать поиск соответствия между конкретным значением отдельной входной переменной системы нечеткого вывода и значением функции принадлежности соответствующего ей термина входной лингвистической переменной. В результате завершения этапа фазификации всем входным переменным должны соответствовать

индивидуальные значения функций принадлежности по каждому из лингвистических термов, которые были использованы в подусловиях базы правил системы нечеткого вывода.

Формально процедура фазификации проходит в определенной последовательности. Перед тем, как начать этап фазификации, предполагается, что конкретные значения всех входных переменных известны, то есть $V' = \{a_1, a_2, \dots, a_n\}$. Обобщенно каждое $a_i \in X_i$, где X_i - универсум лингвистической переменной β_i . Далее отдельно анализируется каждое из подусловий вида " β_i есть a " правил системы нечеткого вывода, где a' - некоторый терм с функцией принадлежности $\mu(x)$. При этом значение a' используется как аргумент $\mu(x)$, тем самым находится количественное значение $\beta'_i = \mu(a_i)$. Данное значение и есть результат фазификации подусловия " β_i есть a ".

Этап фазификации можно считать законченным тогда, когда для каждого из подусловий всех правил значения $\beta'_i = \mu(a_i)$ будут найдены. Это множество значений можно обозначить через $B = \{\beta'_i\}$. Если некоторый терм a' лингвистической переменной β_i отсутствует во всех нечетких высказываниях, то значение функции принадлежности, которое ему соответствует, не находится в процессе фазификации.

Пример: В качестве иллюстрации рассмотрим пример процесса фазификации трех высказываний: «Расход цемента малый», «Расход цемента средний», «Расход цемента высокий» для входной лингвистической переменной β_i - Расход цемента на кубический метр раствора. Им соответствуют нечеткие высказывания первого вида: " β_1 есть a_1 ", " β_1 есть a_2 ", " β_1 есть a_3 ". Предположим, на данный момент расход цемента равен 55 кг, то есть $a_1 = 55 \text{ кг}$.

Тогда фазификация первого нечеткого высказывания дает в результате число 0, означающее степень его истинности и получается в результате

подстановки значения $a_1 = 55 \text{ кг}$ как аргумент функции принадлежности терма a_1 (рисунок А.3.1 А). Фазификация второго высказывания даст в результате число приблизительно равное 0,67, которое означает степень его истинности и получается в результате подстановки значения $a_1 = 55 \text{ кг}$ как аргумент функции принадлежности терма a_2 (рисунок А.3.1 Б). Фазификация третьего высказывания, так же как и в первом случае даст число 0, и будет означать его степень истинности (рисунок А.3.1 В).

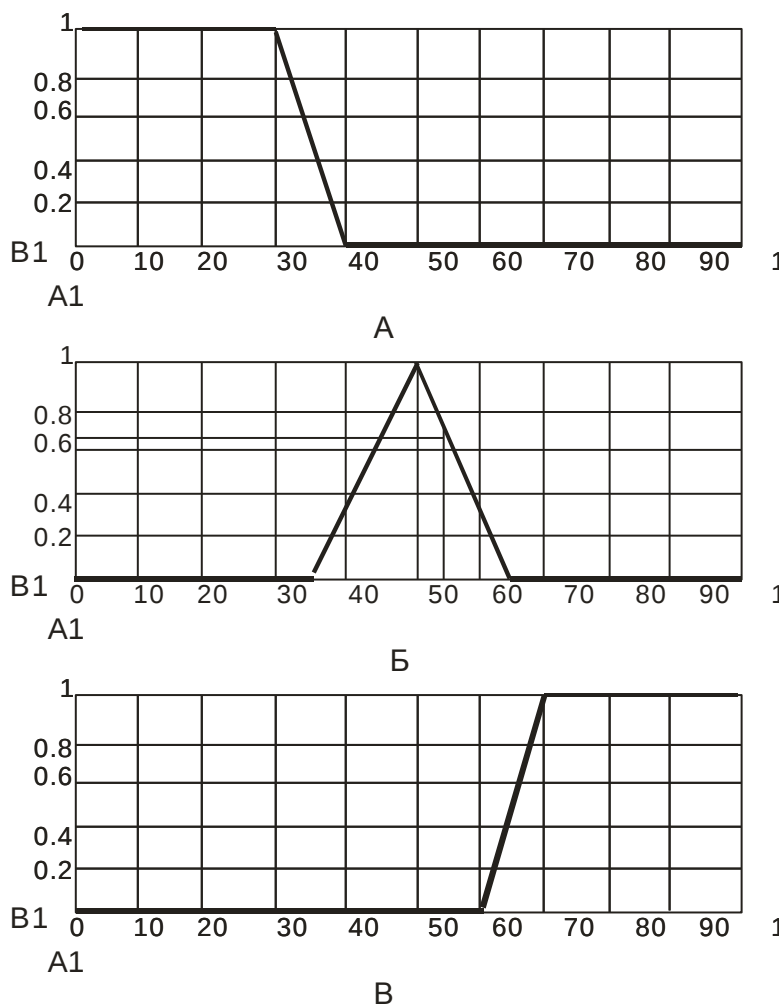


Рисунок А.3.1 – Пример фазификации входной лингвистической переменной “Расход цемента” для трех нечетких высказываний

Дефазификация. Под дефазификацией понимается процедура поиска четкого значения для каждой из входных лингвистических переменных множества $W = \{w_1, w_2, \dots, w_s\}$.

Цель дефазификации заключается в следующем: используя результаты накопления всех выходных лингвистических переменных, необходимо найти четкое число для каждой из выходных переменных. По другому этап дефазификации можно называть приведением к четкости.

Процедура дефазификации. функции принадлежности всех выходных лингвистических переменных, до того как начать этап дефазификации, следует принять известными и записать в форме нечетких множеств:

$$C'_1, C'_2, \dots, C'_s,$$

где s – общее количество выходных лингвистических переменных согласно базы правил. Далее постепенно анализируем каждую выходную лингвистическую переменную $w_j \in W$, к которой относится нечеткое множество C'_j . Результком этапа дефазификации для w_j следует считать количественное значение $y_j \in R$, применив любой из методов.

Этап дефазификации можно считать законченным, когда каждой выходной лингвистической переменной найдено определенное значения в численном виде

$$y_1, y_2, \dots, y_s,$$

где s – количество лингвистических переменных согласно базы правил.

Для нахождения численных значений используют ниже описанные формулы (методы дефазификации).

Метод центра тяжести

Центр тяжести (CoG, COG, Center of Gravity) или центроид площади рассчитывается по формуле:

$$y = \frac{\int_{Min}^{Max} x \cdot \mu(x) dx}{\int_{Min}^{Max} \mu(x) dx},$$

где y является результатом дефазификации; x – соответствующее выходной лингвистической переменной w_s значение переменной; $\mu(x)$ -

соответствующая выходной переменной w после этапа накопления функция принадлежности; Min и Max – левая и правая точки интервала носителя.

В данном методе обычное (четкое) значение выходной переменной равно абсциссе центра тяжести площади, ограниченной графиком кривой функции принадлежности, которая соответствует входной переменной.

Пример дефазификации ФП выходной лингвистической переменной «Расход цемента» выше описанным методом изображен на рисунке А.3.2. В этом случае $y_1 = 40 \text{ кг}$ (приближенное значение).

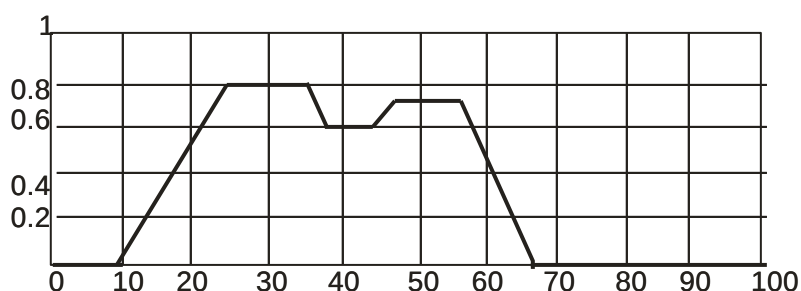


Рисунок А.3.2– Дефазификация методом центра тяжести

Метод центра тяжести для одноточечных множеств.

Центр тяжести для одноточечных множеств (COGS, Centre of Gravity for Singletons) можно рассчитать по формуле:

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)},$$

где n – число одноточечных (одноэлементных) нечетких множеств, каждое из которых характеризует единственное значение рассматриваемой выходной лингвистической переменной.

Пример дефазификации данным методом выходной лингвистической переменной «Расход цемента» изображен на рисунке А.3.3. В этом случае $y_1 = 41 \text{ кг}$ (приближенное значение).

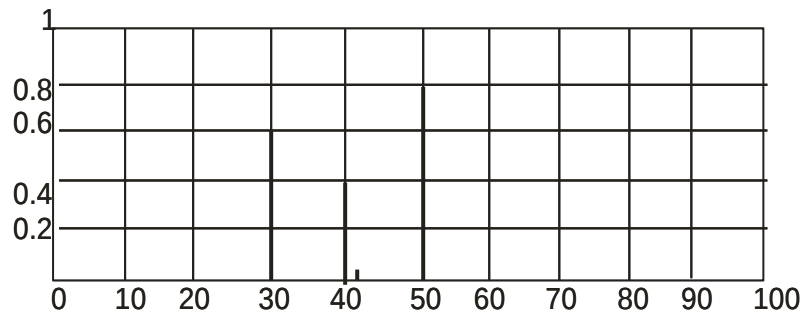


Рисунок А.3.3 – Пример дефазификации выходной лингвистической переменной «Расход цемента» методом центра тяжести для одноточечных множеств

Метод центра площади

Центр площади (CoA, COA, Centre of Area, Bisector of Area)

$y = u$, u определяется уравнением:

$$\int_{\min}^u \mu(x) dx = \int_u^{\max} \mu(x) dx .$$

Пример дефазификации методом центра площади функции принадлежности выходной лингвистической переменной «Расход цемента» изображен на рисунке А.3.4. В этом случае $y_1 = 35$ кг

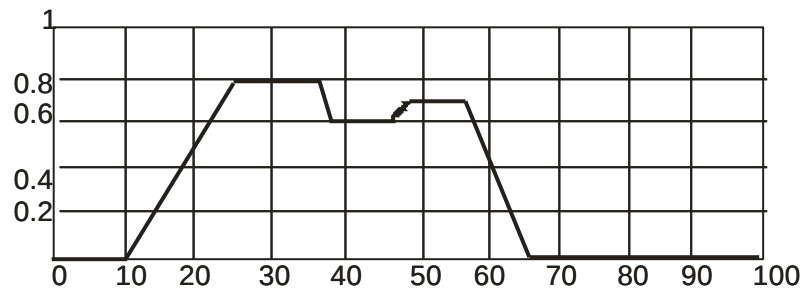


Рисунок А.3.4– Дефазификация выходной лингвистической переменной «Расход цемента» методом центра площади

Иначе говоря центр площади равняется абсциссе, которая делит площадь, ограниченную графиком кривой функции принадлежности соответствующей выходной переменной, на две равные части.

Метод левого модального значения

Левое модальное значение (LM, Left Most Maximum) можно рассчитать по формуле:

$$y = \min\{x_m\},$$

где x_m - модальное значение (мода) нечеткого множества, соответствующего выходной переменной w после аккумуляции.

Другими словами, значение выходной переменной определяется как мода нечеткого множества для соответствующей выходной переменной или наименьшая из мод (самая левая), если нечеткое множество имеет несколько модальных значений.

Пример дефазификации методом левого модального значения функции принадлежности выходной лингвистической переменной «Расход цемента» изображен на рисунке А.3.5. В этом случае $y_1=24$ кг.

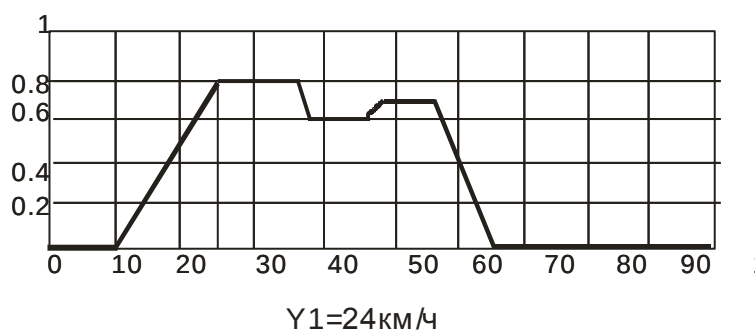


Рисунок А.3.5 – Пример дефазификации выходной лингвистической переменной «Расход цемента» методом левого модального значения

Метод правого модального значения

Правое модальное значение (RM, Right Most Maximum) рассчитывается по формуле:

$$y = \max\{x_m\},$$

где x_m - модальное значение (мода) нечеткого множества, соответствующего выходной переменной w после аккумуляции.

В этом случае значение выходной переменной также определяется как мода нечеткого множества для соответствующей выходной переменной или наибольшая из мод (самая правая), если нечеткое множество имеет несколько модальных значений.

Пример дефазификации методом правого модального значения функции принадлежности выходной лингвистической переменной «Расход цемента» изображен на рисунке А.3.6. В этом случае $y_1=54$ кг (приближенное значение).

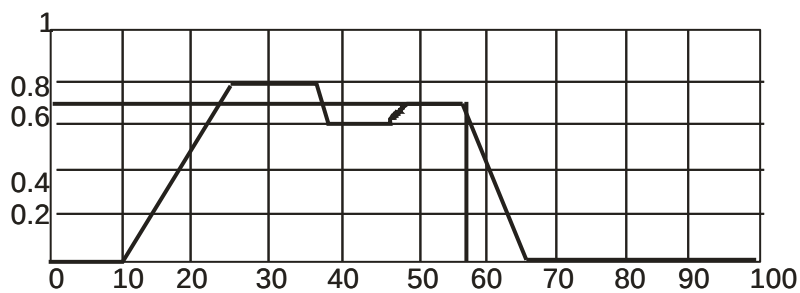


Рис. А.3.6 – Пример дефазификации выходной лингвистической переменной «Расход цемента» методом правого модального значения

4 СТРУКТУРА ЛАБОРАТОРНОГО КОМПЛЕКСА

Данный лабораторный комплекс состоит из микропроцессорного контроллера КРОСС, аналогового вычислительного комплекса АВК-6, персонального компьютера с ОС Windows NT и автоматического выключателя (АВ). В свою очередь, контроллер КРОСС состоит из центрального блока, блока питания, трех модулей УСО (аналогового, дискретного и термопреобразовательного) и соответствующих им терминальных блоков. В состав центрального блока входит центральный процессор, модуль локальной сети ETHERNET (субмодуль SM2-ETH) и модуль питания. Структурная схема комплекса приведена на рисунке А.4, где показаны модули и связи между ними.

Центральный блок контроллера (ЦБ1) связан с модулями УСО шиной SPI-1. Внешние связи могут подключаться к модулям УСО через терминальные блоки, которые содержат клеммные колодки. Модуль УСО АЮ1-8/4 поддерживает связь с объектом управления (АВК-6) через терминальный блок Т1-АЮ.

Питание комплекса производится от сети переменного тока 220 В (50 Гц) через автоматический выключатель. Блок питания ЛОК 4601-2R/P выдает на контроллер постоянное напряжение 24 В, которое преобразуется модулем питания КР-DC24V1 в 5 В, питающего шину SPI-1.

Центральный процессор контроллера SM2-CPU-1,5 соединен с операторской станцией через субмодуль SM2-ETH, обеспечивающей соединение по локальной сети ETHERNET. Субмодуль и плата центрального процессора имеют общую лицевую панель и занимают в ЦБ1 одно посадочное место.

Лабораторный стенд позволяет осуществлять проверку работоспособности систем автоматического регулирования с регуляторами, реализованными программно на контроллере КРОСС. Модель объекта

управления набирается на АВК-6. АВК-6 позволяет моделировать работу объекта до шестого порядка включительно. В процессе проверки работоспособности с помощью экранных форм, реализованных на SCADA, можно менять уставки регуляторов, коэффициенты настройки регуляторов и в графическом виде наблюдать за свойствами САР в режиме реального времени.

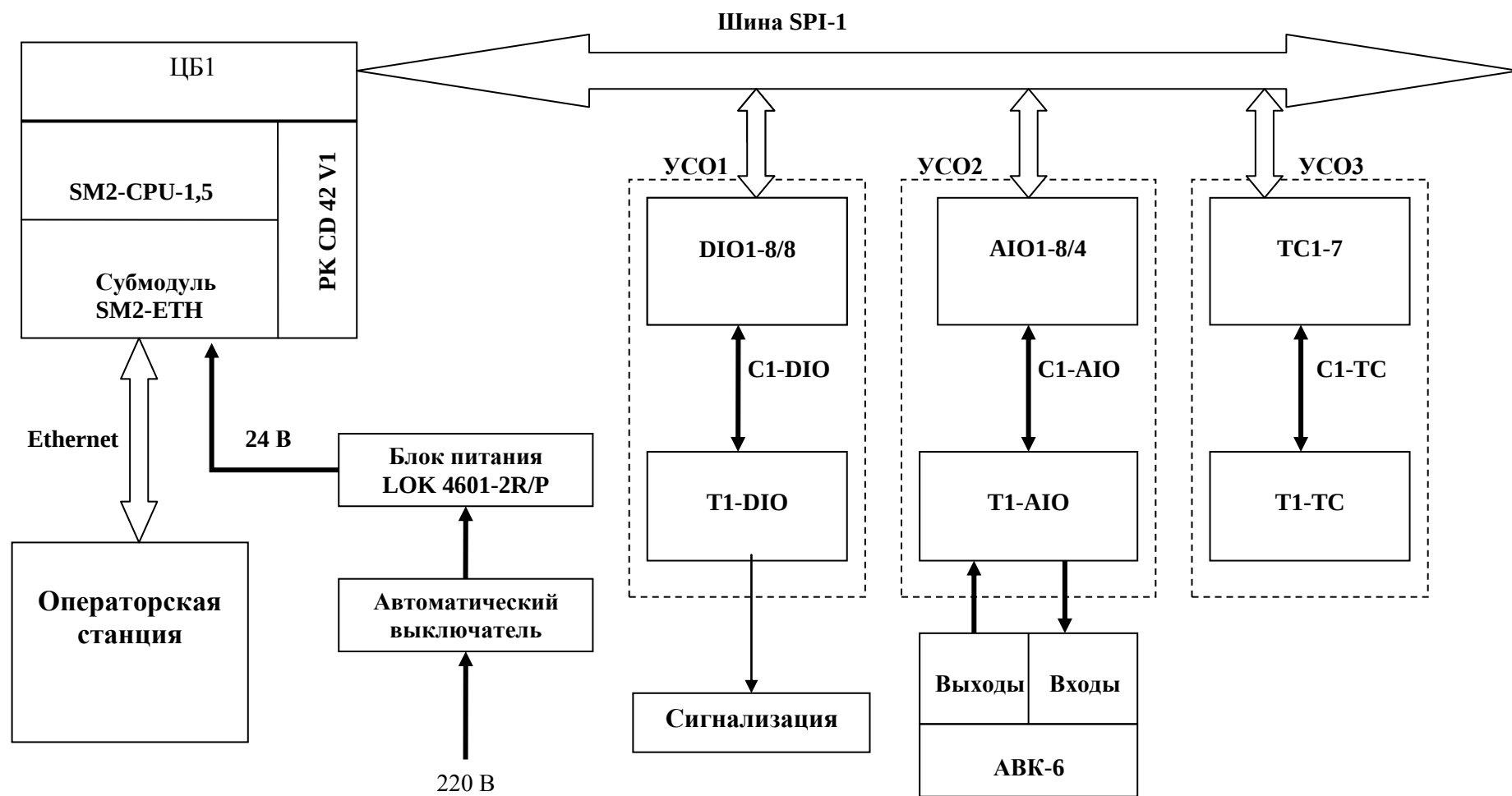


Рисунок А.4 – Структурная схема лабораторного комплекса

5 СИСТЕМА ПРОГРАММИРОВАНИЯ КОНТРОЛЛЕРА

Технологическое программирование контроллера осуществляется в системах ISaGRAF и UltraC/C++.

Рассмотрим систему программирования ISaGRAF, так как для обычного пользователя из промышленности, привыкшего к программированию традиционных программируемых логических контроллеров (PLC), операционная система OS-9 может оказаться не столь простой в изучении. А система ISaGRAF является достаточно доступной и простой.

ISaGRAF – наиболее известная в мире и удачная реализация стандарта МЭК 1131-3, выполненная фирмой CJ International.

Технология программирования традиционных PLC предполагает использование языков программирования, описываемых стандартом МЭК 1131-3:

- язык управляющих последовательностей (SFC);
- язык функциональных блоков (FBD);
- язык релейной логики (LD);
- структурированный текст (ST);
- язык инструкций (IL).

Привычный для многих потребителей язык функциональных блоков FBD в системе ISaGRAF построен на библиотеке элементарных арифметических, логических и динамических функций и может быть расширен библиотекой алгоритмов контроллера Р-130.

Проект ISaGRAF разделен на несколько программных модулей, называемых программами. Программы проекта связаны в древовидную

структуру. Команда **Соединение В/В** запускает редактор соединения переменных ISaGRAF. Этот инструмент используется для создания связей между объявленными в словаре переменными ввода/вывода и соответствующей аппаратурой ввода/вывода.

После написания программы, генерирования кода проекта, соединения ввода/вывода можно симулировать отладку. Команда **Симулировать** открывает отладчик в режиме симуляции. Эта программа очень полезна, когда целевая машина не доступна.

Команда **Установление связей** позволяет пользователю определить параметры связи, используемые для коммуникации между отладчиком на главном компьютере и целевой системой ISaGRAF.

Команда **Отладка** открывает главное окно отладчика при условии, что код сгенерирован и связь с системой установлена.

С помощью средства ISaGRAF «**Прожектор**» пользователь может определить списки наблюдения, которые могут показывать либо графические картинки, либо списки во время отладки. Графические символы должны быть связаны с переменными ISaGRAF. Картинки могут быть определены и анимированы во время работы.

6 SCADA – СИСТЕМА Master SCADA

6.1 Назначение SCADA-систем, виды SCADA-систем

SCADA (сокр. от англ. Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) – процесс сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами.

Применение SCADA-систем позволяет существенно сократить сроки разработки программного обеспечения, обеспечить высокое качество регулирования, при этом при создании программного обеспечения профессиональные программисты могут и не привлекаться.

Все современные SCADA-системы включают три основных структурных компонента, представленные на рисунке А.6.1.1.

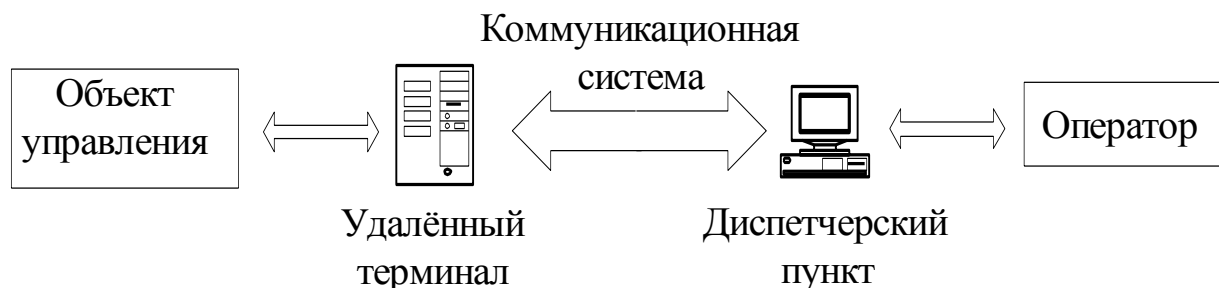


Рисунок А.6.1.1 - Основные структурные компоненты SCADA системы

Удаленный терминал (Remote Terminal Unit - RTU) осуществляет обработку задачи (управление) в режиме реального времени.

Диспетчерский пункт управления (Master Terminal Unit - MTU) осуществляет обработку данных и управление высокого уровня, как правило, в режиме реального времени. Одна из его основных функций – обеспечение интерфейса между человеком-оператором и системой.

Коммуникационная система (каналы связи – Communication System (CS)) необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на диспетчерский пункт (или удаленный объект – в зависимости от конкретного исполнения системы).

К SCADA-системам предъявляются следующие основные требования:

- надежность системы (технологическая и функциональная);
- безопасность управления;
- точность обработки и представления данных;
- простота расширения системы.

В силу тех требований, которые предъявляются к системам SCADA, спектр их функциональных возможностей определен и реализован практически во всех пакетах. Перечислим основные возможности и средства, присущие всем системам и различающиеся только техническими особенностями реализации:

- автоматизированная разработка, дающая возможность создания программного обеспечения (ПО) системы автоматизации без реального программирования;
- средства сбора первичной информации от устройств нижнего уровня;
- средства управления и регистрации сигналов об аварийных ситуациях;
- средства хранения информации с возможностью ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);
- средства обработки первичной информации;
- средства визуализации представления информации в виде графиков, гистограмм и т.п.;
- возможность работы прикладной системы с наборами параметров, рассматриваемых как единое целое (gesire , или установки).

На современном мировом рынке программного обеспечения существует большое количество фирм, выпускаемой продукцией которых являются SCADA-пакеты. Ниже перечислены только некоторые из популярных на западном и российском рынках SCADA-систем, имеющих некоторую поддержку в России (таблица А.6.1).

Таблица А.6.1 - Основные фирмы разработчики SCADA- пакетов

Фирма разработчик	Страна	Фирма распространитель в России	Название SCADA- пакета
AdAstra Res. Group	Россия	AdAstra Res. Group	TRACE MODE
Intellution	США	Индасофт	FIX
Ci Technologies	Австралия	РТСофт	Citect
Wonderware	США	РТСофт	InTouch
Iconics	США	ПроСофт	Genesis
Инсат	Россия	Инсат	VNS MasterSCADA

Перечисленные выше возможности систем SCADA в значительной мере определяют стоимость и сроки создания ПО, а также сроки ее окупаемости.

6.2 Отличительные особенности пакета программ MasterScada

MasterScada - это не просто один из современных SCADA-пакетов, это принципиально новый инструмент разработки АСУТП, в котором реализована совокупность средств и методов, обеспечивающих резкое сокращение трудозатрат и повышение надежности создаваемой системы. MasterScada является полнофункциональным SCADA-пакетом программ с расширяемой функциональностью. Пакет построен на клиент-серверной архитектуре с возможностью функционирования, как в локальных сетях, так и в Интернете. Прием и передача данных и сообщений на основе стандартов OPC встроена в ядро пакета. Максимальная поддержка всех стандартов (XML, HTML, ODBC, OLE, COM/DCOM, ActiveX и др.) и открытые описания интерфейсов и форматов данных обеспечивают все необходимые возможности для стыковки с внешними программами и системами.

Основные преимущества пакета:

1. Единая среда разработки АСУТП.
2. Раздельное конфигурирование структуры АСУТП и логической структуры объекта.
3. Открытость и следование стандартам.
4. Интуитивная легкость освоения.
5. Удобство инструментария.
6. Удобство методики разработки
7. Мощная трехмерная графика и мультимедиа.
8. Неограниченная гибкость вычислительных возможностей.
9. Объектный подход.

Пользовательский интерфейс MasterScada (рис.2.1) построен на идеологии «все в одном». Все модули расширения встроены в общую оболочку. Пользователь всегда работает с простым единым внешним видом программы, состоящим из древовидного проекта, палитры библиотечных элементов и окна редактирования документов и свойств. В зависимости от типа настраиваемого свойства или редактируемого документа в окне редактирования открывается страница настройки нужного свойства, либо необходимый встроенный или внешний редактор. Например, встроенный редактор мнемосхем или внешний редактор текстовых описаний.

Проект состоит из двух разделов: «Система» и «Объект». Раздел «Система» описывает техническую структуру реализуемой системы. Раздел «Объект» описывает иерархическую структуру контролируемого технологического объекта свойства и документы каждого объекта.

Окно проекта состоит из четырех основных частей (рис. А.6.2.1):

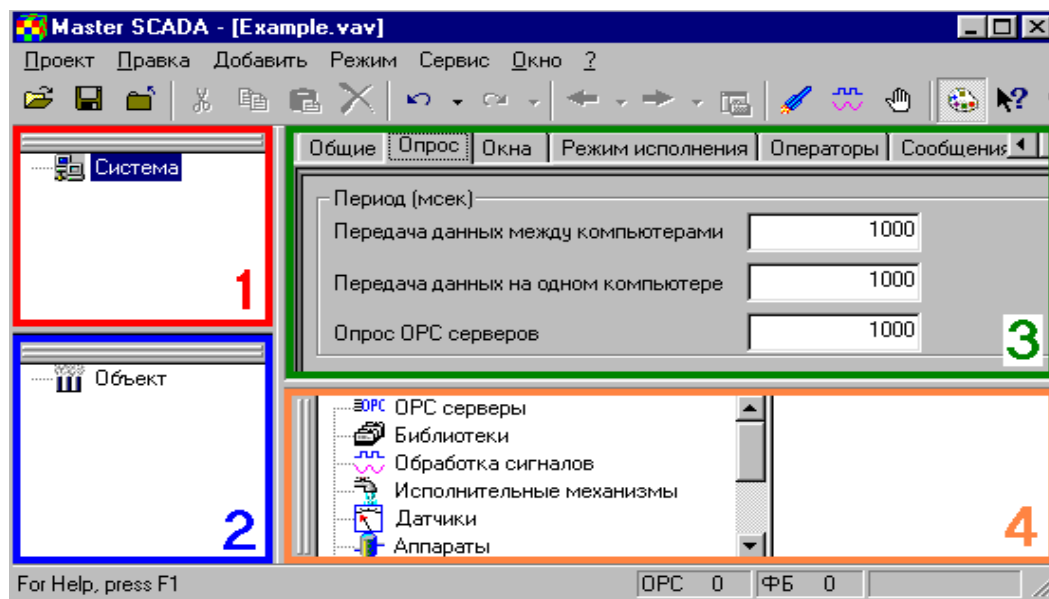


Рисунок А.6.2.1 - Основные структурные компоненты SCADA системы

1. Дерево системы, в котором отображены элементы конфигурации такие, как компьютеры, OPC серверы, и т. д.
2. Дерево объектов, включающее в себя объекты, переменные, группы переменных, функциональные элементы.
3. Страницы свойств элементов, на которых производятся все необходимые настройки элементов.
4. Палитра элементов, из которой берутся библиотечные объекты, функциональные блоки и т. д.

Размеры дерева системы, дерева объектов и палитры можно изменять и они могут располагаться в любом месте экрана. Страницы свойств занимают все оставшееся пространство.

Встроенные программные средства позволяют:

- Вести обработку данных.
- Составлять сценарии наступления событий.

- Создавать схемы и мнемосхемы, тренды.
- Выдавать сообщения и составлять рапорты.
- Вести архивирование.
- Обмениваться данными по сети.
- Ограничивать права доступа и вести контроль действий оператора.
- Отлаживать созданные программы.
- Повышать надежность работы.

Режимы работы

Существует три возможных варианта работы в режиме исполнения.

Рабочий режим. Включается командой "Пуск" из меню "Режим", или кнопкой  на панели инструментов. Это основной режим исполнения. В Рабочем режиме должен быть осуществлен переход к нему на всех компьютерах системы. Программа производит реальное управление технологическим процессом.

Режим отладки. Включается командой "Отладка" из меню "Режим", или кнопкой  на панели инструментов. Этот режим предназначен для отладки проекта на одном компьютере. Независимо от того, сколько компьютеров находится в дереве системы, все объекты, функциональные блоки, OPC серверы создаются на текущем компьютере и все действия производятся на нем.

Режим имитации. Включается командой "Имитация" из меню "Режим", или кнопкой  на панели инструментов. В этом режиме на все входы, не

имеющие связей, вместо констант будет подаваться имитация в соответствии с настройками.

7 OPC сервер

Для связи между каналами и переменными в приложении ISaGRAF используется OPC сервер контроллера КРОСС 500.

Назначение:

CrossOPC сервер (далее OPC-сервер) предназначен для подсоединения и обмена данными между ISaGRAF-приложением контроллера Ремиконт Р-130ISa и SCADA-системами. OPC-сервер работает под управлением ОС Windows NT и предоставляет данные OPC-клиентам через интерфейс, определяемый спецификацией OPC Data Access 2.0).

С ISaGRAF-приложениями OPC-сервер соединяется по последовательному порту или по протоколу TCP/IP. Формат файла (рисунок А.7.1)

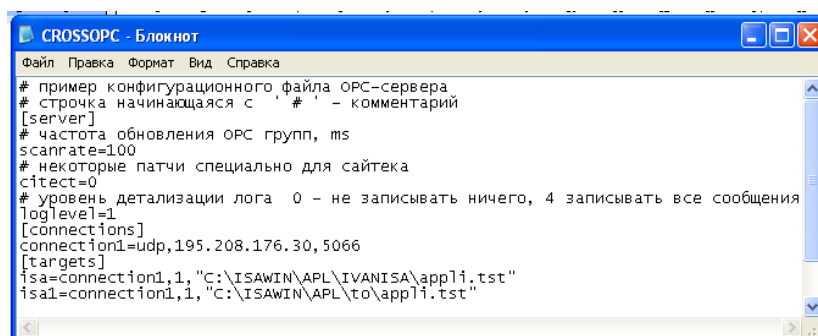


Рисунок А.7.1 – Блокнот CROSSOPC

Это текстовый файл в стиле стандартных ini -файлов windows, расположенный по адресу: C:\Program files\OPC Server\CROSSOPC. Каждая строка содержит либо комментарий, начинающийся с символа "точка с запятой", либо заголовок секции, либо набор параметров секции. Имена параметров к регистру нечувствительны.

Секция [SERVER] может содержать следующие параметры:

LOGFILE = "<file >" Указывает имя файла для записи протокола работы сервера. Имя файла необходимо указывать в кавычках.

LOGLEVEL = <level> Число от 0 до 3, показывающее, какие сообщения будут выводиться в лог. Параметр 0- означает не выводить сообщения совсем, 1- только ошибки, 2- ошибки предупреждения, 3- все сообщения, в том числе и отладочные

SCANRATE = <rate> Целое число, показывающее период опроса контроллера в миллисекундах.

CITECT= <0|1> Если используется SCADA - система Citect, то этот параметр необходимо установить в 1. Этот ключ заставляет OPC- сервер принудительно устанавливать для данных передаваемых в Citect OPC_QUALITY_GOOD, потому что OPC- драйвер Citect работает некорректно: при поступлении хотя бы одного элемента с OPC_QUALITY_BAD остальные элементы тоже рассматриваются как невалидные.

Секция [CONNECTIONS] содержит определения логических подключений. Формат:

<connection_name>=SERIAL,<port >,<speed> для подключения через последовательный порт

< connection_name>=TCP|UDP,<ip_address >,<ip_port > для подключения через TCP/IP

Параметры:

< onnection_name> имя подключения

<port> имя последовательного порта, например com4

<speed>	скорость порта в бодах
<ip_address>	ip – адрес контроллера или dns имя
<ip_port>	ip – порт (по умолчанию 5066)

Секция [TARGETS] содержит определения задач (targets). Под задачей подразумевается исполнительная система ISaGRAF, выполняемая на контроллере. На контроллере может быть запущено несколько задач одновременно, но в стандартной версии на контроллере запускается только одна задача ISaGRAF с логическим номером 1.

Формат:

<targetname>=<connection_name>,<logical_num>,"< appli.tst >"

Таблица А.7.1 Параметры:

<target_name>	имя целевой задачи
<connection_name>	имя подключения, определенное в секции [connections]
<logical_num >	логический номер задачи, обычно 1
<appli.tst >	путь к файлу appli.tst, который содержит определения переменных целевой задачи ISaGRAF

8 ОПИСАНИЕ ПРОГРАММЫ

При выполнении данной лабораторной работы используется программа, реализующая нечеткое ПИ-регулирование. Программа предназначена для изучения цифровых регуляторов, построенных на нечеткой логике. В качестве простого примера возьмем нечеткий ПИ-регулятор,

листинг которой приведен в конце методического пособия. При изменении ошибки регулирования, меняются коэффициенты пропорциональной и интегральной составляющих регулятора. В ходе реализации применялись функции принадлежности, которые приведены ниже:

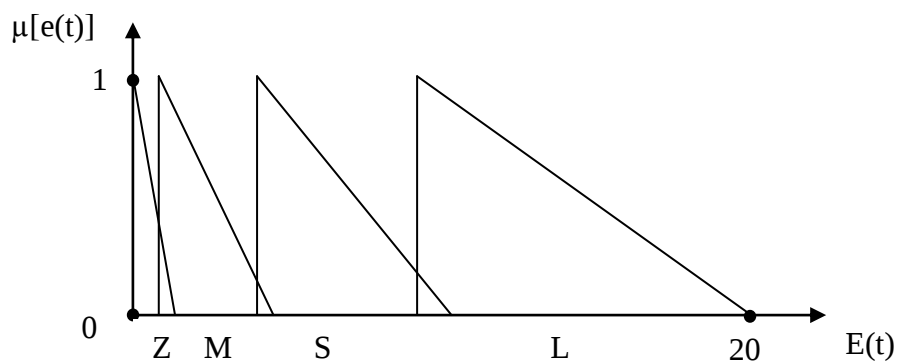


Рисунок А.8.1 Функции принадлежности фазификации по ошибке регулирования

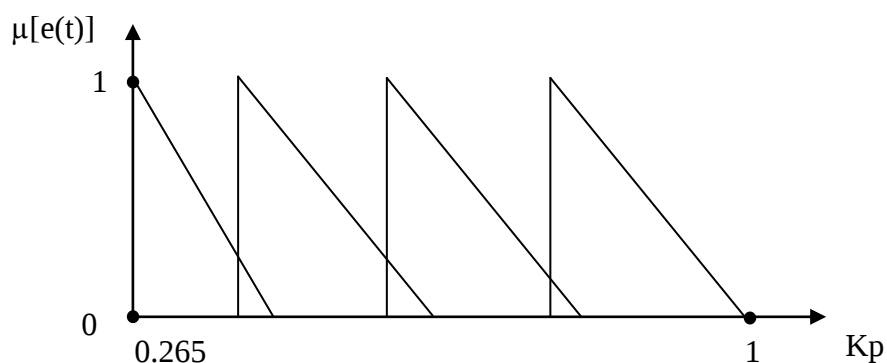


Рисунок А.8.2 Функции принадлежности дефазификации пропорциональной составляющей

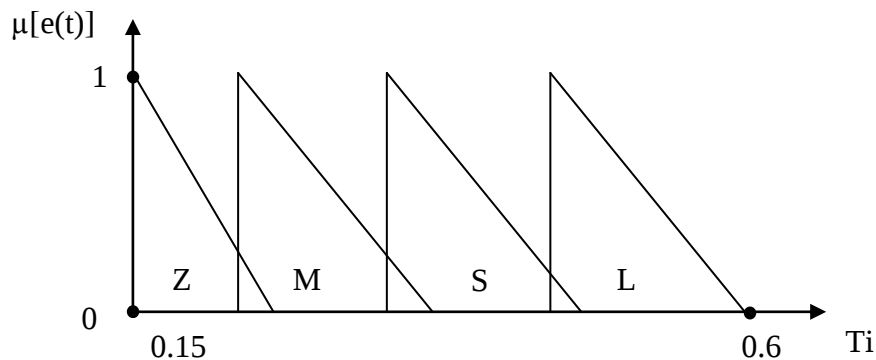


Рисунок А.8.3 Функции принадлежности дефазификации интегральной составляющей

База правил для данного нечеткого ПИ-регулятора выглядит следующим образом:

если $Z \neq 0$ и $M \neq 0$ то M ;

если $M \neq 0$ и $S \neq 0$ то S ;

если $S \neq 0$ и $L \neq 0$ то L ;

если $E=Z$ то $K_p=Z$, $T_i=Z$;

если $E=M$ то $K_p=M$, $T_i=M$;

если $E=S$ то $K_p=S$, $T_i=S$;

если $E=L$ то $K_p=L$, $T_i=L$.

Другими словами, если ошибка очень маленькая, значения коэффициентов принимаем небольшими. При увеличении ошибки, коэффициенты повышаются.

11 ЗАДАНИЕ

1. Набрать модель объекта управления 2 порядка и задать ее параметры (по заданию преподавателя) на АВК-6. Вход и выход модели объекта подключить к адаптеру, обеспечивающему связь с контроллером.

2. На ПК запустить приложение «ПРОЕКТЫ» программы ISaGRAF. Выбрать проект под названием «fuzzyreg» и запустить программу «fuzzyreg».

3. Установить связь с контроллером и запустить отладчик.

4. Задать уставку данной САУ от 0 В до 10 В (по заданию преподавателя).

5. Запустить редактор представления данных, в котором запустить программу «FVVA» в SCADA-системе MASTER SCADA.

6. Изменяя возмущающее воздействие путем изменения коэффициента обратной связи одного из звеньев, обеспечить реализацию переходного процесса и исследовать работу системы с нечетким ПИ-регулятором, то есть проследить в открытой программе, реализованной на Isograf, изменение коэффициентов в зависимости от величины возмущающего воздействия, сравнить работу классического и нечеткого ПИ-регуляторов.

7. Используя графическую форму SCADA-системе MASTER SCADA повторить пункт 6.

8. Скопировать данную программу в новую директорию и на ее основе реализовать нечеткий ПИ-регулятор. Для дефазификации использовать методом максимума, затем повторить пункты 6 и 7.

9. Сравнить результаты реализации данных программ.

Содержание отчета.

1. Цель работы.

2. Обобщенная структура нечеткого регулятора.

3. Структурная схема комплекса.

4. Графики изменения управляющего воздействия нечеткого ПИД-регулятора.

5. Структурная схема созданной программы

6. Ответы на контрольные вопросы.

7. Краткие выводы о проделанной работе.

Контрольные вопросы.

1. В каких системах технологического программирования работает контроллер.

2. Какие существуют методы дефазификации.

3. Что такое фазификация и дефазификация.

4. Для чего предназначено приложение «ПРОЖЕКТОР» в системе ISaGRAF.

5. Назовите состав контроллера.

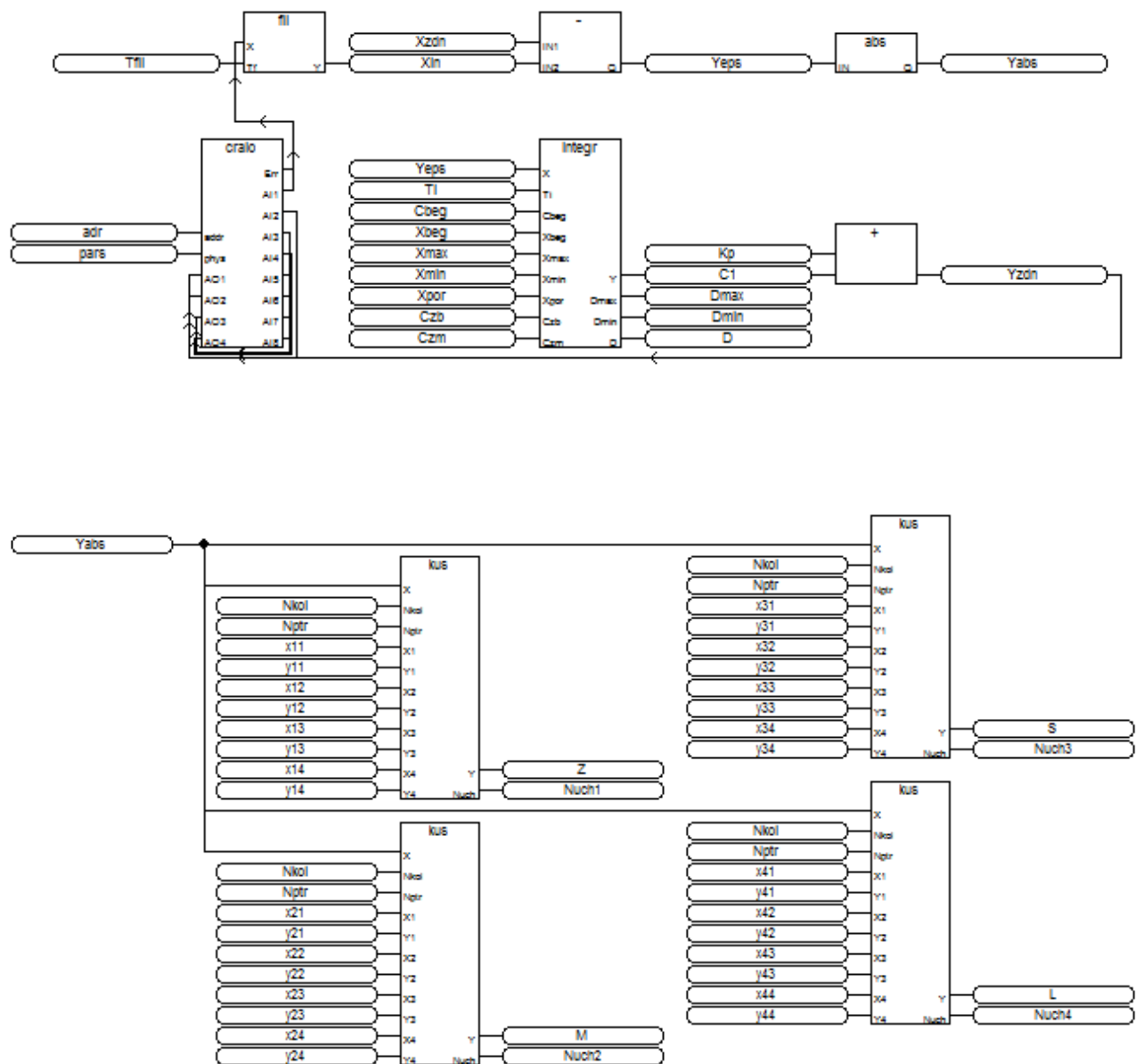
6. Организация SCADA-системы MASTER SCADA

7. Назначение редактора базы каналов, редактора представления данных

Приложение Б

(Обязательное)

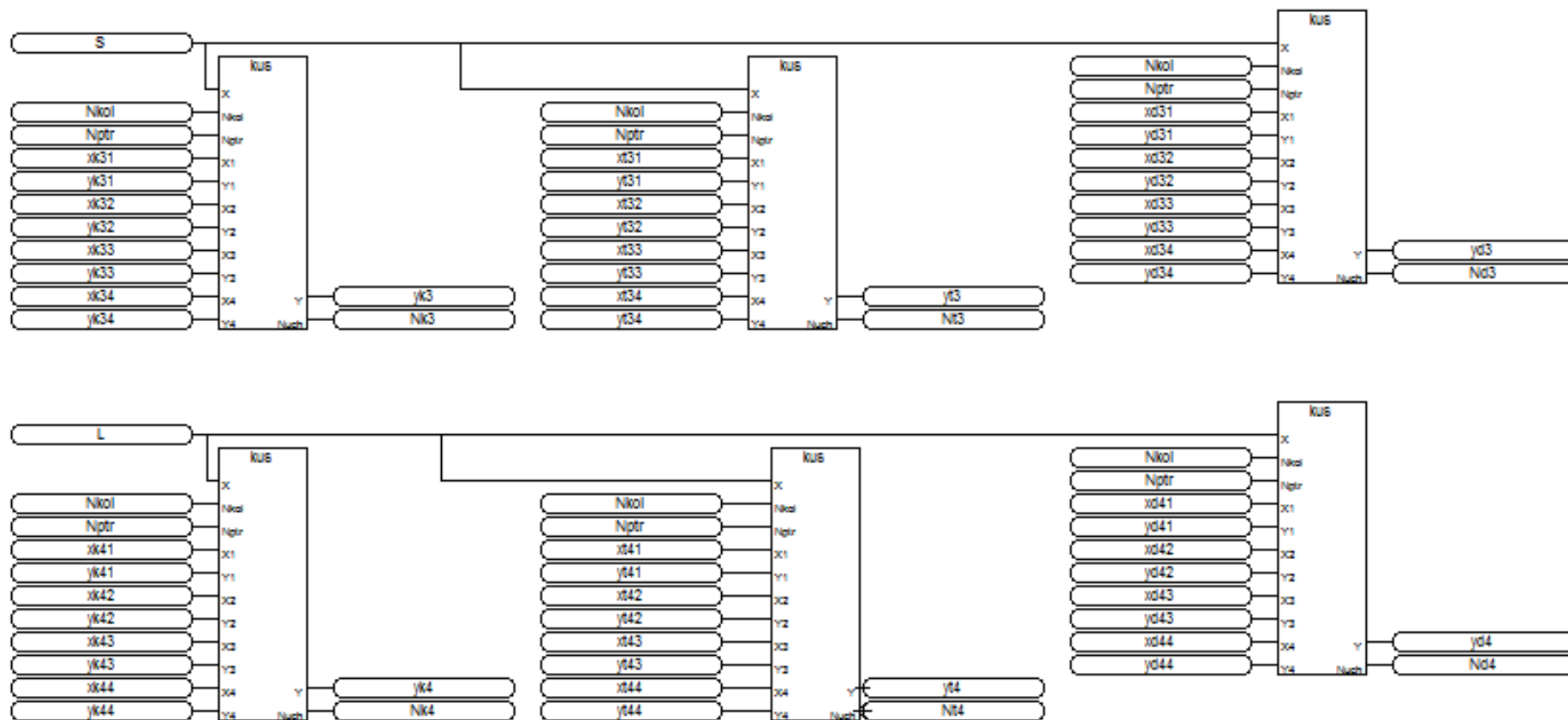
Программа реализации нечеткого ПИ-регулятора



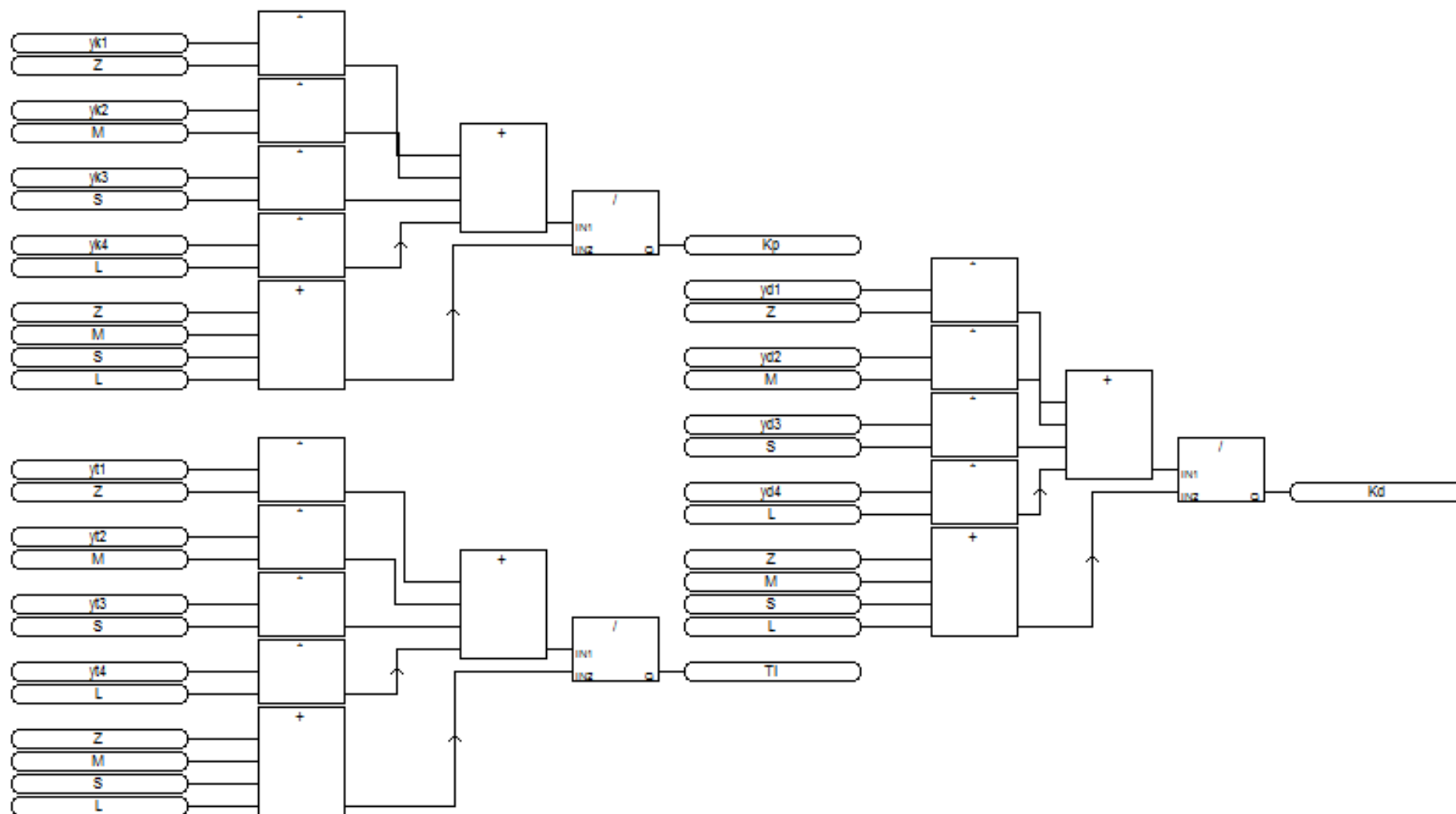
Приложение Б (продолжение)



Приложение Б (продолжение)



Приложение Б (продолжение)



Приложение В
(Обязательное)
Методические указания к выполнению лабораторной работы
«Нечеткий П-регулятор»
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ
Декан АВТФ
_____ Гайворонский С.А.
___ _____ 2016г.

Нечеткий П-регулятор

Методические указания к выполнению лабораторных работ по курсу
“Автоматизированные информационно-управляющие системы”
для студентов
специальности 210201-”Управление и информатика в технических системах”

Томск 2016

Цель работы: Программная реализация нечеткого П-регулятора .

1 Нечеткая логика и ее основные свойства

Согласно одному из самых распространенных определений, логика есть анализ методов рассуждений. Изучая эти методы, логика интересуется в первую очередь формой, а не содержанием доводов в том или ином рассуждении. Истинность или ложность отдельных посылок или заключений не интересует логику. Ее интересует лишь, вытекает ли истинность заключения из истинности посылок. Систематическая формализация и каталогизация правильных способов рассуждений – одна из основных задач логики.

В логике из простых высказываний путем соединения их различными способами можно составить новые, более сложные высказывания.

Используемый в различного рода экспертных и управляющих системах механизм нечетких выводов в своей основе имеет базу знаний, формируемую специалистами предметной области в виде совокупности нечетких предикатных правил вида:

П1: если x есть A_1 , тогда y есть B_1 ,

П2: если x есть A_2 , тогда y есть B_2 ,

.....

П n : если x есть A_n , тогда y есть B_n ,

где x – входная переменная (имя для известных значений данных), y – переменная вывода (имя для значения данных, которое будет вычислено); A и B – функции принадлежности, определенные соответственно на x и y .

Приведем более детальное пояснение. Знание эксперта $A \rightarrow B$ отражает нечеткое причинное отношение предпосылки и заключения, поэтому его можно назвать нечетким отношением и обозначить через R :

$$R = A \rightarrow B,$$

где « $\rightarrow$ » называют нечеткой импликацией. Отношение R можно рассматривать как нечеткое подмножество прямого произведения $X * Y$ полного множества предпосылок X и заключений Y . Таким образом, процесс получения нечеткого результата вывода B' с использованием данного наблюдения A' и знания $A \rightarrow B$ можно представить в виде формулы:

$$B' = A' \circ R = A' \circ (A \rightarrow B),$$

где « $\circ$ » введенная выше операция свертки. Как операцию композиции, так и операцию импликации в алгебре нечетких множеств можно реализовывать по-разному (при этом, естественно, будет разниться и итоговый получаемый результат), но в любом случае общий логический вывод осуществляется за следующие четыре этапа.

1. Нечеткость (введение нечеткости, фазификация, fuzzication). Функции принадлежности, определенные на входных переменных применяются к их фактическим значениям для определения степени истинности каждой предпосылки каждого правила.

2. Логический вывод. Вычисленное значение истинности для предпосылок каждого правила применяется к заключениям каждого правила. Это приводит к одному нечеткому подмножеству, которое будет назначено каждой переменной вывода для каждого правила. В качестве правил логического вывода обычно используются только операции $\min$ (МИНИМУМ) или prod (УМНОЖЕНИЕ). В логическом выводе МИНИМУМА функция принадлежности вывода «отсекается» по высоте, соответствующей вычисленной степени истинности предпосылки правила (нечеткая логика

«И»). В логическом выводе УМНОЖЕНИЯ функция принадлежности вывода масштабируется при помощи вычисленной степени истинности предпосылки правила.

3. Композиция. Все нечеткие подмножества, назначенные к каждой переменной вывода (во всех правилах), объединяются вместе, чтобы формировать одно нечеткое подмножество для каждой переменной вывода. При подобном объединении обычно используются операции $\max$ (МАКСИМУМ) или sum (СУММА). При композиции МАКСИМУМА комбинированный вывод нечеткого подмножества конструируется как поточечный максимум по всем нечетким подмножествам (нечеткая логика «ИЛИ»). При композиции СУММЫ комбинированный вывод нечеткого подмножества конструируется как поточечная сумма по всем нечетким подмножествам, назначенным переменной вывода правилами логического вывода.

4. В заключение (дополнительно) - приведение к четкости (дефазификация, defuzzication), которое используется, когда полезно преобразовать нечеткий набор выводов в четкое число.

Пример: Пусть некоторая система описывается следующими нечеткими правилами:

П1: если x есть A , тогда ϖ есть D ,

П2: если y есть B , тогда ϖ есть E ,

П3: если z есть C , тогда ϖ есть F ,

где x , y и z - имена входных переменных, ϖ - имя переменной вывода, а A , B , C , D , E , F - заданные функции принадлежности (треугольной формы).

Процедура получения логического вывода иллюстрируется рис. 1. Предполагается, что входные переменные приняли некоторые конкретные (четкие) значения - x_0 , y_0 и z_0 .

В соответствии с приведенными этапами, на этапе 1 для данных значений и исходя из функций принадлежности A , B , C , находятся степени истинности $\alpha(x_0)$, $\alpha(y_0)$ и $\alpha(z_0)$ для предпосылок каждого из трех приведенных правил (см. рис. В.1).

На этапе 2 происходит «отсекание» функций принадлежности заключений правил (т. е. D , E , F) на уровнях $\alpha(x_0)$, $\alpha(y_0)$ и $\alpha(z_0)$.

На этапе 3 рассматриваются усеченные на втором этапе функции принадлежности, и производится их объединение с использованием операции $\max$, в результате чего получается комбинированное нечеткое подмножество, описываемое функцией принадлежности $\mu_{\varepsilon}(\varpi)$ и соответствующее логическому выводу для выходной переменной ϖ .

Наконец, на 4-м этапе - при необходимости - находится четкое значение выходной переменной, например, с применением центроидного метода: четкое значение выходной переменной определяется как центр тяжести для кривой $\mu_{\varepsilon}(\varpi)$, то есть

$$\varpi_0 = \frac{\int_{\Omega} \varpi \mu_{\varepsilon}(\varpi) d\varpi}{\int_{\Omega} \mu_{\varepsilon}(\varpi) d\varpi} .$$

Рассмотрим следующие наиболее часто используемые модификации алгоритма нечеткого вывода, полагая, для простоты, что базу знаний реализуют два нечетких правила вида:

П1: если x есть A_1 и y есть B_1 , тогда z есть C_1 ,

П2 если x есть A_2 и y есть B_2 , тогда z есть C_2 ,

где x и y – имена входных переменных, z – имя переменной вывода, $A_1, A_2, B_1, B_2, C_1, C_2$ – некоторые заданные функции принадлежности, при этом четкое значение z_0 необходимо определить на основе приведенной информации и четких значений x_0 и y_0 .

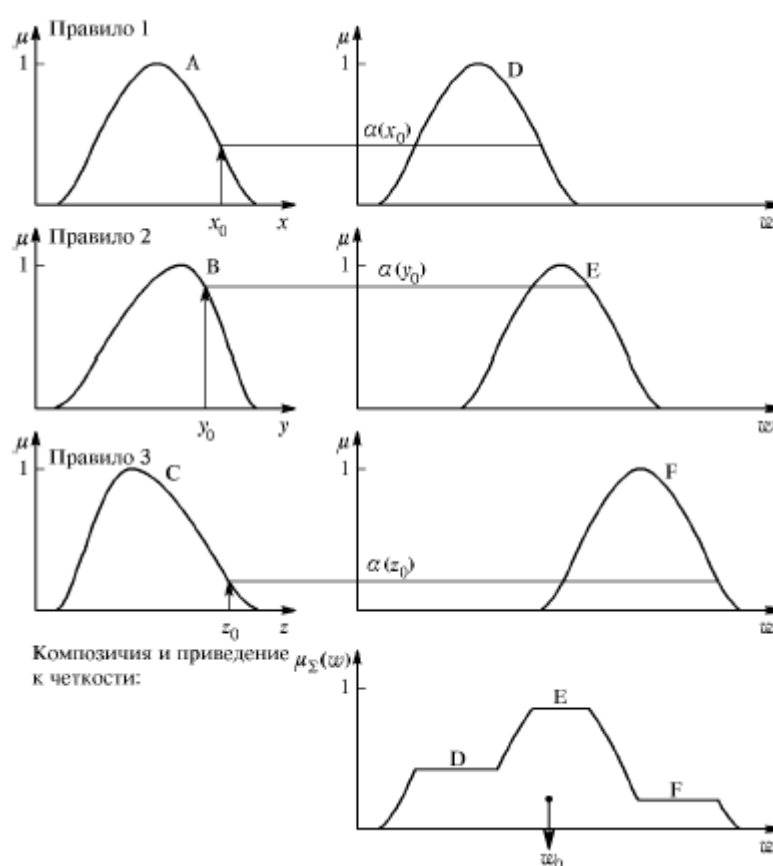


Рисунок В.1– Иллюстрация к процедуре логического вывода

2 Структура САР с нечетким П-регулятором

Особенность нечеткого П-регулятора является изменение в процессе работы коэффициента пропорциональности в соответствии с базой правил. Структурная схема САР с нечетким П-регулятором представлена на рисунке В.2.1.

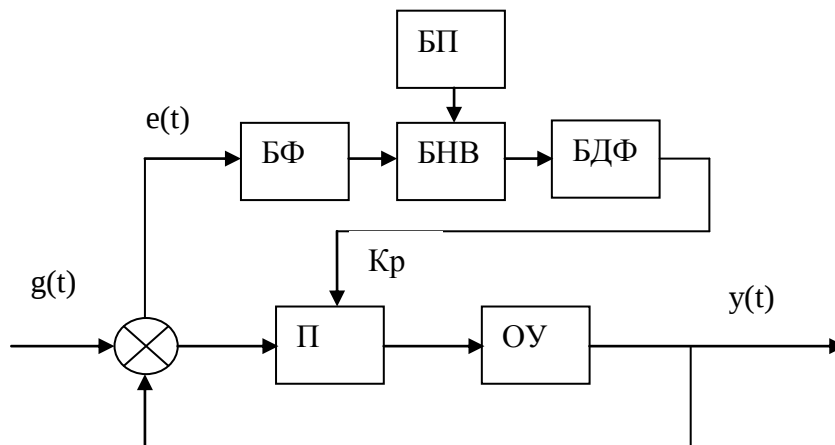


Рис. В.2.1 Структура САР с нечетким П-регулятором

ОУ – Объект управления.

П- П-регулятор.

БФ – Блок фазификации.

БП – База правил.

БНВ – Блок нечеткого вывода.

БДФ – Блок дефазификации.

Под фазификацией понимается процедура нахождения значений функций принадлежности нечетких множеств (термов) на основе обычных (не нечетких) исходных данных. Фазификацию еще называют введением нечеткости. Целью этапа фазификации является установления соответствия между конкретным (обычно – численным) значением отдельной входной переменной системы нечеткого вывода и значением функции принадлежности соответствующего ей терма входной лингвистической переменной. Воспользуемся следующими функциями принадлежности по ошибке регулирования:

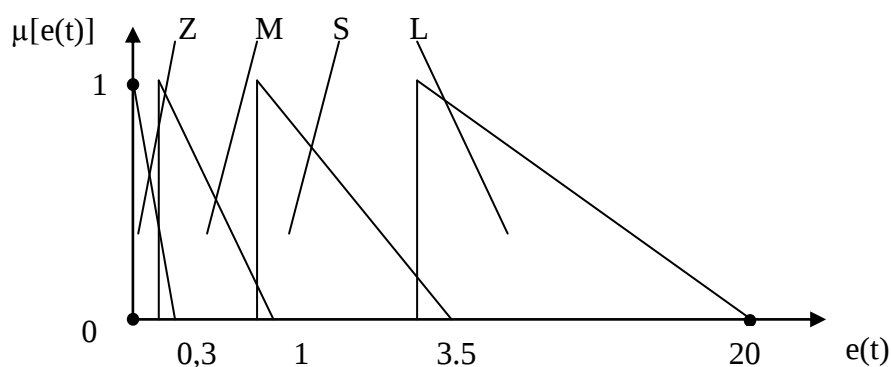


Рисунок В.2.2 Функции принадлежности этапа фазификации

Дефазификация в системах нечеткого вывода представляет собой процедуру или процесс нахождения обычного (не нечеткого) значения для каждой из входных лингвистических переменных множества

Цель дефазификации заключается в том, чтобы, используя результаты аккумуляции всех выходных лингвистических переменных, получить обыкновенное количественное значение каждой из выходных переменных, которое может быть использовано специальными устройствами, внешними по отношению к системе нечеткого вывода. Для этапа дефазификации используем функции принадлежности, изображенные на рис. 4.

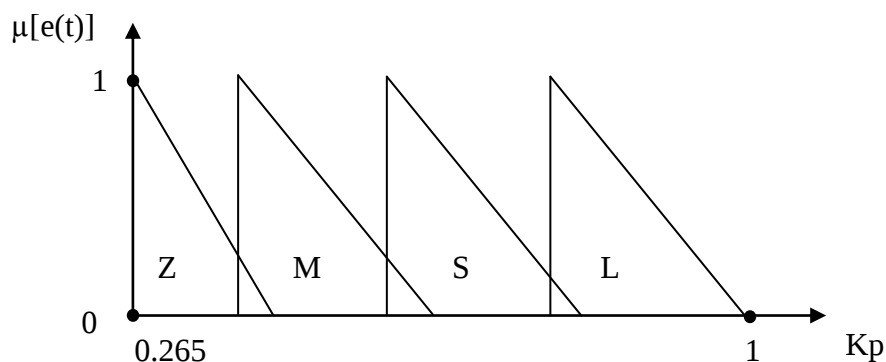


Рисунок В.2.3 Функции принадлежности дефазификации пропорциональной составляющей

База правил для данного нечеткого П-регулятора выглядит следующим образом:

если $Z \neq 0$ и $M \neq 0$ то M ;

если $M \neq 0$ и $S \neq 0$ то S ;

если $S \neq 0$ и $L \neq 0$ то L ;

если $E=Z$ то $K_p=Z$;

если $E=M$ то $K_p=M$;

если $E=S$ то $K_p=S$;

если $E=L$ то $K_p=L$.

Другими словами, если ошибка очень маленькая, значение коэффициента принимаем небольшими. При увеличении ошибки, коэффициент повышается.

3 Программа реализации нечеткого П-регулятора в системе программирования Isograf

Создадим программу в системе ISaGRAF на языке FBD, реализующую регулятор с нечетким ПИ-регулированием. Для этого запускаем приложение **Isograf** через меню **ПУСК/ПРОГРАММЫ** или дважды нажав левую кнопку мыши на приложении **Isograf** на рабочем столе компьютера. В появившемся окне видим список уже существующих программ (рис. В.3.1).

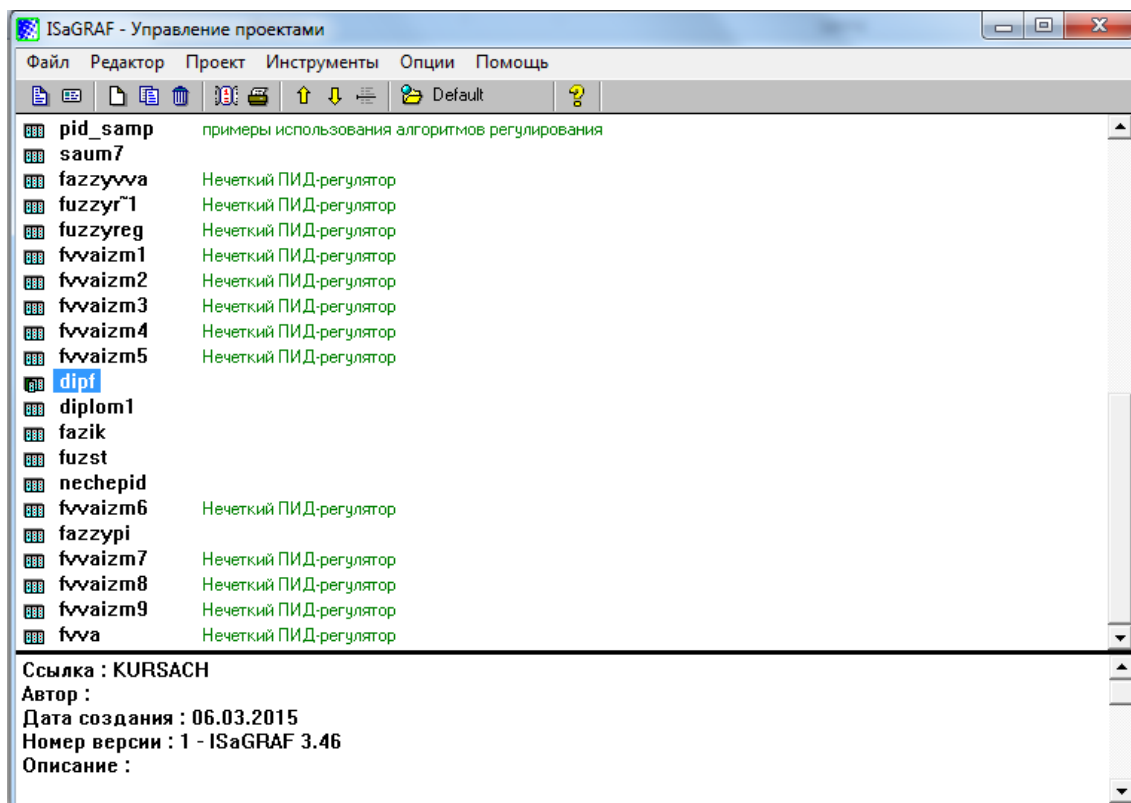


Рисунок В.3.1 – Окно приложения ПРОЕКТЫ

Создадим новый проект с помощью меню **ФАЙЛ/НОВЫЙ** или используя сочетание клавиш **CTRL+N**. После этого на экране появится окно (рис. В.3.2), в котором нужно дать имя проекту, например **FUZZYPI** (обязательно на английском языке).

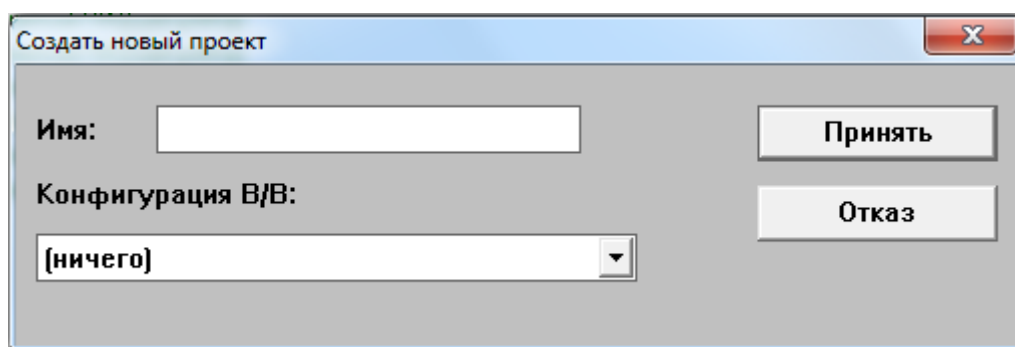


Рисунок В.3.2 – Окно создания нового проекта

После принятия имени, созданный проект отобразится в списке. Нажав два раза левую кнопку мыши на проекте появится окно **ПРОГРАММЫ** (рис. В.3.3), в котором нужно создать новую программу (**ФАЙЛ/НОВЫЙ**).

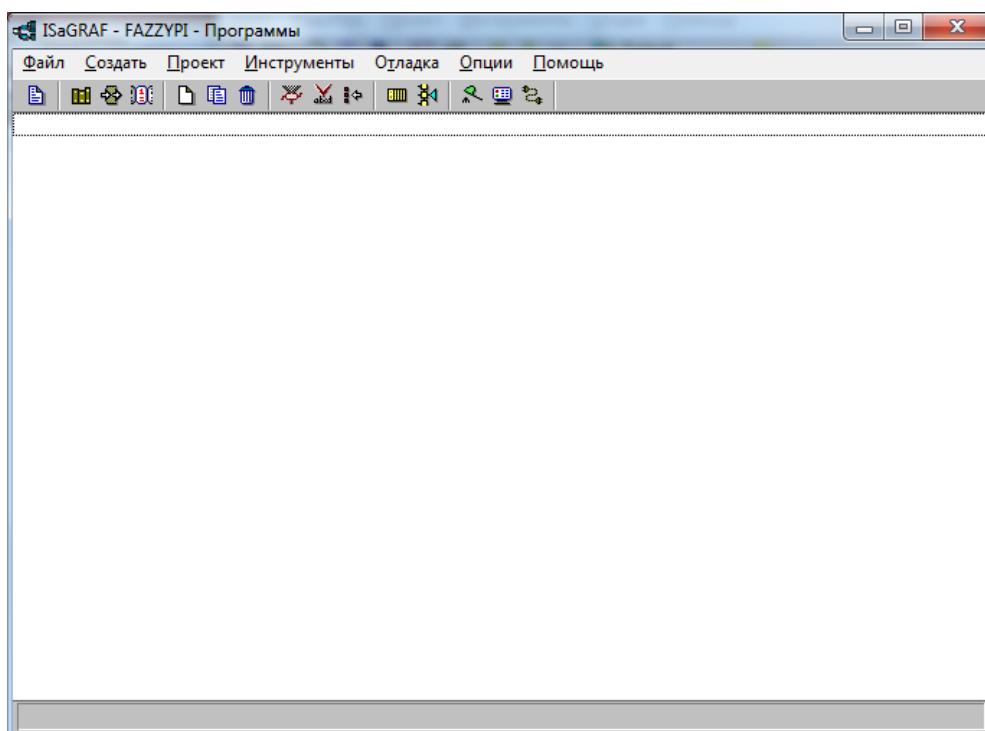


Рисунок В.3.3 – Окно ПРОГРАММЫ

Новой программе необходимо присвоить имя, например **FUZZYPI**, а также указать язык, который будет использоваться. Выбираем **FBD** – **Функциональная блочная диаграмма**, комментарий (по желанию) и стиль **Begin: основная программа** как на рисунок В.3.4

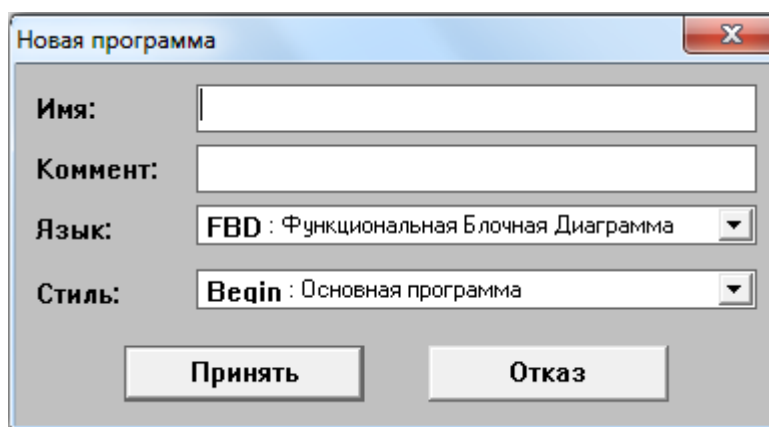


Рисунок В.3.4 – Окно создания новой программы

После того как новая программа создана, в панели инструментов FBD нам необходимо подобрать нужные нам функциональные блоки (рисунок В.3.5)

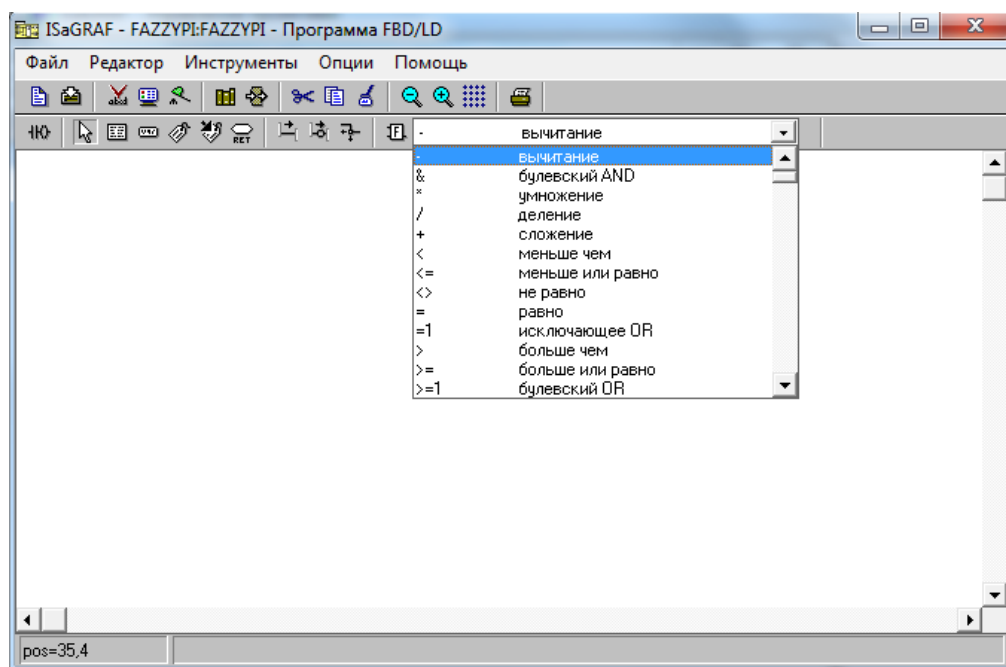
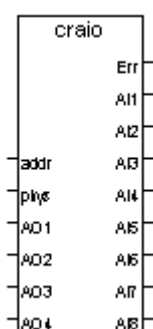


Рисунок В.3.5 – Выбор функциональных блоков

Для реализации нечеткого ПИ-регулятора необходимы следующие функциональные блоки:

CRAIO - Ввод-вывод аналоговых сигналов с модуля УСО КРОСС



Входы: addr – сетевой адрес модуля;

phys – признак необходимости преобразования (TRUE – преобразовывать в

физическую величину, FALSE – не преобразовывать,

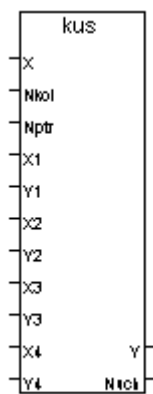
выдавать сигнал в процентах от диапазона измерения);

AI [1-8] – значения входов.

Выходы: AO[1-4] – значения выходов; err – код ошибки.

Данный функциональный блок необходим нам для связи входных и выходных переменных непосредственно с входами-выходами аналогового модуля AIO-8/4.

KUS - кусочно-линейная функция.



Входы: X – основной вход;

Nkol – количество участков;

Nptr – указатель на дополнительные участки;

X1 – абсцисса 1 участка;

Y1 – ордината 1 участка;

X2 – абсцисса 2 участка;

Y2 – ордината 2 участка;

X3 – абсцисса 3 участка;

Y3 – ордината 3 участка;

X4 – абсцисса 4 участка;

Y4 – ордината 4 участка.

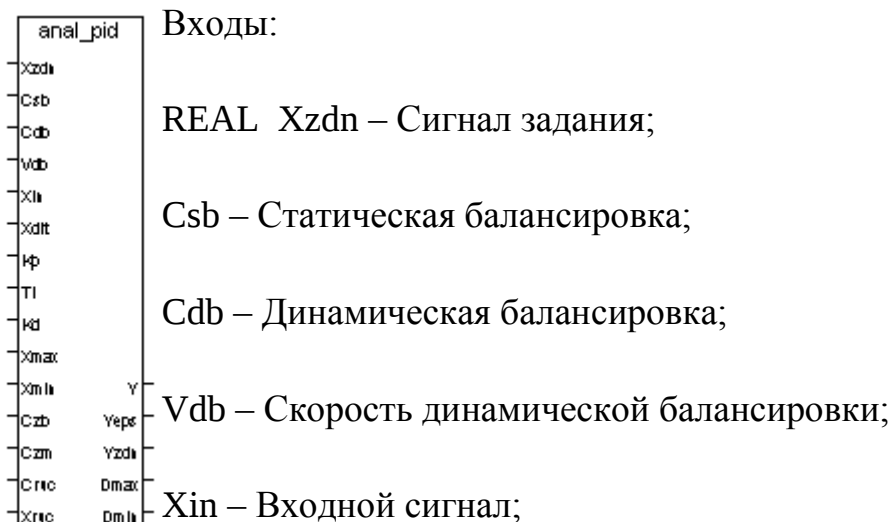
Выходы: Y – основной выход;

Nuch – номер текущего участка.

Алгоритм применяется для линеаризации нелинейной зависимости (например, датчика или регулирующего органа). Алгоритм применяется также для искусственного введения нелинейности (например, зона нечувствительности релейной характеристики и т.п.).

Число отрезков определяется входом Nkol. Координата конца отрезка определяется парой сигналов X_i , Y_i (абсцисса и ордината конца отрезка), которые задаются соответствующими значениями сигналов на настроечных входах алгоритма. При $X < X_1$ и $X > X_m$ выходной сигнал $Y = \text{const}$. На выходе Nuch формируется число, равное номеру текущего участка кусочно-линейной зависимости. Координаты X_i , Y_i могут находиться в любом из четырех квадрантов. Как правило координата $X_i > X_{(i-1)}$ (i-номер отрезка). Алгоритм работает правильно, если $Y_{(i+1)} - Y < 200\%$. При $m=0$ выходной сигнал $Y=0$ независимо от X .

RAN - Аналоговый ПИД-регулятор.



Czm – Сигнал запрета в направлении "меньше";

Cruc – Режим работы алгоритма (0-автоматический, 1-ручной);

Xruc – Ручное задание.

Выходы:

Y_Out – Основной выход регулятора;

Y_Eps – Выход рассогласования;

Y_Zdn – Выход текущего задания;

Dmin – Флаг ограничения по максимуму;

Dmax – Флаг ограничения по минимуму.

Функциональная схема алгоритма содержит несколько звеньев. Звено, выделяющее сигнал рассогласования, вычисляет разницу входных сигналов. Сигнал рассогласования e на выходе этого звена (без учета фильтра) равен:

$$e = X_{zdn} - X_{in},$$

Зона нечувствительности не пропускает на свой выход сигналы, значения которых находятся внутри установленного значения зоны. Сигнал e_2 на выходе этого звена равен:

$$e_2 = 0 \quad \text{при } |e| \leq X_{dlt};$$

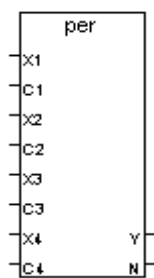
$$e_2 = e \quad \text{при } |e| > X_{dlt},$$

где X_{dlt} - зона нечувствительности; $X_{dlt} < 0$ воспринимается как $X_{dlt} = 0$. ПИД-звено выполняет пропорционально-интегрально-дифференциальное преобразование сигнала и имеет передаточную функцию:

$$W(p) = K_p [1 + 1/(T_i * p) + K_d * T_i * p / (1 + 0.125 * K_d * T_i * p)^2],$$

где K_p , T_i , K_d - соответственно коэффициент пропорциональности, постоянная времени интегрирования и коэффициент времени дифференцирования, равный $K_d = T_d/T_i$. Ограничитель ограничивает выходной сигнал алгоритма по максимуму и минимуму. Уровни ограничения устанавливаются коэффициентами $X_{\max}$, $X_{\min}$. Помимо двух сигнальных входов X_{zdn} и X_{in} , алгоритм имеет настроечные входы, которые задают параметры настройки алгоритма, и два управляющих входа для запрета изменения сигнала в большую или меньшую сторону. Диапазоны параметров настройки – стандартные для алгоритмов. Выход Y_{out} - основной выход алгоритма. На выходе Y_{eps} формируется отфильтрованный сигнал рассогласования. Два дискретных выхода фиксируют момент наступления ограничения выходного сигнала Y_{out} .

PER - переключатель с дискретным управлением.



Входы:

X1 - 1-й переключаемый вход

C1 - Команда выбора входа 1;

X2 - 2-й переключаемый вход;

C2 - Команда выбора входа 2;

X3 - 3-й переключаемый вход;

C3 - Команда выбора входа 3;

X4 - 4-й переключаемый вход;

C4 - Команда выбора входа 4.

Выходы:

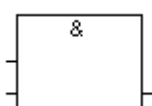
Y - основной выход;

N - порядковый номер выбранного входа.

Алгоритм представляет собой многополюсный переключатель аналоговых сигналов, положение которого определяется дискретными сигналами, поступающими на вход алгоритма. Алгоритм используется для выбора одного из 4-х сигналов. Если на аналоговых входах алгоритма заданы константы, то алгоритм может использоваться для дискретной установки требуемой константы.

Функциональная схема алгоритма представляет собой переключатель аналоговых сигналов. Если на всех дискретных входах C_i , управляющих положением переключателя, сигнал отсутствует (лог. 0), выходной сигнал $Y=0$. Если на какой-либо из дискретных входов подается дискретный сигнал $C_i=1$, выход алгоритма Y подключается к одноименному (по номеру индекса) аналоговому входу X_i . Если дискретные сигналы подаются одновременно на несколько входов, приоритетен вход с младшим номером. Сигнал на выходе Y равен сигналу на выбранном входе. Число на выходе N указывает номер выбранного входа.

Булевский AND

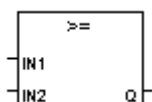


Inputs - основные входы алгоритма(может быть более двух);

Output - основные выходы алгоритма.

Описание: логическое И двух или более входов.

Больше или равно



Входы:

IN1 – основной вход алгоритма;

IN2 – основной вход алгоритма.

Примечание: вход и выход должны иметь один и тот же тип.

Выходы: Q – основной выход алгоритма.

Примечание: Q – TRUE если $IN1 \geq IN2$.

Описание: Проверить, что одна величина больше или равна другой.

Пример реализации фазификации по ошибке регулирования приведен на рисунке ниже:

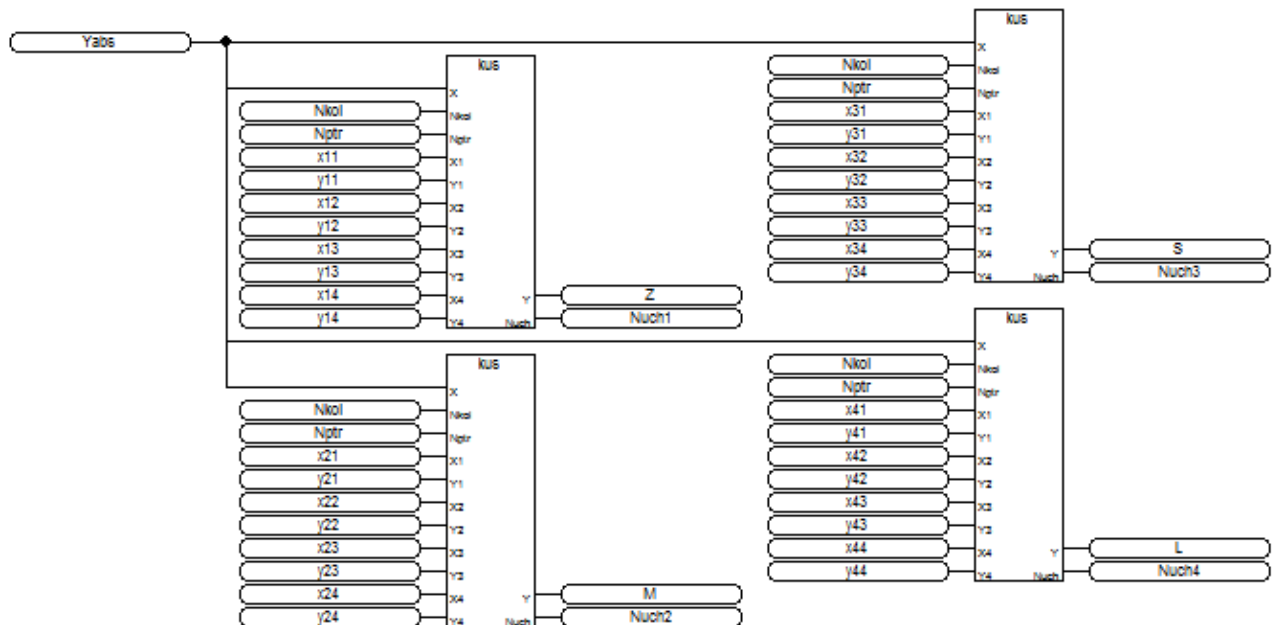


Рисунок В.3.6 Фазификация по ошибке регулирования

4 Задание

1. С помощью системы программирования ISOGRAF составить программу реализации нечеткого П-регулятора.
2. Набрать модель объекта управления 2 порядка и задать ее параметры (по заданию преподавателя) на АВК-6. Вход и выход модели объекта подключить к адаптеру, обеспечивающему связь с контроллером.
3. Установить связь с контроллером и запустить отладчик.
4. Задать уставку данной САУ от 0 В до 10 В (по заданию преподавателя).

5. Запустить прожектор и создать два графика: ошибки регулирования и коэффициента пропорциональности.
6. Изменяя возмущающее воздействие путем изменения коэффициента обратной связи одного из звеньев, обеспечить реализацию переходного процесса и исследовать работу системы с нечетким П-регулятором.
7. Сделать выводы о проделанной работе.

5 Содержание отчета

1. Цель работы.
2. Структурная схема САР с нечетким П-регулятором.
3. Графики изменения управляющего воздействия нечеткого П-регулятора.
4. Ответы на контрольные вопросы.
5. Краткие выводы о проделанной работе.

6 Контрольные вопросы

1. Дать определение фазификации, дефазификации.
2. Какие существуют методы дефазификации.
3. Что такое база правил. Как применяется.
4. Для чего предназначено приложение «ПРОЖЕКТОР» в системе ISaGRAF.

7 Методические указания по установке связи с контроллером и настройке прожектора

После того, как программа нечеткого П-регулятора готова, необходимо проверить ее на работоспособность. Для этого нужно сперва настроить Опции компилятора и связь. В окне программы открыть вкладку «Создать» → «Опции компилятора» (Рис. В.7.1).

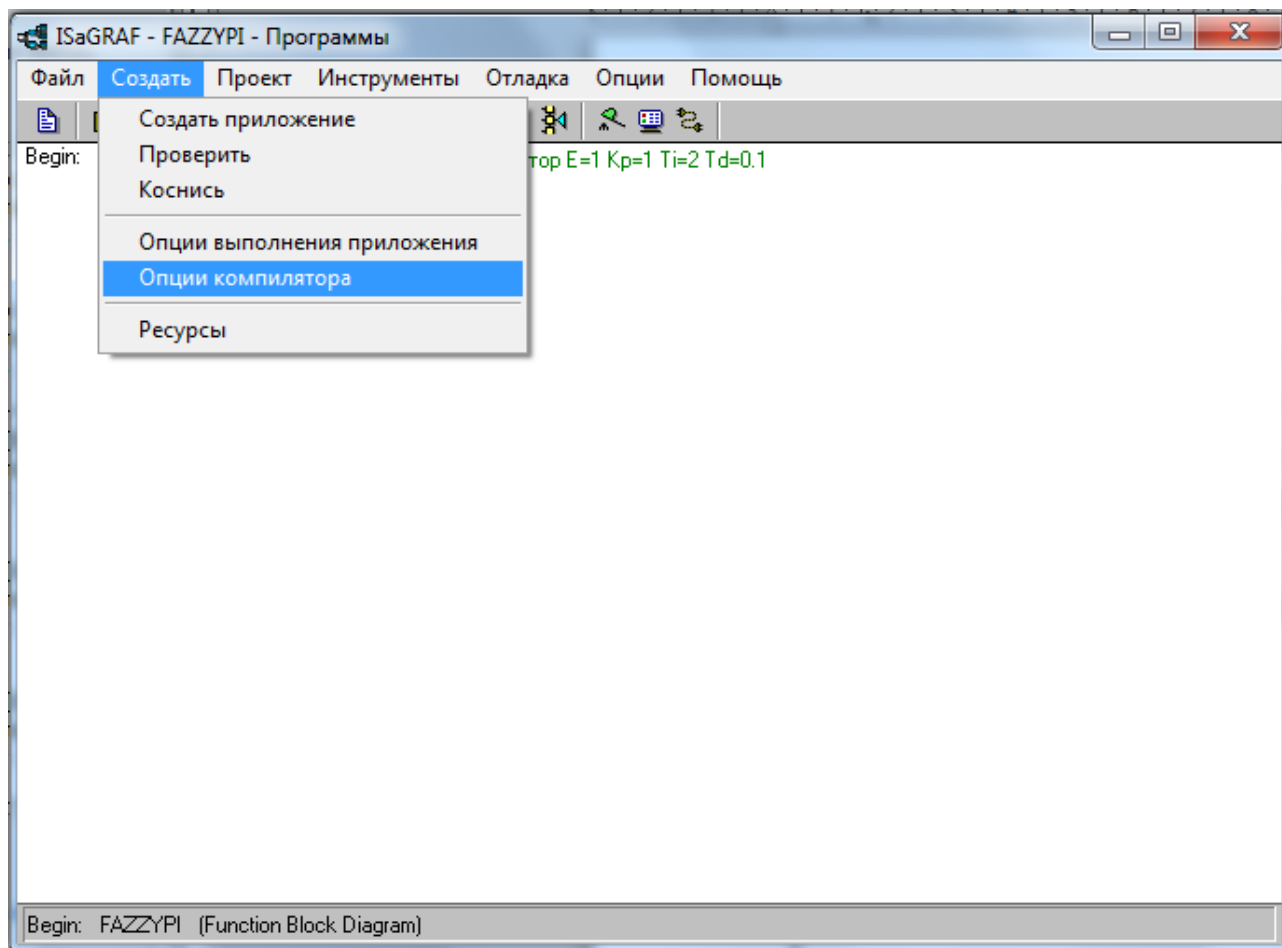


Рисунок В.7.1 Окно программы

В появившемся окне выбрать Опции, как показано на рисунке В.7.2 и нажать кнопку «Принять».

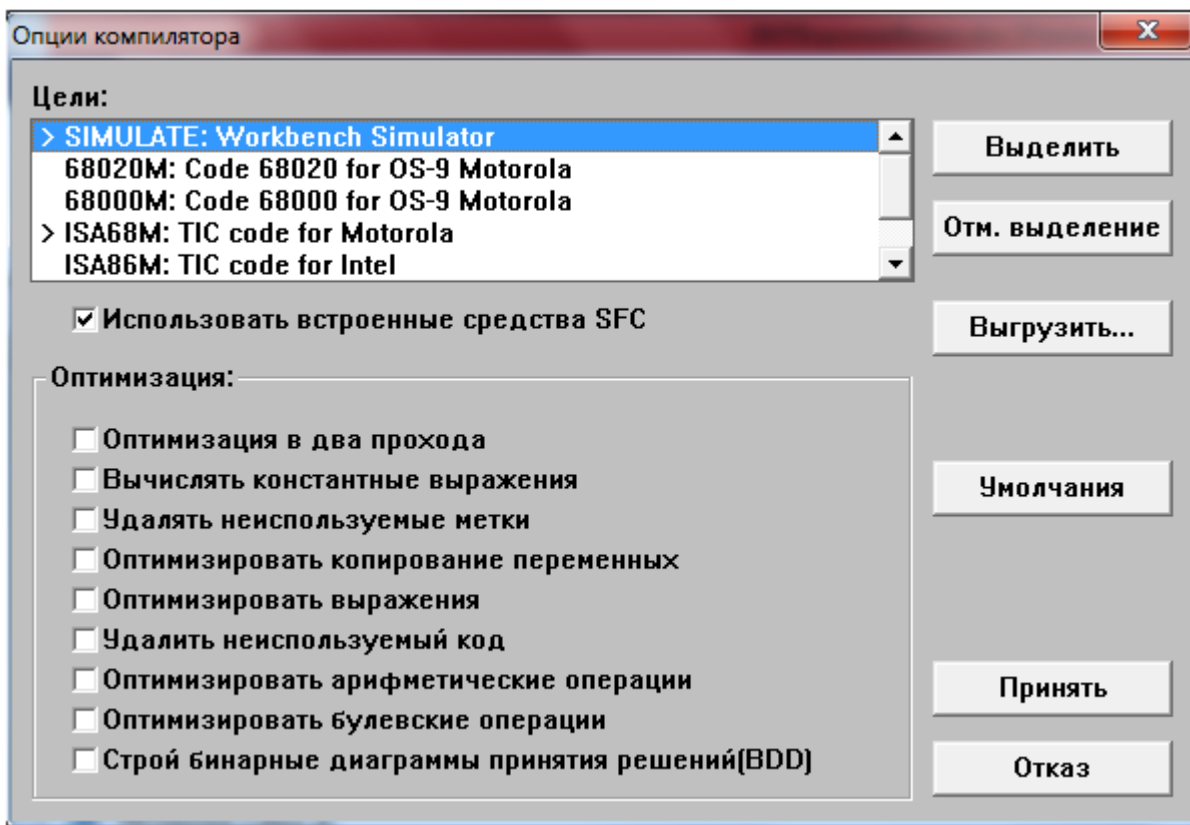


Рисунок В.7.2 Окно опций компилятора

Затем в окне программы выбираем «Отладка» → «Установка связей». В появившемся окне выбираем тип соединения, как показано на рисунке В.7.3, и нажимаем кнопку «Установка».

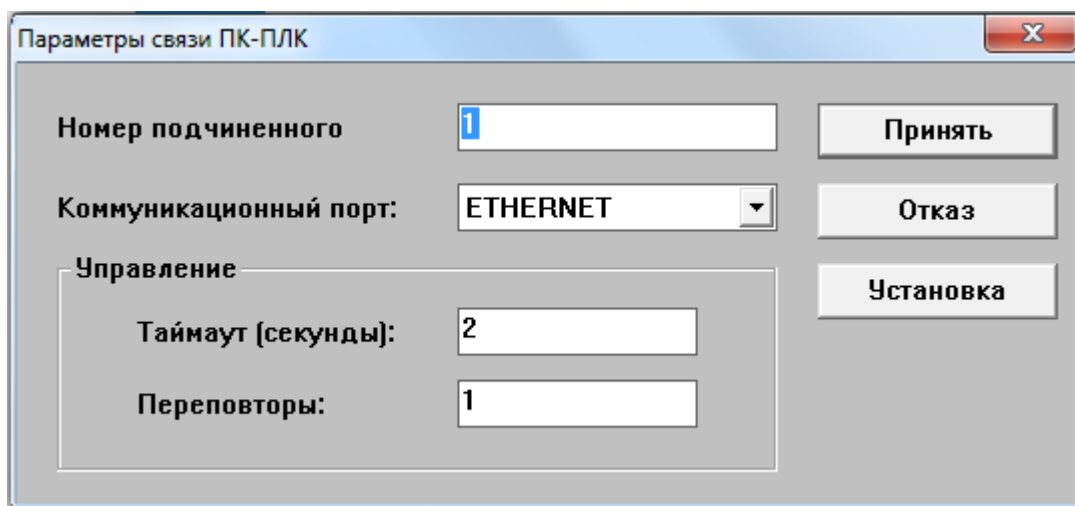
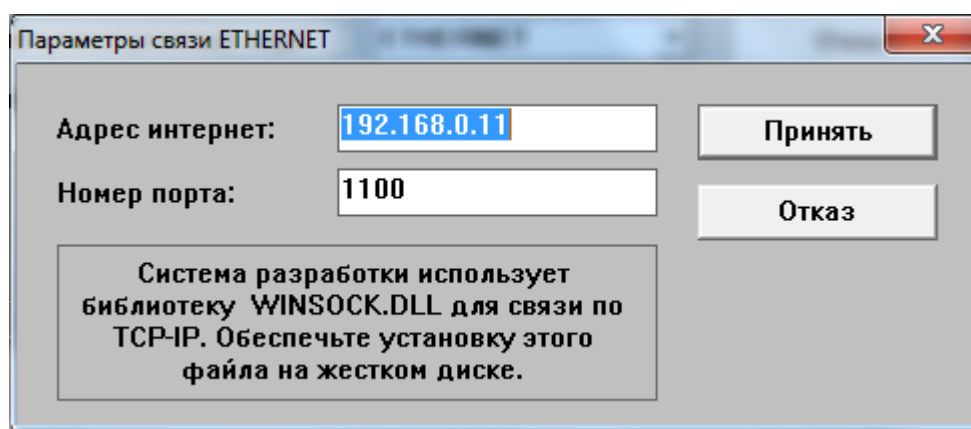


Рисунок В.7.3 Параметры связи

В настраиваем появившееся окно, как показано на рисунке В.7.4, а именно IP-адрес и номер порта для соединения. Затем нажимаем кнопку «Принять».



РисВ.7.4 Параметры связи Ethernet

После того, как настроили параметры связи контроллера с ПК, запускаем отладчик. Не забываем перед каждым запуском отладки заново откомпилировать программу. Далее в окне отладки (рисунок В.7.5) выбираем «Инструменты» → «Прожектор».

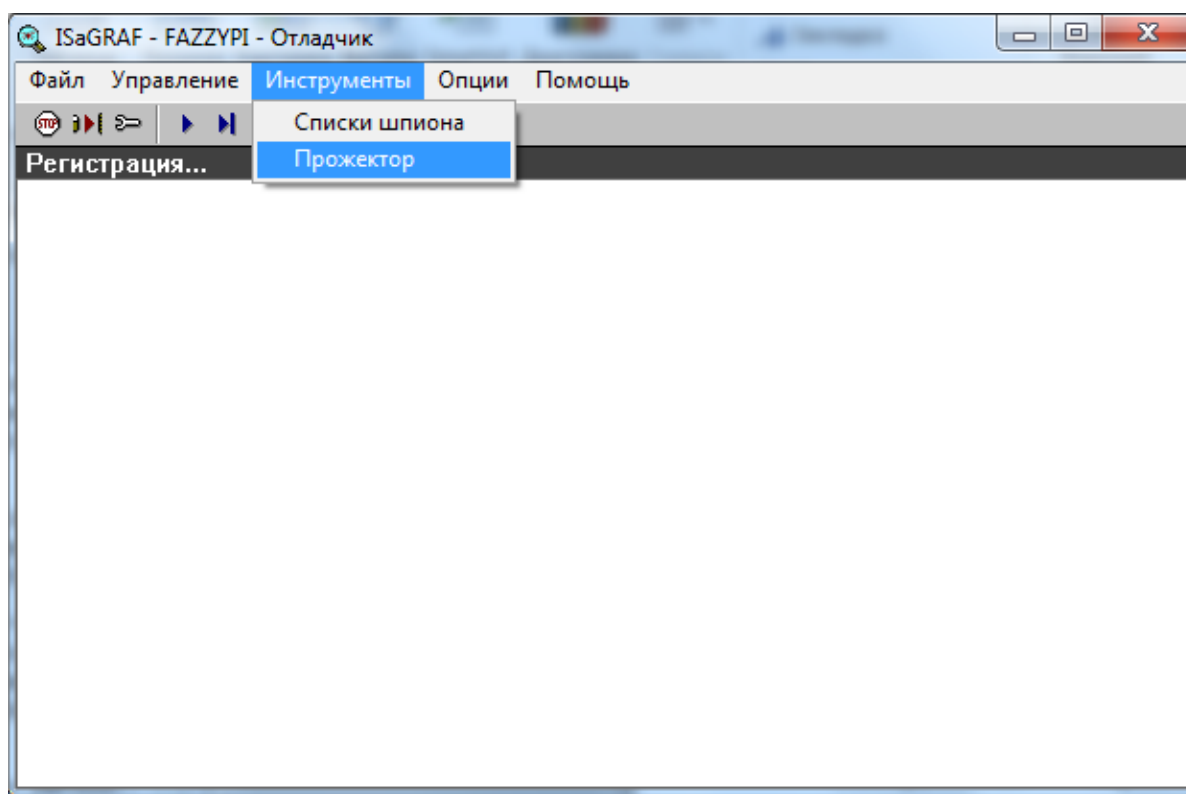


Рисунок В.7.5 Окно отладки

Появится окно прожектора (рисунок В.7.6).

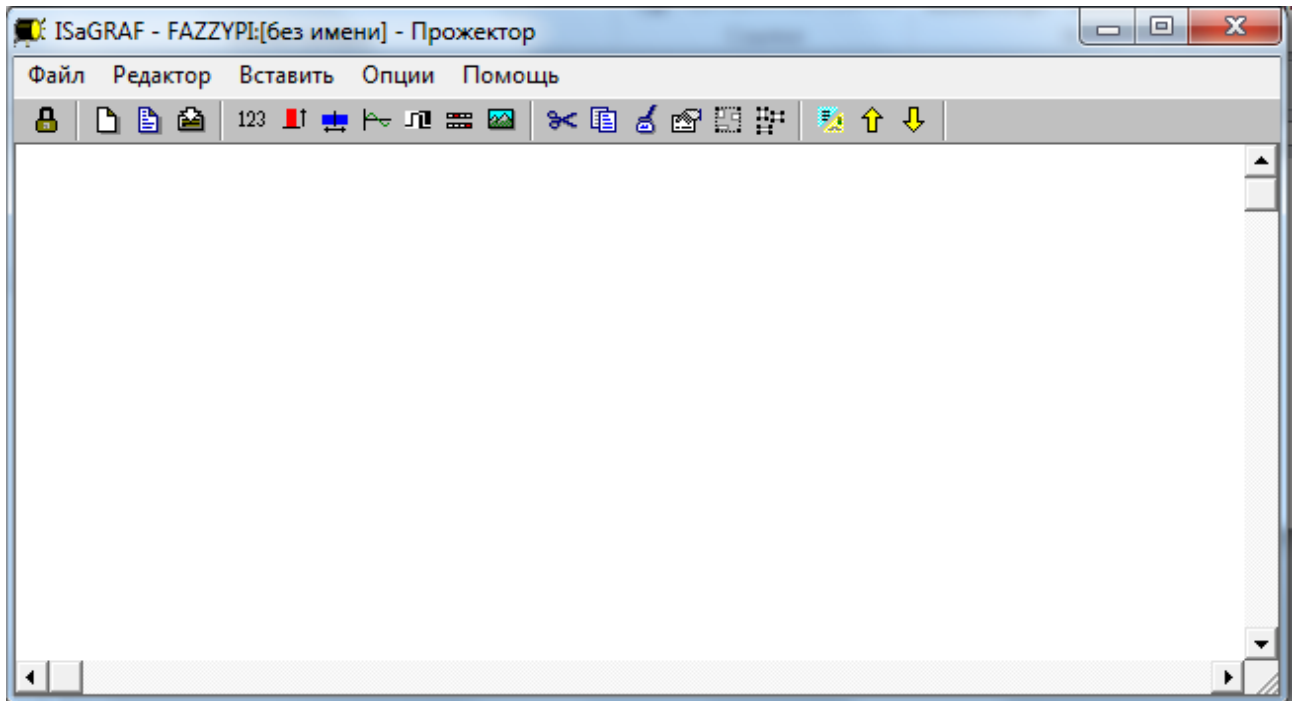


Рисунок В.7.6 Окно прожектора

Рассмотрим пример отображения графика изменения коэффициента пропорциональности нечеткого П-регулятора в прожекторе. Для этого выбираем «Вставить» → «Кривая». Появится окно настройки кривой (Рисунок В.7.7).

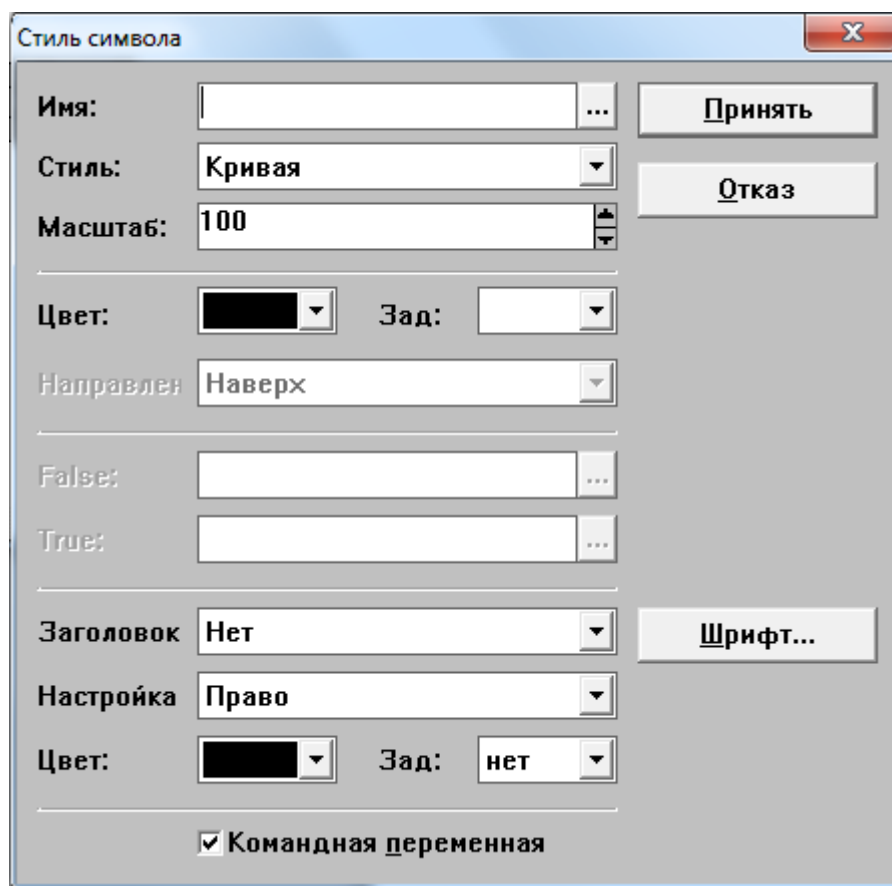
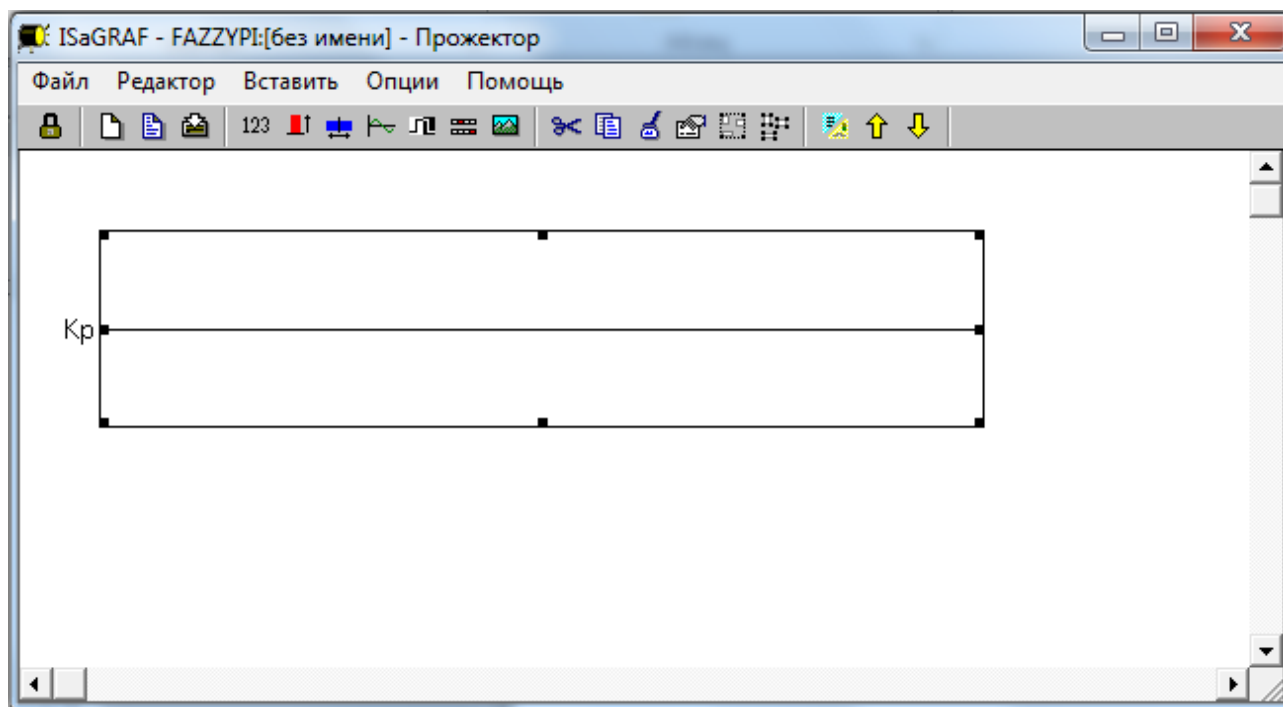


Рисунок В.7.7 Окно настройки кривой прожектора

Выбираем «Имя», нажав на кнопку с тремя точками напротив, обязательно корректируем масштаб в зависимости от диапазона изменения параметра. По желанию корректируем остальные настройки и нажимаем «Принять». Результат динамизации представлен на экранной форме, изображенной на рисунке В.7.8.



РисВ.7.8 Окно прожектора с кривой

Теперь все изменения коэффициента можно будет посмотреть в режиме реального времени с помощью кривой. Аналогично настраиваются другие компоненты визуализации данного инструмента.