

Евразийский национальный университет имени Л.Н. Гумилёва

УДК 62-529

На правах рукописи

**БИГАЛИЕВА АЛЬФИЯ ЗАМИРОВНА**

**Разработка программного комплекса интеллектуального управления  
технологическим процессом тонкого помола**

6D070400 – Вычислительная техника и программное обеспечение

Диссертация на соискание степени  
доктора философии (PhD)

Научный консультант  
доктор технических наук,  
профессор  
С.К. Атанов

Зарубежный консультант  
доктор технических наук,  
профессор  
Л.С. Лисицына  
(НИУ ИТМО, Санкт-Петербург)

Республика Казахстан  
Нур-Султан, 2021

## СОДЕРЖАНИЕ

<b>ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ</b> .....	3
<b>ВВЕДЕНИЕ</b> .....	4
<b>1 ПРОБЛЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ ПОМОЛА</b> .....	10
1.1 Проблемы переработки техногенных минеральных образований.....	10
1.2 Устройство и принцип работы турбулентного измельчителя.....	14
1.3 Особенности реализации автоматического управления технологическими процессами.....	18
1.4 Моделирование динамических процессов для турбулентного измельчителя.....	21
Выводы по первому разделу.....	25
<b>2 ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТУРБУЛЕНТНЫМ ИЗМЕЛЬЧИТЕЛЕМ</b> .....	27
2.1 Принципы построения интеллектуальных систем .....	27
2.2 Создание стенда и исследование режимов работы турбулентного измельчителя породы.....	33
2.3 Моделирование в MATLAB системы управления на базе нечёткой логики.....	38
Выводы по второму разделу.....	47
<b>3 ПРОГРАММНО-АППАРАТНАЯ РЕАЛИЗАЦИЯ УПРАВЛЕНИЯ НА ОСНОВЕ НЕЧЁТКОЙ ЛОГИКИ</b> .....	49
3.1 Структурная схема микропроцессорного управления технологическим процессом тонкого помола .....	49
3.2 Программный комплекс управления технологическим процессом тонкого помола.....	50
3.3 Аппаратный комплекс управления технологическим процессом тонкого помола.....	54
Выводы по третьему разделу.....	63
<b>ЗАКЛЮЧЕНИЕ</b> .....	64
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b> .....	65
<b>ПРИЛОЖЕНИЕ А – Свидетельства о государственной регистрации прав на объект авторского права</b> .....	70
<b>ПРИЛОЖЕНИЕ Б – Акт внедрения</b> .....	72
<b>ПРИЛОЖЕНИЕ В – Коды программы</b> .....	74

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

FLC	– регулятор нечеткой логики
tf-форма	– форма передаточной функции
USB	– универсальная последовательная шина
ЛП	– лингвистическая переменная
НЧ	– нечёткое число
ТМО	– техногенные минеральные образования
ШИМ	– широтно-импульсная модуляция

## ВВЕДЕНИЕ

Диссертационная работа посвящена повышению эффективности технологического процесса извлечения ценных компонентов с использованием интеллектуальной системы управления в турбулентных измельчителях.

### **Актуальность**

Накопление техногенных отходов в Казахстане началось осуществляться с середины XIX века и идет в настоящее время их накопление.

Существующие и широко применяемые в цветной металлургии способы переработки шлаков, хвостов обогащения введут значительного расхода энергоресурсов и низкой извлекаемости металла, а также по требованиям к охране окружающей среды не отвечают современным требованиям комплексной энергосберегающей технологией переработки минерального сырья.

Перспективным методом извлечения тонких частиц меди из шлаков медеплавильного производства является разработка экологически чистой технологии с использованием положения механохимии, которая имеет интеллектуальное управление.

Сырьевой базой для осуществления предлагаемой технологии служат исторические и современные продукты в виде шламов, шлаков, вскрышных пород, забалансовые руды в отвалах и складах. Для условий медной корпорации «Казахмыс» в зависимости от типа медной руды в суммарной стоимости доля сопутствующих компонентов составляет от 24 до 50%, или в денежном выражении – 80\$...120\$ за тонну без учета стоимости меди. В случае организации собственного производства по переработке техногенных отходов в стоимость сырья будет нулевой. Стоимость извлеченного металла ориентировочно с учетом производственных затрат будет в пределах 200\$-250\$ с тонны.

Способ прямого извлечения металла из рудного и техногенного сырья заключается в сверхтонком помоле (порядок 80...125 микрон минеральной составляющей). С последующей воздушной и магнитной классификацией помольной среды с разделением металлической и минеральной составляющих.

Во всех технологиях основной операцией, определяющей весь последующий ход технологии, является тонкий или даже сверхтонкий помол исходного сырья.

Таким образом, актуальной представляется задача разработки алгоритмов управления процессом тонкого помола, обеспечивающих повышение качества помола.

Развитие компьютерных технологий определяет возможность практического использования регуляторов для управления специализированными механическими устройствами.

Проблема использования классических регуляторов заключается в том, что не всегда есть возможность представления входных и выходных данных, опираясь в логике управления на знания и опыт эксперта.

Эффективным решением является применение нечётких регуляторов в качестве управления исполнительными механизмами. Нечёткие регуляторы обладают возможностью работать с быстро изменяющимися, сложными динамическими процессами, учитывать неполную и неоднозначную информации о процессе.

Управление должно быть «интеллектуальным» и обеспечивать высокую точность при простой и недорогой аппаратной реализации.

Диссертационная работа представляет собой результат в формировании теории и практических методов синтеза систем интеллектуального управления помолом в турбулентных измельчителях. Реализация управления технологическим процессом была выполнена с использованием микроконтроллера Arduino Mega 2560 на базе Fuzzy Logic.

**Цель диссертационного исследования** – повышение эффективности технологического процесса извлечения ценных компонентов с использованием интеллектуальной системы управления.

**Задачи исследования:**

- исследовать влияние архитектурных особенностей измельчителей на технологический процесс извлечения ценных компонентов;
- разработать систему управления, учитывающую особенности архитектуры турбулентного измельчителя и сложные нелинейные термомеханические процессы в исследуемой среде;
- разработать программную модель интеллектуальной системы управления и провести численное моделирование;
- разработать и реализовать микроконтроллерный программно-аппаратный комплекс с использованием нечёткой логики для опытно-экспериментальных исследований.

**Методы исследования**

Поставленные в диссертационной работе задачи решены на основе технологий искусственного интеллекта. Использовались методы классической теории множеств и формальной логики. Моделирование, исследование и верификация разрабатываемых алгоритмов, постановка исследования и экспериментов проводились с использованием математических пакетов MatLAB, Fuzzy Logic Toolbox, языков программирования Си/Си++.

Проверка теоретических положений и адекватности моделей, разработанных в диссертационной работе, осуществлялась с помощью компьютерного моделирования на ПЭВМ и экспериментальных исследований на экспериментальных стендах.

**Основные положения, выносимые на защиту:**

- результаты исследования влияния архитектурных особенностей измельчителей на технологический процесс извлечения ценных компонентов;
- программно-аппаратный комплекс разработанной интеллектуальной системы управления.

### **Научные результаты:**

– исследовано влияние архитектурных особенностей измельчителей на технологический процесс извлечения ценных компонентов;

разработана интеллектуальная система управления измельчителя, учитывающего особенности архитектуры, на базе нечёткой логики в среде MATLAB, и исследованы его технико-эксплуатационные характеристики;

– разработан и реализован микроконтроллерный программно-аппаратный комплекс с использованием нечёткой логики для опытно-экспериментальных исследований.

### **Научная новизна:**

– разработана интеллектуальная система управления турбулентным измельчителем породы;

– разработан интеллектуальный программный комплекс в среде MATLAB;

– реализован микроконтроллерный программно-аппаратный комплекс с использованием нечёткой логики.

**Практическое значение полученных результатов** данной работы заключается в том, что благодаря разработке, апробации и внедрению программного комплекса интеллектуального управления технологическим процессом тонкого помола стало возможным обеспечить повышение качества помола.

**Личный вклад соискателя** заключается в непосредственном участии в постановке задач исследования, получении исходных данных, в проведении основного объема теоретических и экспериментальных исследований, изложенных в диссертационной работе, анализе и оформлении результатов работы в виде научных публикаций и докладов на научных конференциях.

### **Апробация результатов диссертации:**

1. Семинары докторантов кафедры «Компьютерная и программная инженерия».

2. Семинар Мегафакультета компьютерных технологий и управления Университета ИТМО (Санкт-Петербург, 2019).

3. VIII Конгресс молодых учёных (Санкт-Петербург, 2019). Диплом в номинации «Лучший доклад».

4. Международная научно-техническая конференция молодых ученых БГТУ им. В.Ш. Шухова (Белгород, 2019).

5. XXV Международная научно-техническая конференция студентов, аспирантов и молодых ученых «Научная сессия ТУСУР-2020» (Томск, 2020).

### **Публикации результатов исследования:**

– публикации в изданиях, входящих в наукометрические базы данных Scopus

1. Вопросы управления процессом тонкого помола в планетарной мельнице // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. – 2020. Т. 16, Вып. 3. – С. 277-292 (База данных Scopus, процентиль по общей информатике 29%).

**– статьи, опубликованные в научных журналах, рекомендуемых ККСОН МОН РК:**

1. Расчёт степени помола сырья с применением компьютерных технологий ANETR5 на примере планетарной мельницы // Университетінің еңбектері – Труды университета. – 2018. – №4. – С. 139-144.

2. Разработка и моделирование оптимального контроллера LQG для управления процессом помола // Вестник КазНУ. – 2019. – №6. – С. 526-533.

3. Математическая модель процесса помола в планетарной мельнице // Университетінің еңбектері – Труды университета. – 2020. – №1. – С. 148-153.

4. Построение фильтра Калмана для восстановления недостающей информации о состоянии процесса помола в планетарной мельнице // Университетінің еңбектері – Труды университета. – 2020. – №2. – С. 133-138.

5. Синтез LQG регулятора для интеллектуального управления технологическим процессом тонкого помола // Вестник Национальной инженерной академии Республики Казахстан. – 2020. – №4. – С. 21-27.

**– публикации, опубликованные в трудах международных конференций:**

1. Расчёт степени помола сырья с применением прикладного программного обеспечения ANETR5 на примере планетарной мельницы. // Технический журнал «Автоматизация. Современные технологии». – 2019. – №2. – С. 155-160.

2. Интеллектуальные системы управления для неопределённых ситуаций управления // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. - [2019, электронный ресурс]. - Режим доступа: <https://kmu.itmo.ru/digests/article/674>, своб.

3. Разработка экспертной системы по управлению технологическим процессом на основе нечёткой логики // Сборник докладов «Международная научно-техническая конференция молодых ученых БГТУ им. В.Ш. Шухова», г. Белгород. – 2019. – С.2868-2873.

4. Интеллектуальное управление технологическим процессом тонкого помола // Сборник избранных статей научной сессии ТУСУРа по материалам XXIV Международной научно-технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР-2019». – 2019. – С. 104-106.

5. Интеллектуальное управление технологическим процессом тонкого помола LQR регулятором // Материалы докладов Международной научно-практической конференции «Электронные средства и системы управления». – 2019. – С. 61-63.

6. Вопросы управления процессом помола с применением фильтра Калмана // Сборник избранных статей научной сессии ТУСУРа по материалам XXV Международная научно-техническая конференция студентов, аспирантов и молодых ученых «Научная сессия ТУСУР-2020». – 2020. – С. 18-21.

7. Определение значения внутримельничного заполнения шарами и акустический анализ // Материалы Международной научной конференции

«Теоретические и прикладные вопросы математики, механики и информатики». – 2019. – С. 126-127.

– **свидетельство о государственной регистрации на объект авторского права**

1. Программа для ЭВМ: Интеллектуальное управление технологическим процессом тонкого помола. Свидетельства, о внесении сведений в государственный реестр прав на объекты, охраняемые авторским правом №5042 от 27.08.2019 (Приложение А).

2. Программа для ЭВМ: Интеллектуальное управление технологическим процессом тонкого помола. Свидетельства, о внесении сведений в государственный реестр прав на объекты, охраняемые авторским правом №15286 от 18.02.2021 (Приложение А).

– **акт внедрения**

2 акта внедрения программного обеспечения «Интеллектуальное управление технологическим процессом тонкого помола» (Приложение Б).

По теме диссертации опубликовано 13 работ, отражающих основные результаты исследования, в том числе 1 публикации в изданиях, входящих в наукометрические базы данных Scopus, 5 публикаций в журналах из перечня ККСоН, 7 публикаций в рецензируемых российских научных журналах и трудах международных конференций. Имеется акт о внедрении результатов диссертации и два свидетельства о государственной регистрации прав на объект авторского права.

**Объем и структура диссертации.** Диссертация состоит из введения, трех разделов, заключения и списка литературы. Объем работы составляет 88 страниц, включает 41 рисунок, 8 таблиц, 3 приложения. Список использованных источников содержит 81 наименование.

**Во введении** представлен научный аппарат исследования, обоснование актуальности исследования, степень ее изученности в теории и практике, определены цель, задачи, объект и предмет исследования, раскрывается научная новизна, теоретическая и практическая значимость работы, определены методы исследования, представлены положения, выносимые на защиту, личный вклад автора, список публикаций и апробаций результатов работы.

**Раздел 1** включает в себя анализ проблемы переработки техногенных минеральных образований. Приводится анализ существующих способов управления технологическим процессом помола. Рассматривается устройство и принцип работы турбулентного измельчителя.

**Раздел 2** включает в себя основные принципы построения интеллектуальных систем в нечеткой среде. Теорию нечетких множеств. Разработку структуры регулятора на базе нечеткой логики. Рассматривается построение нечеткой логики в системе MATLAB.

**Раздел 3** Рассматривается аппаратно-программный комплекс управления технологическим процессом тонкого помола на базе Arduino.

**В заключении** обобщены результаты исследования, сформулированы

основные выводы, подтверждающие и доказывающие истинность положений, выносимых на защиту.

**В приложении** представлены практические материалы исследования.

Автор выражает глубокую благодарность своему научному руководителю, д.т.н., профессору кафедры «Компьютерная и программная инженерия», Атанову Сабыржану Кубейсиновичу и зарубежному консультанту, д.т.н., профессору Университета ИТМО, Лисициной Любовью Сергеевне за неоценимую работу и консультации в ходе исследований.

# 1 ПРОБЛЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ ПОМОЛА

## 1.1 Проблемы переработки техногенных минеральных образований

Переработка ТМО могла бы стать перспективным направлением развития бизнеса в Казахстане и внести важный вклад в улучшение экологической обстановки.

«Техногенные минеральные образования (ТМО) – это вторичное сырье, которое имеет хорошую перспективу для дальнейшего использования», – такое определение дает заместитель исполнительного директора Республиканской ассоциации горнодобывающих и горно-металлургических предприятий (АГМП) Максим Кононов [1].

Если говорить подробнее, то в соответствии со ст.13 Кодекса РК «О недрах и недропользовании», техногенными минеральными образованиями признаются скопления отходов горнодобывающих, горно-перерабатывающих и энергетических производств, содержащие полезные компоненты или полезные ископаемые.

На современном этапе устойчивого развития нашего государства в условиях роста масштаба производства и потребления минерального сырья проблема эффективного и рационального использования промышленных и техногенных образований имеет исключительно важное народнохозяйственное значение. Актуальность этой проблемы обусловлена ограниченностью и невозможностью месторождений полезных ископаемых, усложнением горно-геологических условий залегания рудных тел и удорожанием их добычи, ухудшением количественного и качественного состава добываемых из недр руд, колебанием цен сырья на мировом рынке, негативным влиянием накапливаемых техногенных ресурсов на окружающую природную среду и т.п. Все это определяет глобальное и общегосударственное значение вопросов, связанных с утилизацией и ликвидацией огромных промышленных техногенных объектов.

Накопление техногенных отходов в РК началось осуществляться с середины 19 века и идет их накопление в настоящее время. Особенно интересны для переработки металлургические шлаки периода 1930-1955 годов Балхашского и Жезказганского медеплавильных заводов.

Всего на сегодняшний день по актуальным данным комитета геологии и недропользования министерства по инвестициям и развитию Казахстана, в стране имеется 1 406 объектов ТМО общей массой 47,4 млрд. тонн. Среди них 258 объектов находятся в государственной собственности, 744 – в собственности недропользователей и оставшиеся 404 объекта – в форме разделительного баланса (то есть скапливание на них ТМО началось до 30 мая 1992 года и продолжалось после этой даты) [1].

В таблице 1.1 приведены данные о сопутствующих компонентах в суммарной ценности минерального сырья.

Таблица 1.1 – Характеристика сопутствующих компонентов в суммарной ценности минерального сырья

Тип сырья	Ценные компоненты		Для сопутствующих в суммарной ценности, %
	<i>основные</i>	<i>сопутствующие</i>	
Бокситы	Глинозем	Ga, V	15
Алуниты	Глинозем	Na, K, S, Ga, V	44
Полиметаллические руды	Свиней, цинк, медь	S, Ca, Au, Ag, In, Te, Ga, Ge, Co	38
Медно-молибденовые руды	Медь	Mo, S, Au, Ag, Re, Se, Te, Ca	45
Медно-цинковые руды	Медь, цинк	S, Au, Ag, Ca, Se, Te, In, Ge, Ta	50
Медно-сульфидные руды	Медь, никель	Ca, S, Pt, V, Ru, Os, Ir, Au, Ag	24

Во всем мире нарастают темпы расширения использования техногенных минеральных образований. В переработку вовлекаются отвалы пород и отвалы вскрышных работ, хвосты обогатительных фабрик, шлаки металлургического производства, удержанные фильтрами твердые составляющие дыма.

Зарубежный опыт управления отходами широко изучен и изложен на примере организации системы управления отходами в Германии, Польше, США, Японии и Бразилии [2].

В решении задачи переработки техногенных отходов с целью доизвлечения редких и благородных групп металлов требуется измельчать его до определённой крупности [3-5].

Механоактивация – это сложный процесс [6-9], ход которого можно регулировать, варьируя параметрами, оказывающими влияние на различные характеристики. К управляющим параметрам относятся тип и геометрия используемой мельницы; скорость вращения водила для планетарных активаторов или частота колебаний; амплитуда для вибромельниц; материал и размеры мелющих тел; материал барабана; суммарная масса мелющих тел; отношение объемов обрабатываемого порошка и мелющих тел; атмосфера, в которой проводится обработка и давление в барабане; поверхностно-активное вещество. Тип используемой мельницы определяет, как характер движения шаров, так и количество получаемого на выходе продукта [10-12].

В части экологических характеристик, простоте конструкции, удобства обслуживания и возможности работы без строительства капитальных зданий и сооружений является шаровый турбулентный измельчитель с фрикционной передачей вращения отводила к мелющим барабанам.

В турбулентных измельчителях обычно имеются 3 или 4 барабана, вращающихся вокруг центральной оси и одновременно вокруг собственных осей в противоположном направлении (подобно движению планет вокруг Солнца). В барабаны загружают измельчаемый материал и мелющие тела (обычно шарики).

Частицы измельчаемого материала претерпевают множество соударений с

мельющими телами и стенками барабана. Эффективность турбулентных измельчителей обусловлена высокой кинетической энергией мельющих тел благодаря большой скорости их движения создающих высокие напряжения в активизируемом веществе.

При сравнении турбулентных измельчителей с традиционным измельчительным оборудованием обычно приводят следующие факты. При вращении обычной вращающейся шаровой мельницы мельющие тела измельчают материал, падая под действием силы тяжести. Гравитационное ускорение в данном случае составляет 1g. В турбулентных измельчителях используется центробежное движение барабанов с высокими ускорениями. Ускорение в лабораторных турбулентных измельчителях составляет 28...60g и выше. Ускорение в турбулентных измельчителях промышленного типа составляет примерно 20g. Ясно, что силы, действующие на измельчаемый материал в турбулентных измельчителях, не менее чем в десятки раз превышают силу воздействия на материал в традиционном измельчительном оборудовании. Процесс измельчения в турбулентных измельчителях происходит значительно быстрее и эффективнее. Например, в производстве тонкого порошка карбида вольфрама (WC) рядовым событием является измельчение порошка в шаровой мельнице в течение 130 часов. В турбулентном измельчителе характерное время измельчения WC до субмикронного размера составляет 15-30 мин. Энергонапряженность (в англоязычной литературе – energy density, плотность энергии) мельницы можно оценить, разделив их мощность на объем рабочей камеры. Энергонапряженность турбулентного измельчителя может в 1000 раз превышать соответствующие параметры шаровых мельниц. Турбулентные измельчители, использующие высокие ускорения, характеризуются наиболее высокой энергонапряженностью по сравнению с атриторами, вибро-мельницами, струйными мельницами и дезинтеграторами. На рынке лабораторного оборудования известны турбулентные измельчители зарубежных производителей [13, с. 278; 14, с. 51; 15, с. 29, 16].

Измельчение различных материалов имеет свои особенности. Исследованиями установлено [17-19], что измельчение продукта и широкой мельнице неизбежно сопровождается противоположным по своему характеру процессом агрегирования мелких частиц. Мелкие комочки, слипшиеся одна с другой, называют флокулами. Чем мельче становится продукт в работающей мельнице, тем больше получается флокул. Происходит также образование наслоений – подушек из слипшихся частиц па мельющих шарах, а также на стенках мельницы. При этом возникают непроизводительные, холостые удары шаров. Вместе с тем усиливается работа трения; часть механической энергии, затрачиваемой на помол, переходит в тепловую, что вызывает повышение температуры размалываемого материала. В целом производительность мельницы уменьшается. Существует образное выражение, что агрегирование – это «злейший враг» процесса помола. Предложены различные гипотезы для объяснения причин агрегирования диспергируемых твердых частиц и их налипания на другие тела. Природу этого явления, происходящего при

измельчении твердых тел и возникновении новых поверхностей, подробно изучали В. В. Дерягин и Н. А. Кротова, П. В. Крагельский, Боуден и др.

В настоящее время информационные технологии – необходимый атрибут управления и повышения эффективности деятельности любой отрасли. Проникая во все сферы производственной деятельности информационные технологии обеспечивают возможность построения эффективной системы управления с дальнейшим увеличением количества выполняемых работ, сокращением сроков и повышением качества работ. Каждая сфера деятельности человека вращается вокруг автоматизированных процессов, которые практически не требуют вмешательства человека, что в свою очередь, устраняет вероятности сбоев вследствие человеческой ошибки [20].

Есть две основные причины автоматизации процесса помола:

1) поддержание требуемых режимов измельчения и классификация в условиях изменяющегося качества измельчаемого сырья и других условий измельчения;

2) снижение издержек производства и максимизация прибыли.

Для интеллектуального управления любым процессом необходимо детально изучить все аспекты области, к которой принадлежит объект управления. Изучению подлежат все факторы, возникающие в процессе, затем необходимо определить те, которые имеют решающее влияние, и те, которыми можно пренебречь [21,22].

Изучая научные исследования в области измельчения, был сделан вывод о том, что для достижения поставленной цели научной работы необходимо контролировать размер частиц на выходе, так как контроль других параметров, по мнению автора, дают меньшую гарантию получения качественного продукта на выходе.

Интеллектуальное управление получило широкое распространение в мире, но в сфере измельчительного оборудования вопросов остаётся очень много. Использование промышленных мельниц непрерывного действия с производительностью от 5т/ч с интеллектуальным управлением встречается в работах [23,24]. Производственные предприятия, которые постоянно следят за производительностью и качеством, часто сталкиваются с проблемами, связанными с высокой производственной сложностью и неопределенностью. Это те области, в которых традиционные производственные парадигмы неэффективны из-за ограничения способности операторов-людей справляться с этими сложностями, неопределенности, понимания/запоминания больших данных, а также их неспособности принимать решения, требующие времени. Интеллектуальные производственные системы, с другой стороны, могут давать превосходные результаты по сравнению с традиционными производственными системами, поскольку они способны анализировать, самообучаться, воспринимать сложности, а также способны хранить и анализировать большие объемы данных для получения повышенного качества продукта.

## 1.2 Устройство и принцип работы турбулентного измельчителя

В качестве базового объекта управления, рассматриваемого в данной работе, принят турбулентный измельчитель, который расположен в г. Караганде, Казахстан, и используется ТОО «КМЗ им. Пархоменко». С его помощью решаются различные практические задачи в области переработки техногенных отходов медеплавильного производства для выделения металлической меди.

В Испытательной лаборатории инженерного профиля «Комплексное освоение ресурсов минерального сырья» (Республиканское Государственное Предприятие на Праве Хозяйственного Ведения) Карагандинского государственного технического университета установлено [3, с. 49], что наибольшая извлекаемость металлов из шлака медеплавильного производства достигается при помоле с тониной 80...125 мкм.

Турбулентный измельчитель МПП-04 представляет собой сложную конструкцию, состоящую из корпуса, внутри которого находится блок помольных барабанов в количестве 4 штук (рисунок 1.1). Помольные барабаны крепятся на крестовине, которая насажена на центральный вал. Сверху корпус закрыт крышкой, что обеспечивает безопасную работу установки. В качестве привода в МПП-04 используется асинхронный электродвигатель мощностью 22 кВт, работающий в режиме постоянной или переменной скорости [3, с. 55].

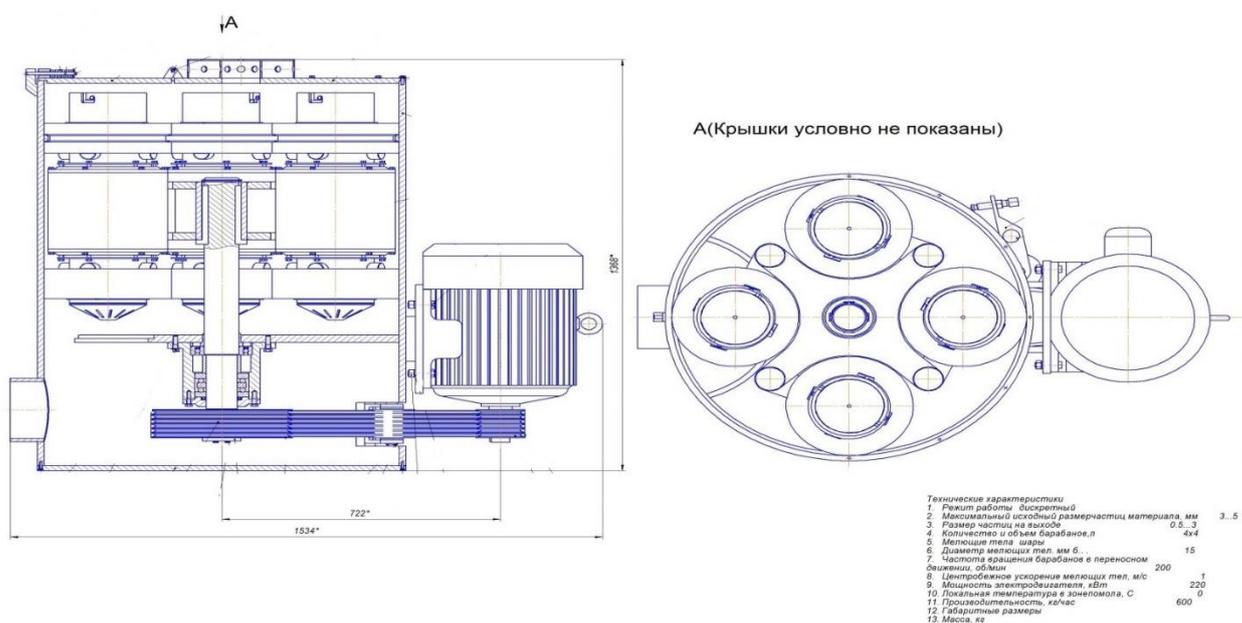


Рисунок 1.1 – Общий вид турбулентного измельчителя МПП-04

Работа в МПП-04 осуществляется следующим образом: к центральному валу посредством рычажной системы присоединены помольные барабаны. Через регулятор загрузки (шиберная заслонка) подается исходное сырье, которое по транспортным линиям через верхние отверстия поступает в барабаны. При вращении центрального вала помольные барабаны под действием центробежной силы прижимаются к корпусу. За счет сил трения между корпусом и боковой поверхностью помольного барабана осуществляется вращение последних вокруг

собственной оси. Таким образом, помольный барабан получает два вида движения: переносное вокруг центрального вала и относительное вокруг собственной оси (рисунок 1.2).

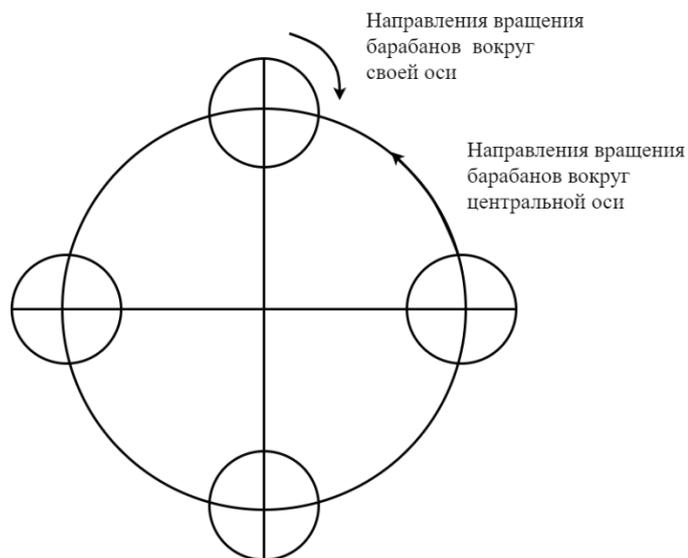


Рисунок 1.2 – Принцип действия турбулентного измельчителя

Центробежное ускорение мелющих тел, находящихся в барабане, определяется угловой скоростью вращения центрального вала и угловой скоростью собственного вращения барабана с учетом кориолисова ускорения. Измельченный материал просыпается через нижние отверстия в бункер-накопитель, где путем воздушной классификации, а затем магнитной сепарации происходит отделение минеральной части от металлических коррекций (рисунок 1.3).

**Схема технологического процесса**

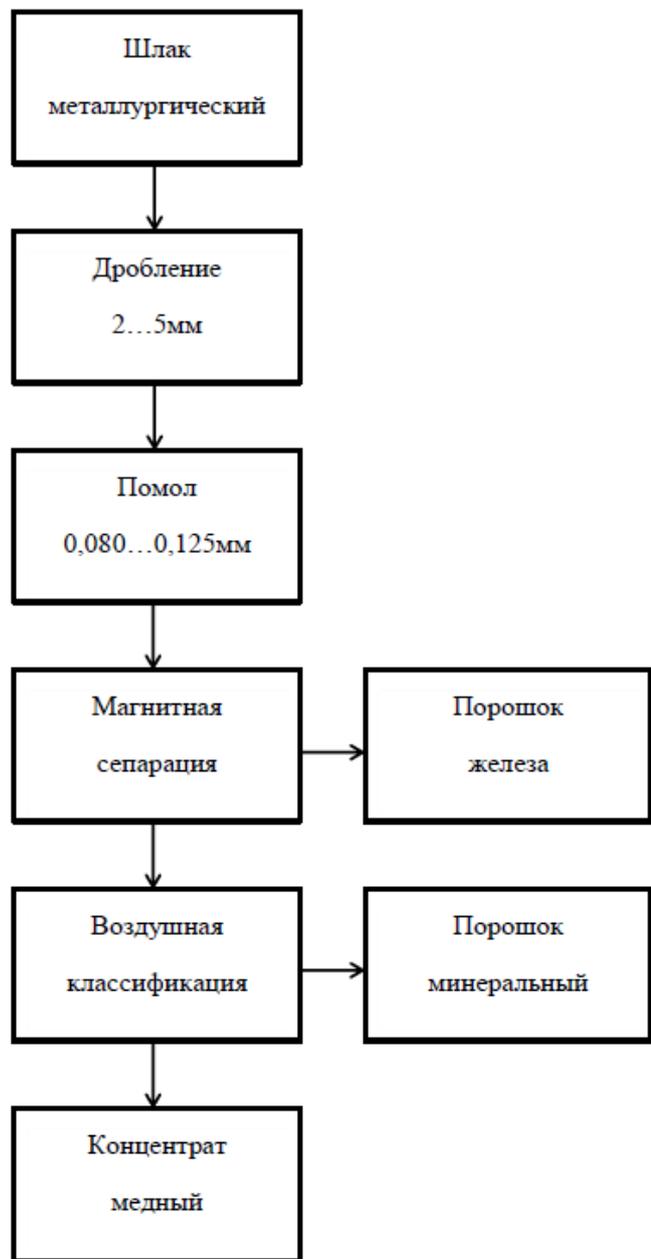


Рисунок 1.3 – Технологический процесс помола

В таблицах 1.2, 1.3 представлены основные технические характеристики и выходные параметры турбулентного измельчителя МПП-04.

Таблица 1.2 – Основные технические характеристики турбулентного измельчителя МПП-04

Наименование параметра	Значение
1	2
Привод	Электродвигатель Р=22кВт, n=1500 об/мин, тип 180S4
Передаточное число ременной передачи	3

Продолжение таблицы 1.2

1	2
Скорость вращения барабанов в переносном движении	480 об/мин
Скорость вращения барабанов в относительном движении	1460 об/мин
Ускорение мелющих шаров	100g
Размер мелющих шаров в планетарной секции барабана	25 мм
Размер мелющих шаров в вибрационной секции барабана	10 мм
Степень заполнения мелющими шарами планетарной секции барабана	25%
Степень заполнения мелющими шарами вибрационной секции барабана	80%
Режим работы мельницы	Циклический
Удельное давление шаров на рабочую поверхность барабанов	200 Мпа
Температура в зоне контакта шара с рабочей поверхностью барабана	900 <sup>0</sup> С
Предельные размеры подлежащего помолу исходного сырья	5 мм
Твердость подлежащего помолу материала по шкале проф. Протодяконова	15
Число рабочих барабанов	4
Вес изделия	1860 кг

Таблица 1.3 – Выходные параметры турбулентного измельчителя МПП-04

Наименование параметра	Значение
Производительность	40 кг/мин
Размеры частиц материала	(2...4)*10 <sup>-6</sup> м
Удаление молотого материала из мельницы	Вакуум-насос
Способ помола	Сухой
Условия работы	Взрывоопасные по пыли и газу
Уровень шума	100 Дб

Мельница оснащена прибором гранулометром, предназначенным для непрерывного автоматического контроля гранулометрического состава шлака без отбора проб. Этот контроль производится непосредственно в потоке, что обеспечивает непрерывное определение и отображение на дисплее измеряемой информации. Она представляет собой величины процентного содержания контрольных классов крупности частиц в пульпе. Далее соответствующие сигналы выдаются на регулирующие устройства с передачей данных по каналам связи.

В работе [3, с. 29] подробно описаны результаты экспериментов, на основе которых был определён размер частиц, при котором оптимально можно выделить конкреции металлической меди из шлаков из её минеральной части. На основании математической обработки экспериментальных данных в [3, с. 29; 25-27] описывается получение уравнения регрессии, анализ которого

позволил установить оптимальные параметры процесса измельчения. На основе экспериментальных исследований по помолу таких видов техногенного сырья, как шлаки медеплавильного и литейного производств, было показано, что, используя объемную деформацию минеральной части шлаков и пластическую деформацию металлических конкреций, можно практически полностью извлекать металл из указанных видов техногенного сырья.

Эти данные могут быть непосредственно использованы специалистами по управлению для реализации системы управления помолом с помощью обратной связи.

Турбулентные измельчители являются специализированными механическими устройствами, используемыми на промышленном производстве для осуществления тонкого помола различных материалов.

Таким образом, актуальной представляется задача разработки алгоритмов управления процессом тонкого помола, обеспечивающих повышение качества помола, быстроедействие, снижение себестоимости конечного продукта.

Развитие современной теории управления, а также компьютерной техники и компьютерных технологий определяет возможность практического использования регуляторов с многоцелевой структурой [28] для управления турбулентными измельчителями.

### **Особенности реализации автоматического управления технологическими процессами**

Современная теория управления имеет дело с системами, которые обладают качествами самоорганизации, приспособляемости, робастности, оптимальности и обучаемости.

Различные автоматические устройства позволяют улучшать потребительские качества выпускаемой продукции.

Процесс облегчения труда человека за счет выдающихся достижений компьютерных технологий вызывают важные социальные изменения: по способности собирать и обрабатывать информацию компьютеры всё больше приближаются к человеческому мозгу.

Автоматизация позволяет увеличить производительность и повысить качество выпускаемой продукции.

На современном этапе автоматизацию можно определить, как технологию, использующую запрограммированные команды, воздействующие на некоторый объект или процесс, и обратную связь, с помощью которой определяется, правильно ли исполнены эти команды.

После автоматизации процесс может функционировать без помощи или вмешательства человека. Фактически большинство автоматизированных систем способны выполнять свои функции с большей точностью и намного быстрее, чем это было при ручном управлении.

Процессы обратной связи являются критическими аспектами большинства инженерных систем. Обратная связь возникает, когда выход системы влияет на вход той же системы.

Управление с обратной связью – это процесс создания контура обратной связи для изменения поведения динамической системы через отклонение от нормы, определяемое измерениями системы [29].

Окружающий мир активно формируется процессами обратной связи – от извилистого пути к регулированию, происходящему внутри каждого элемента тела.

Рисунок 1.4 иллюстрирует общую схему системы управления с обратной связью. Физическая система контролируется датчиками  $y$  и управляется исполнительными механизмами  $u$  через управляющую логику.

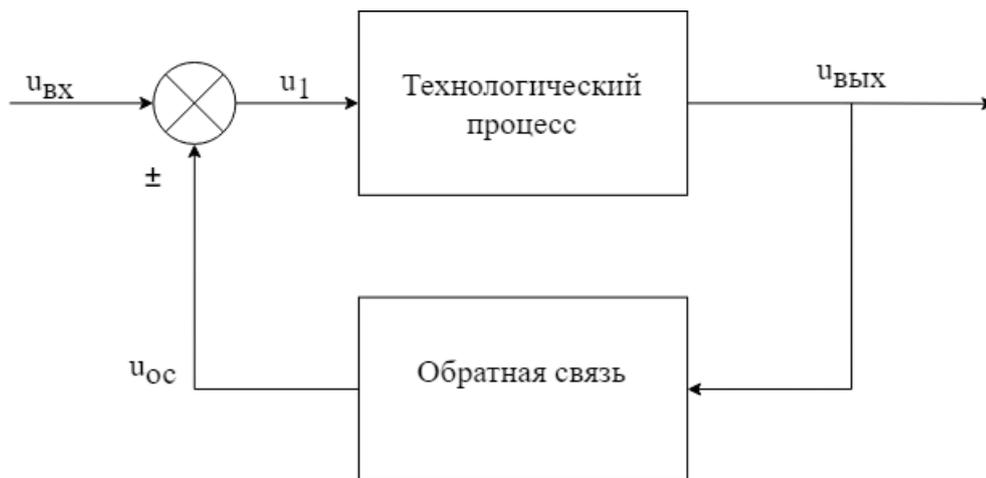


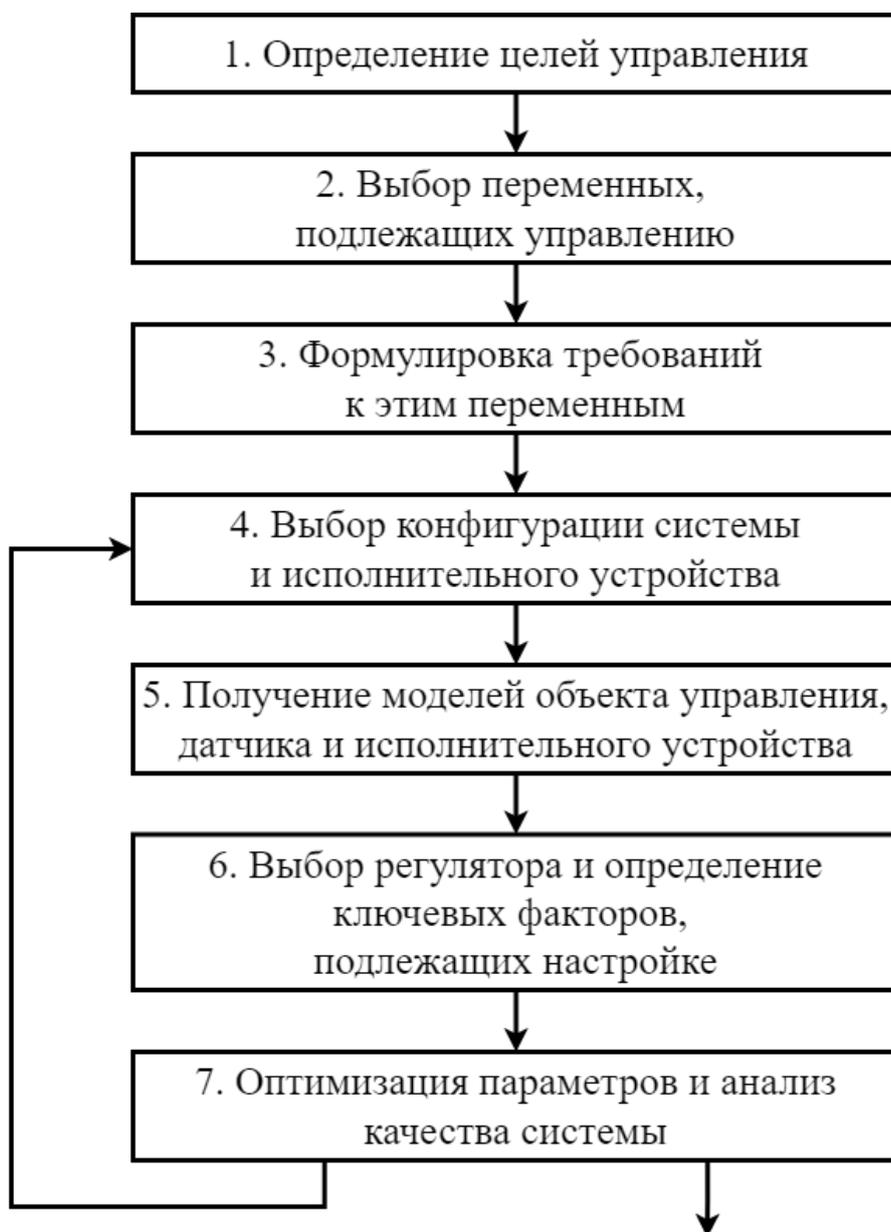
Рисунок 1.4 – Общая схема оптимизации для контроля обратной связи

Поведение физической системы модифицируется управляющим сигналом (входы,  $u$ ) через закон управления, основанного на измерениях датчиков системы (выходы,  $y$ ). Управляющая логика предназначена для формирования реакции замкнутого контура от внешних возмущений к цели, кодируемой квадратичным интегральным функционалом  $J$  [30-35].

Подводя итог можно сказать, что цель проектирования системы управления состоит в определении конфигурации системы, требований, которым она должна удовлетворять, и задании основных параметров, удовлетворяющих предъявляемым к системе требованиям.

Процесс синтеза системы управления при заданной модели объекта управления и установленными целями управления можно представить следующей схемой, представленной на рисунке 1.5.

выб



Если качество системы не удовлетворяет предъявленным требованиям, изменить ее конфигурацию и, возможно, выбрать другое исполнительное устройство

Если качество системы удовлетворяет предъявленным требованиям, закончить процедуру синтеза

Рисунок 1.5 – Процесс синтеза системы управления

## Моделирование динамических процессов для турбулентного измельчителя

Исследовав технологический процесс помола в турбулентном измельчителе, стало возможным выделить присущие особенности:

– в настоящее время турбулентный измельчитель относится к классу сложных систем, в которых оптимальное управление осуществляется в условиях неопределенности из-за отсутствия достаточной статистики поведения контролируемых объектов;

– анализируемая среда представляет собой вихревое неоднородное нелинейное механическое поле;

– такие факторы как давление на стенки барабана, влияние условий эксплуатации, влияние структуры мельницы, расщепление и структурные дефекты и т.д., влияющие на управление помолом не имеют возможности быть учтены;

– для построения модели на основе дифференциальных уравнений динамического процесса измельчения необходимо провести ряд сложных и дорогостоящих экспериментов.

Учитывая вышеописанное следует понимать, что формула, полученная ниже в данном разделе, работает только при постоянных параметрах и носит весьма приближенный характер и может отображать лишь некоторые тенденции технологического объекта.

В данном случае для построения модели системы будем использовать метод параметрической идентификации [36-41].

В работе [3, с. 30] описан ход экспериментальных исследований, проводившихся на турбулентном измельчителе. В качестве входной переменной  $x$  примем отклонение тонкости помола от желаемого значения, а в качестве выходной переменной  $y$  – отклонение объема входящего в мельницу потока от номинального значения. В таблице 1.4 представлены значения переходной характеристики объекта, полученные экспериментально, в виде его реакции на ступенчатое единичное входное воздействие.

Таблица 1.4 – Значения переходной характеристики объекта в виде его реакции на ступенчатое единичное входное воздействие

$t_1, c$	0	$t_9, c$	40	$t_{17}, c$	80
$y_1, \%$	0	$y_9, \%$	76,75	$y_{17}, \%$	96,36
$t_2, c$	5	$t_{10}, c$	45	$t_{18}, c$	85
$y_2, \%$	12,92	$y_{10}, \%$	81,00	$y_{18}, \%$	96,10
$t_3, c$	10	$t_{11}, c$	50	$t_{19}, c$	90
$y_3, \%$	26,10	$y_{11}, \%$	86,53	$y_{19}, \%$	95,70
$t_4, c$	15	$t_{12}, c$	55	$t_{20}, c$	95
$y_4, \%$	39,85	$y_{12}, \%$	91,88	$y_{20}, \%$	95,10
$t_5, c$	20	$t_{13}, c$	60	$t_{21}, c$	100
$y_5, \%$	49,26	$y_{13}, \%$	97,23	$y_{21}, \%$	92,30

$t_6, \text{ с}$	25	$t_{14}, \text{ с}$	65	$t_{22}, \text{ с}$	105
$y_6, \%$	58,49	$y_{14}, \%$	97,79	$y_{22}, \%$	90,15
$t_7, \text{ с}$	30	$t_{15}, \text{ с}$	70	$t_{23}, \text{ с}$	110
$y_7, \%$	67,71	$y_{15}, \%$	97,82	$y_{23}, \%$	89,90
$t_8, \text{ с}$	35	$t_{16}, \text{ с}$	75	$t_{24}, \text{ с}$	115
$y_8, \%$	71,40	$y_{16}, \%$	98,00	$y_{24}, \%$	87,60

Для формирования структуры передаточной функции предварительно введем в рассмотрение дифференциальное уравнение, описывающее динамику тонкого помола, принимая его порядок равным трем, что вполне достаточно для грубого описания процесса. Это уравнение имеет следующий вид:

$$\begin{aligned} \frac{d^3 y(t)}{dt^3} + a_2 \frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_0 y(t) = \\ = b_2 \frac{d^2 u(t)}{dt^2} + b_1 \frac{du(t)}{dt} + b_0 u(t). \end{aligned} \quad (1.1)$$

Записывая уравнение (1.1) в изображениях по Лапласу при нулевых начальных условиях, получим передаточную функцию  $H(s)$  объекта управления в виде:

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}, \quad (1.2)$$

где постоянные коэффициенты  $a_2, a_1, a_0, b_2, b_1, b_0$  подлежат поиску с использованием представленных выше экспериментальных данных.

В принципе, такой поиск может быть выполнен с использованием любой процедуры параметрической идентификации [42]. Однако частность рассматриваемой ситуации позволяет применить специализированный грубый, но достаточно простой практический подход, существо которого состоит в следующем.

Прежде всего, используя любой метод дискретизации модели (1.1) (по Эйлера, по Тастину и др.) с периодом  $\Delta t = h$ , осуществим переход к разностному уравнению, представляющему объект, функционирующий в дискретном времени  $n \in N^1$ . С использованием понятия обратных конечных разностей  $\nabla y[n] = y[n] - y[n-1]$ ,  $\nabla u[n] = u[n] - u[n-1]$ , это уравнение примет вид:

$$\begin{aligned} y[n] + \alpha_1 y[n-1] + \alpha_2 y[n-2] + \alpha_3 y[n-3] = \\ = \beta_0 u[n] + \beta_1 u[n-1] + \beta_2 u[n-2] + \beta_3 u[n-3]. \end{aligned} \quad (1.3)$$

Особо отметим, что коэффициенты этого уравнения взаимно однозначно связаны с искомыми коэффициентами передаточной функции (1.2) (эта связь



$$\mathbf{c}_e^0 = (-1.235 \quad 0.1029 \quad 0.2175 \quad 7.713)^T$$

минимизирующий норму невязки для системы (1.4). Этот вектор определяет значения коэффициентов модели (1.3), которые в среднем наилучшим образом приближают ее выходные значения к экспериментальным данным. Передаточная функция для дискретной системы (1.3) с найденными величинами коэффициентов имеет вид:

$$H_d(z) = \frac{7.713z^3}{z^3 - 1.235z^2 + 0.1029z + 0.2175}. \quad (1.5)$$

И, наконец, от передаточной функции (1.5) перейдем к соответствующей системе непрерывного времени (функция `d2c` в MATLAB, реализующая метод Тастина), т.е. к искомой передаточной функции (2.2):

$$H(s) = \frac{3.638s^3 + 4.365s^2 + 1.746s + 0.2328}{s^3 + 0.9026s^2 + 0.07618s + 0.002577}. \quad (1.6)$$

Однако заметим, что найденная функция (1.6) не является строго правильной дробью, что не совсем удобно для дальнейших расчетов. В связи с этим, осуществим в ручном режиме корректировку ее коэффициентов в числителе и знаменателе. В результате получим:

$$H(s) = \frac{141.0s^2 - 1.500s + 2.400}{s^3 + 8.500s^2 + 0.5500s + 0.02600}. \quad (1.7)$$

Передаточная функция (1.7) и принимается далее в качестве характеристики математической модели в `tf`-форме:

$$y = H(s)u \quad (1.8)$$

рассматриваемого объекта управления.

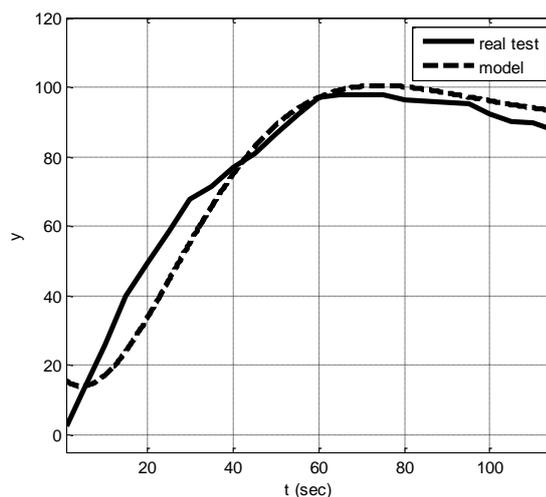


Рисунок 1.10 – Графики динамики эксперимента и идентифицированной модели

Сравнение свидетельствует, что динамика идентифицированной модели не в достаточной мере приближена к поведению объекта, наблюдаемого в ходе эксперимента, и не рекомендуется для использования для промышленных целей.

### Выводы по первому разделу

- проблема эффективного и рационального использования промышленных и техногенных образований имеет исключительно важное значение;

- в решении задачи переработки техногенных минеральных образований требуется измельчать их до определённой крупности;

- для повышения эффективности технологического процесса извлечения ценных компонентов необходимо использовать системы управления, которые обладают возможностью работать с быстро изменяющимися, сложными динамическими процессами, учитывать неполную и неоднозначную информации о процессе;

- управление должно обеспечивать высокую точность при простой и недорогой аппаратной реализации;

- при реализации традиционного метода, из-за сложности или невозможности измерения, ряд параметров и показателей технологического процесс помола для турбулентного измельчителя не описывается количественно, что затрудняет использование детерминированных математических методов для моделирования и оптимизации их режимов работы;

- в настоящее время турбулентный измельчитель относится к классу сложных систем, в которых управление осуществляется в условиях неопределенности из-за отсутствия достаточной статистики поведения контролируемых объектов;

- традиционные методы моделирования не приводят к удовлетворительным результатам, когда первоначальное описание решаемой проблемы неясно и неполно;

– для построения модели на основе дифференциальных уравнений динамического процесса измельчения, учитывающий все физико-механические изменения, необходимо провести ряд сложных и дорогостоящих экспериментов. При этом полученная математическая модель носит весьма приближенный характер и может отображать лишь некоторые тенденции технологического объекта;

– к полученной приближенной математической модели добавляются новые ошибки, связанные вычислительными процедурами и методами;

– при применении длинных вычислений процесс может выходить из необходимого диапазона, что препятствует достижению требуемого качества управления измельчением;

– при быстро текущих технологических процессах данный метод неэффективен – построенная модель была одномерной, что не может в полной мере отразить процессы в многомерной нелинейной системе турбулентного измельчителя породы;

– теория нечетких множеств – один из способов преодоления вышеописанных недостатков для построения системы управления сложным технологическим процессом.

## 2 ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТУРБУЛЕНТНЫМ ИЗМЕЛЬЧИТЕЛЕМ

### Основные принципы построения интеллектуальных систем

При управлении объектом обеспечивается стабильность той или иной физической величины, характеризующей состояние объекта, или изменение этой величины и окружающей среды в соответствии с некоторым законом на основе достоверной информации о состоянии объекта. Это делается с использованием автоматизированной системы управления (АСУ) с обратной связью.

АСУ с обратной связью состоит из двух основных частей: объекта управления и устройства управления. Объектом управления может быть любое управляемое техническое устройство или любой технологический процесс.

На рисунке 2.1 приведена классическая функциональная схема АСУ.

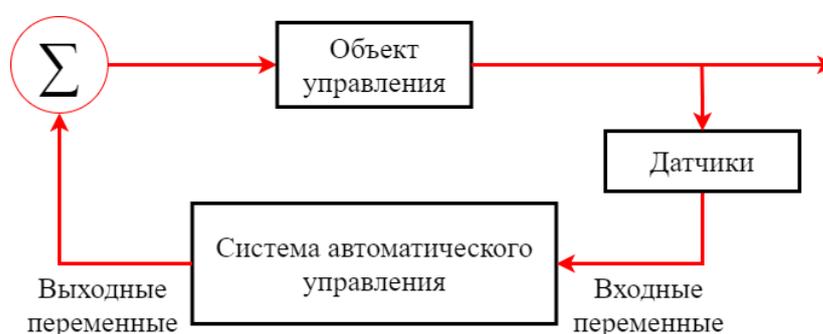


Рисунок 2.1 – Классическая функциональная схема АСУ

В большинстве случаев проектирования и эксплуатации АСУ установлено, что принятие априорности некоторой неизменной математической модели объекта управления не соответствует реальной ситуации [59]. В одних случаях это результат того, что из-за сложности процессов, происходящих в объекте управления, получение его математической модели, основанной на известных физических законах, оказывается неразрешимой задачей, в других это может быть результатом, изменяющим сам объект управления в процессе эксплуатации АСУ под воздействием внешних или внутренних неконтролируемых факторов.

Традиционно система определяется как сложная, если априорной информации недостаточно для построения адекватной модели [60]. Это означает, что порядок сложной системы на самом деле зависит от факторов, влияние которых неизвестно. Поэтому в случае, когда объектом управления является сложная система, можно говорить о ее модели, что неизвестны не только коэффициенты или параметры аналитических выражений, характеризующих объект управления, но и сами эти выражения.

В данной работе объект управления, а именно турбулентный измельчитель породы, представляет собой сложную систему, в которой связь между наблюдаемыми параметрами и состоянием системы имеет сложный и неоднозначный характер. Традиционный метод является применимым, в тех случаях, когда объектом управления является устойчивая линейная система, и

внешними воздействиями в которой можно пренебречь. Однако в случае, когда объектом управления является сложная, нелинейная, многопараметрическая система, такая как турбулентный измельчитель, традиционный метод становится неэффективным.

В моделях сложных динамических систем необходимо включать использование интеллектуальных процессов принятия решений для создания управляющих воздействий, чтобы поддерживать определенный уровень производительности.

Интеллектуальная система – это техническая или программная система, способная решать задачи, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы. Структура интеллектуальной системы включает объект управления, исполнительные механизмы, датчики и системы управления, которые содержат три основных блока – базу знаний, механизм вывода решений и интеллектуальный интерфейс (рисунок 2.2) [61].



Рисунок 2.2 – Общая структура интеллектуальных систем

Интеллектуальное управление – включают в себя различные методы и подходы искусственного интеллекта, такие как искусственные нейронные сети, нечёткая логика, эволюционные вычисления и генетические алгоритмы [62].

Методологии интеллектуального управления применяются в робототехнике и автоматизации, связи, производстве, управлении дорожным движением, и это лишь некоторые области применения.

Дальнейшее развитие этих направлений показало очень эффективное решение этих задач, в случае слабо детерминированных нелинейных или сложно математически описываемых задач.

Искусственная нейронная сеть – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами.

Несмотря на широкий спектр возможностей искусственной нейронной сети, решению задач с их помощью сопутствует ряд недостатков:

- большинство подходов к проектированию искусственной нейронной сети являются эвристическими и часто не приводят к однозначным решениям;
- для построения модели объекта на основе искусственной нейронной сети требуется многоцикловая регуляция внутренних элементов и связей между ними;
- проблемы, возникающие при подготовке учебного образца, связаны с трудностями нахождения достаточного количества примеров упражнений;
- сетевое обучение в некоторых случаях сводится к тупику;
- длительные затраты времени на выполнение тренировочной процедуры часто не позволяют использовать искусственную нейронную сеть в системах реального времени.

Нечёткая логика (англ. fuzzy logic) – раздел математики, являющийся обобщением классической логики и теории множеств, базирующийся на понятии нечёткого множества, впервые введённого Лотфи Заде в 1965 году как объекта с функцией принадлежности элемента ко множеству, принимающей любые значения в интервале  $[0,1]$ , а не только 0 или 1. На основе этого понятия вводятся различные логические операции над нечёткими множествами и формулируется понятие лингвистической переменной, в качестве значений которой выступают нечёткие множества [63].

Недостатки нечётких систем являются:

- отсутствие стандартных методов построения сложных систем;
- невозможность математического анализа сложных систем существующими методами;
- использование неоднозначного подхода по сравнению с вероятностью не приводит к увеличению точности расчетов.

Эволюционные вычисления, в том числе генетический алгоритм – направление в искусственном интеллекте, которое использует и моделирует процессы естественного отбора [64].

Недостатками эволюционных исчислений являются:

- нет недавней гарантии нахождения оптимального решения – алгоритм выполняет локальную оптимизацию;
- слабый теоретический базис – алгоритм является эвристическим, т.е. точность и строгость изложения приносят в жертву осуществимость;
- сложность использования – может потребоваться настройка параметров модели, оптимизированной эволюционным алгоритмом;
- высокие вычислительные затраты – алгоритм не масштабируется.

По результатам исследования для реализации управления технологическим процессом помола была выбрана методика с базой знаний на

основе нечеткой логики, соединяющей в себе знания и опыт экспертов данной предметной области.

Основанием выбора нечеткого управления можно считать следующее:

- формализуется и объединяется опыт операторов и разработчиков в настройке регулирования;

- предлагается простой метод управления для сложных процессов;

- постоянно учитывается опыт по управлению процессами данного типа, принимая во внимание исключения разного рода и особенности системы;

- учитываются исходные данные разного рода и производятся объединение разных исходных данных;

- работает с несколькими входами и может обеспечивать несколько выходов, поскольку работа происходит исключительно на основе правил;

- не состоит из сложного математического анализа, поэтому очень легко проектируется;

- не требует высокоточных датчиков, компенсирует точность использованием интуитивного управления;

- работает стабильно в нелинейных системах.

Нечеткая логика имеет особые рабочие области: при принятии этих решений важно учитывать практический опыт, нюансы и нелинейные и субъективные параметры.

Теоретическим и практическим вопросам нечетких систем посвящены работы зарубежных и отечественных ученых: Заде Л.А. [65-67], Мамдани Е.А. [68,69], Томески М.Л. [70], Деменкова Н.П. [71], Васильева В.И. [72] и др. Мировой опыт показывает целесообразность применения непонятных методов для повышения эффективности управления сложными технологическими процессами, что находит подтверждение в работах Такаги и Сугено [73] и др.

В результате анализа научных трудов было установлено, что неустановленный контроль используется в нелинейных системах, идентификация которых требует слишком больших усилий, кроме того, в соответствии с условием задачи он необходим для использования знаний эксперта [74].

Нечеткая логика позволяет упростить создание и расширить возможности традиционных методик (рисунок 2.3).



Рисунок 2.3 – Систематизация выбора метода управления по критериям объекта

Человек часто решает сложные задачи на основе примерных исходных данных. Однако точность всех параметров исходной информации не требуется.

В промышленности операторы очень часто легко решают очень сложные и трудные проблемы, не прорабатывая сначала возможную проблему и не моделируя систему.

Из вышеизложенного можно сделать следующие выводы:

- часто проще и полезнее моделировать поведение оператора системы управления, чем моделировать работу самой системы;
- вместо использования точных математических расчетов и уравнений более эффективно использовать качественные оценки ситуации и применять соответствующие меры обработки.

Разработка нечеткой базы правил – это интерактивный процесс. По большей части это набор знаний и опыта. Одним из преимуществ нечеткой логики является возможность построения базы правил, которая утверждается экспертами перед проверкой ее работы в реальных условиях.

Создание базы знаний состоит из трех этапов:

- составление списка переменных, которые следует учитывать: они станут лингвистическими переменными в базе данных нечетких правил;
- составление списка качественных оценок, которые принимаются во внимание, и определения, когда они верны, а когда ложны: эти оценки станут лингвистическими терминами базы правил;

– составление формулировок правил: они должны описывать поведение в каждом конкретном случае.

Обоснование базы знаний состоит из нескольких этапов:

– презентация и обсуждение базы правил с экспертами, которые помогут собрать накопленные знания в этой области. Цель обсуждения – выявить упущенные моменты и обеспечить ясность правил;

– моделирование «разомкнутого цикла»: эксперты сравнивают поведение базы правил с ожидаемым поведением;

– если процесс можно смоделировать, то также выполняется моделирование «замкнутого цикла».

При реализации нечеткой логики необходимо выполнить следующие шаги [75]:

– изучить, как система базы правил и оператор могут быть применены к этому управлению;

– понять концепцию функций принадлежности и лингвистических переменных;

– проанализировать управляемую систему и решить, какие переменные состояния следует рассматривать в качестве входных данных для системы;

– сформировать понимание того, как управлять переменными состояниями, чтобы получить необходимый контроль системой;

– сформировать базу правил для лингвистических переменных входов и выходов;

– оптимизировать функции принадлежности, чтобы сделать управление более эффективным;

– интегрировать нечеткий регулятор в систему и проверить результат.

Управление нечеткой логикой в основном зависит от правил, формируемых лингвистическими переменными. Нечетко-логическое управление не требует сложных численных вычислений, в отличие от других методов. Он использует только простые математические вычисления для управления моделью. Несмотря на то, что он опирается на базовый математический анализ, он обеспечивает хорошую производительность в системе управления. Следовательно, этот метод является одним из лучших доступных методов, а также более простым для управления системой. Управление нечеткой логикой основано на теории нечетких множеств. В теории нечетких множеств каждый элемент имеет степень принадлежности, с которой он принадлежит какому-либо конкретному множеству. Можно сказать, что нечеткие множества подобны классическим множествам без гораздо более четких границ. Регулятор нечеткой логики чаще используется, когда требуемая точность является умеренной и система должна быть лишена сложного математического анализа [76].

Тремя основными компонентами нечеткого логического регулятора являются (рисунок 2.4) [77]:

1. Фазификация – процесс преобразования числовой переменной состояния в нечеткую входную лингвистическую переменную;

2. База нечетких правил и нечеткий логический вывод;
3. Дефазификация – методология предоставления количественного результата в нечеткой форме.

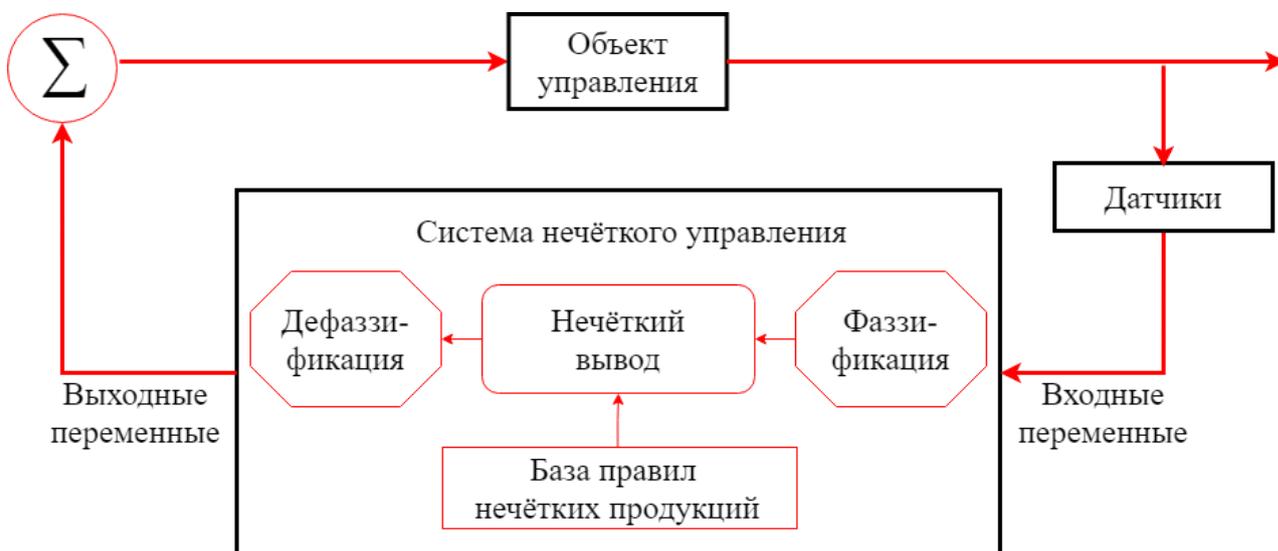


Рисунок 2.4 – Архитектура компонентов процесса нечёткого управления

## 2.2 Создание стенда и исследование режимов работы турбулентного измельчителя породы

В ходе исследований было установлено, что механоактивация – сложный процесс, ход которого можно регулировать, изменяя большое количество управляющих параметров, влияющих на энергетические, температурные и деформационные характеристики. Эти параметры определяют качество продукта, получаемого на выходе.

Ранее, в первой главе, были описаны проблемы переработки техногенных минеральных образований, обоснование выбора турбулентного измельчителя, а также его устройство и принцип работы.

Анализируя результаты исследований, в работе был сделан выбор в пользу интеллектуального управления, основанного на нечёткой логике. Обоснование выбора нечёткой логики и результаты работы синтезированного нечёткого управления представлено в разделе 2.1.

Для возможности перехода от компьютерной модели к практической реализации необходимо подтвердить адекватность выбранных параметров и их зависимость на процесс тонкого помола.

Следуя исходным данным, рассматриваемыми нами ранее во второй главе, факторами эксперимента определены следующие параметры:

X1 – частота вращения двигателя, об/мин – 300, 600, 900, 1200, 1500.

X2 – подача шлама в барабан, кг/ч – 200, 400, 600, 800, 1000.

Функциями отклика Y являются:

Y1 – Тонкость помола, мкм – 40...160

Y2 – Фоновый шум, дБ – 20...100

У3 – Общая загрузка барабана, т/ч – 1...5

У4 – Фоновая температура шлака, °С – 40...100

При исследовании факторов, влияющих на тонкий помол минеральной составляющей техногенного сырья, была выбрана шлак металлургического производства. Для эксперимента использовалась фракция не крупнее 5 мм, для чего исходный шлак подвергался дроблению на щековой дробилке.

Выбор шлака металлургического производства в качестве объекта исследования определяется следующими факторами. Во-первых, прочность данного материала по шкале прочности Протодяконова (5-7 единиц) соответствует прочности минеральной части некоторых видов техногенного сырья (например, шлаков медеплавильного и литейного производств). Во-вторых, минеральная часть указанных видов техногенного сырья представлена широким диапазоном фракционного состава – от пылевидного до указанных размеров (то есть до 5мм). Таким образом можно утверждать, что в случае полного измельчения в ходе эксперимента исследуемого шлака металлургического производств следует ожидать также и полное измельчения минеральной части техногенного сырья, которое уже частично измельчено.

Выбор пределов факторов эксперимента основан на литературных данных [7; 31, стр. 23; 32, стр. 167] и на данных предварительных экспериментов [3, стр. 30]. Степень заполнения помольных барабанов мелющими телами составлял примерно 25...30 %, а объем измельчаемой массы был равен примерно половине объема мелющих тел. Объем мелющих тел был выбран равным 30% от объема помольного барабана. В качестве мелющих тел использовались стальные шары диаметром 19 мм.

Эксперименты проводились на лабораторной установке (рисунок 2.5).



Рисунок 2.5 – Лабораторная установка

Проведение натурных экспериментов на лабораторной установке весьма дорогостоящий и трудоемкий процесс, и приходится учитывать большое число факторов, влияющих на эксперимент. По этой причине появилась необходимость разработки системы оптимального управления на базе имеющихся экспериментальных данных.

Цель данного раздела – подтвердить правильность выбранных входных и выходных параметров для проектируемой системы интеллектуального управления на основе нечеткой логики. Полученные экспериментальные данные позволили установить их зависимости относительно друг друга и стали базой знаний для формирования правил работы нечеткого регулятора.

В таблицах 2.1-2.4 показаны зависимости исследуемых параметров, полученных в результате натурных экспериментов.

Таблица 2.1 – Величина тонкости помола в зависимости от частоты вращения двигателя и скорости подачи шлака, мкм

Частота вращения двигателя, об/мин	Подача шлака в барабан, кг/ч				
	200	400	600	800	1000
300	140	128	119	106	96
600	132	119	112	104	91
900	128	112	106	96	82
1200	115	108	93	84	75
1500	107	99	97	78	66

Таблица 2.2 – Величина фонового шума в зависимости от частоты вращения двигателя и скорости подачи шлака, дБ

Частота вращения двигателя, об/мин	Подача шлака в барабан, кг/ч				
	200	400	600	800	1000
300	40	52	58	71	79
600	43	54	61	73	82
900	47	57	65	75	87
1200	51	59	68	77	90
1500	54	65	72	82	94

Таблица 2.3 – Величина общей загрузки барабана в зависимости от частоты вращения двигателя и скорости подачи шлага, кг/ч

Частота вращения двигателя, об/мин	Подача шлага в барабан, кг/ч				
	200	400	600	800	1000
300	3,075	3,3	3,5	3,65	3,9
600	2,95	3,25	3,4	3,55	3,8
900	2,9	3,2	3,3	3,5	3,7
1200	2,8	3,1	3,2	3,375	3,6
1500	2,7	3,0	3,1	3,3125	3,55

Таблица 2.4 – Величина фоновой температуры шлага в зависимости от частоты вращения двигателя и скорости подачи шлага, °С

Частота вращения двигателя, об/мин	Подача шлага в барабан, кг/ч				
	200	400	600	800	1000
300	52	56	57	59	64
600	56	59	61	63	68
900	61	64	65	71	73
1200	68	72	75	79	80
1500	76	80	83	88	90

На рисунках 2.6-2.9 представлены результаты экспериментальных измерений при заданных режимах

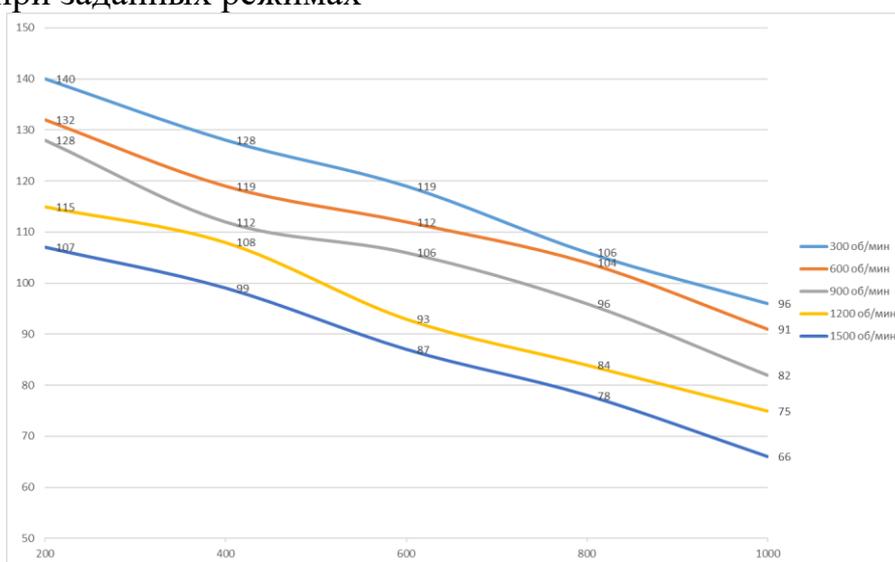


Рисунок 2.6 – Зависимость тонкости помола от оборотов двигателя и подачи шлага

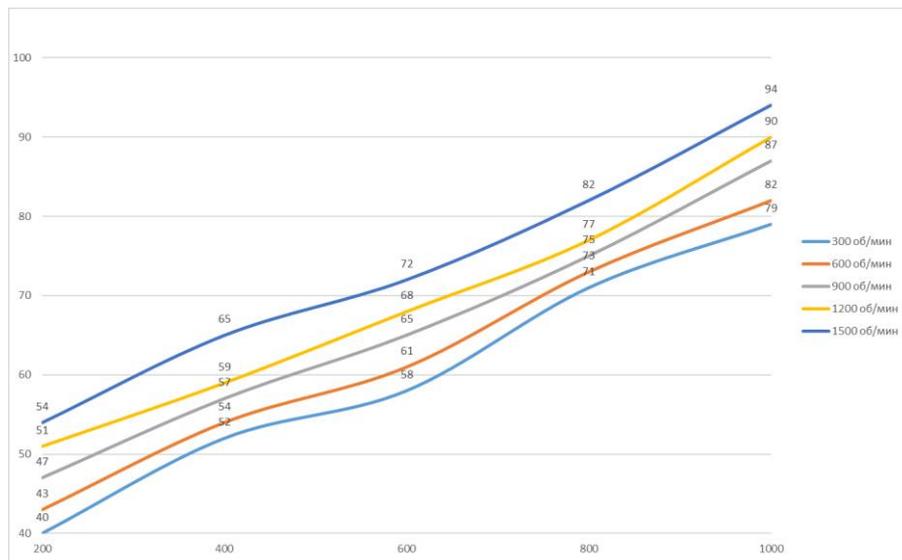


Рисунок 2.7 – Зависимость шума от оборотов двигателя и подачи шлага

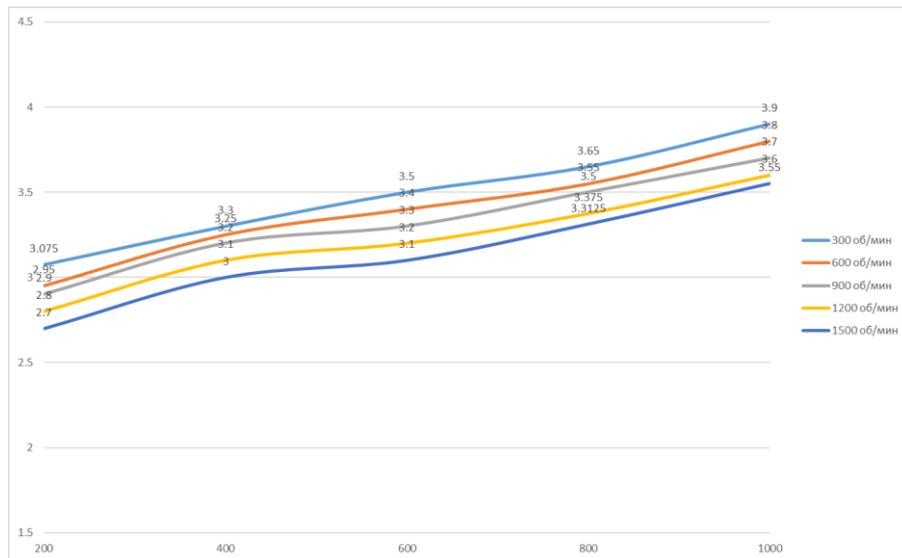


Рисунок 2.8 – Зависимость загрузки от оборотов двигателя и подачи шлага

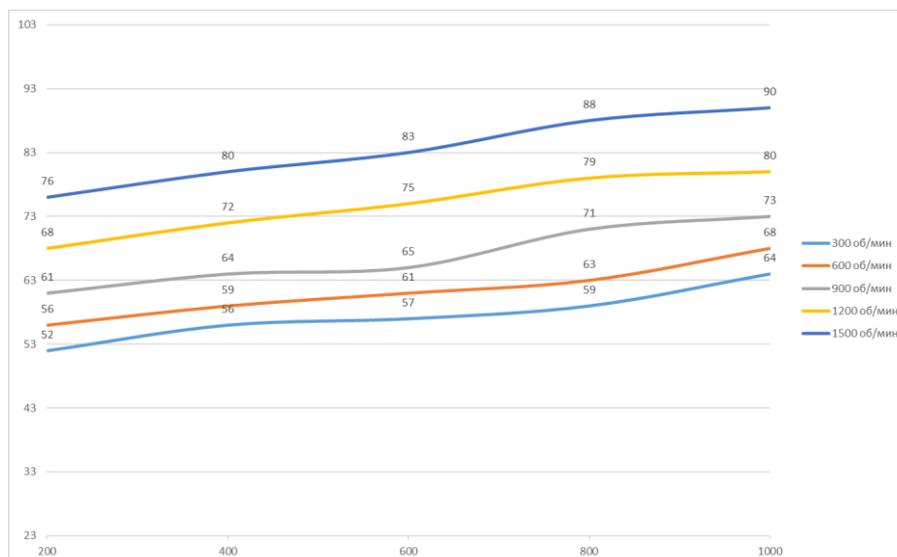


Рисунок 2.9 – Зависимость температуры от оборотов двигателя и подачи шлага

Анализируя полученные результаты были сделаны следующие заключения:

- для составления управляющих правил нечетких логических регуляторов требуется хорошее знание объекта управления и интуиция выступающего в роли эксперта оператора- технолога, т.е. накопленный им эксплуатационный опыт или множественная серия экспериментальных данных;

- на основании этих данных есть возможность смоделировать в MATLAB систему управления на базе нечёткой логики и реализовать лабораторной образец управления турбулентным измельчителем породы на микроконтроллере.

### **Моделирование в MATLAB системы управления на базе нечёткой логики**

Используя набор инструментов MATLAB «Fuzzy logic Designer» [78,79], показанный на рисунке 2.10, была смоделирована интеллектуальная система управления турбулентным измельчителем породы. Характеристики модели содержит приведены в таблице 2.5.

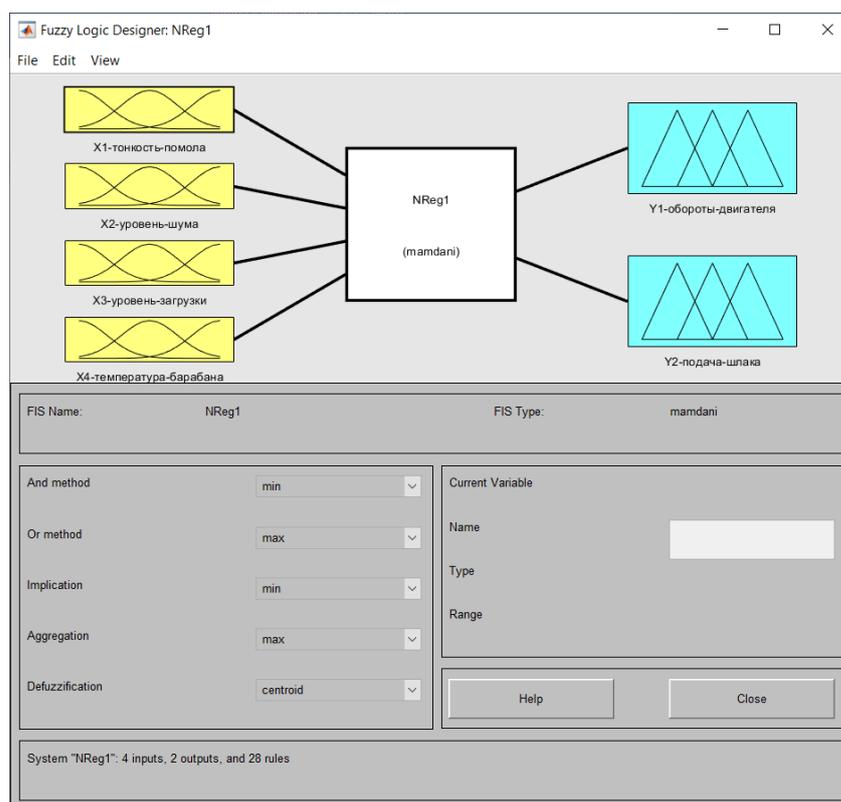


Рисунок 2.10 – Графический пользовательский интерфейс набора инструментов нечеткой логики в среде MATLAB «Fuzzy logic Designer»

Таблица 2.5 – Характеристики модели интеллектуальной системы управления турбулентным измельчителем породы

Controller type	Mamdani
And method	Min
Or method	Max
Implication	Min
Defuzzification	Centroid

Согласно процедуре построения управляющей модели [80,81], лингвистическая шкала по каждой из входных переменных была разбита на три и выходных переменных – на пять значений. Соответствие смысловых и числовых значений по используемым ЛП приведено ниже (рисунок 2.11 и таблица 2.6).

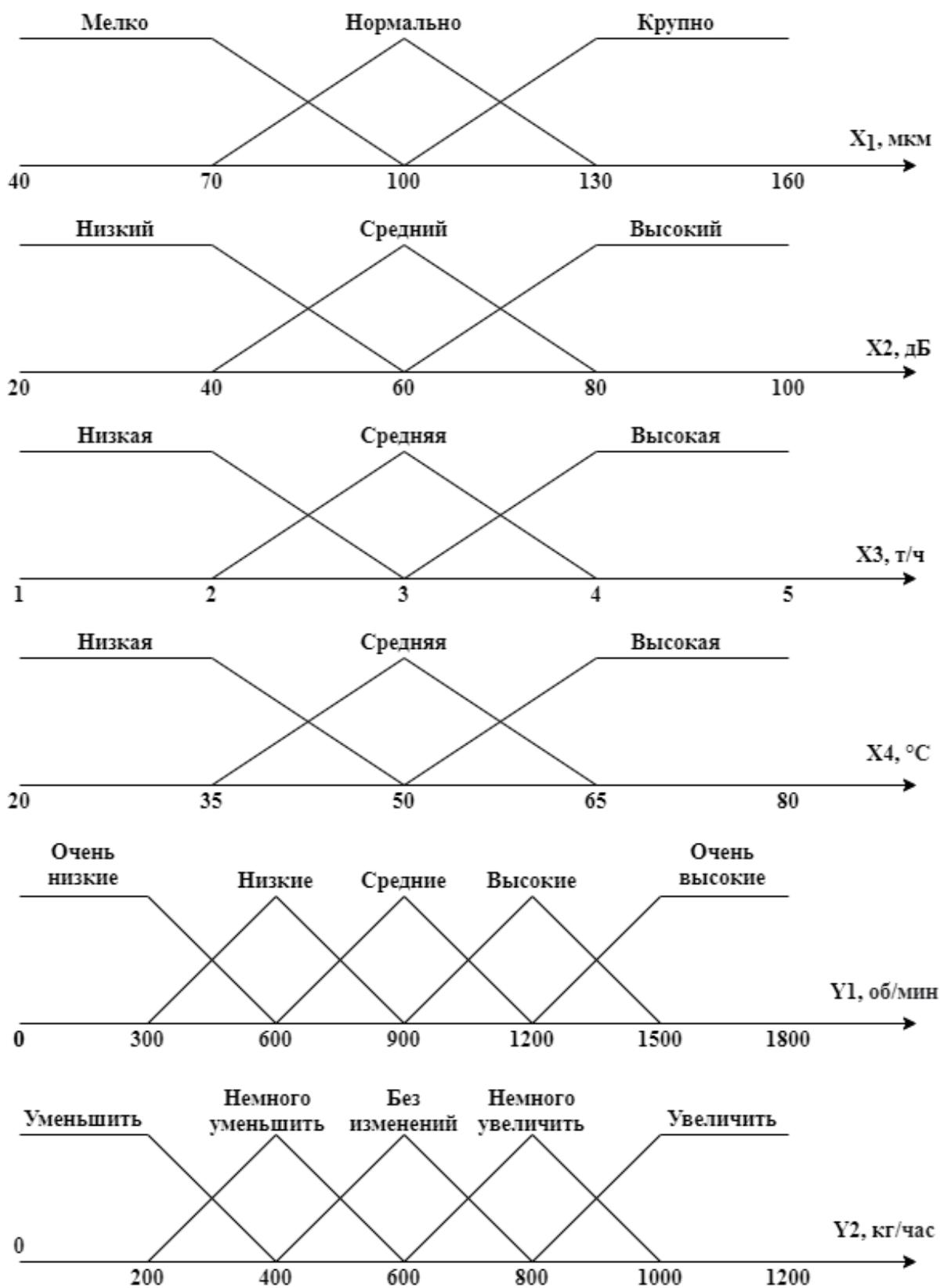


Рисунок 2.11 – Представление входных  $X_i$  и выходных  $Y_i$  переменных в виде ЛП

Таблица 2.6 – Соответствие значений ЛПП (по данным операторов – технологов)

Условное обозначение	Наименование ЛПП	Множество значений	Множество НЧ (L-R)-типа
X <sub>1</sub>	Тонкость помола	Мелко	(40,0,30)
		Нормально	(70,30,30)
		Крупно	(100,30,0)
X <sub>2</sub>	Уровень шума	Низкий	(40,0,20)
		Средний	(60,20,20)
		Высокий	(80,20,0)
X <sub>3</sub>	Уровень загрузки барабана	Низкая	(2,0,1)
		Средняя	(3,1,1)
		Высокая	(4,1,0)
X <sub>4</sub>	Внутренняя температура барабана	Низкая	(35,0,15)
		Средняя	(50,15,15)
		Высокая	(65,15,0)
Y <sub>1</sub>	Обороты двигателя	Очень низкие	(300,0,300)
		Низкие	(600,300,300)
		Средние	(900,300,300)
		Высокие	(1200,300,300)
		Очень высокие	(1500,300,0)
Y <sub>2</sub>	Подача шлама в мельницу	Уменьшить	(200,0,200)
		Немного уменьшить	(400,200,200)
		Без изменений	(600,200,200)
		Немного увеличить	(800,200,200)
		Увеличить	(1000,200,0)

Используемые функции принадлежности показаны на рисунках 2.12-2.17.

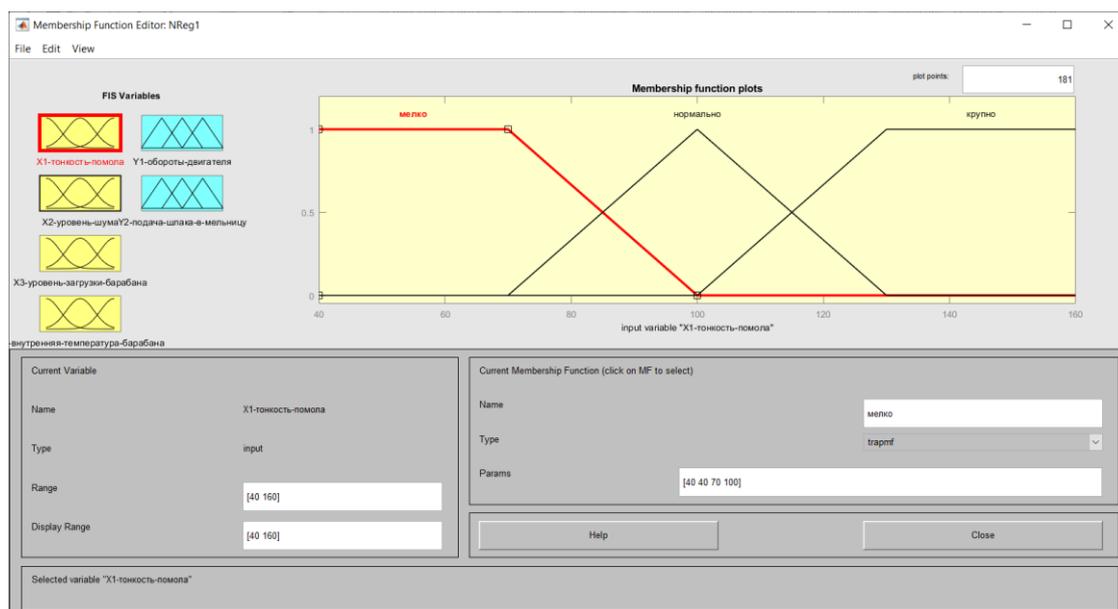


Рисунок 2.12 – Функции принадлежности Тонкость помола

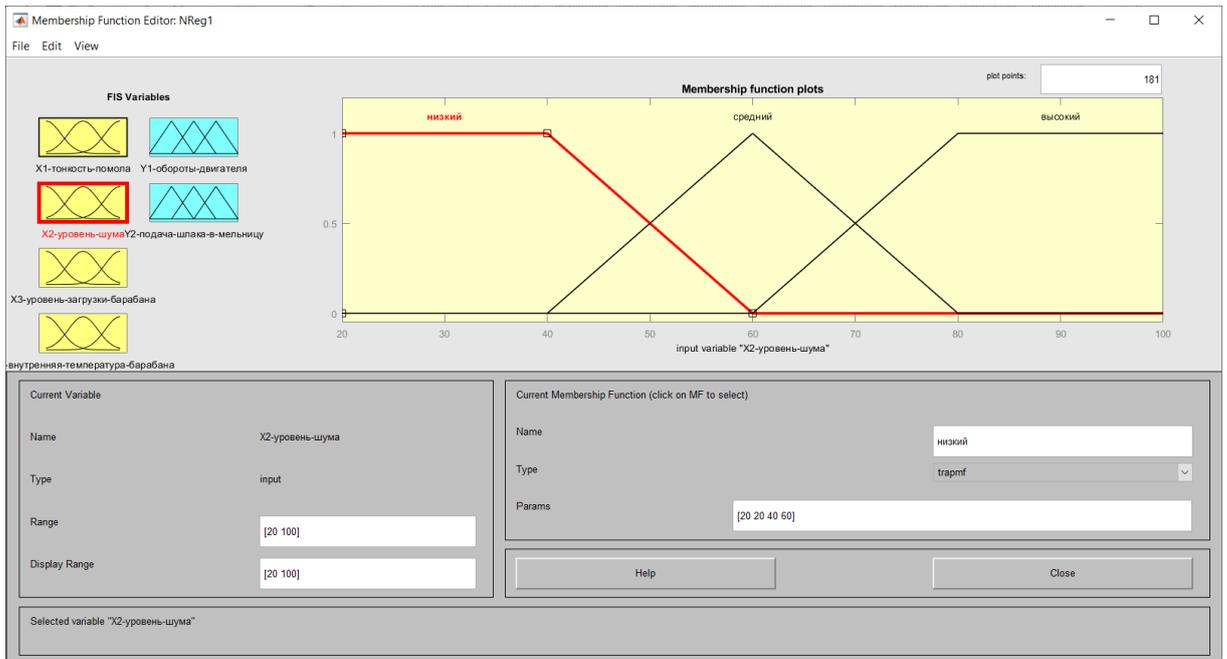


Рисунок 2.13 – Функции принадлежности Уровень шума

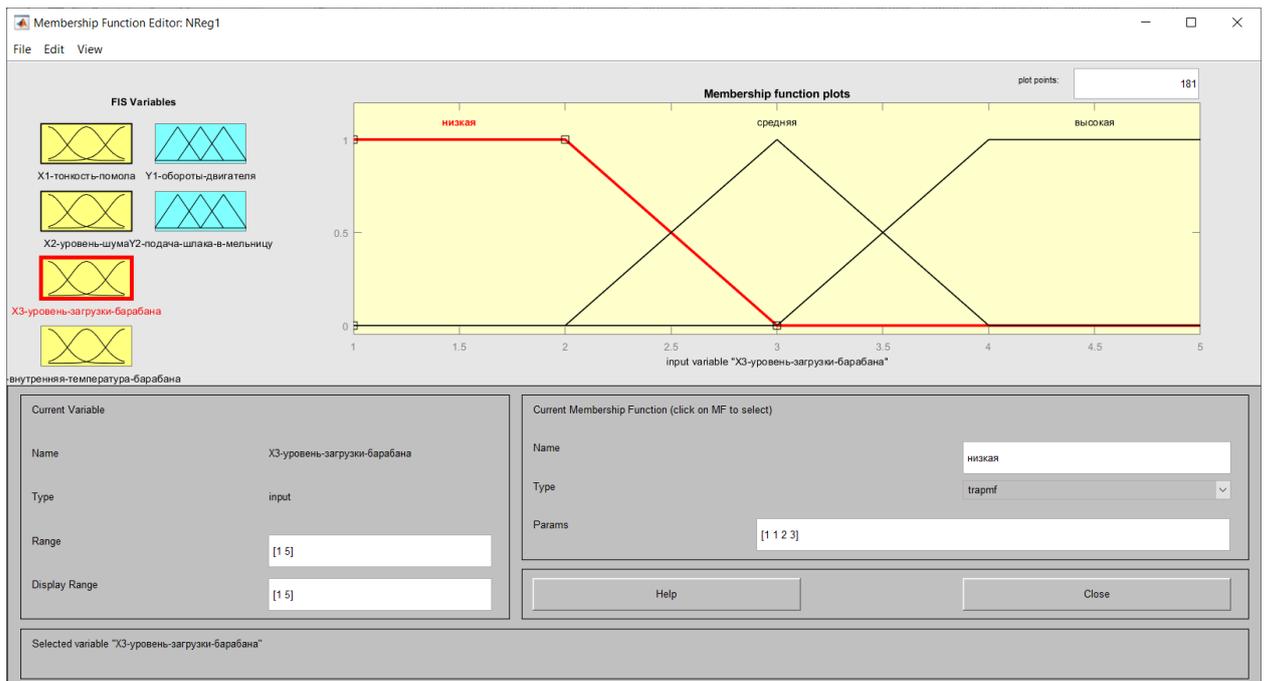


Рисунок 2.14 – Функции принадлежности Уровень загрузки барабана

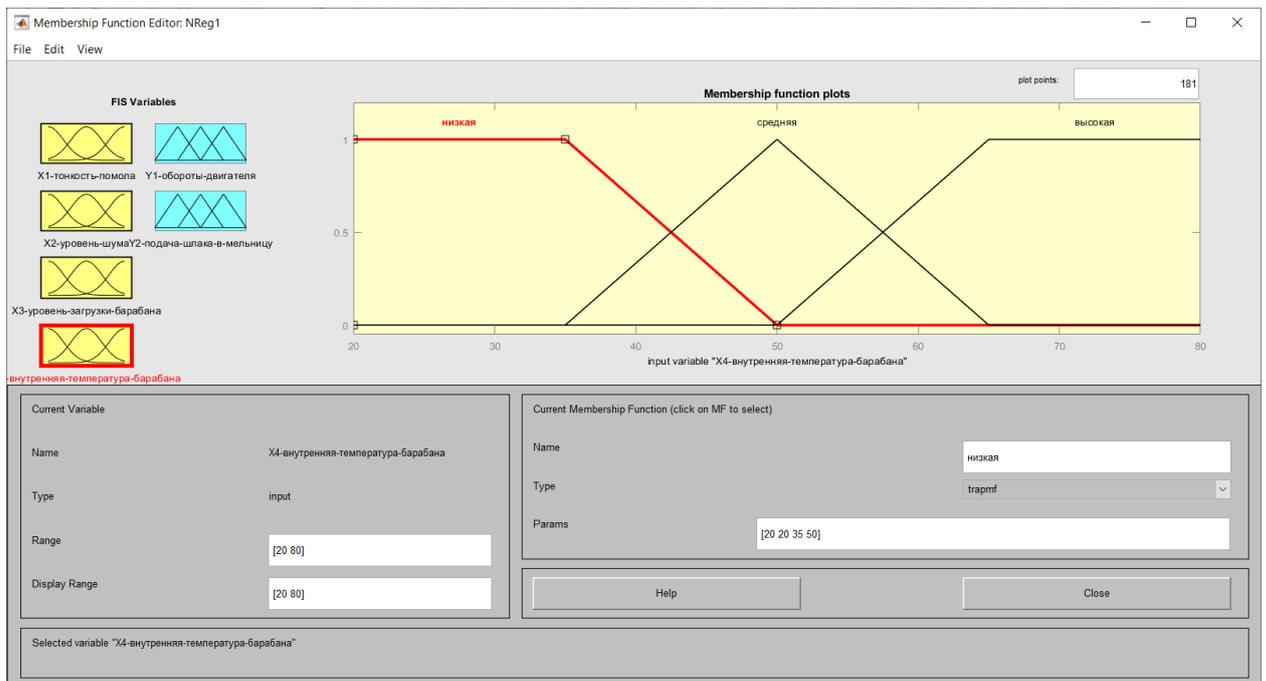


Рисунок 2.15 – Функции принадлежности Внутренняя температура барабана

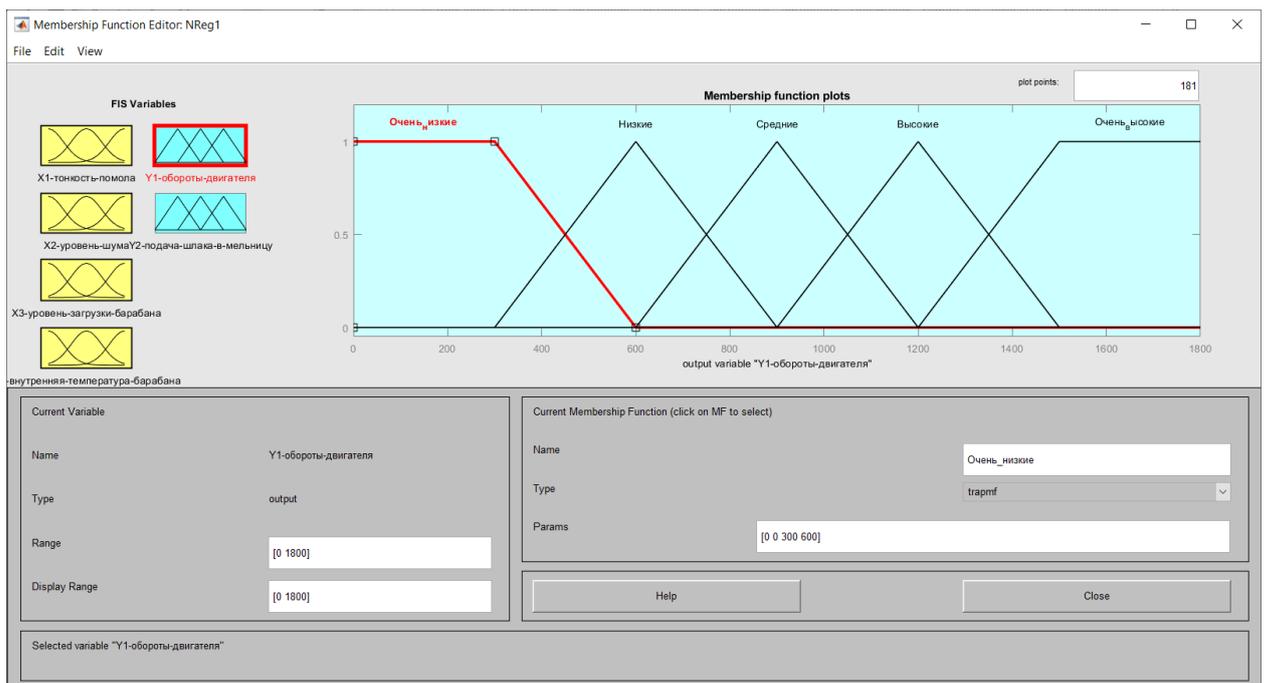


Рисунок 2.16 – Функции принадлежности Обороты двигателя

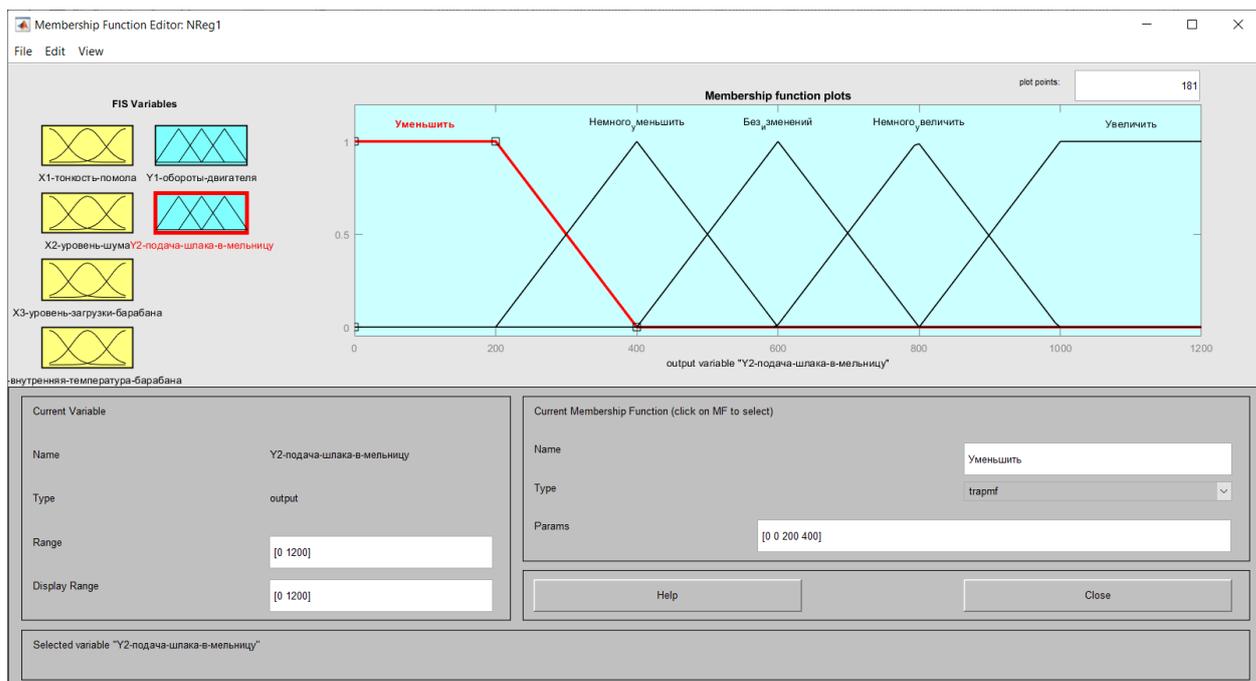


Рисунок 2.17 – Функции принадлежности Подача шлака в мельницу

В процессе создания правил эксперту предлагались ситуации в следующей форме:

«Если тонкость помола МЕЛКО и уровень шума НИЗКИЙ и уровень загрузки барабана НИЗКИЙ и внутренняя температура барабана НИЗКАЯ, то изменить обороты двигателя и подачу шлака – ?».

Результаты опроса представлен в таблице 2.7.

Таблица 2.7 – Матрица опрашиваемых ситуаций и ответов эксперта

Тонкость помола, $X_1$	Уровень шума, $X_2$	Уровень загрузки барабана, $X_3$	Внутренняя температура барабана, $X_4$	Обороты двигателя, $Y_1$	Подача шлака в мельницу, $Y_2$
1	2	3	4	5	6
мелко	низкий	низкая	низкая	очень низкие	увеличить
нормально	низкий	низкая	низкая	средние	увеличить
крупно	низкий	низкая	низкая	очень высокие	увеличить
мелко	средний	низкая	низкая	очень низкие	немного увеличить
мелко	высокий	низкая	низкая	очень низкие	немного уменьшить
мелко	низкий	средняя	низкая	очень низкие	без изменений
мелко	низкий	высокая	низкая	очень низкие	уменьшить
мелко	низкий	низкая	средняя	очень низкие	немного увеличить

Продолжение таблицы 2.7

1	2	3	4	5	6
мелко	низкий	низкая	высокая	очень низкие	немного уменьшить
нормально	средний	низкая	низкая	низкие	немного увеличить
нормально	средний	средняя	низкая	средние	без изменений
нормально	средний	средняя	средняя	низкие	без изменений
мелко	низкий	средняя	средняя	очень низкие	немного уменьшить
мелко	низкий	высокая	высокая	очень низкие	уменьшить
мелко	низкий	высокая	средняя	низкие	уменьшить
мелко	низкий	средняя	высокая	очень низкие	немного уменьшить
крупно	высокий	высокая	высокая	низкие	уменьшить
крупно	высокий	высокая	низкая	высокие	уменьшить
крупно	высокий	высокая	средняя	низкие	уменьшить
крупно	высокий	низкая	высокая	низкие	немного уменьшить
крупно	высокий	средняя	высокая	низкие	уменьшить
крупно	высокий	низкая	низкая	средние	без изменений
крупно	высокий	низкая	средняя	средние	немного увеличить
крупно	средний	высокая	высокая	средние	уменьшить
крупно	низкий	высокая	высокая	средние	уменьшить
крупно	средний	высокая	средняя	высокие	уменьшить
крупно	средний	высокая	низкая	высокие	уменьшить
крупно	средний	низкая	низкая	высокие	немного увеличить

Составленные правила, их общая интерпретация, а также просмотр поверхности нечетких правил, установленных для нечеткого управления, показаны на рисунке 2.18-2.20.

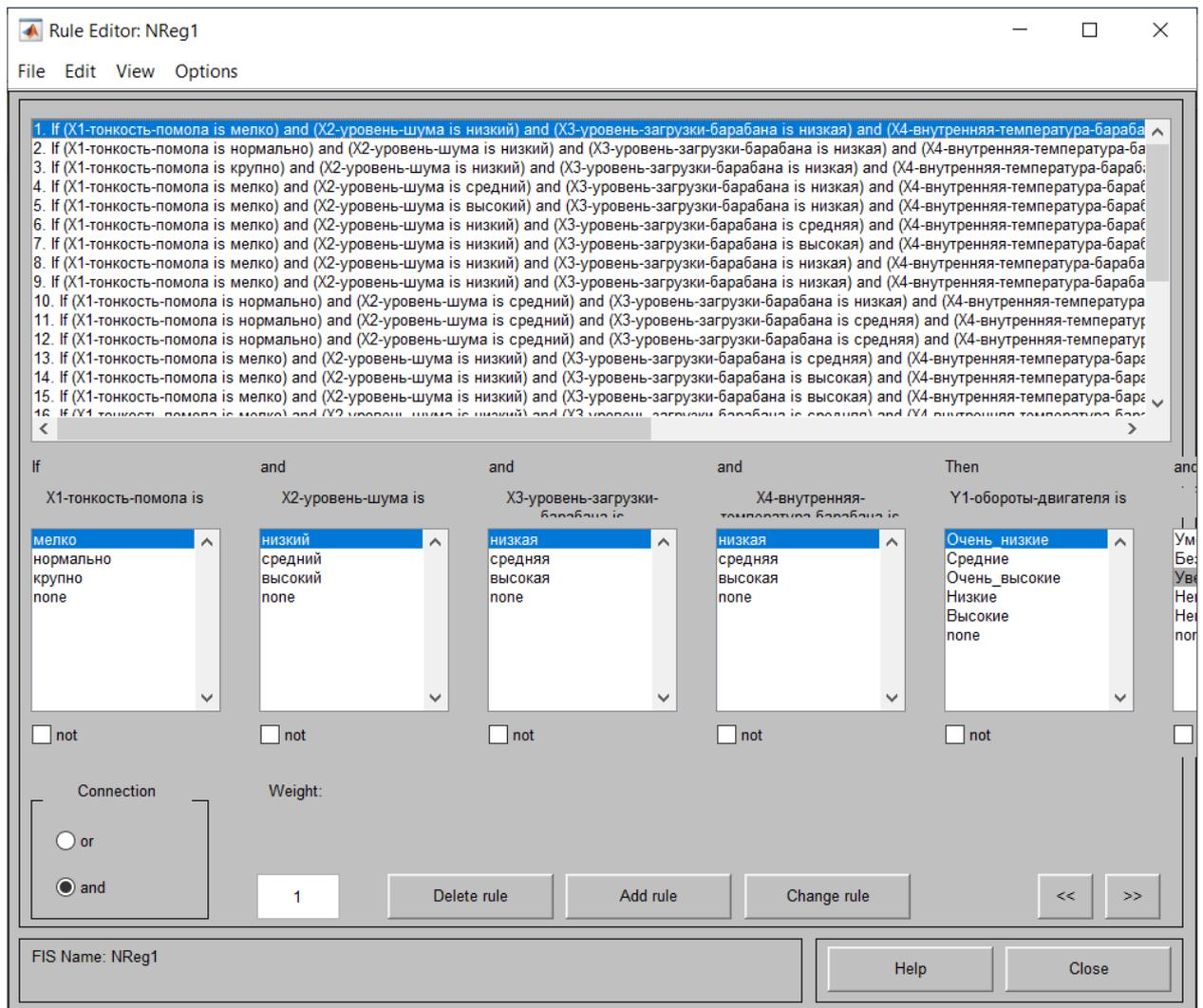


Рисунок 2.18 – Редактор нечетких правил в среде MATLAB

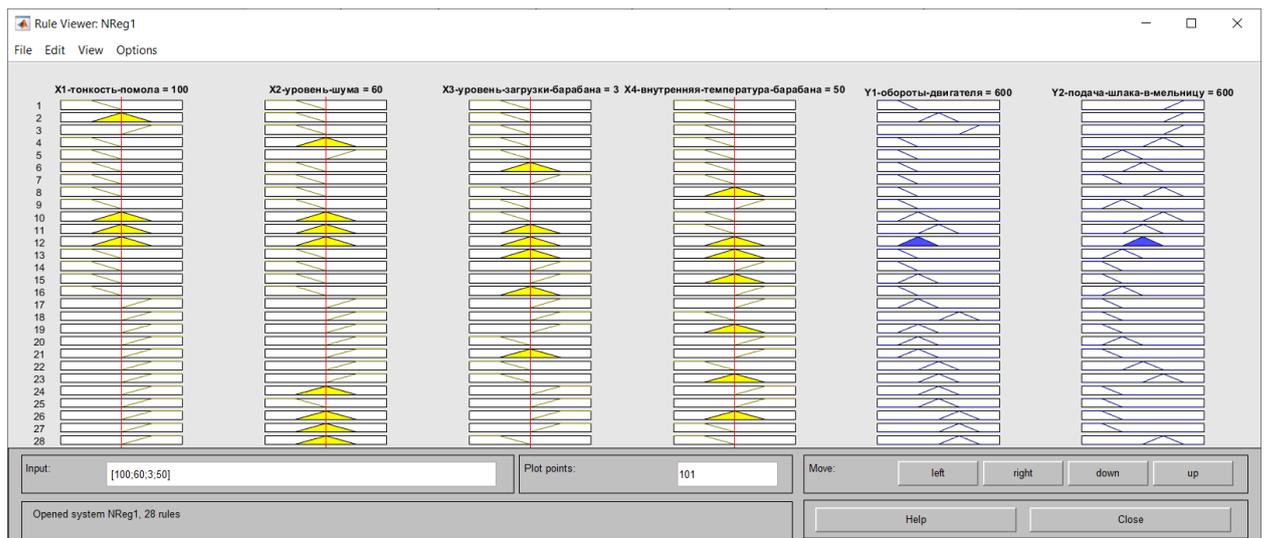


Рисунок 2.19 – Просмотр правил и имитационное моделирование различных режимов технологического процесса в среде MATLAB



- нечёткий регулятор обеспечивает большую стабильность, чем традиционный регулятор, и эффективнее контролирует нелинейный процесс;
- сложность построения систем на основе нечеткого регулирования значительно ниже, по сравнению с традиционными, что приводит к высокой надежности при промышленном использовании;
- нечеткий логический регулятор менее чувствителен к внешним возмущениям, по сравнению с классическими.

### 3 ПРОГРАММНО-АППАРАТНАЯ РЕАЛИЗАЦИЯ УПРАВЛЕНИЯ НА ОСНОВЕ НЕЧЁТКОЙ ЛОГИКИ

#### Структурная схема микропроцессорного управления технологическим процессом тонкого помола

В данном разделе рассмотрены структурное решение для построения интеллектуального управления системы.

Все описанные ниже блоки реализованы посредством микроконтроллера на базе Arduino MEGA 2560.

Структурная схема, реализующая управление технологическим процессом измельчения материалов в турбулентном измельчителе частиц интеллектуальной системы, представлена на рисунке 3.1.

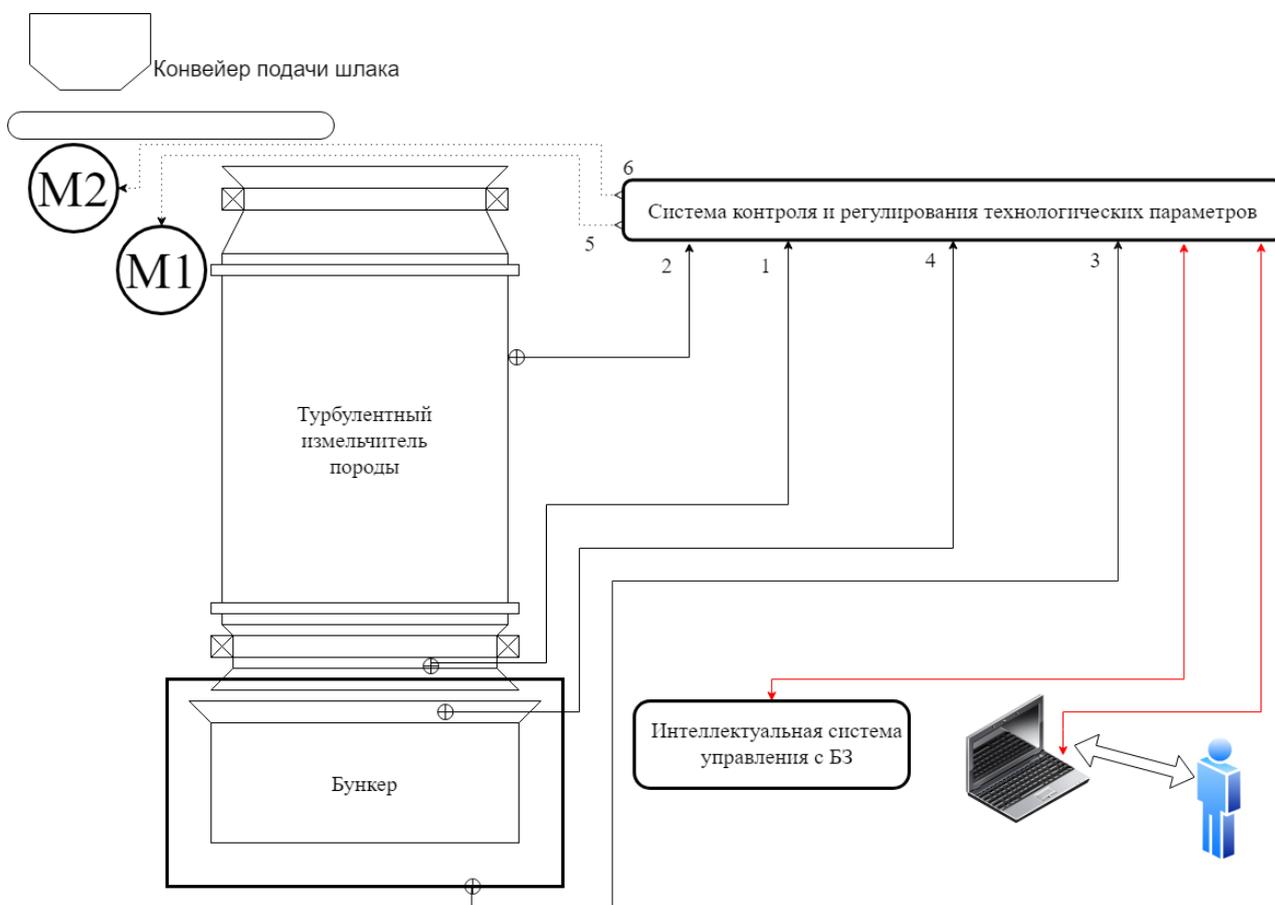


Рисунок 3.1 – Структурная схема управления технологическим процессом измельчения в турбулентном измельчителе частиц интеллектуальной системы

Назначение системы – повышение эффективности технологического процесса извлечения ценных компонентов. Основой управляющего алгоритма интеллектуальной системы является база знаний, для построения и проверки адекватности которой применены знания и опыт операторов-технологов, а также экспертов ТОО «КМЗ им. Пархоменко», г. Караганда, Казахстан и научно-

производственная фирма «Техника и Технология Дезинтеграции», г. Санкт-Петербург, РФ.

На рисунке 3.2 приведена общая конструкция микроконтроллера.

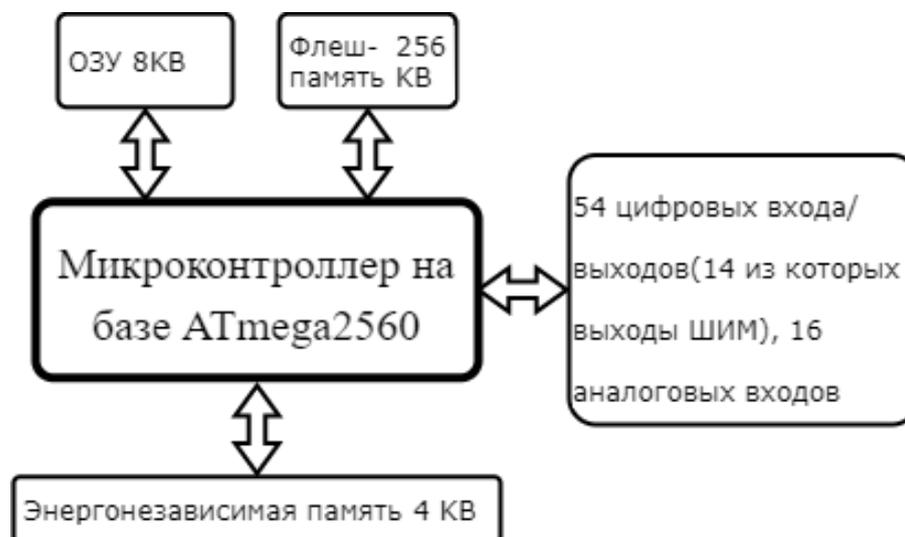


Рисунок 3.2 – Общая конструкция микроконтроллера

Реализация вывода данных на монитор ПК организована через шнур USB (виртуальный СОМ-порт), а на панель оператора – через выходы Arduino, называемые I2C. Используются два дисплея, один для входных данных, один для выходных. Для отображения данных на мониторе ПК было написано отдельное приложение.

### **Программный комплекс управления технологическим процессом тонкого помола**

Для написания приложения, отображающего данные на мониторе ПК, была использована среда разработки Embarcadero Rad Studio 2010. Работа с СОМ-портом осуществлялась посредством библиотеки TComPort.

Для решения задачи интеллектуальности было принято решение использовать базу правил нечеткой логики.

На рисунке 3.3 приведена схема алгоритма реализации ввода вывода ключевой НЛ.

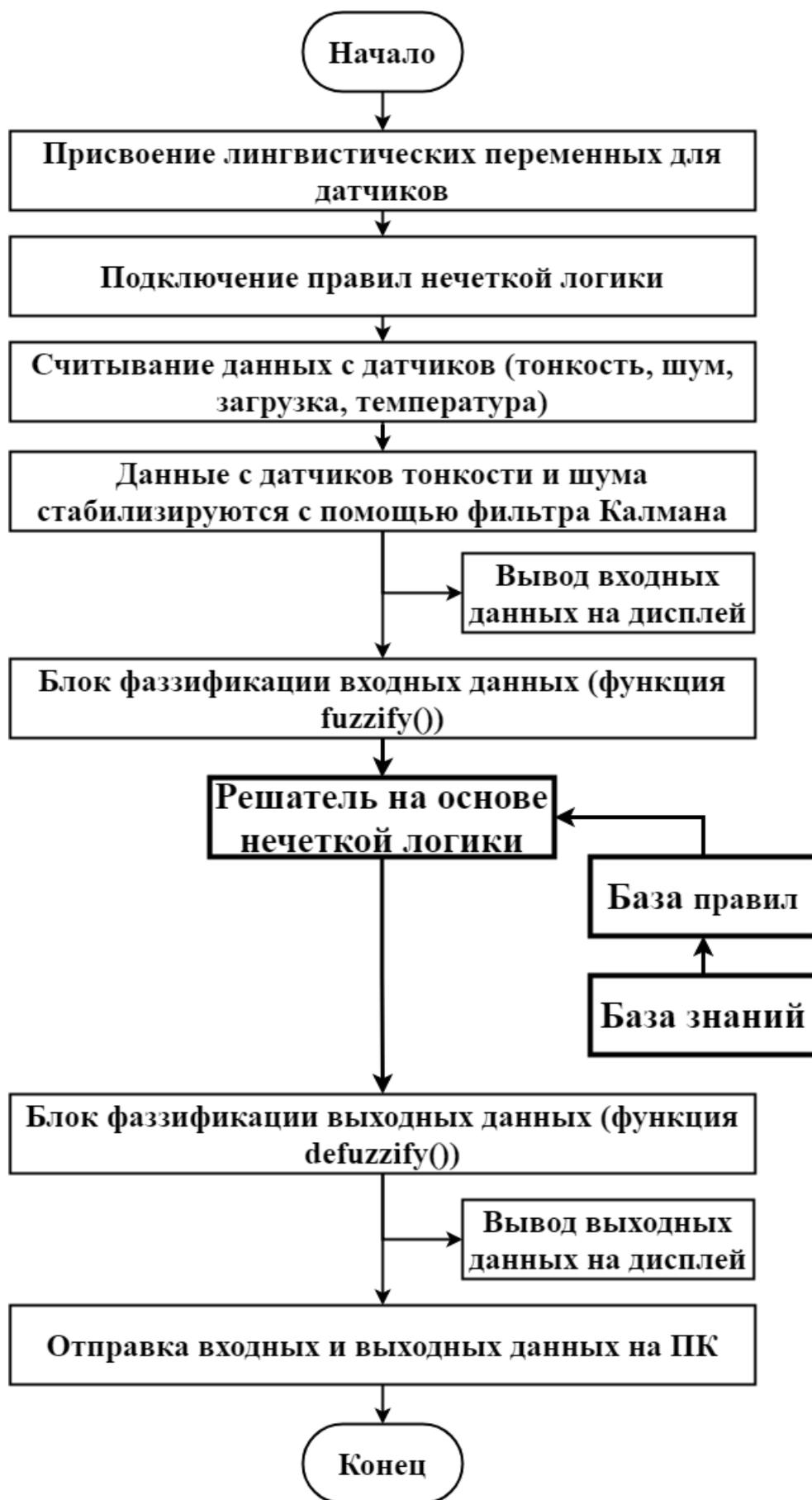


Рисунок 3.3 – Схема алгоритма реализации ввода вывода ключевой НЛ

База правил прописана в самом микроконтроллере, принципиальная схема указана на рисунке 3.4, где

входные данные X1 – тонкость помола, X2 – шум в турбулентный измельчитель, X3 – загрузка барабана, X4 – температура барабана;

выходные данные Y1 – обороты двигателя, Y2 – подача шлака.

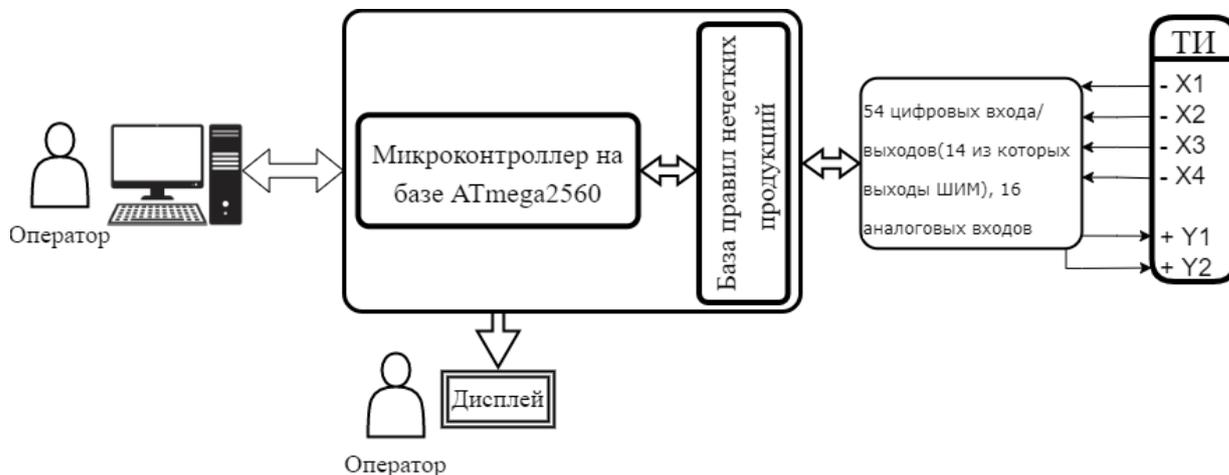


Рисунок 3.4 – Принцип работы программно-аппаратного комплекса

На микроконтроллере организован вывод всей необходимой информации как на панель оператора, так и передача этой же информации на ПК оператора для отображения в дополнительном программном обеспечении. Передача информации выполняется по USB проводу. Подключение в операционной системе Window организован в виде виртуального COM-порта. Скорость передачи данных – 9600 кбит/с. Окно программного обеспечения представлено на рисунке 3.5. При отсутствии соединения с оборудованием программа выдает соответствующее сообщение.

Передача данных на ПК оператора реализовано по следующему принципу. В COM-порт отсылается строка в следующем формате:

- Символ «В» - обозначающий начало строки передачи данных;
- Данные по тонкости помола;
- Символ «&» - разделитель входных данных;
- Данные по шуму;
- Символ «&» - разделитель входных данных;
- Данные по загрузке;
- Символ «&» - разделитель входных данных;
- Данные по температуре;
- Символ «S» - разделитель входных и выходных данных;
- Данные по обороту двигателя;
- Символ «!» - разделитель выходных данных;
- Данные по подаче шлака;
- Символ «Е» - обозначающий конец строки передачи данных.

Например, B95.68&102.33&1.55&48.56S750!300E

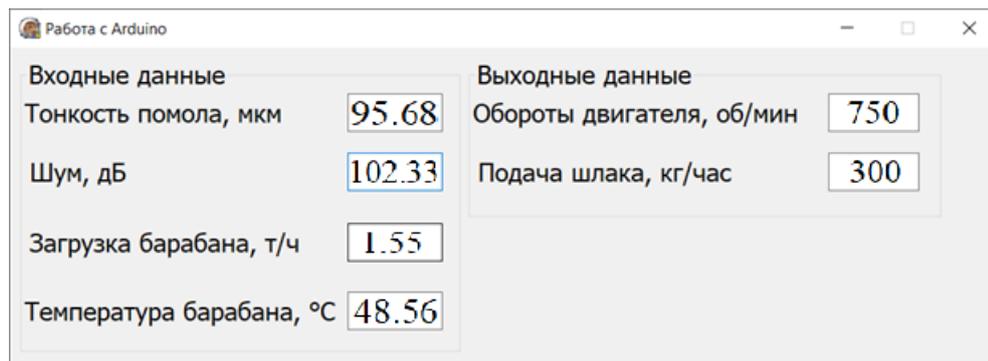


Рисунок 3.5 – Программное обеспечение в системе Windows.

Алгоритм работы системы осуществляется следующим образом: полученные по информационным каналам данные с 1, 2, 3, 4 измерительных каналов, которые отражают текущее состояние тонкости помола, акустический шум и объемную загрузку барабана, а также температуру шлама, поступают на вход системы. Система кодирует входные параметры и рассчитывает скорость оборотов двигателя М1 и загрузки М2 в турбулентный измельчитель породы.

Данные обрабатываются согласно правилами, после чего определяются выходные данные, т.е. обороты двигателя и скорость подачи шлама. Скорость подачи шлама тоже регулируется оборотами маломощного электродвигателя. У этих электродвигателей есть свои регуляторы оборотов, к которым мы можем подключиться напрямую и управлять ими, нужно только подавать ШИМ-сигнал (широтно-импульсная модуляция). А как нам известно на нашем микроконтроллере 14 из 54 цифровых входов/выходов могут быть реализованы как широтно-импульсная модуляция – процесс управления мощностью методом пульсирующего включения и выключения потребителя энергии (рисунок 3.2).

Все расчеты система выводит на графический интерфейс оператора-технолога.

Исполнение выработанных системой воздействий производится исполнительными механизмами (5 – оборот двигателя турбулентного измельчителя и 6 – подача шлама).

С целью сглаживания существенных флуктуаций входных параметров в системе использован фильтр Калмана.

Arduino – это плата микроконтроллера с открытым исходным кодом программного и аппаратного обеспечения. На рынке существует несколько различных плат Arduino. Однако эти разные платы Arduino имеют общую особенность – легкость разработки программ для них. Для данной работы Arduino является хорошим аппаратным обеспечением, так как он может работать с большим количеством датчиков, как аналоговых так и цифровых. На выходе можно управлять ШИМ – сигналом, что дает возможность регулировать обороты двигателя. Arduino MEGA 2560 показан на рисунке 3.6.

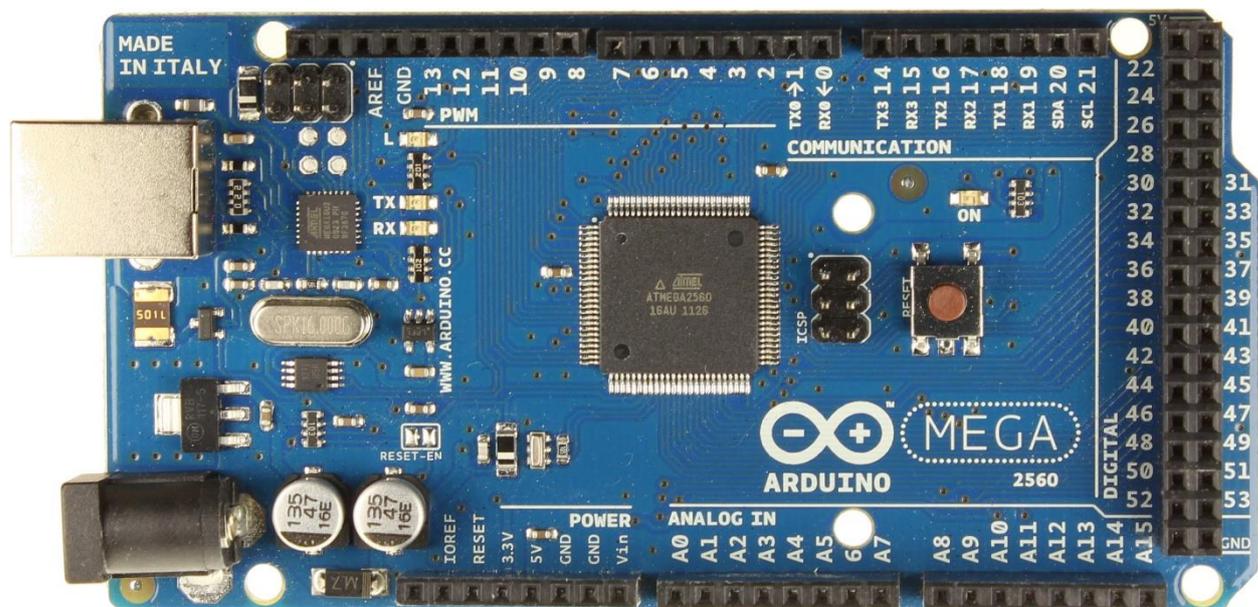


Рисунок 3.6 – Плата Arduino MEGA 2560

ШИМ-контакты – это контакты в верхнем ряду, отмеченные как PWM, а аналоговые контакты – это контакты в нижнем ряду от A0 до A15.

В данной работе интегрированная среда разработки Arduino (IDE) используется для написания программного кода под Arduino. IDE является производным от программ с открытым исходным кодом, таких как Processing, и написана на Java. В среде Java есть текстовый редактор, в котором можно писать свой код. Эта среда проста в использовании, для создания исполняемой программы ей нужны две функции: первая – `setup()`, где инициализируются переменные и запускается только один раз при пуске программы, а вторая – `loop()`, которая выполняет ее содержимое до тех пор, пока плата Arduino не будет отключена. Arduino подключается к компьютеру через USB, и через USB эскизы загружаются в Arduino. Код написан на C или C++. Стандартные конструкции, поддерживаемые компилятором C/C++, работают на Arduino. IDE также содержит компилятор, который используется перед загрузкой кода в Arduino. Если в коде что-то не так, появится сообщение об ошибке с объяснением ошибки. При последовательной связи через USB можно отправлять команды во время работы программы. Существует окно под названием “Serial monitor”, где пользователь может отображать результаты и отправлять команды на Arduino.

### 3.3 Аппаратный комплекс управления технологическим процессом тонкого помола

При написании программы есть возможность включить библиотеки с открытым исходным кодом от сторонних разработчиков, и есть множество библиотек, доступных для скачивания на сайте [www.Arduino.cc](http://www.Arduino.cc). Среди которых есть библиотека eFLL, которая предназначена для работы с нечеткой логикой.

Для реализации нечеткой логики с использованием базы правил была применена плата Arduino Mega 2560, так как для работы с большими цифрами типа int и float ресурсов других плат семейства Arduino недостаточно.

На рисунке 3.7 показан фрагмент подключения библиотеки для работы с нечеткой логикой, строка:

```
#include <Fuzzy.h>.
```

Далее определяются все диапазоны, строки:

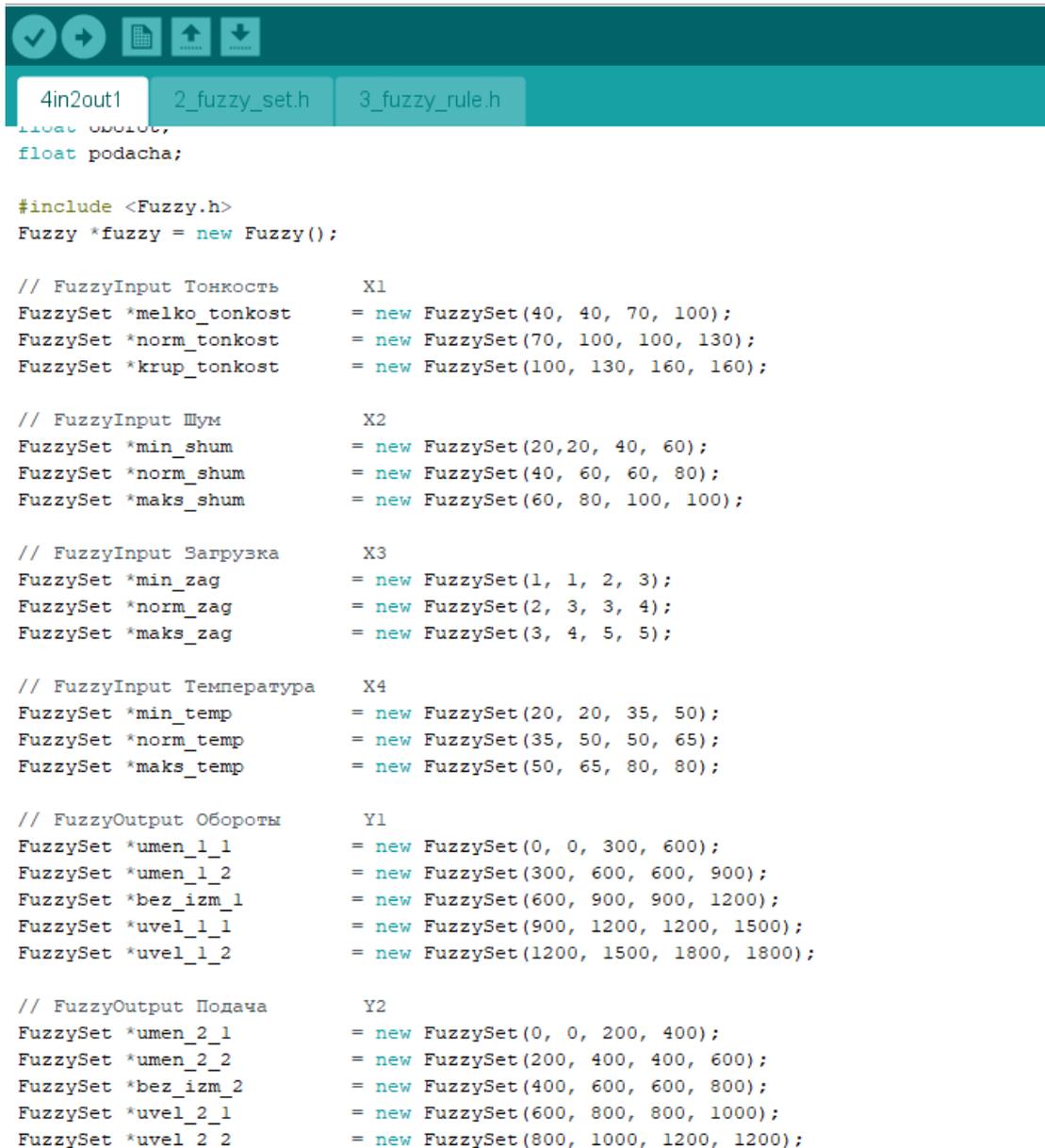
```
FuzzySet *melko_tonkost = new FuzzySet(40, 40, 70, 100).
```

...

```
FuzzySet *uvel_2_2 = new FuzzySet(800, 1000, 1200, 1200).
```

4in2out1 | Arduino 1.8.8

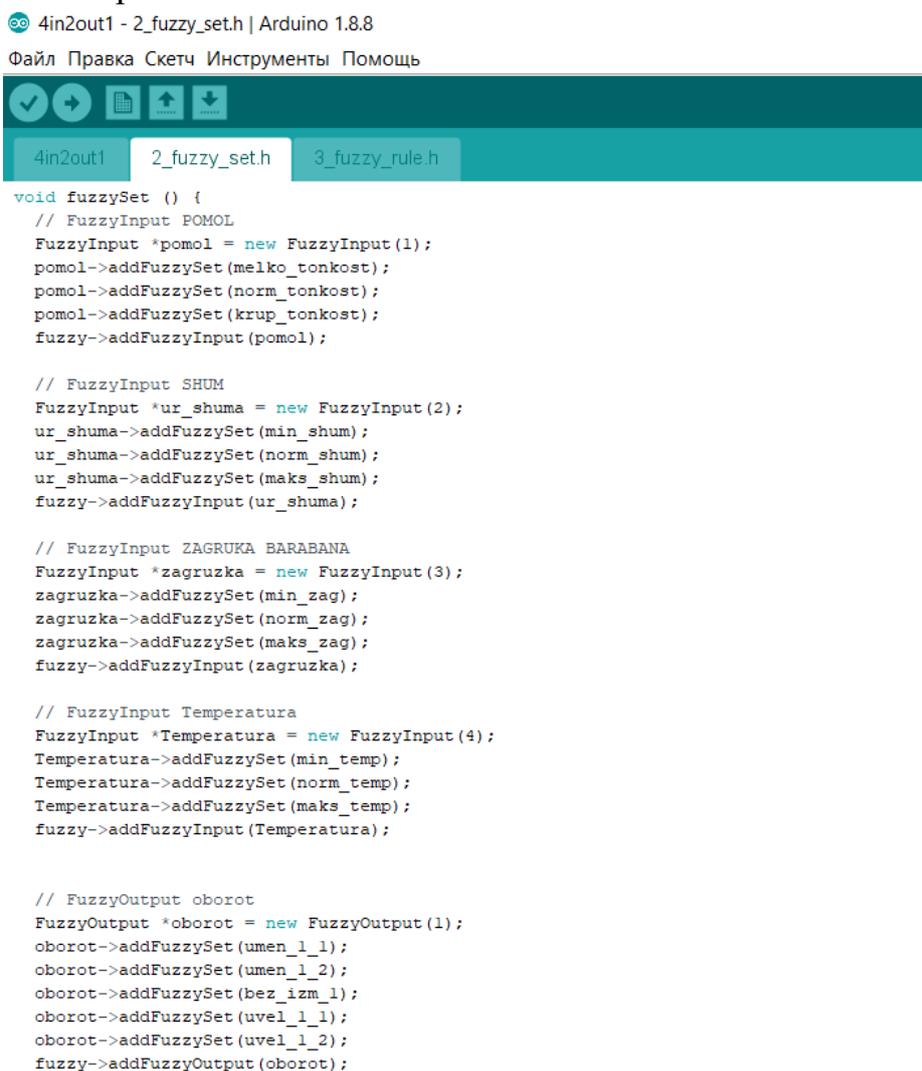
Файл Правка Скетч Инструменты Помощь



```
float obrot0;  
float podacha;  
  
#include <Fuzzy.h>  
Fuzzy *fuzzy = new Fuzzy();  
  
// FuzzyInput Тонкость X1  
FuzzySet *melko_tonkost = new FuzzySet(40, 40, 70, 100);  
FuzzySet *norm_tonkost = new FuzzySet(70, 100, 100, 130);  
FuzzySet *krup_tonkost = new FuzzySet(100, 130, 160, 160);  
  
// FuzzyInput Шум X2  
FuzzySet *min_shum = new FuzzySet(20, 20, 40, 60);  
FuzzySet *norm_shum = new FuzzySet(40, 60, 60, 80);  
FuzzySet *maks_shum = new FuzzySet(60, 80, 100, 100);  
  
// FuzzyInput Загрузка X3  
FuzzySet *min_zag = new FuzzySet(1, 1, 2, 3);  
FuzzySet *norm_zag = new FuzzySet(2, 3, 3, 4);  
FuzzySet *maks_zag = new FuzzySet(3, 4, 5, 5);  
  
// FuzzyInput Температура X4  
FuzzySet *min_temp = new FuzzySet(20, 20, 35, 50);  
FuzzySet *norm_temp = new FuzzySet(35, 50, 50, 65);  
FuzzySet *maks_temp = new FuzzySet(50, 65, 80, 80);  
  
// FuzzyOutput Обороты Y1  
FuzzySet *umen_1_1 = new FuzzySet(0, 0, 300, 600);  
FuzzySet *umen_1_2 = new FuzzySet(300, 600, 600, 900);  
FuzzySet *bez_izm_1 = new FuzzySet(600, 900, 900, 1200);  
FuzzySet *uvel_1_1 = new FuzzySet(900, 1200, 1200, 1500);  
FuzzySet *uvel_1_2 = new FuzzySet(1200, 1500, 1800, 1800);  
  
// FuzzyOutput Подача Y2  
FuzzySet *umen_2_1 = new FuzzySet(0, 0, 200, 400);  
FuzzySet *umen_2_2 = new FuzzySet(200, 400, 400, 600);  
FuzzySet *bez_izm_2 = new FuzzySet(400, 600, 600, 800);  
FuzzySet *uvel_2_1 = new FuzzySet(600, 800, 800, 1000);  
FuzzySet *uvel_2_2 = new FuzzySet(800, 1000, 1200, 1200);
```

Рисунок 3.7 – Объявление лингвистических переменных в среде Arduino IDE

На рисунке 3.8 показано добавление лингвистических переменных для использования в правилах.



```
4in2out1 - 2_fuzzy_set.h | Arduino 1.8.8
Файл Правка Скetch Инструменты Помощь
4in2out1 2_fuzzy_set.h 3_fuzzy_rule.h
void fuzzySet () {
  // FuzzyInput POMOL
  FuzzyInput *pomol = new FuzzyInput(1);
  pomol->addFuzzySet(melko_tonkost);
  pomol->addFuzzySet(norm_tonkost);
  pomol->addFuzzySet(krup_tonkost);
  fuzzy->addFuzzyInput(pomol);

  // FuzzyInput SHUM
  FuzzyInput *ur_shuma = new FuzzyInput(2);
  ur_shuma->addFuzzySet(min_shum);
  ur_shuma->addFuzzySet(norm_shum);
  ur_shuma->addFuzzySet(maks_shum);
  fuzzy->addFuzzyInput(ur_shuma);

  // FuzzyInput ZAGRUKA BARABANA
  FuzzyInput *zagruzka = new FuzzyInput(3);
  zagruzka->addFuzzySet(min_zag);
  zagruzka->addFuzzySet(norm_zag);
  zagruzka->addFuzzySet(maks_zag);
  fuzzy->addFuzzyInput(zagruzka);

  // FuzzyInput Temperatura
  FuzzyInput *Temperatura = new FuzzyInput(4);
  Temperatura->addFuzzySet(min_temp);
  Temperatura->addFuzzySet(norm_temp);
  Temperatura->addFuzzySet(maks_temp);
  fuzzy->addFuzzyInput(Temperatura);

  // FuzzyOutput oborot
  FuzzyOutput *oborot = new FuzzyOutput(1);
  oborot->addFuzzySet(umen_1_1);
  oborot->addFuzzySet(umen_1_2);
  oborot->addFuzzySet(bez_izm_1);
  oborot->addFuzzySet(uvel_1_1);
  oborot->addFuzzySet(uvel_1_2);
  fuzzy->addFuzzyOutput(oborot);
}
```

Рисунок 3.8 – Создание экземпляра объекта нечеткого множества в среде Arduino IDE

На рисунке 3.9 представлены имеющиеся правила, они идентичны правилам в среде MATLAB.

```

void fuzzyRule () {
  // Building FuzzyRule//////////////////////////////////// 1
  FuzzyRuleAntecedent *Pr1_1_1 = new FuzzyRuleAntecedent();
  Pr1_1_1->joinWithAND(melko_tonkost, min_shum);

  FuzzyRuleAntecedent *Pr2_1_1 = new FuzzyRuleAntecedent();
  Pr2_1_1->joinWithAND(min_zag, min_temp);

  FuzzyRuleAntecedent *Pravilo_1 = new FuzzyRuleAntecedent();
  Pravilo_1->joinWithAND(Pr1_1_1, Pr2_1_1);

  FuzzyRuleConsequent *P_1 = new FuzzyRuleConsequent();
  P_1->addOutput(umen_1_1);
  P_1->addOutput(uvel_2_2);

  FuzzyRule *fuzzyRule1 = new FuzzyRule(1, Pravilo_1, P_1);
  fuzzy->addFuzzyRule(fuzzyRule1);

  // Building FuzzyRule//////////////////////////////////// 2
  FuzzyRuleAntecedent *Pr1_2_1 = new FuzzyRuleAntecedent();
  Pr1_2_1->joinWithAND(norm_tonkost, min_shum);

  FuzzyRuleAntecedent *Pravilo_2 = new FuzzyRuleAntecedent();
  Pravilo_2->joinWithAND(Pr1_2_1, Pr2_1_1);

  FuzzyRuleConsequent *P_2 = new FuzzyRuleConsequent();
  P_2->addOutput(bez_izm_1);
  P_2->addOutput(uvel_2_2);

  FuzzyRule *fuzzyRule2 = new FuzzyRule(2, Pravilo_2, P_2);
  fuzzy->addFuzzyRule(fuzzyRule2);

  // Building FuzzyRule//////////////////////////////////// 3
  FuzzyRuleAntecedent *Pr1_3_1 = new FuzzyRuleAntecedent();
  Pr1_3_1->joinWithAND(krup_tonkost, min_shum);

  FuzzyRuleAntecedent *Pravilo_3 = new FuzzyRuleAntecedent();
  Pravilo_3->joinWithAND(Pr1_3_1, Pr2_1_1);

  FuzzyRuleConsequent *P_3 = new FuzzyRuleConsequent();
  P_3->addOutput(uvel_1_2);

```

Рисунок 3.9 – Код программы демонстрирующий набор правил в среде Arduino IDE

На рисунке 3.10 представлен прототип программно-аппаратного комплекса интеллектуального управления, который выводит результаты генерации входных данных. Входные данные имитируются с помощью потенциометров. Код программы (Приложение В).

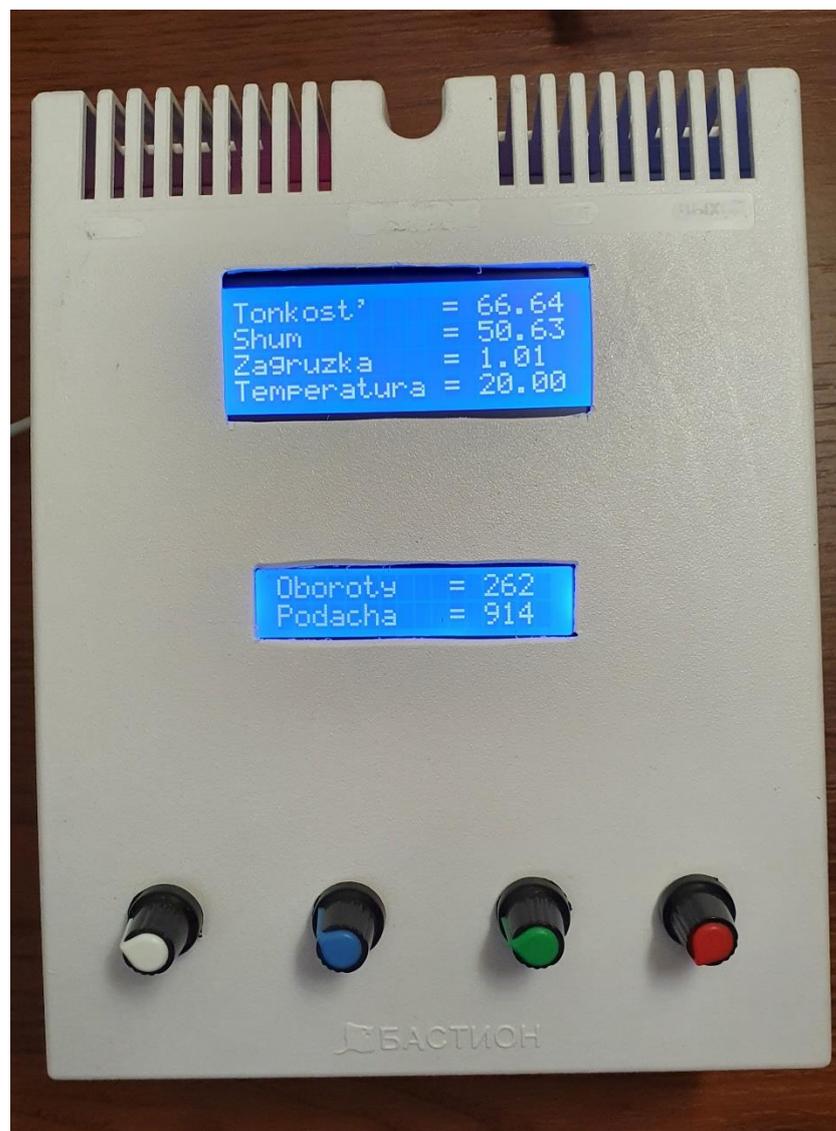


Рисунок 3.10 – Прототип программно-аппаратного комплекса интеллектуального управления

На рисунке 3.11 представлен лабораторный стенд программно-аппаратного комплекса интеллектуального управления при исследовании режимов работы турбулентного измельчителя.



Рисунок 3.11 – Лабораторный стенд

Ниже приведен перечень оборудования, использованного для экспериментальных исследований и его основные характеристики.

Виброакустический анализатор загрузки мельницы «ВАЗМ-1М» (рисунок 3.12) предназначен для своевременного и быстрого контроля технологической нагрузки, а также прогнозирует момент перегрузки в измельчителях, данные на ПК передавались по USB-каналу. Основные характеристики анализатора представлены в таблице 3.1.



Рисунок 3.12 – Виброакустический анализатор загрузки мельницы «ВАЗМ-1М»

Таблица 3.1 – Основные характеристики

Количество виброакустических каналов измерения	2
Частотный диапазон измерения, Гц	20 - 10000
Амплитудный диапазон измерения, мВ	5 - 500
Диапазон выходных токовых сигналов, мА	4 - 20
Интерфейс связи	RS-485, Ethernet
Электропитание и потребляемая мощность	~220В/50Гц/100ВА
Габаритные размеры и масса	500×500×210мм/25кг
Конструктивное исполнение	IP65 по ГОСТ 14254

Гранулометр «ПИК-074П» (рисунок 3.13) предназначен для непрерывного автоматического контроля гранулометрического состава пульпообразных и сухих продуктов процессов измельчения и классификации, транспортируемых по трубопроводам в пульподелителях и емкостях, без отбора проб, непосредственно в потоке и обеспечивает непрерывное определение и отображение на дисплее величины процентного содержания контрольных классов крупности частиц в пульпе, выдачу стандартного сигнала на регулирующие устройства и передачу данных по каналам связи. Основные характеристики гранулометра представлены в таблице 3.2.



Рисунок 3.13 – Гранулометр «ПИК-074П»

Гранулометр «ПИК-074П» обеспечивает:

- возможность просмотра на дисплее измеренных и вычисленных значений;
- выдачу стандартного сигнала на регулирующие устройства;
- передачу данных по каналам связи;
- установку режимов работы, выбор диапазонов, ввод коэффициентов калибровки и настройки при помощи цифровой клавиатуры;
- на показания измерений не влияют пузырьки воздуха, включения магнитных фракций, изменения плотности пульпы и температуры.

Таблица 3.2 – Основные характеристики

Диапазон измерения крупности частиц, мм	от 0,028 до 1,5
Количество контролируемых классов крупности	до 2
Среднеквадратичная погрешность контролируемой величины, %	не более 2
Давление, атм	до 1
Температура, С <sup>0</sup>	от +10 до +100
Выходной сигнал, мА	0-5; 0-20; 4-20
Интерфейс связи с ЭВМ	RS-485
Питание от сети переменного тока частотой 50 Гц:	
Напряжение, В	220
Потребляемая мощность, ВА	100
Степень защиты	IP-65
Габаритные размеры, мм	
Шкаф с микропроцессорным преобразователем МИП-3	500x400x210
Датчик крупности ДКП-05 (для штанги длиной 160 мм)	372x300x200
Масса, кг	
Шкаф с микропроцессорным преобразователем МИП-3, не более	17
Датчик крупности ДКП-05, не более	12

Структура комплекса представлена на рисунке 3.14.

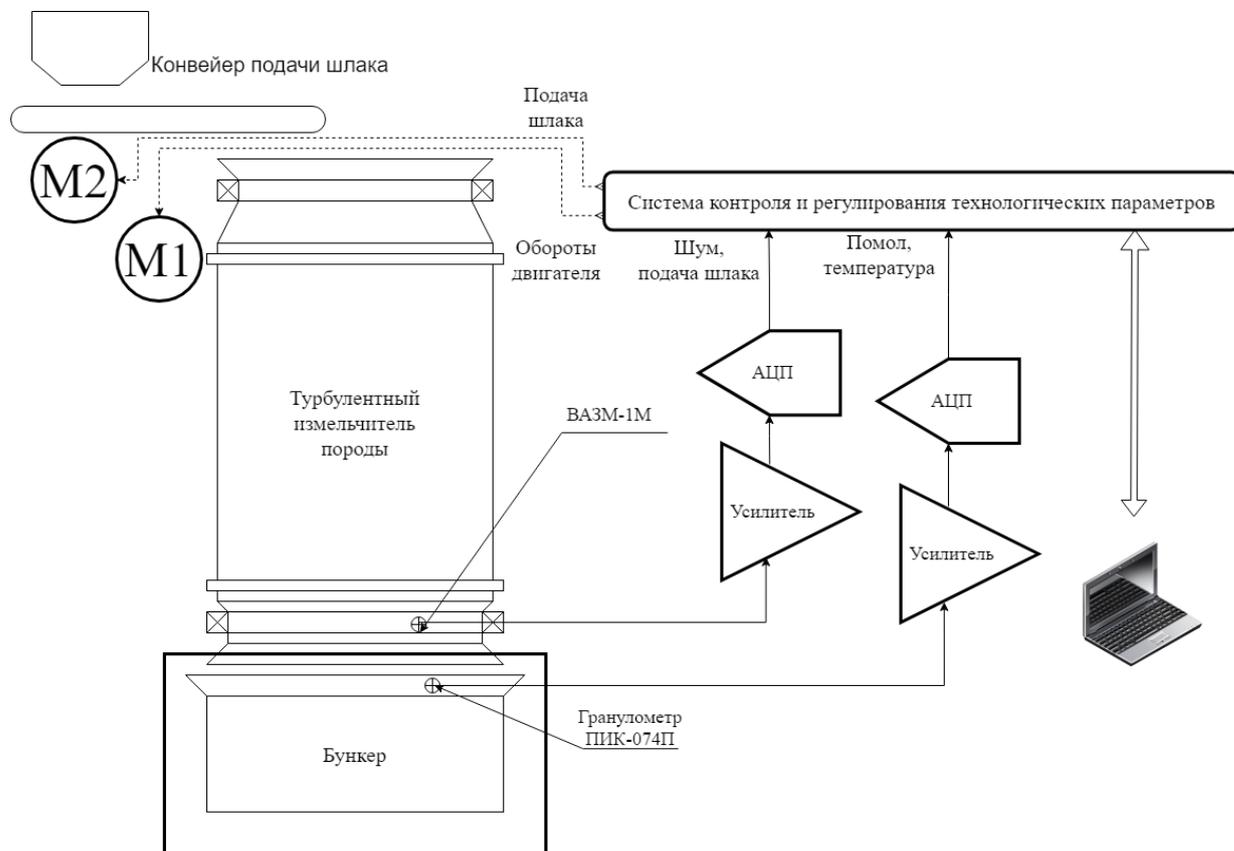


Рисунок 3.14 – Структура комплекса

Ниже, на рисунке 3.15, приведено сравнение результатов нечёткого регулятора в среде MATLAB и Arduino. В среде MATLAB в поле Input идентичные входные данные, как в микроконтроллере Arduino Mega. В соответствии с рисунком 3.15, в последних двух столбцах указаны выходные данные (обороты двигателя и подача шлама в мельницу).

Сопоставление полученных результатов свидетельствует о достаточно высоком качестве построенной имитационной модели и микроконтроллера. Отклонение в результатах составляет менее 1%, что является допустимым для инженерных расчетов.

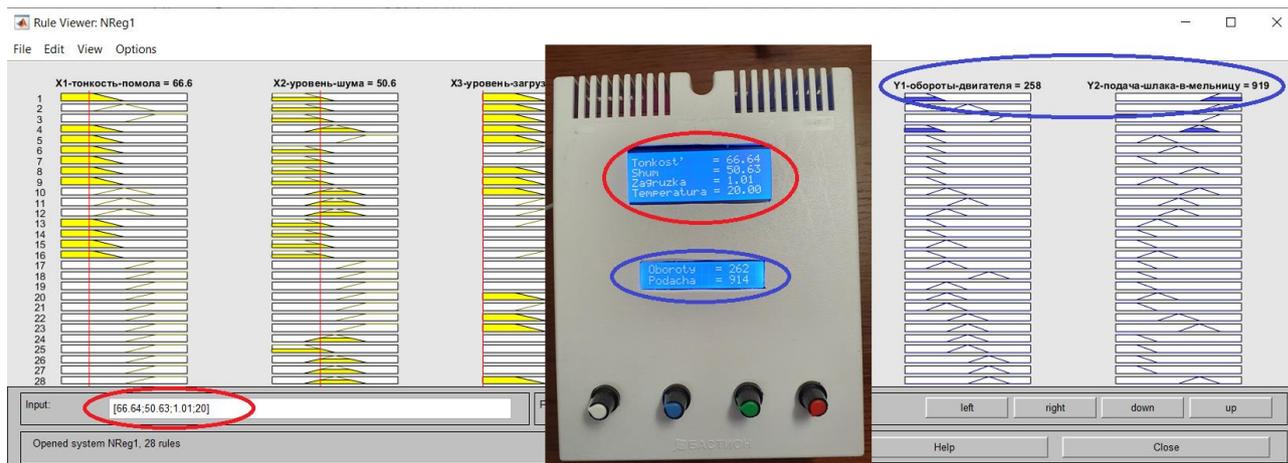


Рисунок 3.15 – Сравнение результатов нечёткого регулятора в среде MATLAB и Arduino

В качестве эффективности интеллектуальной подсистемы можно привести следующие показатели:

- для создания интеллектуального управления для турбулентного измельчителя породы на основе нечёткой логики нет необходимости проведения экспериментов на реальном технологическом объекте, связанных с построением адекватной математической модели процесса измельчения на основе дифференциальных уравнений. Это позволяет избежать ряд издержек, связанных с изменением режимов работы реального технологического комплекса;

- на основании полученных результатов можно утверждать, что синтезированный способ управления по сравнению с традиционным (ручным) способом управления технологическим процессом тонкого помола позволяет получить существенную экономию по энергозатратам, ориентировочно 10%, а также повысить эффективность технологического процесса извлечения ценных компонентов;

- при стоимости электроэнергии, по состоянию на 1 января 2021 года равной 22,14 за 1кВтч, годовая экономия электроэнергии составляет 250 тыс. тг;

- стоимость полученного полиметаллического концентрата по предлагаемой технологии ориентировочно будет составлять 200\$-250\$ с одной тонны сырья. Стоимость сырья при его покупке составляет в пределах 80\$-120\$

за одну тонну, при переработке на собственных производствах стоимость сырья нулевая;

– срок окупаемости организации производства по прямому извлечению металла из техногенного сырья при производительности турбулентного измельчителя в 5 тонн в час составит не более 12 месяцев.

Стоит обратить внимание, что при тонком измельчении затраты электроэнергии растут прямо пропорционально тонине продукта (таблица 3.3).

Таблица 3.3 – Сводные экономические показатели эффективности внедрения интеллектуальной системы

Наименование показателей	До внедрения	После внедрения
Затраты электроэнергии на измельчение 1т породы, кВт/ч	4,4	3,74
Экономия электроэнергии на 1 т. породы, кВтч	-	0,66
Годовая экономия в денежном выражении, тг	-	250602

### **Выводы по третьему разделу**

– с помощью способа управления технологическими процессами на основе нечёткой логики появляется возможность анализировать и проектировать системы управления реальными объектами, с быстро изменяющимися, сложными динамическими процессами, учитывать неполную и неоднозначную информации о процессе;

– на основании знаний и опыта операторов-технологов, выступающих в качестве экспертов, формализовано и представлено в виде лингвистических переменных факторное пространство управления технологическим процессом тонкого помола в турбулентном измельчителе породы;

– сравнительный анализ результатов нечёткого регулятора в MATLAB и Arduino показал, что сложность алгоритма систем управления на основе нечётких контроллеров значительно ниже, по сравнению с классическим системами. На реальных объектах это приводит систему к «интеллектуальности» и обеспечивать высокую точность при простой и недорогой аппаратной реализации;

– результаты компьютерного моделирования и экспериментальные исследования на лабораторной установке показали высокую эффективность разработанных алгоритмов нечеткого управления.

## ЗАКЛЮЧЕНИЕ

Для повышения эффективности технологического процесса извлечения ценных компонентов необходимо использовать системы управления, которые обладают возможностью работать с быстро изменяющимися, сложными динамическими процессами, учитывать неполную и неоднозначную информации о процессе.

Управление должно обеспечивать высокую точность при простой и недорогой аппаратной реализации.

При реализации традиционных методов возникают сложности измерения ряда параметров и показателей технологические процесс помола для турбулентного измельчителя не описывается количественно, что затрудняет использование детерминированных математических методов для моделирования и оптимизации их режимов работы. Для построения модели на основе дифференциальных уравнений динамического процесса измельчения, учитывающий все физико-механические изменения, необходимо провести ряд сложных и дорогостоящих экспериментов. При этом полученная математическая модель носит весьма приближенный характер и может отображать лишь некоторые тенденции технологического объекта.

Турбулентный измельчитель породы является нелинейной, многопараметрической системой, в которой связь между наблюдаемыми параметрами и состоянием системы имеет сложный и неоднозначный характер. Нечеткая логика показала очень эффективное решение слабо детерминированных нелинейных и математически сложно описываемых задач. Нечеткое регулирование позволяет повысить адекватность процессов управления и учесть множество факторов, влияющих на процессы принятия решений для управления турбулентным измельчителем.

С помощью способа управления технологическими процессами на основе нечёткой логики появляется возможность анализировать и проектировать системы управления реальными объектами, с быстро изменяющимися, сложными динамическими процессами, учитывать неполную и неоднозначную информации о процессе. На основании знаний и опыта операторов-технологов, выступающих в качестве экспертов, формализовано и представлено в виде лингвистических переменных факторное пространство управления технологическим процессом тонкого помола в турбулентном измельчителе породы.

Проведен сравнительный анализ результатов нечёткого регулятора в MATLAB и Arduino, который показал, что сложность алгоритма систем управления на основе нечётких контроллеров значительно ниже, по сравнению с классическими системами. На реальных объектах это приводит систему к «интеллектуальности» и обеспечивать высокую точность при простой и недорогой аппаратной реализации. Результаты компьютерного моделирования и экспериментальные исследования на лабораторной установке показали высокую эффективность разработанных алгоритмов нечеткого управления

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Покидаев Д. Вторичное сырье: почему в Казахстане не подбирают деньги, лежащие под ногами // <https://kursiv.kz/news/kompanii-i-rynki/2018-05/vtorichnoe-syre-pochemu-vkazakhstan-ne-podbirayut-dengi>. 23.04.2018.
- 2 Коротаяев В.Н., Слюсарь Н.Н., Жилинская Я.А. и др. Управление техногенными отходами. – Пермь: ПНИПУ, 2016. – 390 с.
- 3 Газалиев А.М., Апачиди Н.К., Федорченко В.И. Разработка технологии и комплекса технических средств извлечения металла из техногенного сырья: отчет о НИР (заключительный). – Караганда: КарГТУ, 2015. – 70 с. – Инв. №0213РК00332.
- 4 Гусев А.И. Наноматериалы, наноструктуры, нанотехнологии. – М.: Физматлит, 2007. – 416 с.
- 5 Падохин В.А., Зуева Г.А. Стохастические модели измельчения дисперсных материалов // Теоретические основы химической технологии. – 2009. – №5. – С. 586-594.
- 6 Сиденко П.М. Измельчение в химической промышленности. – Изд. 2-е, перер. – М.: Химия, 1977. – 368 с.
- 7 Zhang Y.F., Lu L., Yap S.M. Prediction of the amount of PCA for mechanical milling // Journal of Materials Processing Technology. – 1999. – Vol. 89-90. – P. 260-265.
- 8 Рипан Р., Четяну И. Неорганическая химия. – М.: Химия, 2003. – 360 с.
- 9 Атанов С.К., Бигалиева А.З. Применение компьютерных технологий для расчёта степени помола сырья на примере планетарной мельницы // Тр. университета. – 2018. – №4. – С. 139-144.
- 10 Abdellaoui M., Gaffet E. A mathematical and experimental dynamical phase diagram for ball-milled Ni<sub>10</sub>Zr<sub>7</sub> // Journal of Alloys and Compounds. – 1994. – Vol. 209, Issue 1-2. – P. 351-361.
- 11 Abdellaoui M., Gaffet E. The physics of mechanical alloying in a planetary ball mill: mathematical treatment // Acta Metallurgica et Materialia. – 1995. – Vol. 43, Issue 3. – P. 1087-1098.
- 12 Davis R.M., McDermot B., Koch C.C. Mechanical Alloying of Brittle Materials // Metallurgical Transactions A. – 1988. – Vol. 19, Issue 12. – P. 2867-2874.
- 13 Хархута Н.Я., Капустин М.И., Семенов В.П., Эвентов И.М. Дорожные машины. – Изд. 2-е, доп. и перер. – Л.: Машиностроение, 1976. – 472 с.
- 14 Борщёв В.Я. Оборудование, для измельчения материалов: дробилки и мельницы. – Тамбов: ТГТУ, 2004. – 75 с.
- 15 Еренков О.Ю., Богачев А.П. Оборудование механических процессов в химической технологии. – Хабаровск: ТГУ, 2014. – 80 с.
- 16 Кузьмина В.П. Механоактивация материалов для строительства. Гипс // Строительные материалы. – 2007. – №9. – С. 52-54.
- 17 Дерягин Б.В., Чураев Н.В., Муллер В.М. Поверхностные силы. – М.: Наука, 1985. – 398 с.

- 18 Ребиндер П.А. Поверхностные явления в дисперсных системах. Коллоидная химия. Избранные труды. – М.: Наука, 1978. – 368 с.
- 19 Валеева А.А., Давыдов Д.А. Дезинтеграция крупнокристаллического порошка монооксида ванадия  $VO_y$  // Журнал технической физики. – 2009. – Т. 79, №11. – С. 57-59.
- 20 Душин С.Е., Зотов Н.С., Имаев Д.Х. и др. Теория автоматического управления. – Изд. 2-е, перер. – М.: Высшая школа, 2005. – 567 с.
- 21 Сибирцева Н.Б., Потапенко А.Н., Семернин А.Н. Особенности автоматизации загрузки сырьевой мельницы в составе АСДУ // Известия Самарского научного центра Российской академии наук. – 2011. – Т. 13, №1(3). – С. 641-645.
- 22 Лазарева О.В., Подкаменный Ю.А. Автоматизированный способ управления комплексом измельчения и классификации алмазосодержащих руд. // Вестник ИрГТУ. – 2014. – №4(87). – С. 128-132.
- 23 Uraikul V., Chan C.W., Tontiwachwuthikul P. Artificial intelligence for monitoring and supervisory control of process systems. // Engineering Applications of Artificial Intelligence. – 2007. – Vol. 20, Issue 2. – P. 115-131.
- 24 Liang S., Rajora M., Liu X., Zou P., Wang L. Intelligent manufacturing systems: A review/ // International Journal of Mechanical Engineering and Robotics Research. – 2018. – Vol. 7, Issue 3. – P. 324-330.
- 25 Саутин С.Н. Планирование эксперимента в химии и химической технологии. – Л.: Химия, 1975. – 48 с.
- 26 Королев Л.В., Бытев Д.О. Уравнение кинетики измельчения с произвольным законом распределения времени ожидания // Моделирование и анализ информационных систем. – 2012. – Т. 19, №2. – С. 53-61.
- 27 Пономарев В.Б. Математическое моделирование технологических процессов. – Екатеринбург: ГОУ ВПО УГТУ–УПИ, 2006. – 129 с.
- 28 Веремей Е.И. Среднеквадратичная многоцелевая оптимизация. – СПб.: СПбГУ, 2016. – 407 с.
- 29 Ogata K. Modern control engineering. – Ed. 4th. – New Jersey: Prentice Hall, 2010. – 964 p.
- 30 Анищенко В.С. Сложные колебания в простых системах. – Изд. 2-е. – М.: Наука, 2009. – 320 с.
- 31 Васильев К.К. Теория автоматического управления: следящие системы // [http://tk.ulstu.ru/lib/books/tau\\_1.pdf](http://tk.ulstu.ru/lib/books/tau_1.pdf). 15.09.2020.
- 32 Aström K.J., Murray R.M. Feedback Systems // <http://www.cds.caltech.edu/~murray/amwiki>. 10.09.2020.
- 33 Duriez T., Brunton S.L., Noack B.R. Machine Learning Control – Taming Nonlinear Dynamics and Turbulence. – Ed. 1th. – NY.: Publisher: Springer International Publishing, 2018. – 211 p.
- 34 Веремей Е.И. Линейные системы с обратной связью. – СПб.: Лань, 2013. – 448 с.
- 35 Anderson B.D.O., Moore J.B. Linear Optimal Control. – New Jersey: Prentice-Hall, 1971. – 413 p.

- 36 Bhattacharyya S.P., Chapellat H., Keel L.H. Robust control: the parametric approach. – New Jersey: Prentice Hall, 1995. – 672 p.
- 37 Bhattacharyya S.P., Datta A., Keel L.H. Linear control theory: structure, robustness and optimization. – Boca Raton: CRC Press, 2009. – 924 p.
- 38 Bosgra H., Kwakernaak H., Meinsma G. Design methods for control systems. – Delft, 2003. – 319 p.
- 39 Boyd S., Ghaoui E., Feron E., Balakrishnan V. Linear matrix inequalities in systems and control theory. – Philadelphia: Society for Industrial and Applied Mathematics, 1994. – 193 p.
- 40 Зубов В.И. Динамика управляемых систем. – Изд. 2-е, перер. и доп. – СПб.: СПбГУ, 2004. – 380 с.
- 41 Атанов С.К., Бигалиева А.З. Математическая модель процесса помола в планетарной мельнице // Тр. университета. – 2020. – №1. – С. 148-153.
- 42 Алексеев А.А. Идентификация и диагностика систем. – М.: Академия, 2009. – 351 с.
- 43 Александров А.Ю., Жабко А.П. Устойчивость разностных систем. – СПб.: НИИ химии СПбГУ, 2003. – 112 с.
- 44 Воронов А.А. Устойчивость, управляемость, наблюдаемость. – М.: Наука, 1979. – 336 с.
- 45 Воронов А.А. Введение в динамику сложных управляемых систем. – М.: Наука, 1985. – 304 с.
- 46 Летов А.М. Математическая теория процессов управления. – М.: Наука, 1981. – 127 с.
- 47 Дорф Р. Современные системы управления. – М.: Лаборатория Базовых знаний, 2002. – 832 с.
- 48 Андреев Ю.Н. Управление конечномерными линейными объектами. – М.: Наука, 1979. – 424 с.
- 49 Веремей Е.И. Алгоритмы решения одного класса задач  $H_\infty$ -оптимизации систем управления // Известия РАН. Теория и системы управления. – 2011. – №3. – С. 52-61.
- 50 Кузовков Н.Т. Модальное управление и наблюдающие устройства. – М.: Машиностроение, 1976. – 181 с.
- 51 Калман Р. Очерки по математической теории систем / пер. с англ. – Изд. 2-е, стер. – М.: Едиториал УРСС, 2004. – 400 с.
- 52 Бигалиева А.З., Мурых Е.Л. Вопросы управления процессом помола с применением фильтра Калмана. // Матер. 25-й междунар. науч.-техн. конф. студентов, аспирантов и молодых ученых «Научная сессия ТУСУР-2020». – Томск, 2020. – С. 18-21.
- 53 Атанов С.К., Бигалиева А.З. Построение фильтра Калмана для восстановления недостающей информации о состоянии процесса помола в планетарной мельнице // Тр. университета. – 2020. – №2. – С. 133-138.
- 54 Chrif L., Z.M. Kadda. Aircraft Control System Using LQG and LQR Controller with Optimal Estimation-Kalman Filter Design // Procedia Engineering. – 2014. – Vol. 80. – P. 245-257.

55 Veremey E.I. Efficient Spectral Approach to SISO Problems of H<sub>2</sub>-Optimal Synthesis // Applied Mathematical Sciences. – 2015. – Vol. 9, Issue 79. – P. 3897-3909.

56 Veremey E.I. Dynamical Correction of Positioning Control laws // IFAC Proceedings Volumes (IFAC Papers Online). – 2013. – Vol. 9, Issue 1. – P. 31-36.

57 Атанов С.К., Бигалиева А.З., Апачиди Н.К., Русак А.В. Вопросы управления процессом тонкого помола в планетарной мельнице // Вестник Санкт-Петербургского университета. – 2020. – Т. 16, №3. – С. 277-292.

58 Атанов С.К., Бигалиева А.З. Синтез LQG регулятора для интеллектуального управления технологическим процессом тонкого помола // Вестник Национальной инженерной академии Республики Казахстан. – 2020. – №4. – С. 21-27.

59 Симанков В.С., Луценко Е.В. Адаптивное управление сложными системами на основе теории распознавания образов: Монография. – Краснодар: ТУ КубГТУ, 1999. – 318 с.

60 Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. – М.: Высшая школа, 1989. – 367 с.

61 Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. Толковый словарь по искусственному интеллекту. М.: Радио и связь, 1992. – 256с.

62 Нейронная сеть // Большая российская энциклопедия: [в 35 т.]/гл. ред. Ю. С. Осипов. – М.: Большая российская энциклопедия, 2004-2017.

63 Круглов В.В., Длин М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети. – М.: Физматлит, 2000. – 224 с.

64 Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 432 с.

65 Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. Пер. с англ. Н.И. Ринго под ред. Н.Н. Моисеева и С.А. Орловского. М.: Мир, 1976. – 166 с.

66 Заде Л. Принятие решений в расплывчатых условиях // В сб.: Вопросы анализа и процедуры принятия решений / Под ред. И.Ф. Шахнова, с предисл. Г.С. Поспелова. – М.: Мир, 1976. – С. 172-215.

67 Zadeh L.A. Fuzzy Sets // Inform. And Control, 1965. Vol.8. – P. 338-353.

68 Mamdani E.H. Application of fuzzy algorithms for control of simple dynamic plant. // Proceedings IEEE. 1974. Vol.121. – P. 1585-1588.

69 Mamdani E.H. Advances in the linguistic synthesis of fuzzy controllers. // International Journal of Man-Machine Studies. 1976. – Vol. 8. P. 669-678.

70 Tomescu M.-L., Preitl S., Precup R.-E., J.K.Tar. Stability analysis Method for fuzzy Control Systems Dedicated Controlling Nonlinear Processes. // Acta Polytechnica Hungarica. 2007. – V. 46 Issue 3. – P. 127-141.

71 Деменков Н.П. Нечеткое управление в технических системах. М.: МГТУ им. Н.Э. Баумана. 2005. – 198 с.

72 Васильев В.И., Ильясов Б.Г. Интеллектуальные системы управления. Теория и практика: учебное пособие. – М.: Радиотехника. 2009. – 392 с.

73. Takagi N., Sugeno M. Fuzzy identification of systems and its applications to modeling and control. - IEEE Transactions on Systems, Man and Cybernetics, vol. 15, no. 1, 1985, pp. 116-132.

74 Катков М.С. Непрерывные системы адаптивного управления с идентификаторами. - М.: Изд-во МПИ «Мир книги», 1992. - 385 с.

75 Пегат А. Нечеткое моделирование и управление. – М.: Бином. Лаборатория знаний, 2009. – 798 с.

76 Chabni F., Taleb R., Benbouali A., Bouthiba M. The Application of Fuzzy Control in Water Tank Level Using Arduino // International Journal of Advanced Computer Science and Applications. – 2016. – Vol. 7. – P. 262-265.

77 Усков А.А. Системы с нечеткими моделями объектов управления. – Смоленск: СФРУК, 2013. – 153 с.

78 Дьяконов В.П., Круглов В. Математические пакеты расширения MATLAB. – СПб.: Питер, 2001. – 480 с.

79 Штовба С.Д. Проектирование нечетких систем средствами Matlab. – М.: Горячая линия-Телеком, 2007. – 288 с.

80 Гостев В.И. Проектирование нечетких регуляторов для систем автоматического управления. – СПб.: БХВ-Петербург, 2011. – 416 с.

81 Яхьяева Г.Э. Нечеткие множества и нейронные сети: учеб. пос. – Изд. 2-е, испр. – М.: Бином. Лаборатория знаний, 2012. – 315 с.

## ПРИЛОЖЕНИЕ А

Свидетельства о государственной регистрации прав на объект авторского права

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

**СВИДЕТЕЛЬСТВО**  
**О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР**  
**ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ**

№ 5042 от «27» августа 2019 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):  
**БИГАЛИЕВА А.ЛЬФИЯ ЗАМИРОВНА**

Вид объекта авторского права: **программа для ЭВМ**

Название объекта: **Интеллектуальное управления технологическим процессом тонкого помола**

Дата создания объекта: **20.08.2019**

ҚР АИМ «ҚҰЛТҒАҚ ЗАҢТҚЕРЛІК МЕНШІК ПАСТІҚАТЫ» АҚ  
ҚАЗАҚСТАН РЕСПУБЛИКАСЫ АРНАУЛЫҚ ПАТЕНТ АГЕНТТІГІ  
ҚАЗАҚСТАН РЕСПУБЛИКАСЫ АРНАУЛЫҚ ПАТЕНТ АГЕНТТІГІ



Құжат түпнұсқалығын <http://www.kazpatent.kz/>; сайтының  
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>  
Подлинность документа возможно проверить на сайте [kazpatent.kz](http://www.kazpatent.kz)  
в разделе «Авторское право» <https://copyright.kazpatent.kz>

Подписано ЭЦП

Оспанов Е.К.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ



РЕСПУБЛИКА КАЗАХСТАН

**СВИДЕТЕЛЬСТВО  
О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР  
ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ**

№ 15286 от «18» февраля 2021 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):  
**БИГАЛИЕВА АЛЬФИЯ ЗАМИРОВНА**

Вид объекта авторского права: **программа для ЭВМ**

Название объекта: **Интеллектуальное управление технологическим процессом тонкого помола**

Дата создания объекта: **16.02.2021**



Құжат түпнұсқалығын <http://www.kazpatent.kz/ru> сайтының  
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте [kazpatent.kz](http://kazpatent.kz)  
в разделе «Авторское право» <https://copyright.kazpatent.kz>

Подписано ЭЦП

Оспанов Е.К.

## ПРИЛОЖЕНИЕ Б

### Акт внедрения

УТВЕРЖДАЮ

Директор ТОО «КМЗ им.

А.Пархоменко

Сурина В.В.

« 2 » 05

2020



### АКТ

о внедрении научно-исследовательских результатов диссертационной работы  
Бигалиевой Альфии Замировны  
по теме «Разработка программного комплекса интеллектуального управления  
технологическим процессом тонкого помола»

Настоящий акт подтверждает, что результаты диссертационного исследования по теме «Разработка программного комплекса интеллектуального управления технологическим процессом тонкого помола», полученные докторантом PhD по специальности 6D070400 «Вычислительная техника и программное обеспечение» Бигалиевой А.З., программное обеспечение «Интеллектуального управления технологическим процессом тонкого помола» (авторы Бигалиева А.З.) использованы в рамках научных исследований для непрерывного контроля тонкости (тонины) помола с использованием законов автоматического управления, синтезированных на базе оптимизационного подхода.

Результаты работы внедрены в систему управления при изготовлении промышленного образца планетарной мельницы тонкого помола в условиях ТОО «КМЗ им. А. Пархоменко», что позволило получить существенную экономию по энергозатратам при оптимальной производительности мельницы.

От предприятия

Исполнительный директор

Начальник ИСО



Булгаков Ю.И.

Шломина Е.С.



АКТ

о внедрении научно-исследовательских результатов диссертационной работы  
Бигалиевой Альфии Замировны  
по теме «Разработка программного комплекса интеллектуального управления  
технологическим процессом тонкого помола»

Настоящий акт подтверждает, что результаты диссертационного исследования по теме «Разработка программного комплекса интеллектуального управления технологическим процессом тонкого помола», полученные докторантом PhD по специальности 8D06104 «Вычислительная техника и программное обеспечение» Бигалиевой А.З., программное обеспечение «Интеллектуальное управление технологическим процессом тонкого помола» (авторы Бигалиева А.З.) внедрены в образовательный процесс.

Указанная работа внедрена в учебный процесс 2020-2021 учебном году в следующих лекционных курсах: «Разработка ПО для создания интеллектуальных систем», «Моделирование систем и комплексов» для подготовки бакалавров по образовательной программе «Вычислительная техника и программное обеспечение».

Зав. кафедрой ИТБ

Коккоз М.М.

## ПРИЛОЖЕНИЕ В

### Код программы

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd1(0x27, 16, 2);
LiquidCrystal_I2C lcd2(0x25, 20, 4);

float tonkost;
float shum;
float zagr;
float temper;

int oborot;
int podacha;
String data, data1;
byte k, s1;

#include <Fuzzy.h>
Fuzzy *fuzzy = new Fuzzy();

// FuzzyInput Тонкость    X1
FuzzySet *melko_tonkost   = new FuzzySet(40, 40, 70, 100);
FuzzySet *norm_tonkost    = new FuzzySet(70, 100, 100, 130);
FuzzySet *krup_tonkost    = new FuzzySet(100, 130, 160, 160);

// FuzzyInput Шум        X2
FuzzySet *min_shum        = new FuzzySet(20, 20, 40, 60);
FuzzySet *norm_shum       = new FuzzySet(40, 60, 60, 80);
FuzzySet *maks_shum       = new FuzzySet(80, 80, 100, 100);

// FuzzyInput Загрузка   X3
FuzzySet *min_zag         = new FuzzySet(1, 1, 2, 3);
FuzzySet *norm_zag        = new FuzzySet(2, 3, 3, 4);
FuzzySet *maks_zag        = new FuzzySet(3, 4, 5, 5);

// FuzzyInput Температура X4
FuzzySet *min_temp        = new FuzzySet(20, 20, 35, 50);
FuzzySet *norm_temp       = new FuzzySet(35, 50, 50, 65);
FuzzySet *maks_temp       = new FuzzySet(50, 65, 80, 80);

// FuzzyOutput Обороты   Y1
```

```

FuzzySet *umen_1_1      = new FuzzySet(0, 0, 300, 600);
FuzzySet *umen_1_2      = new FuzzySet(300, 600, 600, 900);
FuzzySet *bez_izm_1     = new FuzzySet(600, 900, 900, 1200);
FuzzySet *uvel_1_1     = new FuzzySet(900, 1200, 1200, 1500);
FuzzySet *uvel_1_2     = new FuzzySet(1200, 1500, 1800, 1800);

```

```

// FuzzyOutput Подача      Y2
FuzzySet *umen_2_1      = new FuzzySet(0, 0, 200, 400);
FuzzySet *umen_2_2      = new FuzzySet(200, 400, 400, 600);
FuzzySet *bez_izm_2     = new FuzzySet(400, 600, 600, 800);
FuzzySet *uvel_2_1     = new FuzzySet(600, 800, 800, 1000);
FuzzySet *uvel_2_2     = new FuzzySet(800, 1000, 1200, 1200);

```

```

#include "2_fuzzy_set.h"
#include "3_fuzzy_rule.h"

```

```

void setup()
{
  Serial.begin(115200);
  lcd1.begin();
  lcd2.begin();
  fuzzySet ();
  fuzzyRule ();
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
}

```

```

void loop()
{
  lcd1.clear();
  lcd2.clear();
  tonkost = random(4000, 16000);
  tonkost=tonkost/100;
  shum = random(2000, 10000);
  shum=shum/100;
  zagr = random(100, 500);
  zagr=zagr/100;
  temper = random(2000, 8000);
  temper=temper/100;
  lcd2.setCursor(0, 0);
  lcd2.print("Tonkost'  = ");
  lcd2.setCursor(14, 0);
  lcd2.print(tonkost);
  lcd2.setCursor(0, 1);

```

```

lcd2.print("Shum    = ");
lcd2.setCursor(14, 1);
lcd2.print(shum);
lcd2.setCursor(0, 2);
lcd2.print("Zagruzka  = ");
lcd2.setCursor(14, 2);
lcd2.print(zagr);
lcd2.setCursor(0, 3);
lcd2.print("Temperatura = ");
lcd2.setCursor(14, 3);
lcd2.print(temper);
fuzzy->setInput(1, tonkost);
fuzzy->setInput(2, shum);
fuzzy->setInput(3, zagr);
fuzzy->setInput(4, temper);
fuzzy->fuzzify();

oborot = fuzzy->defuzzify(1);
podacha = fuzzy->defuzzify(2);
if (oborot==0){
    oborot=900;
}
if (podacha==0){
    podacha=600;
}
lcd1.setCursor(0, 0);
lcd1.print("Oboroty  = ");
lcd1.setCursor(12, 0);
lcd1.print(oborot);
lcd1.setCursor(0, 1);
lcd1.print("Podacha  = ");
lcd1.setCursor(12, 1);
lcd1.print(podacha);

Serial.print("B");
Serial.print(tonkost);
Serial.print("&");
Serial.print(shum);
Serial.print("&");
Serial.print(zagr);
Serial.print("&");
Serial.print(temper);
Serial.print("S");
Serial.print(oborot);

```

```

    Serial.print("!");
    Serial.print(podacha);
    Serial.println("E");
    delay(2500);
}

void fuzzySet () {
    // FuzzyInput POMOL
    FuzzyInput *pomol = new FuzzyInput(1);
    pomol->addFuzzySet(melko_tonkost);
    pomol->addFuzzySet(norm_tonkost);
    pomol->addFuzzySet(krup_tonkost);
    fuzzy->addFuzzyInput(pomol);

    // FuzzyInput SHUM
    FuzzyInput *ur_shuma = new FuzzyInput(2);
    ur_shuma->addFuzzySet(min_shum);
    ur_shuma->addFuzzySet(norm_shum);
    ur_shuma->addFuzzySet(maks_shum);
    fuzzy->addFuzzyInput(ur_shuma);

    // FuzzyInput ZAGRUKA BARABANA
    FuzzyInput *zagruzka = new FuzzyInput(3);
    zagruzka->addFuzzySet(min_zag);
    zagruzka->addFuzzySet(norm_zag);
    zagruzka->addFuzzySet(maks_zag);
    fuzzy->addFuzzyInput(zagruzka);

    // FuzzyInput Temperatura
    FuzzyInput *Temperatura = new FuzzyInput(4);
    Temperatura->addFuzzySet(min_temp);
    Temperatura->addFuzzySet(norm_temp);
    Temperatura->addFuzzySet(maks_temp);
    fuzzy->addFuzzyInput(Temperatura);

    // FuzzyOutput oborot
    FuzzyOutput *oborot = new FuzzyOutput(1);
    oborot->addFuzzySet(umen_1_1);
    oborot->addFuzzySet(umen_1_2);
    oborot->addFuzzySet(bez_izm_1);
    oborot->addFuzzySet(uvel_1_1);
    oborot->addFuzzySet(uvel_1_2);
    fuzzy->addFuzzyOutput(oborot);
}

```

```

// FuzzyOutput podacha
FuzzyOutput *podacha = new FuzzyOutput(2);
podacha->addFuzzySet(umen_2_1);
podacha->addFuzzySet(umen_2_2);
podacha->addFuzzySet(bez_izm_2);
podacha->addFuzzySet(uvel_2_1);
podacha->addFuzzySet(uvel_2_2);
fuzzy->addFuzzyOutput(podacha);
}

void fuzzyRule () {
// Building FuzzyRule//////////////////////////////////// 1
FuzzyRuleAntecedent *Pr1_1_1 = new FuzzyRuleAntecedent();
Pr1_1_1->joinWithAND(melko_tonkost, min_shum);

FuzzyRuleAntecedent *Pr2_1_1 = new FuzzyRuleAntecedent();
Pr2_1_1->joinWithAND(min_zag, min_temp);

FuzzyRuleAntecedent *Pravilo_1 = new FuzzyRuleAntecedent();
Pravilo_1->joinWithAND(Pr1_1_1, Pr2_1_1);

FuzzyRuleConsequent *P_1 = new FuzzyRuleConsequent();
P_1->addOutput(umen_1_1);
P_1->addOutput(uvel_2_2);

FuzzyRule *fuzzyRule1 = new FuzzyRule(1, Pravilo_1, P_1);
fuzzy->addFuzzyRule(fuzzyRule1);

// Building FuzzyRule//////////////////////////////////// 2
FuzzyRuleAntecedent *Pr1_2_1 = new FuzzyRuleAntecedent();
Pr1_2_1->joinWithAND(norm_tonkost, min_shum);

FuzzyRuleAntecedent *Pravilo_2 = new FuzzyRuleAntecedent();
Pravilo_2->joinWithAND(Pr1_2_1, Pr2_1_1);

FuzzyRuleConsequent *P_2 = new FuzzyRuleConsequent();
P_2->addOutput(bez_izm_1);
P_2->addOutput(uvel_2_2);

FuzzyRule *fuzzyRule2 = new FuzzyRule(2, Pravilo_2, P_2);
fuzzy->addFuzzyRule(fuzzyRule2);

// Building FuzzyRule//////////////////////////////////// 3

```

```

FuzzyRuleAntecedent *Pr1_3_1 = new FuzzyRuleAntecedent();
Pr1_3_1->joinWithAND(krup_tonkost, min_shum);

FuzzyRuleAntecedent *Pravilo_3 = new FuzzyRuleAntecedent();
Pravilo_3->joinWithAND(Pr1_3_1, Pr2_1_1);

FuzzyRuleConsequent *P_3 = new FuzzyRuleConsequent();
P_3->addOutput(uvel_1_2);
P_3->addOutput(uvel_2_2);

FuzzyRule *fuzzyRule3 = new FuzzyRule(3, Pravilo_3, P_3);
fuzzy->addFuzzyRule(fuzzyRule3);

// Building FuzzyRule//////////////////////////////////// 4
FuzzyRuleAntecedent *Pr1_1_2 = new FuzzyRuleAntecedent();
Pr1_1_2->joinWithAND(melko_tonkost, norm_shum);

FuzzyRuleAntecedent *Pravilo_4 = new FuzzyRuleAntecedent();
Pravilo_4->joinWithAND(Pr1_1_2, Pr2_1_1);

FuzzyRuleConsequent *P_4 = new FuzzyRuleConsequent();
P_4->addOutput(umen_1_1);
P_4->addOutput(uvel_2_1);

FuzzyRule *fuzzyRule4 = new FuzzyRule(4, Pravilo_4, P_4);
fuzzy->addFuzzyRule(fuzzyRule4);

// Building FuzzyRule//////////////////////////////////// 5
FuzzyRuleAntecedent *Pr1_1_3 = new FuzzyRuleAntecedent();
Pr1_1_3->joinWithAND(melko_tonkost, maks_shum);

FuzzyRuleAntecedent *Pravilo_5 = new FuzzyRuleAntecedent();
Pravilo_5->joinWithAND(Pr1_1_3, Pr2_1_1);

FuzzyRuleConsequent *P_5 = new FuzzyRuleConsequent();
P_5->addOutput(umen_1_1);
P_5->addOutput(umen_2_2);

FuzzyRule *fuzzyRule5 = new FuzzyRule(5, Pravilo_5, P_5);
fuzzy->addFuzzyRule(fuzzyRule5);

// Building FuzzyRule//////////////////////////////////// 6
FuzzyRuleAntecedent *Pr2_2_1 = new FuzzyRuleAntecedent();
Pr2_2_1->joinWithAND(norm_zag, min_temp);

```

```

FuzzyRuleAntecedent *Pravilo_6 = new FuzzyRuleAntecedent();
Pravilo_6->joinWithAND(Pr1_1_1, Pr2_2_1);

FuzzyRuleConsequent *P_6 = new FuzzyRuleConsequent();
P_6->addOutput(umen_1_1);
P_6->addOutput(bez_izm_2);

FuzzyRule *fuzzyRule6 = new FuzzyRule(6, Pravilo_6, P_6);
fuzzy->addFuzzyRule(fuzzyRule6);

// Building FuzzyRule//////////////////////////////////// 7
FuzzyRuleAntecedent *Pr2_3_1 = new FuzzyRuleAntecedent();
Pr2_3_1->joinWithAND(maks_zag, min_temp);

FuzzyRuleAntecedent *Pravilo_7 = new FuzzyRuleAntecedent();
Pravilo_7->joinWithAND(Pr1_1_1, Pr2_3_1);

FuzzyRuleConsequent *P_7 = new FuzzyRuleConsequent();
P_7->addOutput(umen_1_1);
P_7->addOutput(umen_2_1);

FuzzyRule *fuzzyRule7 = new FuzzyRule(7, Pravilo_7, P_7);
fuzzy->addFuzzyRule(fuzzyRule7);

// Building FuzzyRule//////////////////////////////////// 8
FuzzyRuleAntecedent *Pr2_1_2 = new FuzzyRuleAntecedent();
Pr2_1_2->joinWithAND(min_zag, norm_temp);

FuzzyRuleAntecedent *Pravilo_8 = new FuzzyRuleAntecedent();
Pravilo_8->joinWithAND(Pr1_1_1, Pr2_1_2);

FuzzyRuleConsequent *P_8 = new FuzzyRuleConsequent();
P_8->addOutput(umen_1_1);
P_8->addOutput(uvel_2_1);

FuzzyRule *fuzzyRule8 = new FuzzyRule(8, Pravilo_8, P_8);
fuzzy->addFuzzyRule(fuzzyRule8);

// Building FuzzyRule//////////////////////////////////// 9
FuzzyRuleAntecedent *Pr2_1_3 = new FuzzyRuleAntecedent();
Pr2_1_3->joinWithAND(min_zag, maks_temp);

FuzzyRuleAntecedent *Pravilo_9 = new FuzzyRuleAntecedent();

```

```

Pravilo_9->joinWithAND(Pr1_1_1, Pr2_1_3);

FuzzyRuleConsequent *P_9 = new FuzzyRuleConsequent();
P_9->addOutput(umen_1_1);
P_9->addOutput(umen_2_2);

FuzzyRule *fuzzyRule9 = new FuzzyRule(9, Pravilo_9, P_9);
fuzzy->addFuzzyRule(fuzzyRule9);

// Building FuzzyRule//////////////////////////////////// 10
FuzzyRuleAntecedent *Pr1_2_2 = new FuzzyRuleAntecedent();
Pr1_2_2->joinWithAND(norm_tonkost, norm_shum);

FuzzyRuleAntecedent *Pravilo_10 = new FuzzyRuleAntecedent();
Pravilo_10->joinWithAND(Pr1_2_2, Pr2_1_1);

FuzzyRuleConsequent *P_10 = new FuzzyRuleConsequent();
P_10->addOutput(umen_1_2);
P_10->addOutput(uvel_2_1);

FuzzyRule *fuzzyRule10 = new FuzzyRule(10, Pravilo_10, P_10);
fuzzy->addFuzzyRule(fuzzyRule10);

// Building FuzzyRule//////////////////////////////////// 11
FuzzyRuleAntecedent *Pravilo_11 = new FuzzyRuleAntecedent();
Pravilo_11->joinWithAND(Pr1_2_2, Pr2_2_1);

FuzzyRuleConsequent *P_11 = new FuzzyRuleConsequent();
P_11->addOutput(bez_izm_1);
P_11->addOutput(bez_izm_2);

FuzzyRule *fuzzyRule11 = new FuzzyRule(11, Pravilo_11, P_11);
fuzzy->addFuzzyRule(fuzzyRule11);

// Building FuzzyRule//////////////////////////////////// 12
FuzzyRuleAntecedent *Pr2_2_2 = new FuzzyRuleAntecedent();
Pr2_2_2->joinWithAND(norm_zag, norm_temp);

FuzzyRuleAntecedent *Pravilo_12 = new FuzzyRuleAntecedent();
Pravilo_12->joinWithAND(Pr1_2_2, Pr2_2_2);

FuzzyRuleConsequent *P_12 = new FuzzyRuleConsequent();
P_12->addOutput(umen_1_2);
P_12->addOutput(bez_izm_2);

```

```

FuzzyRule *fuzzyRule12 = new FuzzyRule(12, Pravilo_12, P_12);
fuzzy->addFuzzyRule(fuzzyRule12);

// Building FuzzyRule//////////////////////////////////// 13
FuzzyRuleAntecedent *Pravilo_13 = new FuzzyRuleAntecedent();
Pravilo_13->joinWithAND(Pr1_1_1, Pr2_2_2);

FuzzyRuleConsequent *P_13 = new FuzzyRuleConsequent();
P_13->addOutput(umen_1_1);
P_13->addOutput(umen_2_2);

FuzzyRule *fuzzyRule13 = new FuzzyRule(13, Pravilo_13, P_13);
fuzzy->addFuzzyRule(fuzzyRule13);

// Building FuzzyRule//////////////////////////////////// 14
FuzzyRuleAntecedent *Pr2_3_3 = new FuzzyRuleAntecedent();
Pr2_3_3->joinWithAND(maks_zag, maks_temp);

FuzzyRuleAntecedent *Pravilo_14 = new FuzzyRuleAntecedent();
Pravilo_14->joinWithAND(Pr1_1_1, Pr2_3_3);

FuzzyRuleConsequent *P_14 = new FuzzyRuleConsequent();
P_14->addOutput(umen_1_1);
P_14->addOutput(umen_2_1);

FuzzyRule *fuzzyRule14 = new FuzzyRule(14, Pravilo_14, P_14);
fuzzy->addFuzzyRule(fuzzyRule14);

// Building FuzzyRule//////////////////////////////////// 15
FuzzyRuleAntecedent *Pr2_3_2 = new FuzzyRuleAntecedent();
Pr2_3_2->joinWithAND(maks_zag, norm_temp);

FuzzyRuleAntecedent *Pravilo_15 = new FuzzyRuleAntecedent();
Pravilo_15->joinWithAND(Pr1_1_1, Pr2_3_2);

FuzzyRuleConsequent *P_15 = new FuzzyRuleConsequent();
P_15->addOutput(umen_1_2);
P_15->addOutput(umen_2_1);

FuzzyRule *fuzzyRule15 = new FuzzyRule(15, Pravilo_15, P_15);
fuzzy->addFuzzyRule(fuzzyRule15);

// Building FuzzyRule//////////////////////////////////// 16

```

```
FuzzyRuleAntecedent *Pr2_2_3 = new FuzzyRuleAntecedent();
Pr2_2_3->joinWithAND(norm_zag, maks_temp);
```

```
FuzzyRuleAntecedent *Pravilo_16 = new FuzzyRuleAntecedent();
Pravilo_16->joinWithAND(Pr1_1_1, Pr2_2_3);
```

```
FuzzyRuleConsequent *P_16 = new FuzzyRuleConsequent();
P_16->addOutput(umen_1_1);
P_16->addOutput(umen_2_2);
```

```
FuzzyRule *fuzzyRule16 = new FuzzyRule(16, Pravilo_16, P_16);
fuzzy->addFuzzyRule(fuzzyRule16);
```

```
// Building FuzzyRule//////////////////////////////////// 17
FuzzyRuleAntecedent *Pr1_3_3 = new FuzzyRuleAntecedent();
Pr1_3_3->joinWithAND(krup_tonkost, maks_shum);
```

```
FuzzyRuleAntecedent *Pravilo_17 = new FuzzyRuleAntecedent();
Pravilo_17->joinWithAND(Pr1_3_3, Pr2_2_3);
```

```
FuzzyRuleConsequent *P_17 = new FuzzyRuleConsequent();
P_17->addOutput(umen_1_2);
P_17->addOutput(umen_2_1);
```

```
FuzzyRule *fuzzyRule17 = new FuzzyRule(17, Pravilo_17, P_17);
fuzzy->addFuzzyRule(fuzzyRule17);
```

```
// Building FuzzyRule//////////////////////////////////// 18
FuzzyRuleAntecedent *Pravilo_18 = new FuzzyRuleAntecedent();
Pravilo_18->joinWithAND(Pr1_3_3, Pr2_3_1);
```

```
FuzzyRuleConsequent *P_18 = new FuzzyRuleConsequent();
P_18->addOutput(uvel_1_1);
P_18->addOutput(umen_2_1);
```

```
FuzzyRule *fuzzyRule18 = new FuzzyRule(18, Pravilo_18, P_18);
fuzzy->addFuzzyRule(fuzzyRule18);
```

```
// Building FuzzyRule//////////////////////////////////// 19
FuzzyRuleAntecedent *Pravilo_19 = new FuzzyRuleAntecedent();
Pravilo_19->joinWithAND(Pr1_3_3, Pr2_3_2);
```

```
FuzzyRuleConsequent *P_19 = new FuzzyRuleConsequent();
P_19->addOutput(umen_1_2);
```

```

P_19->addOutput(umen_2_1);

FuzzyRule *fuzzyRule19 = new FuzzyRule(19, Pravilo_19, P_19);
fuzzy->addFuzzyRule(fuzzyRule19);

// Building FuzzyRule//////////////////////////////////// 20
FuzzyRuleAntecedent *Pravilo_20 = new FuzzyRuleAntecedent();
Pravilo_20->joinWithAND(Pr1_3_3, Pr2_1_3);

FuzzyRuleConsequent *P_20 = new FuzzyRuleConsequent();
P_20->addOutput(umen_1_2);
P_20->addOutput(umen_2_2);

FuzzyRule *fuzzyRule20 = new FuzzyRule(20, Pravilo_20, P_20);
fuzzy->addFuzzyRule(fuzzyRule20);

// Building FuzzyRule//////////////////////////////////// 21
FuzzyRuleAntecedent *Pravilo_21 = new FuzzyRuleAntecedent();
Pravilo_21->joinWithAND(Pr1_3_3, Pr2_2_3);

FuzzyRuleConsequent *P_21 = new FuzzyRuleConsequent();
P_21->addOutput(umen_1_2);
P_21->addOutput(umen_2_1);

FuzzyRule *fuzzyRule21 = new FuzzyRule(21, Pravilo_21, P_21);
fuzzy->addFuzzyRule(fuzzyRule21);

// Building FuzzyRule//////////////////////////////////// 22
FuzzyRuleAntecedent *Pravilo_22 = new FuzzyRuleAntecedent();
Pravilo_22->joinWithAND(Pr1_3_3, Pr2_1_1);

FuzzyRuleConsequent *P_22 = new FuzzyRuleConsequent();
P_22->addOutput(bez_izm_1);
P_22->addOutput(bez_izm_2);

FuzzyRule *fuzzyRule22 = new FuzzyRule(22, Pravilo_22, P_22);
fuzzy->addFuzzyRule(fuzzyRule22);

// Building FuzzyRule//////////////////////////////////// 23
FuzzyRuleAntecedent *Pravilo_23 = new FuzzyRuleAntecedent();
Pravilo_23->joinWithAND(Pr1_3_3, Pr2_1_2);

FuzzyRuleConsequent *P_23 = new FuzzyRuleConsequent();
P_23->addOutput(bez_izm_1);

```

```

P_23->addOutput(uvel_2_1);

FuzzyRule *fuzzyRule23 = new FuzzyRule(23, Pravilo_23, P_23);
fuzzy->addFuzzyRule(fuzzyRule23);

// Building FuzzyRule//////////////////////////////////// 24
FuzzyRuleAntecedent *Pr1_3_2 = new FuzzyRuleAntecedent();
Pr1_3_2->joinWithAND(krup_tonkost, norm_shum);

FuzzyRuleAntecedent *Pravilo_24 = new FuzzyRuleAntecedent();
Pravilo_24->joinWithAND(Pr1_3_2, Pr2_3_3);

FuzzyRuleConsequent *P_24 = new FuzzyRuleConsequent();
P_24->addOutput(bez_izm_1);
P_24->addOutput(umen_2_1);

FuzzyRule *fuzzyRule24 = new FuzzyRule(24, Pravilo_24, P_24);
fuzzy->addFuzzyRule(fuzzyRule24);

// Building FuzzyRule//////////////////////////////////// 25
FuzzyRuleAntecedent *Pravilo_25 = new FuzzyRuleAntecedent();
Pravilo_25->joinWithAND(Pr1_3_1, Pr2_3_3);

FuzzyRuleConsequent *P_25 = new FuzzyRuleConsequent();
P_25->addOutput(bez_izm_1);
P_25->addOutput(umen_2_1);

FuzzyRule *fuzzyRule25 = new FuzzyRule(25, Pravilo_25, P_25);
fuzzy->addFuzzyRule(fuzzyRule25);

// Building FuzzyRule//////////////////////////////////// 26
FuzzyRuleAntecedent *Pravilo_26 = new FuzzyRuleAntecedent();
Pravilo_26->joinWithAND(Pr1_3_2, Pr2_3_2);

FuzzyRuleConsequent *P_26 = new FuzzyRuleConsequent();
P_26->addOutput(uvel_1_1);
P_26->addOutput(umen_2_1);

FuzzyRule *fuzzyRule26 = new FuzzyRule(26, Pravilo_26, P_26);
fuzzy->addFuzzyRule(fuzzyRule26);

// Building FuzzyRule//////////////////////////////////// 27
FuzzyRuleAntecedent *Pravilo_27 = new FuzzyRuleAntecedent();
Pravilo_27->joinWithAND(Pr1_3_2, Pr2_3_1);

```

```

FuzzyRuleConsequent *P_27 = new FuzzyRuleConsequent();
P_27->addOutput(uvel_1_1);
P_27->addOutput(umen_2_1);

FuzzyRule *fuzzyRule27 = new FuzzyRule(27, Pravilo_27, P_27);
fuzzy->addFuzzyRule(fuzzyRule27);

// Building FuzzyRule//////////////////////////////////// 28
FuzzyRuleAntecedent *Pravilo_28 = new FuzzyRuleAntecedent();
Pravilo_28->joinWithAND(Pr1_3_2, Pr2_1_1);

FuzzyRuleConsequent *P_28 = new FuzzyRuleConsequent();
P_28->addOutput(uvel_1_1);
P_28->addOutput(uvel_2_1);

FuzzyRule *fuzzyRule28 = new FuzzyRule(28, Pravilo_28, P_28);
fuzzy->addFuzzyRule(fuzzyRule28);
}

```