

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н.Ельцина»
Нижнетагильский технологический институт (филиал)
Школа магистратуры
Кафедра информационных технологий

Допустить к защите в ГЭК
И.о. зав. каф., к.п.н.
Р.А. Карелова

« ____ » _____ 2021 г.

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ
УПРАВЛЕНИЯ ТЕМПЕРАТУРОЙ В ПОМЕЩЕНИИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

Пояснительная записка
150406 60800601 003 ПЗ

Руководитель
доцент, к.т.н.

В.В. Гоман

Консультант
инженер, ООО «Иридиум»

Р.М. Мухутдинов

Нормоконтролер
доцент, к.п.н.

Р.А. Карелова

Студент
Группа ТВМ-380701-НТ МиР

Ю.А. Тюхтий

Нижний Тагил

2021

Школа магистратуры
Кафедра информационных технологий
Направление 15.04.06 Мехатроника и робототехника

Образовательная программа Управление мехатронными и робототехническими системами

УТВЕРЖДАЮ

Зав. кафедрой _____

(подпись)

(Ф.И.О.)

« _____ » _____ 201__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы (магистерской диссертации)

студента Тюхтий Юлии Алексеевны группы ТВМ-380701-НТ МиР
(фамилия, имя, отчество)

1. Тема ВКР

Разработка интеллектуальной системы управления температурой в помещении

Утверждена приказом по институту от «01» июня 2020 г. № 28/05 (НТ)

2. Руководитель Гоман Виктор Валентинович, доцент, к.т.н.

(Ф.И.О., должность, ученое звание, ученая степень)

3. Исходные данные к работе с указанием места прохождения преддипломной практики

Преддипломная практика проходила в НТИ (филиал) УрФУ, кафедра информационных технологий; исходные данные к работе: техническое задание заказчика, описания аналогов, научные статьи по теме работы.

4. Содержание пояснительной записки (перечень подлежащих к разработке вопросов)

1. Обзор литературных источников

2. Анализ системы управления температурой в помещении в MATLAB-Simulink

3. Разработка алгоритма прогнозирования температуры для интеллектуального термостата

5. Перечень демонстрационных материалов

Мультимедийная презентация работы

6. Консультанты по проекту (работе) с указанием относящихся к ним разделов проекта

Раздел	Консультант	Подпись, дата	
		здание выдал	здание принял
Практическая часть	Мухутдинов Руслан Маисович	14.12.2020	14.12.2020

7. Календарный план

Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
Обзор литературы	14.12.2020-20.12.2020	
Выполнение основной части	21.12.2020-31.12.2020	
Выполнение графических материалов: чертежей, схем	01.01.2021-10.01.2021	
Подготовка презентации и доклада	11.01.2021-26.01.2021	

Руководитель _____ дата 14.12.2020 г.
(подпись)

Задание принял к исполнению _____ дата 14.12.2020 г.
(подпись)

8. Выпускная квалификационная работа закончена _____ 13.01.2021 г.

Пояснительная записка и все материалы просмотрены.

Оценка консультантов:

1. _____ /Практическая часть/
2. _____ /Нормоконтроль/

Считаю возможным допустить _____ к защите его выпускной квалификационной работы в экзаменационной комиссии.

Руководитель _____
(подпись)

9. Допустить _____ к защите выпускной квалификационной работы в экзаменационной комиссии (протокол заседания кафедры/департамента № _____ от «____» _____ 20__ г.)

Зав. кафедрой

(подпись)

Ф.И.О.

СОДЕРЖАНИЕ

Реферат	5
Введение.....	6
1. Обзор литературных источников	8
1.1 Факторы, определяющие климатические условия в помещении.....	8
1.2 Методика расчета теплотерь для помещения	11
1.3 Современные способы регулирования температуры в помещении	18
1.3.1 традиционные системы управления микроклиматом	19
1.3.2 Автоматизированные системы управления микроклиматом	20
1.3.3 Автоматические системы управления микроклиматом.....	21
1.3.4 Алгоритмы для систем терморегулирования с искусственным интеллектом на базе нечеткой логики	22
1.4 Алгоритм прогнозирования температуры для интеллектуального термостата	28
2. Анализ системы управления температурой в помещении в matlab-simulink	35
3. Разработка алгоритма прогнозирования температуры для интеллектуального термостата	47
3.1 Исходные параметры для моделирования.....	47
3.2 Определение параметров сбора данных для интеллектуальной системы управления температурой в помещении.....	48
3.3 Инструменты для реализации системы.....	49
3.4 Реализации аппаратной части	54
3.5 Реализация программной части	60
Заключение	67
Список использованных источников	69
Приложение А	72

РЕФЕРАТ

Пояснительная записка содержит 71 страницу, 35 рисунков, 2 таблицы, 7 листингов, 23 формулы, 22 библиографических источников, 1 приложение.

Ключевые слова: интеллектуальная система управления, умный термостат, нейронная сеть, управление температурой.

Объектом работы является управление температурным режимом в помещении.

Целью работы является разработка интеллектуальной системы управления температурой в помещении.

В первом разделе работы рассмотрены факторы, определяющие климатические условия в помещении и методика расчета теплотерь. Также приведено описание современных способов регулирования температуры в помещении и рассмотрены два реализованных алгоритма для управления температурным режимом в помещении, основанные на математическом анализе и на базе нечеткой логике.

Во втором разделе рассматривается тепловая модель здания для традиционной системы регулирования температуры, реализованная в Matlab simulink. По рассмотренной модели проведен сравнительный анализ использования различного вида регуляторов и его выбор для реализации интеллектуальной системы управления температурным режимом.

В третьем разделе описана реализация алгоритма для предикции температуры на базе нейронных сетей. А также представлено описание реализации аппаратной и программной части интеллектуальной системы.

ВВЕДЕНИЕ

На сегодняшний день концепция Умный дом является стремительно развивающейся областью, множество работ [5,8,9,14,18] посвящено одной из важнейших функциональных составляющих Умного дома автоматическому климат-контролю, который формирует оптимальные показатели микроклимата в помещении с эффективными показателями энергосбережения. Так в работах [12,22] рассмотрены аспекты энергосбережения в умных домах.

От климатических условий в помещении напрямую зависит здоровье и комфортное проживание людей. Поэтому неотъемлемым элементом системы управления микроклиматом Умного дома является интеллектуальный термостат, главная функция которого заключается в управлении температурным режимом в помещении.

«Умный» термостат это современное электронное устройство, которое подключается к улучшенной энергетической системе дома и работает в рамках систем охлаждения, отопления и вентиляции.

Идея данного проекта заключается в том, чтобы разработать такую интеллектуальную систему управления температурой в помещении, которая бы не только регулировала температуру в ручном режиме, но и подбирала показатели температуры автоматически при минимальных энергетических затратах и минимальной общей стоимости потребленной энергии, при этом решение должно быть доступным.

Объектом работы является управление температурным режимом в помещении.

Предметом работы является интеллектуальная система управления температурой в помещении.

Целью работы является разработка интеллектуальной системы управления температурой в помещении.

В соответствии с поставленной целью определены следующие задачи:

- проанализировать предметную область;

- разработать тепловую модель помещения для обеспечения возможности модели ориентированной разработки;
- изучить возможности применения регулятора температуры на основе нечеткой логики;
- определить инструменты и технологии для реализации программы;
- разработать интеллектуальный алгоритм обучения термостата на основе анализа тепловых параметров здания и поведения пользователя;
- реализовать аппаратную часть;
- реализовать программную часть;
- оценить результаты выполненной работы.

Теоретическая значимость работы заключается в том, что предложенная архитектура, а также описание алгоритмов для интеллектуальной системы управления температурой в помещении могут быть использованы для разработки подобных систем.

Практическая значимость работы заключается в том, что разработанная интеллектуальная система управления температурой в помещении, может быть использован с целью автоматического изменения температурного режима в помещении при минимальных энергетических затратах и минимальной общей стоимости потребленной энергии.

По теме диссертации опубликована работа Тюхтий Ю.А., Гоман В.В. «Обзор интеллектуальных термостатов для управления микроклиматом помещений» [6].

По данной тематике источником является список используемой литературы, а также привлечены ресурсы Интернет-источников.

1. ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

В главе «Обзор литературных источников» описаны факторы, определяющие климатические условия в помещении, методика расчета теплопотерь в помещении, а также существующие решения в области систем управления температурой в помещении, и возможные методы моделирования алгоритма для автоматического управления температурой.

1.1 Факторы, определяющие климатические условия в помещении

В помещении с высокой или низкой температурой воздуха и недостаточным уровнем его влажности ни один человек не будет чувствовать себя комфортно. Здоровья человека постоянно прибывающего в помещении напрямую зависит от его микроклимата. Если тело человека получает и выделяет равное количества тепла, то климатически условия в помещении считаются оптимальными. Отклонение параметров микроклимата от нормы вызывает у человека плохое самочувствие, и в последствие сказывается на его здоровье.

Параметры микроклимата формируются в результате воздействия на помещение наружной среды, технологического процесса в помещении и систем отопления-охлаждения и вентиляции или кондиционирования воздуха (рисунок 1.1).



Рисунок 1.1 – Структурная схема формирования микроклимата в помещении

Наружная среда оказывает влияние на микроклимат помещения через ограждающие конструкции и внутренние перегородки в помещении. Технологический процесс играет важную роль в формировании микроклимата. При таком процессе непосредственно в помещении происходит выделение влаги, пыли, газов, что напрямую влияет на формирование температурного режима и состава воздуха. Системы обеспечения микроклимата активно формируют внутренний климат помещения, нейтрализуя отрицательное воздействие наружной среды и технологического процесса.

Оборудовав помещение качественной системой регулирования микроклиматом можно предотвратить появления различных заболеваний и поспособствовать укреплению иммунитета организма человека.

«ГОСТ 30494-2011. Здания жилые и общественные. Параметры микроклимата в помещениях» [1] устанавливает параметры микроклимата обслуживаемой зоны жилых помещений, а также качества воздуха в обслуживаемой зоне указанных помещений и общие требования к оптимальным и допустимым показателям микроклимата и качеству воздуха.

Микроклимат помещения это состояние внутренней среды помещения, оказывающее воздействие на человека, характеризуемое показателями температуры воздуха и ограждающих конструкций, влажностью и подвижностью воздуха.

Допустимые параметры микроклимата это сочетания значений показателей микроклимата, которые при длительном и систематическом воздействии на человека могут вызвать общее и локальное ощущение дискомфорта, ухудшение самочувствия и понижение работоспособности при усиленном напряжении механизмов терморегуляции и не вызывают повреждений или ухудшения состояния здоровья.

Пунктом 4.2 ГОСТ 30494-2011 [1] определены параметры, характеризующие микроклимат в жилых и общественных помещениях:

- температура воздуха;

- скорость движения воздуха;
- относительная влажность воздуха;
- результирующая температура помещения;
- локальная асимметрия результирующей температуры.

Оптимальные и допустимые параметры микроклимата в обслуживаемой зоне помещений жилых зданий следует принимать для соответствующего периода года в пределах значений параметров, приведенных в таблицах 1.

Таблица 1 - Оптимальные и допустимые нормы температуры, относительной влажности и скорости движения воздуха в обслуживаемой зоне помещений жилых зданий и общежитий

Период года	Наименование помещения	Температура воздуха, °С		Результирующая температура, °С		Относительная влажность, %		Скорость движения воздуха, м/с	
		оптимальная	допустимая	оптимальная	допустимая	оптимальная	допустимая, не более	оптимальная, не более	допустимая, не более
Холодный	Жилая комната	20-22	18-24	19-20	17-23	45-30	60	0,15	0,2
	Жилая комната в районах с температурой наиболее холодной пятидневки минус 31°С и ниже	21-23	20-24	20-22	19-23	45-30	60	0,15	0,2
	Кухня	19-21	18-26	18-20	17-25			0,15	0,2
	Туалет	19-21	18-26	18-20	17-25			0,15	0,2
	Ванная, совмещенный санузел	24-26	18-26	23-27	17-26			0,15	0,2
	Помещения для отдыха и учебных занятий	20-22	18-24	19-21	17-23	45-30	60	0,15	0,2
	Межквартирный коридор	18-20	16-22	17-19	15-21	45-30	60		
	Вестибюль, лестничная клетка	16-18	14-20	15-17	13-19				
	Кладовые	16-18	12-22	15-17	11-21				

продолжение таблицы 1

Период года	Наименование помещения	Температура воздуха, °С		Результирующая температура, °С		Относительная влажность, %		Скорость движения воздуха, м/с	
		оптимальная	допустимая	оптимальная	допустимая	оптимальная	допустимая, не более	оптимальная, не более	допустимая, не более
Теплый	Жилая комната	22-25	20-28	22-24	18-27	60-30	65	0,2	0,3

Поддерживать оптимальные параметры микроклимата в помещении можно при помощи интеллектуальной системы регулирования температуры. Комплект электрифицированного оборудования обеспечивает централизованное управление всеми присутствующими в доме системами климата и их отдельными элементами.

Для поддержания комфортных параметров микроклимата в помещении и снижения, затраченных на создание благоприятного микроклимата, энергоресурсов, необходимо определить и рассчитать факторы, при которых создаются теплопотери в помещении.

1.2 Методика расчета теплопотерь для помещения

Теплопотери в простом понимании это количество тепла, которое теряет помещение. Тепловые потери возникают в помещении из-за разницы внешней и внутренней температуры воздуха. Для снижения теплопотерь в помещении производится утепление ограждающих конструкций и налаживание системы отопления, кондиционирования и вентиляции. Главной задачей является поддержание комфортных параметров микроклимата в помещении и снижение затрат на энергоресурсы.

Расчет теплопотерь необходимо обязательно производить при проектировании систем отопления, систем вентиляции, воздушных отопительных систем. Температуры для расчетов берут из нормативных документов.

Исходные данные для расчета: внешняя и внутренняя температура воздуха, конструкция стен, пола, перекрытий, назначение каждого помещения, географическая зона строения. Все тепловые потери напрямую зависят от термического сопротивления ограждающих конструкций, чем оно больше, тем меньше теплопотери.

Для начала необходимо определить потери тепла отапливаемого помещения по следующей формуле 1.1.

$$Q = F_{\text{ок}} \cdot \frac{1}{R_{\text{ок}}} \cdot (t_{\text{в}} - t_{\text{н}}) \quad (1.1)$$

где Q – теплопотери через ограждение, в ккал/час;

$F_{\text{ок}}$ – площадь ограждающей конструкции, в м^2 ;

$R_{\text{ок}}$ – сопротивление конструкции ограждения, в $\text{м}^2 \text{ час } ^\circ\text{C}/\text{ккал}$;

$t_{\text{в}}$ – температура воздуха в помещении;

$t_{\text{н}}$ – температура наружного воздуха.

Ограждения бывают однородными или многослойными. Если ограждение многослойное, необходимо сложить сопротивление всех слоев.

Сопротивление однородного ограждения вычисляется по формуле 1.2.

$$R = \frac{\delta}{\lambda} \quad (1.2)$$

где δ – толщина слоя в м;

λ – коэффициент теплопроводности материала.

Помимо основных теплопотерь существуют также добавочные теплопотери через ограждающие конструкции, которые учитывают: ориентацию ограждения по сторонам света, наличие в помещении двух и более наружных стен, высоту помещений, воздействие ветра на ограждения и охлаждение через открываемые двери.

Рассмотрим подробно виды теплопотерь в помещении и их расчет.

Теплопотери на стены. Расчет теплопотерь помещения через ограждающие конструкции производится по формуле 1.3.

$$Q = AK \cdot (t_{\text{п}} - t_{\text{н}}) \cdot n \cdot (1 + \sum B) \quad (1.3)$$

где Q – дополнительные и основное теплопотери, Вт;

A – расчетная площадь ограждающих конструкций;

K – коэффициент теплопередачи отдельного ограждения;

$t_{\text{п}}$ – температура помещения, °С;

$t_{\text{н}}$ – температура наружного воздуха для холодного периода года, °С;

B – добавочные потери теплоты в долях от основных потерь, Вт;

n – коэффициент учета положения наружной поверхности ограждения по отношению к наружному воздуху.

Коэффициент теплопередачи отдельного ограждения K определяется по формуле 1.4.

$$K = \frac{1}{\frac{1}{\alpha_1} + \frac{\delta_{\text{ст}}}{\lambda_{\text{ст}}} + \frac{1}{\alpha_2}} \quad (1.4)$$

где α_1 – коэффициент теплоотдачи со стороны наружного пространства;

α_2 – коэффициент теплоотдачи со стороны внутреннего помещения;

$\delta_{\text{ст}}$ – толщина ограждающей конструкции, м;

$\lambda_{\text{ст}}$ – теплопроводность ограждающей конструкции.

Определение коэффициентов теплоотдачи α рассчитывается по формуле 1.5.

$$\alpha = \frac{Nu \cdot \lambda_{\text{возд}}}{l} \quad (1.5)$$

где Nu – число Нуссельта;

$\lambda_{\text{возд}}$ – теплопроводность воздуха;

l – длина характерного участка, м.

Число Нуссельта находится по следующей формуле 1.6.

$$Nu = 0.008 \cdot Re^{0.9} \cdot Pr^{0.43} \quad (1.6)$$

где Re – критерий Рейнольдса;

Pr – число Прандтля Критерий.

Критерий Рейнольдса Re задается формулой 1.7.

$$Re = \frac{W \cdot l}{\nu} \quad (1.7)$$

где W – скорость среды;

ν – кинематическая вязкость;

l – длина участка, м.

Коэффициент теплоотдачи для внутренней поверхности принимаем из СП 50.13330.2012 «Тепловая защита зданий».

Теплопотери со стороны грунта. Температура грунта под зданием зависит в первую очередь от теплопроводности и теплоемкости самого грунта и от температуры окружающего воздуха в данной местности в течение года. Для расчета теплопотери через грунт необходимо найти коэффициент теплоотдачи со стороны грунта по формуле 1.8.

$$\alpha_3 = \frac{2 \cdot \lambda_{\text{мат}}}{l * \ln\left(\frac{2 \cdot h_0}{l} + \sqrt{\frac{2 \cdot h_0}{l} + 1}\right)} \quad (1.8)$$

где $\lambda_{\text{мат}}$ – теплопроводность материала;

h_0 – толщина фундамента, м;

l – глубина заложения фундамента, м.

Теплопотери в оконные заполнения. Сквозь окна может уходить до 10% тепла. Утечка тепла из помещения посредством оконных конструкций происходит по нескольким направлениям:

- сквозь блок рамы и элементы переплета;
- за счет теплопроводности воздушных масс и конвекции между стеклами;
- вследствие теплового излучения.

Теплопотери через окна рассчитываются по той же схеме, что и через ограждающие конструкции.

Для расчетов необходимо знать такие параметры как: количество камер и переплетов, наличие покрытия и заполнение газом. Приведенное сопротивление теплопередаче для выбранных окон представлено в СП 23–101–2004 «Проектирование тепловой защиты зданий».

Теплопотери на двери и дверные проемы. При расчете заполнения дверных проемов необходимо учитывать добавку на запуск холодного воздуха через наружные двери, не оборудованные воздушными или воздушно-тепловыми завесами, при открывании их на короткие периоды времени. Теплопотери не учитываются, если двери являются летними или запасными, то есть не открываются постоянно.

Инфильтрация воздуха через ограждающие конструкции. Инфильтрация – это перемещение воздуха через ограждающие конструкции из окружающей среды в помещения за счет ветрового и теплового напоров, формируемых разностью температур и перепадом давления воздуха снаружи и внутри помещений. Она

происходит через небольшие щели в дверных и оконных рамах. Воздух поступает в помещение также из неотапливаемых частей здания – чердаков, подвалов и так далее. Он проникает через отверстия в стенах, полах и потолках, таких как трещины в местах сопряжения двух стен или стены и потолка. Для определения количества фильтрующегося воздуха через окна и стены необходимо найти разность давлений воздуха на наружной и внутренней стороне ограждающей конструкции 1.9.

$$\Delta P = (H - h_i) \cdot (\rho_n - \rho_v) \cdot g + 0.5 \cdot \rho_n \cdot v_n^2 \cdot (c_n - c_z) \cdot k - P_0 \quad (1.9)$$

где H – высота здания, м;

h_i – высота расчетной конструкции от уровня земли, м;

ρ_n – плотность воздуха на наружной поверхности.

Плотность воздуха на наружной поверхности определяется по формуле 1.10.

$$\rho_n = \frac{353}{273 + t_n} \quad (1.10)$$

Плотность воздуха на внутренней поверхности ρ_v , определяемая по формуле 1.11.

$$\rho_v = \frac{353}{273 + t_v} \quad (1.11)$$

где g – ускорение свободного падения;

v_n – скорость ветра в январе;

c_n, c_z – аэродинамические коэффициенты, 0.8 и -0.6 соответственно;

k – коэффициент учета изменений давлений ветра, 0.58;

P_0 – условно-постоянное давление воздуха, которое находится по формуле 1.12.

$$P_0 = 0.5 \cdot H \cdot g \cdot (\rho_H - \rho_B) + 0.25 \cdot v_H^2 \cdot \rho_H \cdot (c_H - c_3) \cdot k \quad (1.12)$$

Расход инфильтрующегося воздуха через ограждения находится по формуле 1.13.

$$G = 0.216 \cdot \frac{\Delta P^{\frac{2}{3}}}{R_u} \cdot A \quad (1.13)$$

где R_u – сопротивление воздухопроницанию, которое находится из формулы 1.14.

$$R_u = \frac{1}{G_n} \cdot \left(\frac{\Delta p^{\frac{2}{3}}}{\Delta p_0} \right) \quad (1.14)$$

где Δp – разность давлений воздуха на наружной и внутренней поверхностях ограждающих конструкций;

Δp_0 – 10 Па – разность давлений воздуха на наружной и внутренней поверхностях светопрозрачных ограждающих конструкций, при которой экспериментально определяется сопротивление воздухопроницанию конструкций выбранного типа; G_n – поперечная воздухопроницаемость [3].

Теплопотери для каждого помещения рассчитываются по формуле 1.15.

$$Q_{\text{пом}} = Q_{\text{огр}} + Q_{\text{вент/инф}} - Q_{\text{быт}} \quad (1.15)$$

где $Q_{\text{пом}}$ – теплопотери всего помещения, Вт;

$Q_{\text{огр}}$ – сумма всех теплопотерь ограждающих конструкций, Вт;

$Q_{\text{вент/инф}}$ – теплопотери на вентиляцию/ инфильтрацию, Вт;

$Q_{\text{быт}}$ – тепловые выделения бытовыми приборами в помещении, Вт.

Сумма тепловых потерь $Q_{\text{огр}}$ складывается из теплотерь ограждений таких как: стена, пол, потолок, окно, дверь.

Тепловые выделения приборов $Q_{\text{быт}}$ таких как: холодильник, стиральная машина, плита, чайник, микроволновка, пылесос, телевизор и пр. Эти электроприборы, потребляя электроэнергию, вырабатывают тепло, которое попадает в помещение, и его нагревает. Почти вся потребляемая электроэнергия (более 90%) превращается в тепло.

В различных инженерных документах $Q_{\text{быт}}$ для жилых помещений с постоянным пребыванием людей находится в диапазоне от 10 до 20 Вт. на кв. метр. Рассчитать тепловые выделения можно по формуле 1.16.

$$Q_{\text{быт}} = A_{\text{пом}} \cdot W_{\text{пом}} \quad (1.16)$$

где $A_{\text{пом}}$ – площадь помещения, м²;

$W_{\text{пом}}$ – количество тепловой энергии на квадратный метр площади, Вт/м².

Количество тепловой энергии по статистике принимает значение в пределах $10 \text{ Вт/м}^2 \leq W_{\text{пом}} \leq 20 \text{ Вт/м}^2$.

Для компенсации теплотерь и поддержания комфортного микроклимата в помещении с наименьшими затратами энергоресурсов существуют современные способы регулирования и управления температурным режимом.

1.3 Современные способы регулирования температуры в помещении

Практический каждый человек хочет создать комфортные условия проживания. Основными пунктами обеспечения комфорта в доме являются правильная температура и чистота воздуха. Идеальные условия можно создать с помощью системы микроклимата для дома.

Воздух в помещении считается комфортным, если он чистый и увлажненный в нормальных пределах. Искусственно его можно создать с помощью системы фильтров, нагрева, охлаждения, увлажнения и рекуперации.

Проанализировав научно-техническую литературу по существующим системам управления микроклиматом можно разделить их на три условные группы: традиционные, автоматизированные и автоматические. Рассмотрим каждую группу подробнее.

1.3.1 Традиционные системы управления микроклиматом

В традиционных системах управление системой микроклимата осуществляется посредством датчиков температуры и механических преобразователей.

Механический терморегулятор – это устройство, работающее по принципу расширения газов при изменении температуры, или биметаллические пластины, изменяющие свою форму от нагревания или охлаждения. Механическое устройство являются одновременно и датчиком температуры, и исполнительным органом, объединёнными в единый узел. В нагревательных устройствах используется биметаллическая пластина, которая представляет собой термопару из двух металлов с разным коэффициентом теплового расширения [7].

Например, для управления системой тёплых полов используется механический регулятор. Для его включения достаточно нажать на кнопку и повернуть ручку регулятора на необходимую температуру. При понижении температуры ниже заданного значения прибор начнет работать, если же температура станет выше заданного значения, то прибор отключится для охлаждения помещения. Прибор сможет работать при очень низких температурах.

Для управления радиатором отопления в традиционных системах контроля температуры в помещении используются термостатические вентили и ручные или механические термоголовки.

Ручные термоголовки работают по принципу стандартного крана, с помощью которого можно регулировать объем теплоносителя транспортируемого в радиатор отопления. Такой вид термоголовок устанавливается обычно вместо шарового крана. Недостаток ручной термоголовки в том, что при частом ее переключении может ослабнуть моховик вентиля и термоголовка выйдет из строя.

Механические термоголовки представляют собой более сложную конструкцию. В основе прибора размещен сифон цилиндрической формы, внутри которого размещен тепловой агент в жидком или газообразном виде. Установив необходимую температуру на термоголовки, устройство при помощи расширения или сужения цилиндра регулирует подачу теплоносителя в радиатор отопления, тем самым регулируя температуру.

Основные достоинства таких систем – простота конструкции и дешевизна, а также большая ремонтпригодность. К основным недостаткам относится низкая точность поддержания заданных параметров, поэтому для таких систем необходим постоянный контроль и высокое энергопотребление. Используются наиболее часто, что связано с их относительной простотой реализации.

1.3.2 Автоматизированные системы управления микроклиматом

Автоматизированные системы управления базируются на использовании микроконтроллеров таких как, Siemens Logo, Pixel. В такой системе механическое реле заменено электронным «программируемым выключателем», который позволяет чётко задавать требуемые параметры работы в соответствии с заранее заданным алгоритмом. Электронный терморегулятор состоит из основного блока и датчика температуры, который подает сигналы об увеличении или понижении заданной температуры в системе.

Для управления радиатором отопления в автоматизированных системах управления температурой в помещении используются электронные термоголовки с встроенным микропроцессором, который управляет штоком регулирования подачи теплоносителя в систему. В таком устройстве появляются

дополнительные функции, например выставлении температуры по времени, или программирование температуры по дням недели. Работает термоголовка от батареек, которых хватает на достаточное время, но вес же необходимо отслеживать их заряд.

К достоинствам следует отнести возможность точного поддержания параметров в заданных пределах, относительно невысокую стоимость. Недостатки заключаются в следующем: некорректное управление микроклиматом при режимах, которые заранее не заданы в алгоритмах функционирования; необходимость смены алгоритмов при изменении условий функционирования или добавлении новых элементов в систему управления.

1.3.3 Автоматические системы управления микроклиматом

Под автоматическими системами управления понимают комплекс взаимодействующих между собой механизмов управляемого объекта, предназначенных для управления объектом без вмешательства человека. В настоящее время автоматические системы управления разрабатывают с использованием методов искусственного интеллекта. В этом случае выделяют только входные и выходные параметры управления, не выделяя в явном виде физические принципы построения модели управления. Такая модель является некоторой аппроксимацией наблюдаемых процессов. К таким системам управления можно отнести всевозможные системы, разработанные на основе: дифференциальных уравнений теплового баланса здания; нечеткой логики; нейронных сетей; мультиагентных систем; генетических алгоритмов и др.

Примером автоматической системой управления температурным режимом в помещении является система, состоящая из интеллектуального термостата и датчиков температуры и влажности.

1.3.4 Алгоритмы для систем терморегулирования с искусственным интеллектом на базе нечеткой логики

Рассмотрим алгоритм для системы контроля температуры в помещении построенный на базе нечеткой нейронной сети. Регулирование объектами системы осуществляется контролером Fuzzy Controller, работающим на основе выводов правил нечеткой логики.

Алгоритм позволяет минимизировать невязки между поступающим вектором состояния переменных и идеальным вектором состояния переменных на основе нечетких правил.

Нечеткая логика предназначена для формализации человеческих способностей к неточным или приближенным рассуждениям, которые позволяют более правильно описывать ситуации с неопределённостью. В классической логике проблема неопределенности ни как не решается, так как все рассуждения и высказывания могут быть «истиной» (И, 1) или «ложью» (Л, 0), что эквивалентно человеческому «да» или «нет». Однако человеком могут быть приняты различные решения такие как «Возможно да», «Не могу сказать», «Возможно нет», «Уже нет» и другие, поэтому истинность рассуждений в нечеткой логике оценивается как некая степень, которая может принимать разные значения отличные от «И» или «Л».

Для возможности выражения неопределенных значений необходимо выстроить некую логическую систему, которая позволит какому-либо высказыванию иметь истинное значение, которое отличается от бинарного «И» или «Л». Существует подход расширения множества истинных значений {И, Л}, который позволит высказываниям принимать дополнительные значения истинности.

Л. Заде предложил вариант определения нечеткой логике множество истинных значений высказываний обобщать до интервала действительных значений $[0,1]$, тем самым высказывание может принимать любое значение из заданного интервала. Полученное численное значение это и есть количественная

оценка степени истинности выражения, о котором точно нельзя сказать относиться оно к истине или лжи. Такой подход позволил построить логическую схему, по которой возможно находить рассуждения с неопределенностью и оценивать истинность выражения, например, таких как «Скорость поезда довольно высокая» и другие.

В системе нечеткого управления и нечеткой логике центральное место занимает нечеткий вывод. Система нечеткого вывода – это процесс получения нечетких заключений о требуемом управлении объектом на основе нечетких условий или предпосылок, представляющих собой информацию о текущем состоянии объекта. Этот процесс соединяет в себе все основные концепции теории нечетких множеств: функции принадлежности, лингвистические переменные, нечеткие логические операции, методы нечеткой импликации и нечеткой композиции.

Системы нечеткого вывода предназначены для реализации процесса нечеткого вывода и служат концептуальным базисом всей современной нечеткой логики. Системы нечеткого вывода позволяют решать задачи автоматического управления, классификации данных, распознавания образов, принятия решений, машинного обучения и многое другое.

Архитектура нечеткого управления основана на замене классической системы управления системой нечеткого управления, в качестве которой используются системы нечеткого вывода (рисунок 1.2).



Рисунок 1.2 – Модель нечеткого управления

На рисунке 1.3 представлена диаграмма процесса нечеткого вывода.



Рисунок 1.3 – Диаграмма процесса нечеткого вывода

Для формирования системы нечеткого вывода необходимо сформировать базу правил нечетких продукций и реализовывать нечеткий вывод заключений на основе посылок или условий, представленных в форме нечетких лингвистических высказываний.

Основными этапами нечеткого вывода являются:

1. Фаззификация входных переменных – это этап выполнения нечеткого вывода и процедура нахождения значений функций принадлежности нечетких множеств (термов) на основе обычных исходных данных $\mu_A^i(x_j)$.
2. Агрегирование функций принадлежности находится по формуле 1.20.

$$w_i = \mu_A^i(x_j) \quad (1.20)$$

Агрегирование представляет собой процедуру определения степени истинности условий по каждому из правил системы нечеткого вывода.

3. Формирование сигналов по каждому правилу находится по формуле 1.21.

$$y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij}x_j \quad (1.21)$$

4. Аккумуляция заключений – это умножение сигналов $y_i(x)$ на веса w_i (нахождение функции принадлежности для каждой из выходных лингвистических переменных).
5. Дефазификация – это вычисление взвешенного среднего. Дефазификация определяется по формуле 1.22.

$$\bar{x} = \sum_{i=1}^n w_i * x_i \quad (1.22)$$

6. Вывод результата.

При построении нечеткого регулятора необходимо сформировать базу правил в форме ЕСЛИ (предпосылка), ТО (вывод) и базу данных с функциями принадлежности для предпосылок, то есть необходимо определить все лингвистические правила с лингвистическими переменными и термами. На рисунке 1.4 показан пример составления базы правил для управления температурным режимом в помещении.

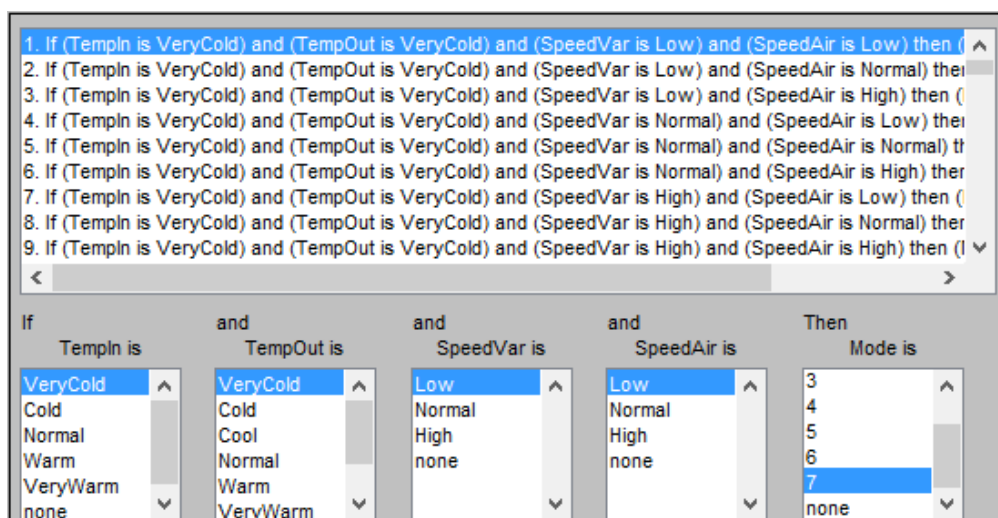


Рисунок 1.4 – База лингвистических правил нейронной сети

В качестве переменных вектора входных данных выбраны значения показателей:

- температура воздуха внутри помещения x_1 ;
- температура воздуха вне помещения x_2 ;
- скорость изменения температуры x_3 ;
- скорость потоков воздуха в комнате x_4 .

Составим вектор выходных данных:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Вектор идеальных значений показателей необходим для приведения текущего значения температуры воздуха в помещении к желаемому:

- желаемая температура воздуха внутри помещения x_{1_0} , от 18 до 22 °С;
- желаемая температура воздуха вне помещения x_{2_0} , которая варьируется в связи погодными условиями;
- желаемая скорость изменения температуры x_{3_0} , которая варьируется в зависимости от физических и химических процессов происходящих в помещении;
- желаемая скорость потоков воздуха в комнате x_{4_0} , которая варьируются от 0,3 до 0,5 м/с.

Составим вектор идеальных значений:

$$X = \begin{pmatrix} x_{1_0} \\ x_{2_0} \\ x_{3_0} \\ x_{4_0} \end{pmatrix}$$

В качестве выходного параметра используется режим работы кондиционера.

Вектор Δ – это рассогласование по входу/выходу системы. Это разность между уставными и имеющимися параметрами.

Вектор X' является выходным вектором для исполнительного устройства, который подается на датчик температуры.

Нечеткая нейронная сеть представлена следующими слоями.

Первый слой выполняет отдельную фаззификацию каждой переменной, определяя для каждого правила вывода значение коэффициента принадлежности $\mu_A^i(x_j)$ в соответствии с применяемой функцией фаззификации. Вторым слоем выполняется агрегирование функций принадлежности элементов вектора, определяя результирующее значение $w_i = \mu_A^i(x_j)$ для вектора x в соответствии с формулой 1.23.

$$\mu_A^i(x_j) = \prod_{j=1}^N \left[\frac{1}{1 + \left(\frac{x_j - c_j^{(i)}}{\sigma_j^{(i)}} \right)^{2b_j^{(i)}}} \right] \quad (1.23)$$

где $c_j^{(i)}, \sigma_j^{(i)}, b_j^{(i)}$ – параметры адаптации.

Третий слой представляет собой генератор функции Сугено-Такаги, рассчитывающий значения по формуле 1.24.

$$y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij}x_j \quad (1.24)$$

В четвертом слое происходит умножение сигналов $y_i(x)$ на значения w_i , сформированные во втором слое.

На последнем пятом слое происходит вывод значения нейронной сети.

Основным элементом Simulink-модели системы контроля температуры является блок *Fuzzy Logic Controller with Ruleviewer*. Данный блок содержит обученную нейронную сеть (рисунок 1.5) [4].

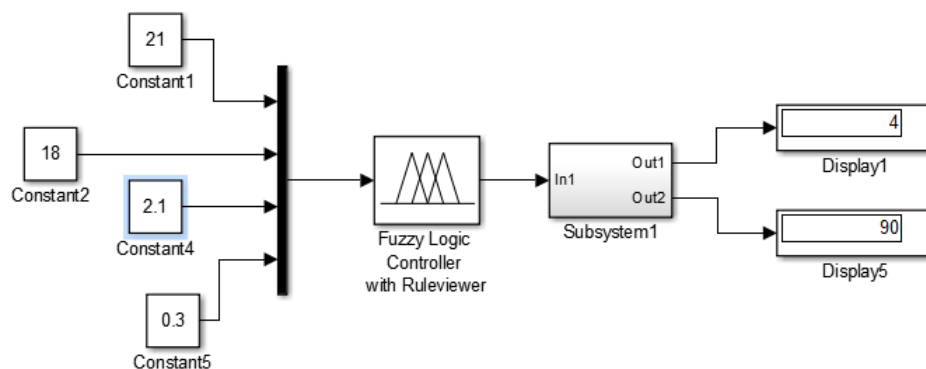


Рисунок 1.5 – Simulink-модель системы управления температурой воздуха в помещении

Нечеткие нейронные сети основаны на составление правил нечеткой логики. Из этого следует их недостаток – ограниченность использования. Это связано с тем, что база правил – это четкие указания зависимостей лингвистических и численных переменных, о которых нужно иметь априорное представление. В таких сетях существует четкая зависимость между входными параметрами и выходными.

1.4 Алгоритм прогнозирования температуры для интеллектуального термостата

В современных зданиях устанавливают различные системы отопления, вентиляции и кондиционирования. Обычно, каждая из установленных систем имеет собственную систему управления и не имеет связи с другими системами.

Например, в помещении установлены радиаторы водяного отопления, вентиляция и система кондиционирования. Радиаторы управляются термостатической головкой, имеющей свою заданную температуру. Система вентиляции имеет постоянную производительность и может быть включена или выключена. Система кондиционирования имеет свой терморегулятор со своей заданной температурой. Все вышеперечисленные системы, установленные в помещении, не имеют связи, и управление этими системами осуществляется независимо друг от друга.

Так как заданная температура термостатической головки и заданная температура терморегулятора кондиционера независимы, то в случае превышения заданной температуры термостатической головки над заданной температурой терморегулятора кондиционера отопление и кондиционирование будут работать друг против друга. При этом менее мощное устройство будет работать постоянно, а более мощное либо постоянно, либо прерывисто. Подобное терморегулирование вовсе неэффективно, так как необходимые затраты энергии складываются из энергии, необходимой для обогрева или охлаждения помещения и энергии, необходимой для противодействия систем отопления и кондиционирования.

Описанная выше ситуация характерна и для многих других сочетаний систем отопления, вентиляции и кондиционирования, таких как воздушное отопление, теплые полы, прямое электрическое отопление, кондиционеры с тепловыми насосами и другие. Вышеизложенная проблема связана с неэффективным управлением при совместной работе перечисленных систем.

Кроме того, различные системы, отапливающие или охлаждающие помещение, имеют различную производительность и удельную стоимость тепловой энергии, зависящие от внешних факторов. Так, например, при определенных соотношениях между температурой в помещении и температурой наружного воздуха, вентиляция оказывается энергетически более выгодной, чем кондиционирование.

Еще одним фактором оптимизации управления температурным режимом помещений является планирование режима работы и учет фактической необходимости поддержания температурного режима в соответствии с пребыванием людей в помещении.

В заключение вышесказанного, задачу оптимального управления температурным режимом помещений, можно сформулировать следующим образом: оптимальный регулятор должен обеспечивать заданное качество регулирования температуры, при минимальных энергетических затратах и минимальной общей стоимости потребленной энергии [2].

Для решения вышеописанных проблем с неэффективным управлением систем отопления, вентиляции и кондиционирования разработаны алгоритмы прогнозирования температуры для интеллектуального термостата.

Рассмотрим основные требования, относящиеся к самообучающимся системам управления температурой помещения с использованием интеллектуального термостата.

К одному из основных требований относится экономия затрат на электроэнергию. Классические термостаты позволяют устанавливать температуру до одного значения, которое должно быть изменено вручную, что не может быть названо эффективным использованием энергии. Назовем это нулевым уровнем интеллектуализации. Классические программируемые модели позволяют планировать, какая температура в помещении должна быть в определенные часы в течение дня, что позволяет снизить ее, когда в помещении нет людей, обеспечивая тем самым значительную экономию электроэнергии (первый уровень интеллектуализации). Например, термостат Siemens REA23 [19] работает в режиме самообучения, автоматически настраивая параметры адаптивного пропорционально-интегрально-дифференциального (ПИД) регулятора. Однако контроль температуры осуществляется в соответствии с расписанием, установленным пользователем, а не формируется автоматически на основе анализа поведения пользователя.

Интеллектуальный термостат автоматически формирует и динамически изменяет график работы в зависимости от активности и присутствия пользователей с учетом заданных пользователем приоритетов, что позволяет достичь наилучшей экономической эффективности (второй уровень интеллектуализации). Некоторые модели термостатов изучают предпочтения человека во времени, чтобы построить свой собственный график и адаптироваться к изменениям. Другие модели не предсказывают расписание на основе предыдущих действий, но автоматически реагируют на действия пользователя. Модели с автоматическим планированием используют привычки пользователя, его ответы на ряд простых вопросов и местные погодные условия.

Например, интеллектуальный термостат Nest Learning Thermostat адаптируется под алгоритм жизни человека и работает соответственно с его расписанием, графиком жизни и температурными предпочтениями, то есть фактически программирует себя в течение недели, создавая тем самым персонализированный график. Устройство постоянно адаптируется ко всем изменениям в расписании владельца, например, функция «Auto-Away» автоматически распознает, если дома никого нет, тогда термостат автоматически регулирует температуру, чтобы избежать нагрева пустого дома [17, 16]. В случае если необходимо определить объем экономии электроэнергии, то технология Nest Leaf позволит определить, насколько были эффективны настройки изменения температурного режима в помещении за определенный период [17].

Интеллектуальные термостаты марки Honeywell [15, 16], T9 [20], Ecobee SmartThermostat [21] и ряд других интеллектуальных термостатов имеют аналогичную функциональность.

Также интеллектуальный термостат должен иметь возможность интегрирования компонентов в систему умного дома. Некоторые модели термостатов имеют функцию подключения к другим устройствам экосистемы умного дома. Например, можно подключить термостат к умному замку, который активирует режим отсутствия людей и произведет запираение двери.

Формирование и отправка уведомлений, напоминаний, а также поддержка пользователей. Большинство термостатов имеют функцию оповещения различных типов, уведомляя владельца, например, когда пришло время провести базовое техническое обслуживание кондиционера или заменить воздушные фильтры в системе охлаждения. Также уведомление может предупредить о проблеме или сбое в системе отопления или охлаждения. Некоторые модели могут обеспечивать связь с представителем сервисного центра, который обслуживает термостат.

По результатам вышеизложенного, были рассмотрен подход построения алгоритма для работы и самообучения интеллектуального термостата.

Рассмотрим алгоритм моделирования, описанный в статье «Development of the temperature prediction algorithm for smart thermostat» [10].

Для прогнозирования температурных условий в помещении предполагает создание надежной тепловой модели с однозначно определяемыми параметрами. Модель должна, с одной стороны, достоверно описывать физические процессы теплообмена, а с другой – быстро адаптироваться к реальным условиям функционирования и изменению тепловых параметров помещения.

Рассмотрим формулу теплообмена для стандартного помещения с установленной системой отопления радиатором, кондиционером и теплым полом 1.17.

$$\frac{dt_v}{d\tau} = -\frac{L_0}{V} \cdot (t_v - t_0) - \frac{L_c}{V} \cdot (t_v - t_c) + \frac{\alpha_0 \cdot F_{ra}}{\rho \cdot c_r \cdot V} \cdot (t_v - t_0) + \frac{\alpha_{uf} \cdot F_{uf}}{\rho \cdot c_r \cdot V} \cdot (t_{uf} - t_v) + \frac{\alpha_r \cdot F_r}{\rho \cdot c_r \cdot V} \cdot (t_r - t_v) \quad (1.1.7)$$

где α_0 – коэффициент теплоотдачи помещения,

α_{uf} – коэффициент теплого пола;

α_r – коэффициент радиатора отопления;

t_0 – температура наружного воздуха;

t_v – температура воздуха в помещении;

L_0 – приток наружного воздуха;

L_c – приток воздуха от кондиционера;

ρ – плотность воздуха;

c_r – теплоемкость вещества при постоянном давлении;

V – объем помещения;

t_c – температура воздуха кондиционера;

t_r – температура радиатора отопления;

t_{uf} – температура теплого пола;

F_{ra}, F_{uf}, F_r – площадь поверхности помещения, теплого пола и радиатора отопления.

В формуле коэффициент теплопередачи α_0 является неизвестным значением, который выбирается таким образом, чтобы решение по формуле совпадало с экспериментальными значениями температуры воздуха внутри помещения.

Рассмотрим формулу расчета температуры радиатора отопления. По своей форме тепловая модель для радиатора аналогична модели температуры воздуха в помещении, но с меньшим числом слагаемых 1.18.

$$\frac{dt_r}{d\tau} = \frac{L_{hw}}{V_r} \cdot (t_{hw} - t_r) + \frac{\alpha_{r0} \cdot F_r}{\rho_w \cdot c_{rw} \cdot V_r} \cdot (t_v - t_r) \quad (1.18)$$

где L_{hw} – расход горячей воды в радиаторе;

V_r – объем отопительного радиатора;

t_{hw}, t_r – температура горячей воды в системе отопления и радиаторе отопления соответственно;

ρ_w, c_{rw} – плотность и теплоемкость при постоянном давлении воды в радиаторе отопления;

α_{r0} – коэффициент теплопередачи радиатора отопления;

F_r – площадь внутренней поверхности радиатора отопления.

В формуле параметры $L_{hw}, \rho_w, c_{rw}, \alpha_{r0}, F_r$ являются неизвестными значениями, которые должны быть выбраны таким образом, чтобы решение совпадало с экспериментальными значениями температуры радиатора отопления.

Определим параметры модели. Для определения коэффициент теплопередачи α_0 составим функционал, в котором рассчитывается сумма квадратов отклонений экспериментальной температуры воздуха в помещении (t_{vi}^{exp}) от расчетных значений температуры ($t_{vi}^{mod}(\alpha_0)$) по формуле 1.19.

$$M(\alpha_0) = \sum_{i=n}^{n-4} (t_{vi}^{exp} - t_{vi}^{mod}(\alpha_0))^2 \quad (1.19)$$

В формуле сумма вычисляется по последним 5 значениям температуры. Для определения α_0 , использовался одномерный метод минимизации – метод золотого сечения [13]. Аналогичный подход использовался и для определения неизвестных параметров в формуле для определения температуры радиатора отопления.

2. АНАЛИЗ СИСТЕМЫ УПРАВЛЕНИЯ ТЕМПЕРАТУРОЙ В ПОМЕЩЕНИИ В MATLAB-SIMULINK

В данной работе была спроектирована имитационная модель автоматической системы управления температуры внутри помещения. В данной модели предусматривается наличие подсистем, которые моделируют температуру окружающей среды, термических характеристик помещения, здания и моделирование системы отопления и кондиционирования. В качестве среды проектирования и моделирования системы используется программная среда визуального программирования Matlab Simulink.

Термодинамическая модель дома позволяет моделировать изменение температуры внутри жилого здания в результате влияния температуры окружающей среды, работы системы отопления и кондиционирования под управлением регулятора. Модель базируется на дифференциальных уравнениях термодинамических процессов, протекающих внутри здания.

Рассмотрим модернизированную тепловую модель дома, построенную на основе библиотечной модели Matlab Simulink, рисунок 2.1.

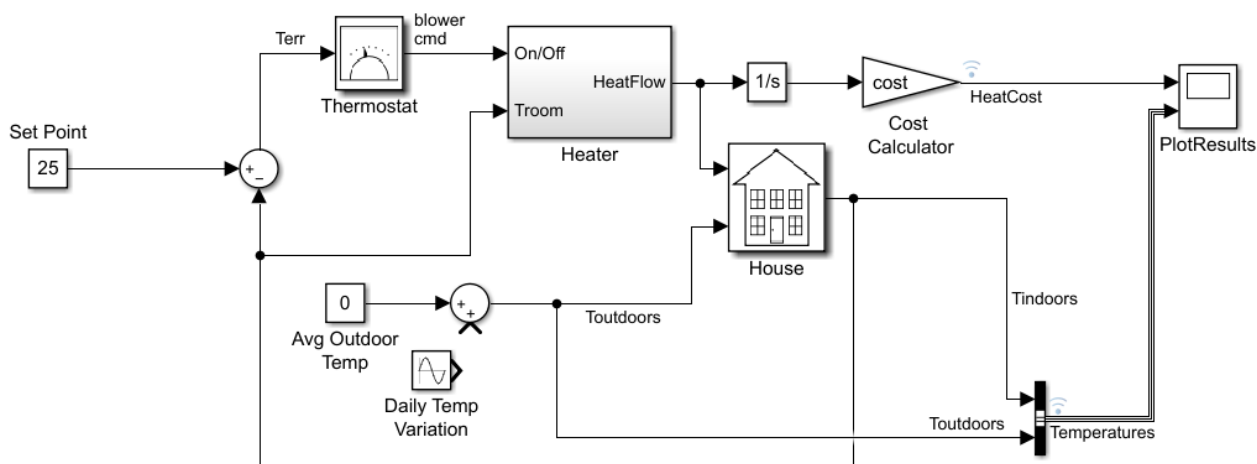


Рисунок 2.1 – Тепловая модель дома

Модель рассчитывает расходы на отопление некоего типового дома. При инициализации модели выполняются функции:

- определения геометрии дома (размер, количество окон);
- определение тепловых свойств материала дома;

- расчета теплового сопротивления дома;
- определение стоимости электроэнергии;
- установка начальной комнатной температуры.

Рассмотрим подробнее, из каких компонентов состоит модель системы управления:

1. Set Point (установленное значение) – блок указывает температуру, которую необходимо поддерживать в помещении. По умолчанию установлено значение 25°C.
2. Thermostat (Термостат) – это подсистема, которая содержит блок реле. Термостат допускает колебания на 0,5°C выше или ниже желаемой комнатной температуры (рисунок 2.2). Если температура воздуха опускается ниже 24,5°C, то термостат включает нагреватель.

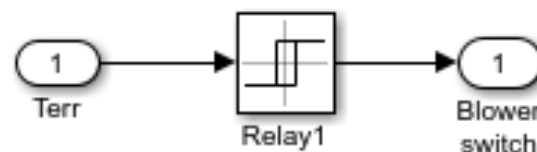


Рисунок 2.2 – Подсистема термостата

3. Heater (Нагреватель) – это подсистема с постоянной скоростью воздушного потока «Mdot», которая определяется в загрузочном файле `sldemo_househeat_data.m` (рисунок 2.3). Сигнал термостата включает или выключает нагреватель. Когда нагреватель включен, он нагнетает горячий воздух при температуре «THeater» с постоянной скоростью потока «Mdot» (1 кг / сек = 3600 кг / час по умолчанию). Тепловой поток выражается уравнением, приведенным в формуле 2.1.

$$\frac{dQ}{dt} = (T_{heater} - T_{room}) \cdot M_{dot} \cdot c \quad (2.1)$$

где $\frac{dQ}{dt}$ – тепловой поток;

T_{heater} – температура горячего воздуха от нагревателя;
 T_{room} – текущая температура воздуха в помещении;
 M_{dot} – поток воздуха проходящий через нагреватель (кг / час);
 c – теплоемкость воздуха при постоянном давлении.

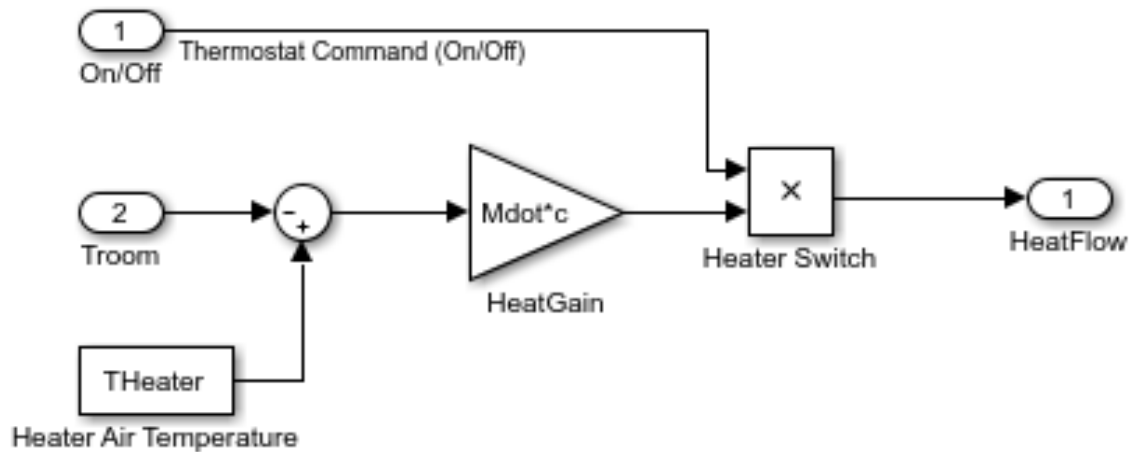


Рисунок 2.3 – Подсистема нагревателя

4. Cost Calculator (калькулятор стоимости) – данный блок интегрирует тепловой поток с течением времени и умножает его на стоимость энергии. Стоимость энергии за отопления отображается на графике «PlotResults».
5. House (типовой дом) – это подсистема, которая вычисляет колебания температуры в помещении (рисунок 2.4). Она учитывает тепловой поток от нагревателя и тепловые потери в окружающей среде. Тепловые потери выражаются уравнением (2.2, 2.3):

$$\left(\frac{dQ}{dt}\right) = \frac{T_{room} - T_{out}}{R_{eq}} \quad (2.2)$$

$$\frac{dT_{room}}{dt} = \frac{1}{M_{air} * c} * \left(\frac{dQ_{heater}}{dt} - \frac{dQ_{losses}}{dt}\right) \quad (2.3)$$

где M_{air} – масса воздуха внутри дома;

R_{eq} – эквивалентное тепловое сопротивление дома.

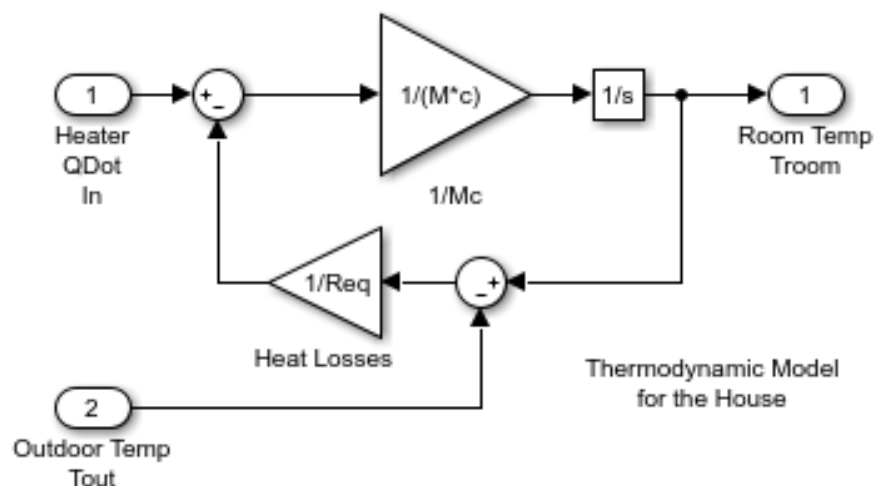


Рисунок 2.4 – Подсистема дома

Моделирование окружающей среды. В системе окружающая среда моделируется как теплоотвод с бесконечной теплоемкостью и изменяющейся во времени температурой «Tout». Постоянный блок «Avg Outdoor Temp» определяет среднюю температуру воздуха на улице.

Смоделируем тепловую модель дома при следующих параметрах:

- средняя температура воздуха на улице 0°C;
- установленная температура в здании 20°C;
- установленное значение для поддержания температуры в помещении 25°C;
- стоимость электроэнергии 4,28 руб./кВтч;
- размер помещения 30·10·4 м³, 6 окон 1·1 м²

Рассмотрим в тепловой модели дома нагрев помещения при различных регуляторах релейный регулятор, П-, ПИ- и ПИД регулятор и фаззи регулятор.

В модели термостата установлен релейный регулятор. При заданных параметрах температура в здании устанавливается за 0,23 часа, затем термостат поддерживает двухпозиционное регулирование температуры от 24,5°C до 25,5 °C (рисунок 2.5).

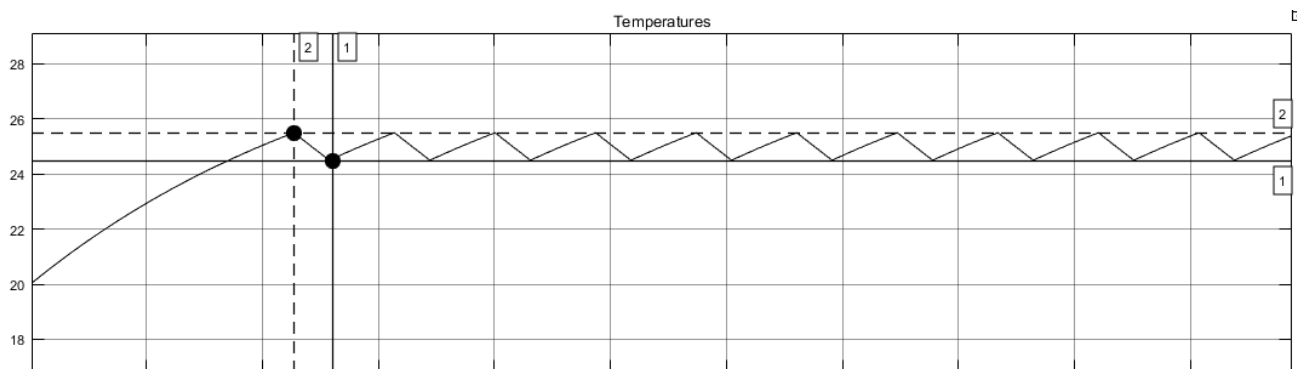


Рисунок 2.5 – Нагрев помещения с релейным регулятором

При уменьшении уставки в релейном регуляторе, например, с 0,5 до 0,1, термостат поддерживает более точную температуру, но чаще происходит включение и выключение нагревателя (рисунок 2.6).

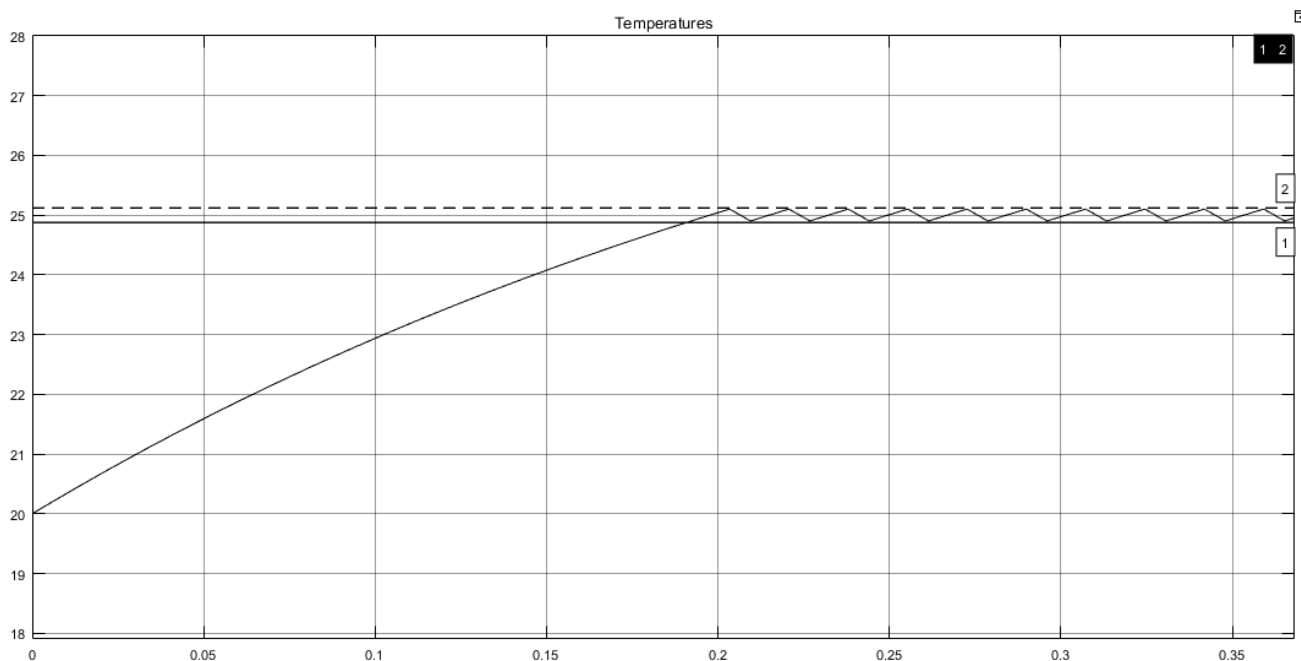


Рисунок 2.6 – Нагрев помещения до заданной температуры с уставным значением $\pm 0,1$

При изменении температуры на улице от -17°C и более, нагревателю недостаточно мощности для нагрева помещения с заданными параметрами, температура в помещении достигает до $22,5^{\circ}\text{C}$ – $23,5^{\circ}\text{C}$. Для достижения результата необходимо улучшить параметры помещения, например, уменьшить количество окон, увеличить толщину стен (рисунок 2.7).

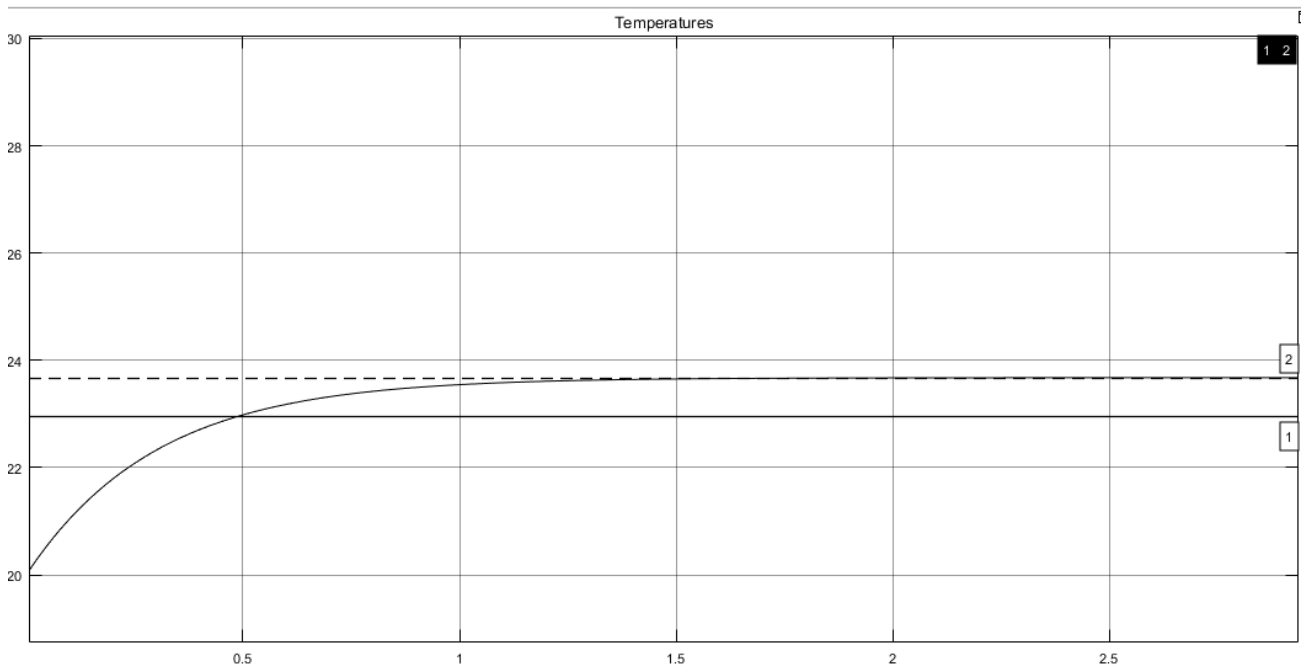


Рисунок 2.7 – Нагрев помещения при температуре на улице -17°C

В модели термостата установлен П-регулятор. Выполним настройку блока. Установим коэффициент регулирования 100 и установим пределы от -1 до 1 (рисунок 2.8).

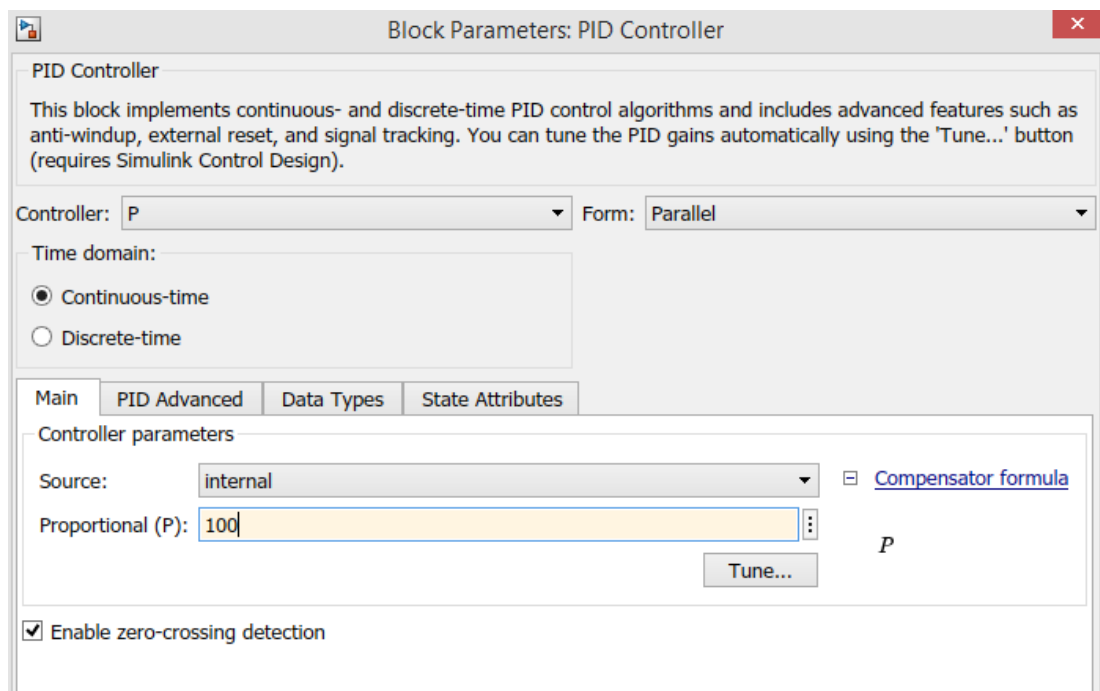


Рисунок 2.8 – Настройка П-регулятора

При заданных параметрах температура в здании устанавливается за 0,21 часа, затем термостат поддерживает заданное значение 25°C (рисунок 2.9).

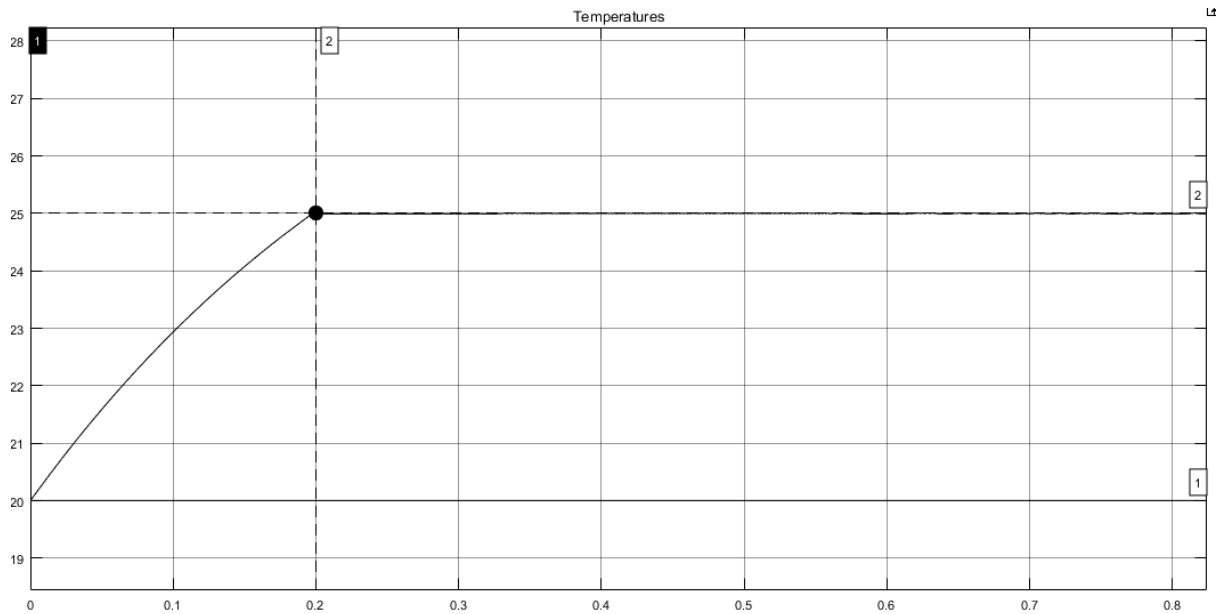


Рисунок 2.9 – Нагрев помещения до заданной температуры с П-регулятором

С установленным в термостате П-регулятором нагрев помещения происходит за более короткий период, чем с релейным регулятором. Чем точнее будет выбран коэффициент регулирования в П-регуляторе, тем ближе будет установившееся значение к заданному с малой погрешностью. Так при низком коэффициенте регулирования, например, 1 установившееся значение достигает до 24,4 °С (рисунок 2.10). Таким образом, при маленьком коэффициенте регулирования увеличивается статическая ошибка.

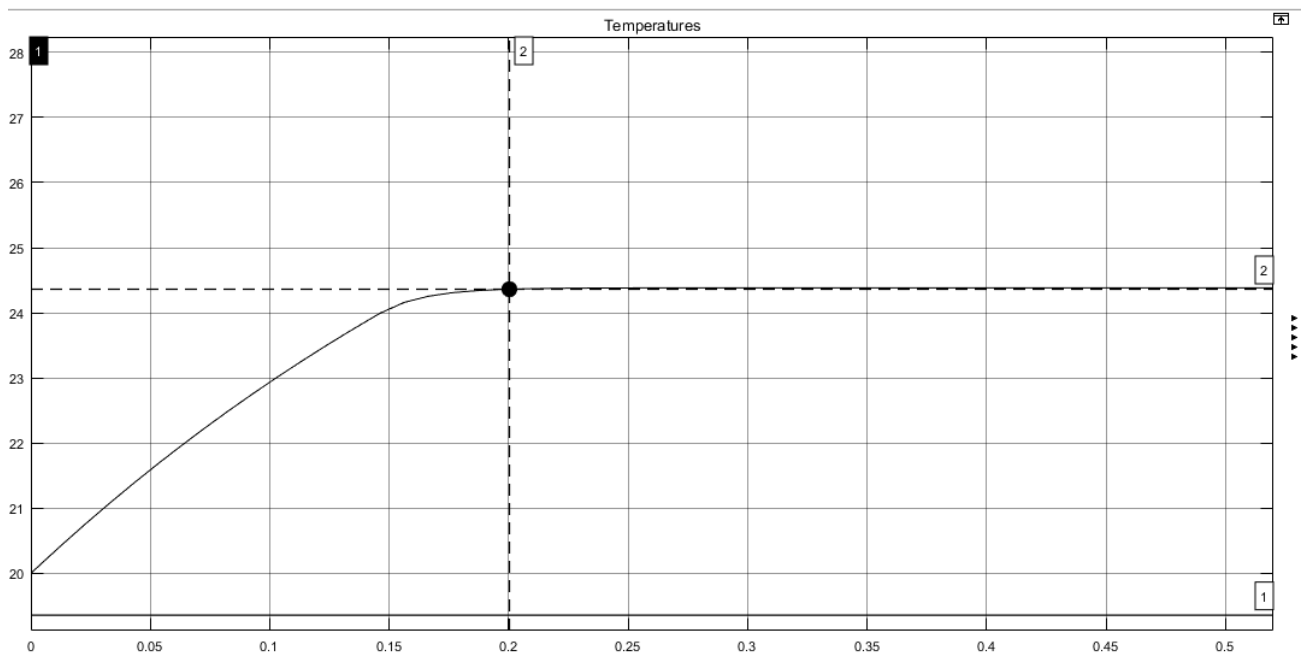


Рисунок 2.10 – Нагрев помещения до заданной температуры с коэффициентом регулирования 1

В модели термостата установлен ПИ-регулятор. Выполним автоматическую настройку блока. Установим время регулирования 0.07273 и степень устойчивости 0.838 (рисунок 2.11).

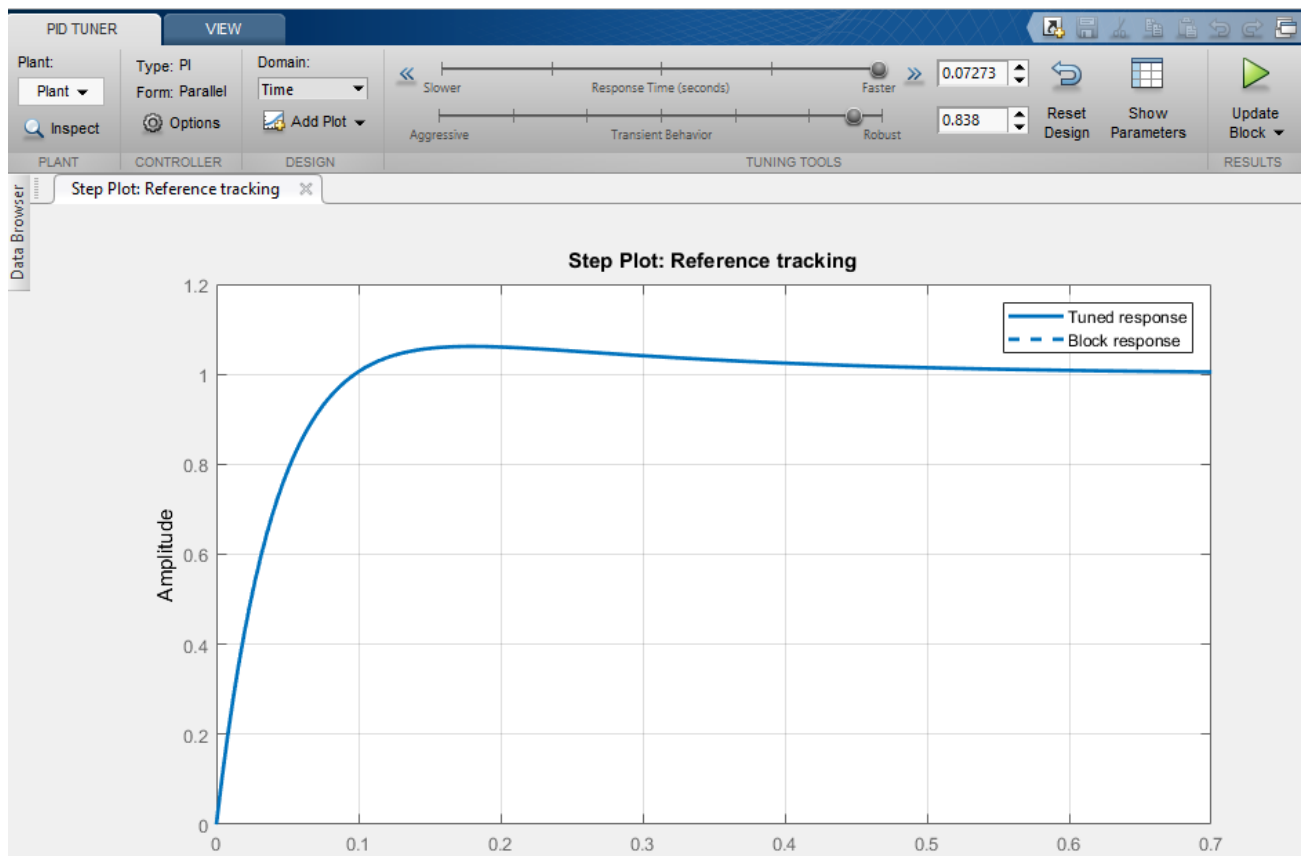


Рисунок 2.11 – Настройка ПИ-регулятора

С установленным в термостате ПИ-регулятором нагрев помещения происходит за более длительный период – 0,286 часа, чем с установленным релейным регулятором или П-регулятором. Однако установившееся значение равно заданному значению и равно 25 °С (рисунок 2.12).

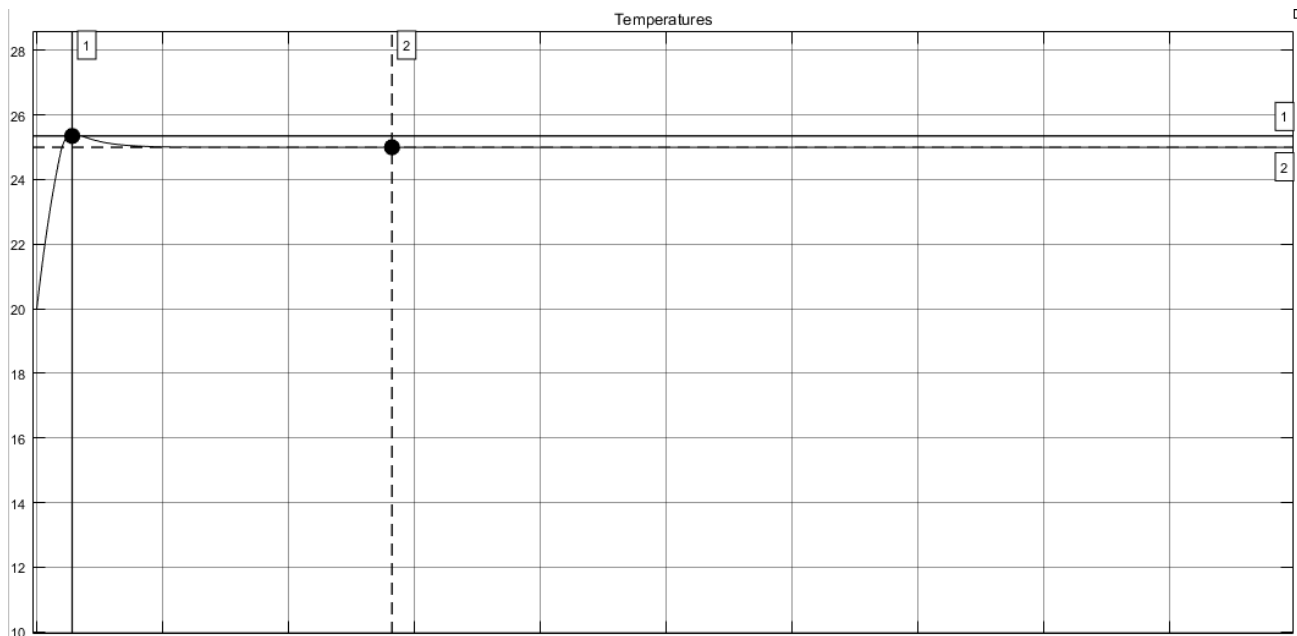


Рисунок 2.12 – Нагрев помещения до заданной температуры с ПИ-регулятором

Определим максимальное отклонение управляемой величины от заданного значения и выражается в процентах – перерегулирование равно 1,4%.

В модели термостата установлен ПИД-регулятор. Выполним автоматическую настройку блока. Установим время регулирования 0,1341 и степень устойчивости 0,5 (рисунок 2.13).

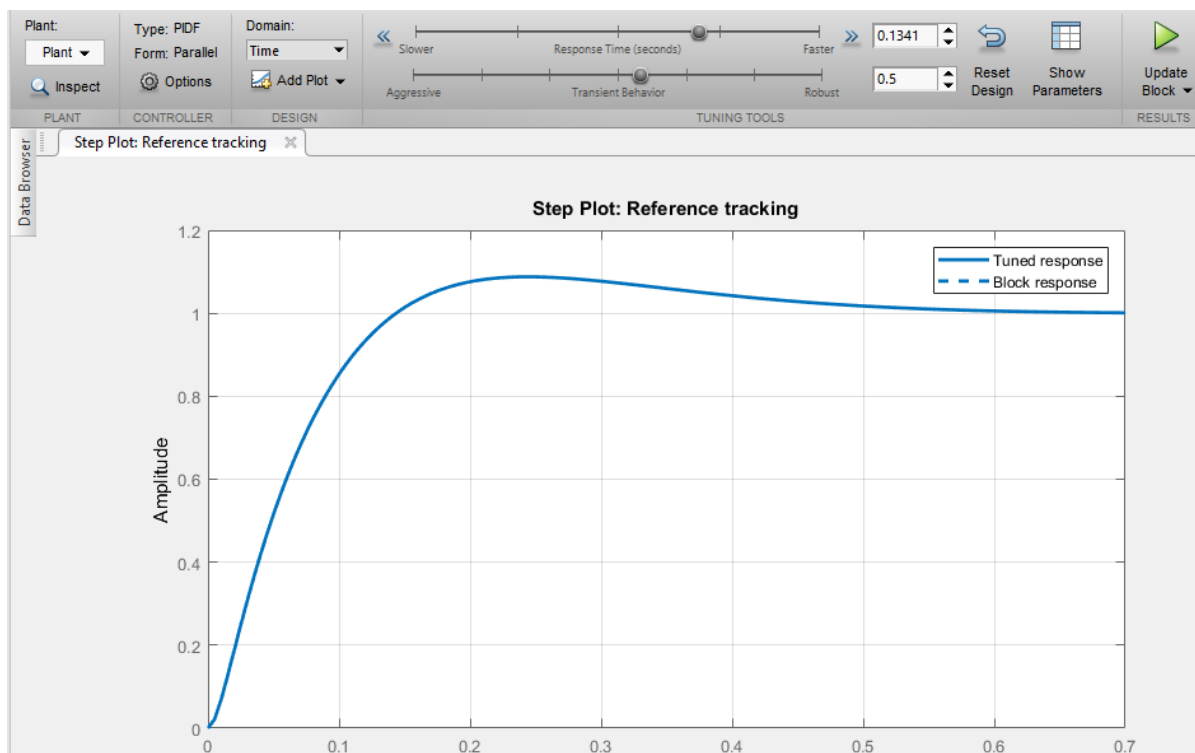


Рисунок 2.13 – Настройка ПИД-регулятора

С установленным в термостате ПИД-регулятором нагрев помещения происходит за период 0,275 часа и установившееся значение равно заданному значению 25°C, как и с ПИ-регулятором (рисунок 2.14). Максимальное отклонение управляемой величины от заданного значения равно 1,3%.

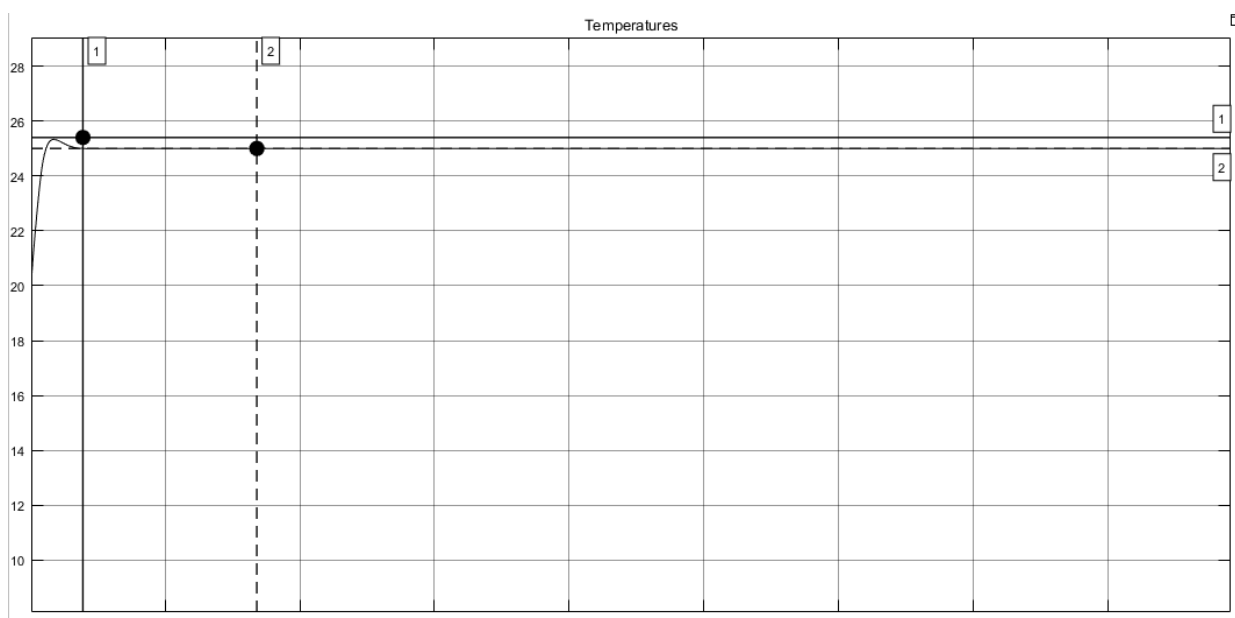


Рисунок 2.14 – Нагрев помещения до заданной температуры с ПИД-регулятором

ПИД-регулятор обеспечивает большее быстродействие, чем ПИ-регулятор.

В модели термостата установлен нечеткий регулятор. Выполним настройку блока.

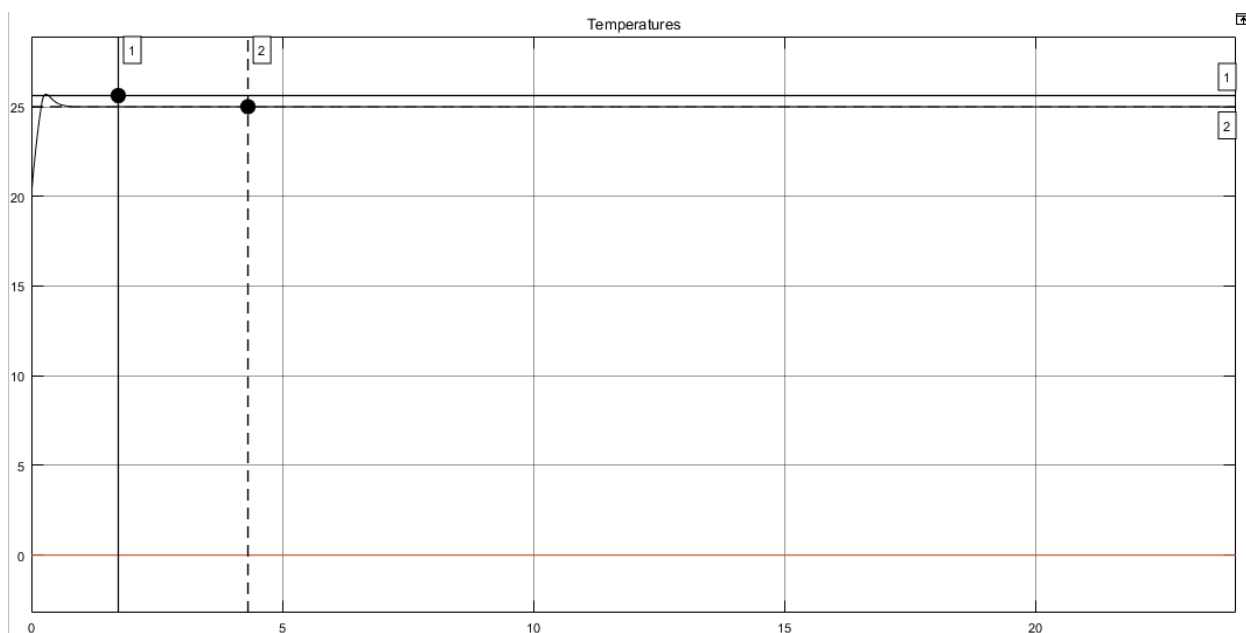


Рисунок 2.14 – Нагрев помещения до заданной температуры нечетким регулятором

С установленным нечетким регулятором нагрев помещения происходит за период 0,279 часа и установившееся значение равно заданному значению 25°C. Максимальное отклонение управляемой величины от заданного значения равно 0,5%.

Рассмотрим сравнительную таблицу №2 применения различных регуляторов в схеме управления температурным режимом в помещении.

Таблица 2 – Сравнительная таблица показателей регуляторов

№	Регулятор	Точность поддержания температуры, %	Время регулирования, час	Перерегулирование, %
1	Релейный регулятор	98% - 102%	0,230	0%
2	П- регулятор	97,6%	0,210	0%
3	ПИ- регулятор	100%	0,286	1,4%
4	ПИД- регулятор	100%	0,275	1,3%
5	Нечеткий регулятор	100%	0.279	0,5%

В рассмотренной имитационной модели автоматической системы управления температуры внутри помещения с заданными параметрами при помощи системы Matlab Simulink, проведен анализ изменения параметров в тепловой модели дома при различных регуляторах: релейный регулятор, П-, ПИ-, ПИД- и нечеткий регулятор. Наилучшие параметры системы поддерживаются при ПИД- и нечетком регуляторе.

В данной системе регулирования нечеткий регулятор не дает явного преимущества по сравнению с ПИД-регулятором. Он обеспечивает несколько меньшее перерегулирование, но при этом несколько большее время регулирования.

Проведенный анализ по имитационной модели автоматической системы управления температуры внутри помещения, позволит в дальнейшем разработать алгоритмы работы и самообучения интеллектуального термостата, а именно разработать интеллектуальный алгоритм обучения термостата на основе анализа тепловых параметров здания и поведения пользователя;

3. РАЗРАБОТКА АЛГОРИТМА ПРОГНОЗИРОВАНИЯ ТЕМПЕРАТУРЫ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО ТЕРМОСТАТА

В данной главе представлено описание алгоритма прогнозирования температуры для интеллектуального термостата, описана реализация аппаратной и программной частей технического решения.

3.1 Исходные параметры для моделирования

Для моделирования алгоритма прогнозирования температуры рассмотрим помещение размерами: ширина 3 метра, длина 6 метров, высота 2,5 метра, с установленным окном их ПВХ пластика с двухкамерным стеклопакетом размером 1,5 · 1,3 метра.

Ограждающая конструкция представляет собой кирпичную стену толщиной 640 мм с внутренней известковой штукатуркой. Толщина штукатурки 15 мм. Коэффициенты теплопроводности: для внутренней известковой штукатурки $\lambda = 0,6$ ккал/м час град; для кирпичной кладки на тяжелом растворе $\lambda = 0,7$ ккал/м час град.

Сопротивление теплопереходу R_B для внутренних поверхностей стен, полов, а также потолков, имеющих гладкую поверхность составляет $0,133 \text{ м}^2 \text{ час град/ккал}$. Сопротивление теплопереходу для поверхностей, соприкасающихся непосредственно с наружным воздухом, составляет $0,05 \text{ м}^2 \text{ час град/ккал}$

Удельная теплоемкость воздуха $c_B = 1,005 \text{ кДж/кгК}$. Плотность воздуха при температуре $20 \text{ }^\circ\text{C}$: $\rho_B = 1,2041 \text{ кг/м}^3$.

Коэффициент конвективной теплоотдачи воздуха при естественной конвекции $\alpha_B = 5 + 25 \text{ Вт/м}^2$.

Плотность жидкости при температуре 95°C : $\rho_{\text{ж}} = 962 \text{ кг/м}^3$. Удельная теплоемкость жидкости $c_{\text{ж}} = 4,22 \text{ кДж/кгК}$.

Для обогрева помещения установлен чугунный радиатор МС-140 состоящий из 10 секций. Диаметр ниппеля равен 1, 25 дюйма, т.е. 0,03175 м. Согласно СТО НП «АВОК» 4.2.2-2006 «Радиаторы и конвекторы отопительные. Общие

технические условия» толщина стенки литого чугунного радиатора, соприкасающейся с водой, должна быть не менее 2,7 мм. Толщину стенки радиатора примем равной 4 мм. Поверхность нагрева одной секции для данного типа радиатора составляет 0,254 м.

3.2 Определение параметров сбора данных для интеллектуальной системы управления температурой в помещении

Для работы интеллектуальной системы управления температурой в помещении необходимо получать и обрабатывать данные, передаваемые с установленных датчиков.

Необходимо получать такие параметры как:

- температура наружного воздуха, t_n °С;
- температура воздуха в помещении, t_b °С;
- температура горячей воды в системе отопления и радиаторе отопления соответственно, t_p °С;
- состояние термоголовки, установленной на радиаторе отопления;
- заданная уставная температура в помещении, $t_{уст}$ °С;

Датчик для измерения наружной температуры должен быть установлен таким образом, чтобы он был защищен от ветра, влаги и нагрева от солнечных лучей.

Датчик для измерения воздуха в помещении также необходимо устанавливать в местах, где отсутствуют технологические процессы с выделением тепла и не вблизи приборов обогрева или охлаждения.

Датчик для определения температуры радиатора отопления устанавливается непосредственно на одну из первых секций радиатора.

3.3 Инструменты для реализации системы

Для моделирования интеллектуальной системы управления температурой в помещении был спроектирован тестовый стенд, состоящий из такого оборудования как:

- интегральный датчик температуры DS18B20;
- термостат, реализованный на Arduino;
- термоголовка для радиатора отопления с сервоприводом;
- Wi-Fi модуль;
- сеть интернет.

Интегральный датчик температуры DS18B20 – это цифровой измеритель температуры, с разрешением преобразования от 9 до 12 разрядов и функцией контроля температурой с тревожным сигналом (рисунок 3.1). Параметры контроля могут задаваться пользователем и сохраняться в энергонезависимой памяти датчика.

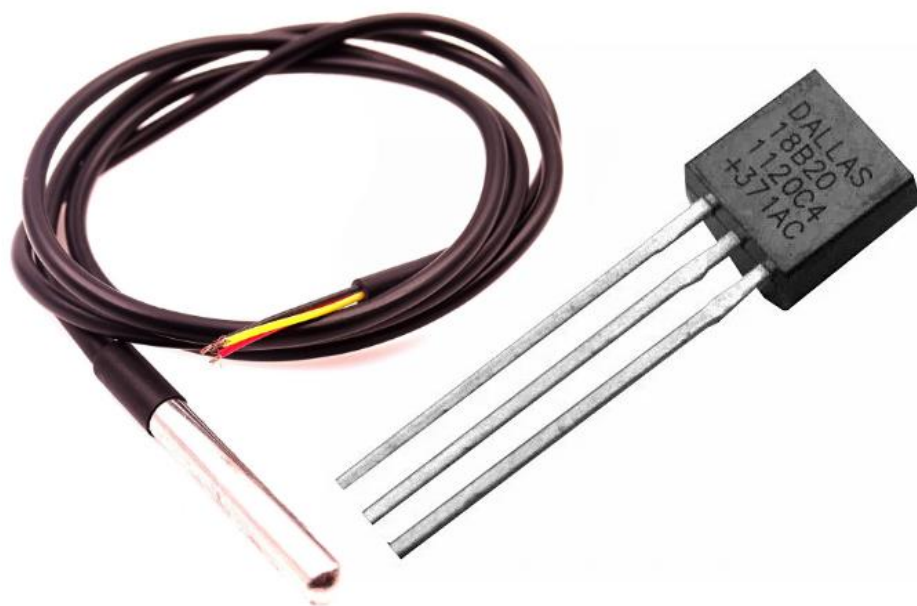


Рисунок 3.1 – Интегральный датчик температуры DS18B20

DS18B20 обменивается данными с микроконтроллером по однопроводной линии связи, используя протокол интерфейса 1-Wire.

Диапазон измерения температуры составляет от -55 до +125 °С. Для диапазона от -10 до +85 °С погрешность не превышает 0,5 °С.

У каждой микросхемы датчика DS18B20 есть уникальный серийный код длиной 64 разряда, который позволяет нескольким датчикам подключаться на одну общую линию связи. Таким образом, через один порт микроконтроллера можно обмениваться данными с несколькими датчиками, распределенными на значительном расстоянии [11].

На рисунке 3.2 представлена схема подключения датчика:

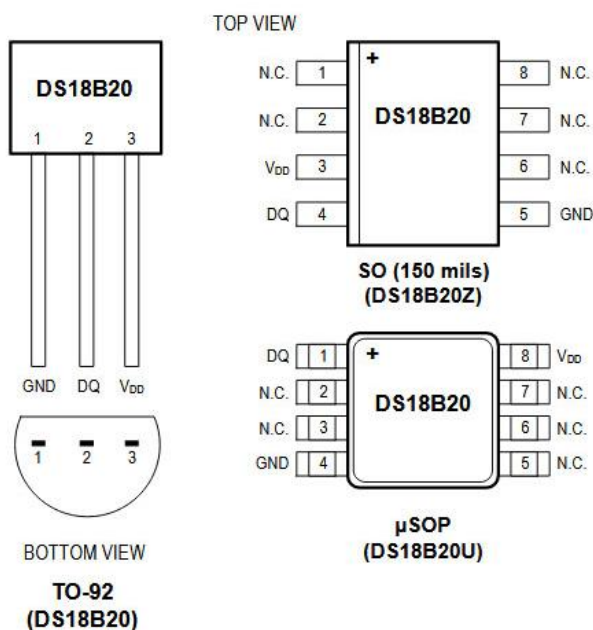


Рисунок 3.2 – Схема подключения датчика DS18B20

Сигнал GND (черный) – земля. Сигнал DQ (желтый, белый или синий) является выводящим сигналом данных (вход/выход), выход типа открытый коллектор интерфейса 1-Wire. Также через сигнал DQ происходит питание за счет подключенного адаптера. Сигнал V_{DD} (красный) предназначен для вывода внешнего питания. Если питание идет от подключенного оборудования, то данный сигнал должен быть подключен к земле.

Для моделирования системы используется термостат, реализованный на платформе Arduino.

Для вывода температуры с датчика DS18b20, была использована плата NodeMCU на основе модуля ESP8266 с интерфейсом передачи данных Wi-Fi.

Плата NodeMCU имеет следующие технические параметры:

- одиннадцать цифровых входов и выходов;

- один аналоговый вход с максимальным входным напряжением 3,3В;
- micro USB разъем;
- поддержка Arduino IDE;
- flash память 4 Мб;
- беспроводная загрузка программного кода по интерфейсу Wi-Fi.

На рисунке 3.3 представлена плата NodeMCU на основе модуля ESP8266.

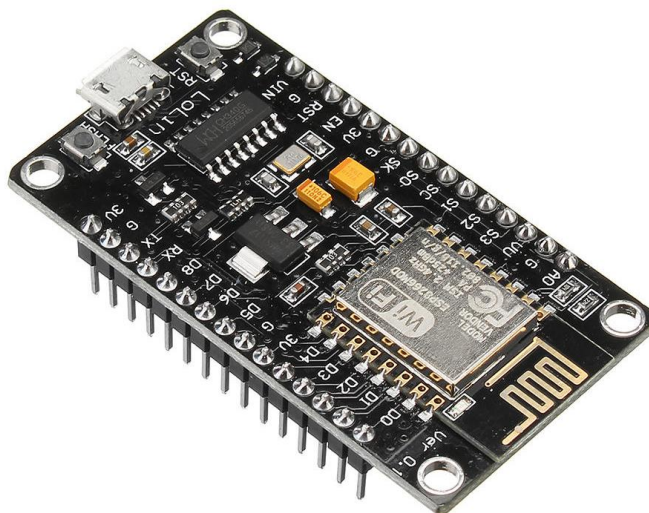


Рисунок 3.3 – NodeMCU на основе модуля ESP8266

Модуль ESP8266 является популярным инструментом для организации беспроводной связи в проектах «Интернета вещей». Модуль ESP8266 имеет следующие параметры:

- напряжение питания 3.3В;
- сила тока в режиме передачи до 215 мА;
- сила тока во время приема до 62 мА;
- протокол 802.11 b/g/n;
- режимы сохранения энергии и сна;
- встроенный микроконтроллер;
- температура функционирования от -40°C до +125°C;
- максимальная дистанция связи до 100 м.

Главное преимущество модуля ESP8266 в сравнительно низкой стоимости.

Для создания беспроводной сети в пределах помещения используется чип ESP8266. С его помощью организуется связь по Wi-Fi, осуществляются сбор информации, дистанционное управление и выход в интернет.

Все данные, поступающие с датчиков, будут храниться в реляционной базе данных.

Электротермический двухпозиционный сервопривод применяется для автоматического управления радиаторным термостатическим клапаном систем водяного отопления и охлаждения помещения (рисунок 3.4).

Действие привода основано на расширении заполняющего сиффон армированного парафина, при протекании электрического тока через встроенный нагревательный элемент по сигналу от комнатного термостата или контроллера.



Рисунок 3.4 – Термоголовка радиатора отопления

Для тестового стенда был выбран электротермический двухпозиционный сервопривод VT.TE3043.

Исходя из функциональной модели, была спроектирована архитектура проекта, которая представлена на рисунке 3.5.

Устройство сбора и передачи данных (термостат) подключается к порту датчиков, которые установлены в помещении, с наружи помещения и на

радиаторе отопления. Термостат передает данные на веб-сервер по средствам связи Wi-Fi.

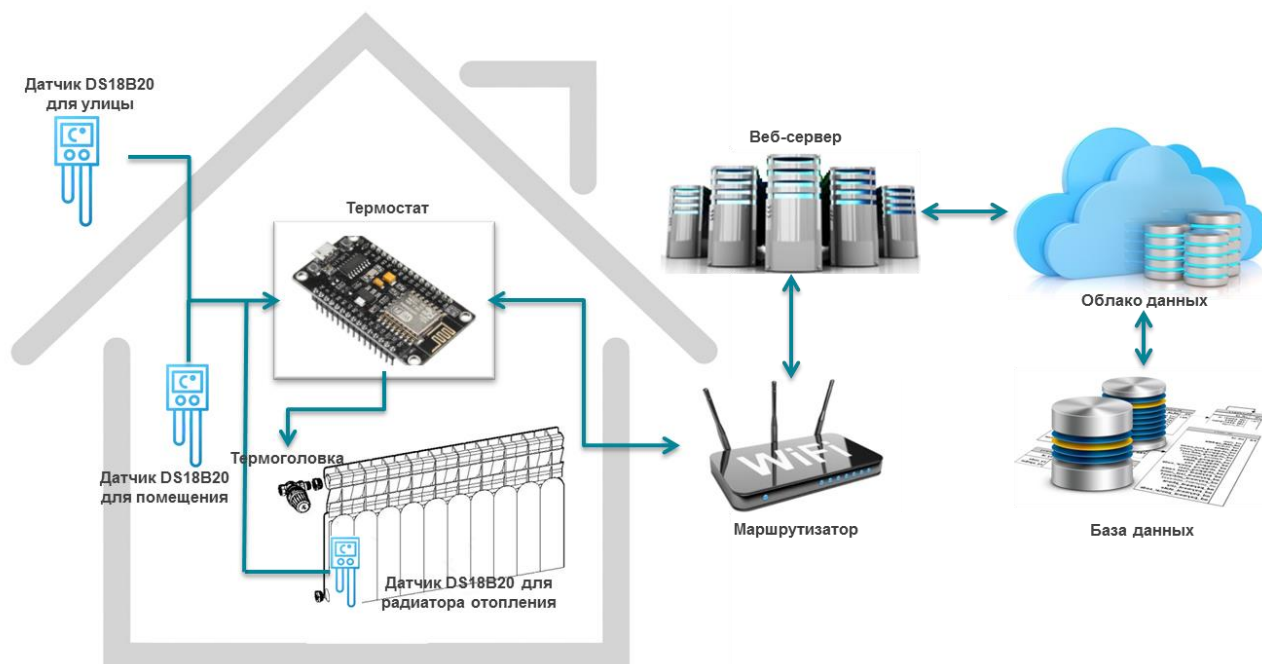


Рисунок 3.5 – Структурная схема проекта

Модуль ESP8266 подключается к данному порту и преобразует импульсы, передаваемые с датчиков, в значения температуры. Данный модуль оснащен Wi-Fi передатчиком, который подключается к маршрутизатору по средствам беспроводной сети.

Роутер имеет доступ к сети интернет и перенаправляет данные от модуля ESP8266 на сервер по протоколу передачи данных HTTP. При отключении сети интернет, все данные хранятся во встроенной памяти устройства, до восстановления сети интернет. После восстановления, все накопленные данные передаются на сервер.

На сервере развернуто веб-приложение, которое принимает данные, обрабатывает их, и устанавливает соответствие между установившейся температурой в помещении и уставной температурой, затем по разработанному алгоритму в помещении корректируется температура воздуха до уставной температуры в определенный момент к заданному времени. По сигналу от термостата регулируется состояние термостатической головки для регулирования температуры радиатора.

В качестве веб-сервера для развертывания сервиса были рассмотрены Apache и Nginx. Основное различие данных веб-серверов заключается в том, как обрабатываются соединения. Apache использует разветвленное многопоточное решение, которое поддерживает соединение для каждого пользователя. Nginx использует неблокирующий цикл событий, который объединяет соединения, работающие асинхронно через рабочие процессы. Таким образом, был выбран веб-сервер Nginx, так как для обработки каждого нового соединения не создаются дополнительные процессы, и даже при высокой нагрузке, CPU и оперативная память не очень сильно расходуются при таком подходе.

Для реализации базы данных была выбрана свободная реляционная система управления базами данных MySQL.

Для реализации веб-сервиса был выбран язык программирования Python, так как он имеет открытый исходный код и распространяется по PSFL лицензии.

При разработке серверной части применяется фреймворк Flask.

Flask – это бесплатный фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблоны Jinja2.

3.4 Реализации аппаратной части

Для реализации аппаратной части необходимо следующее:

- плата NodeMCU с встроенным модулем ESP8266;
- 3 штуки интегральных датчиков температуры DS18B20;
- приложение Arduino IDE.

Необходимо подключить датчик температуры к плате следующим образом, как показано на рисунке 3.6.

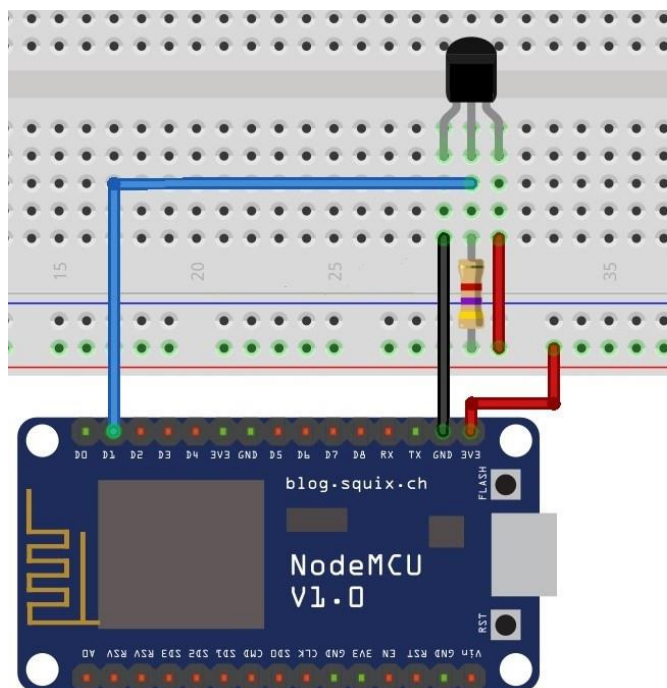


Рисунок 3.6 – Схема подключения датчика температуры к плате

После того, как подключены датчики температуры к плате – необходимо настроить оборудование. Для этого необходимо скачать приложение Arduino IDE версии 1.8.0 или выше, и установим его на персональный компьютер (рисунок 3.7).

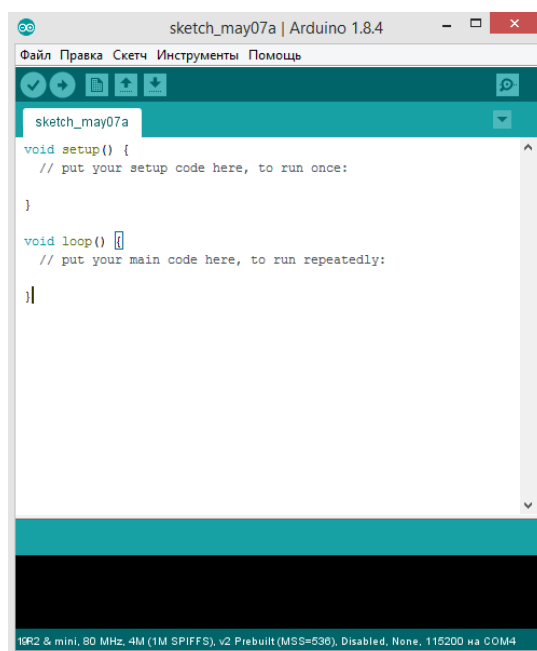


Рисунок 3.7 – Главное окно приложения Arduino IDE

Далее необходимо подключить плату к USB-порту компьютера при помощи micro USB кабеля и скачать драйвер для приложения Arduino, затем установить его (рисунок 3.8).



Рисунок 3.8 – Установка драйвера

После того как установлено приложение и необходимые драйвера, нужно перейти в раздел «Настройки». В строке «Дополнительная ссылка для Менеджера плат» ввести URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json.

Затем, как показано на рисунке 3.9, нажать на кнопку рядом с полем ввода URL для того, что бы убедиться, что адрес был добавлен и завершить настройку, нажав кнопку «Ок».

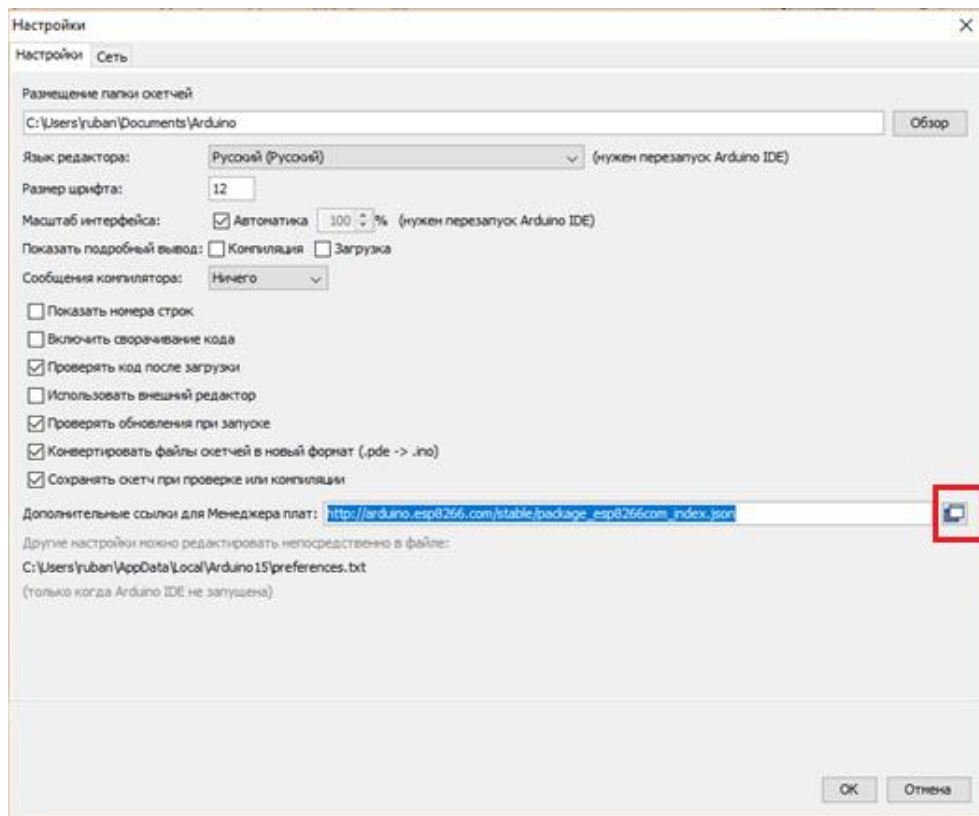


Рисунок 3.9 – Настройка приложения Arduino

Далее необходимо перейти во вкладку «Инструменты» и выбрать «Менеджер плат» (рисунок 3.10).

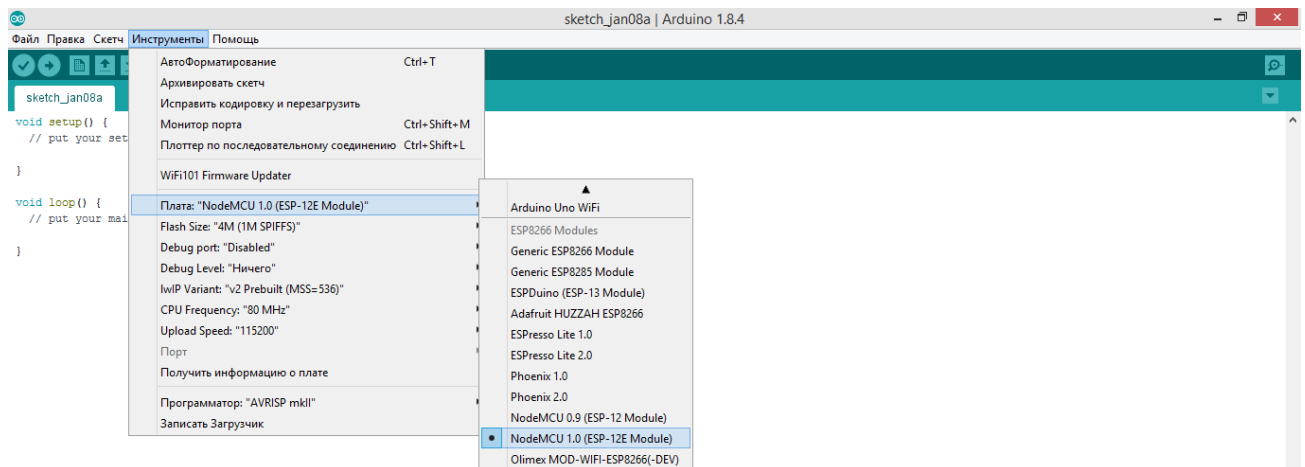


Рисунок 3.10 – Переход в раздел «Менеджер плат»

В открывшемся окне «Менеджер плат» выбрать модуль ESP8266, как показано на рисунке 3.11, установить его, и после того как загрузились все необходимые модули нажать кнопку «Заккрыть».

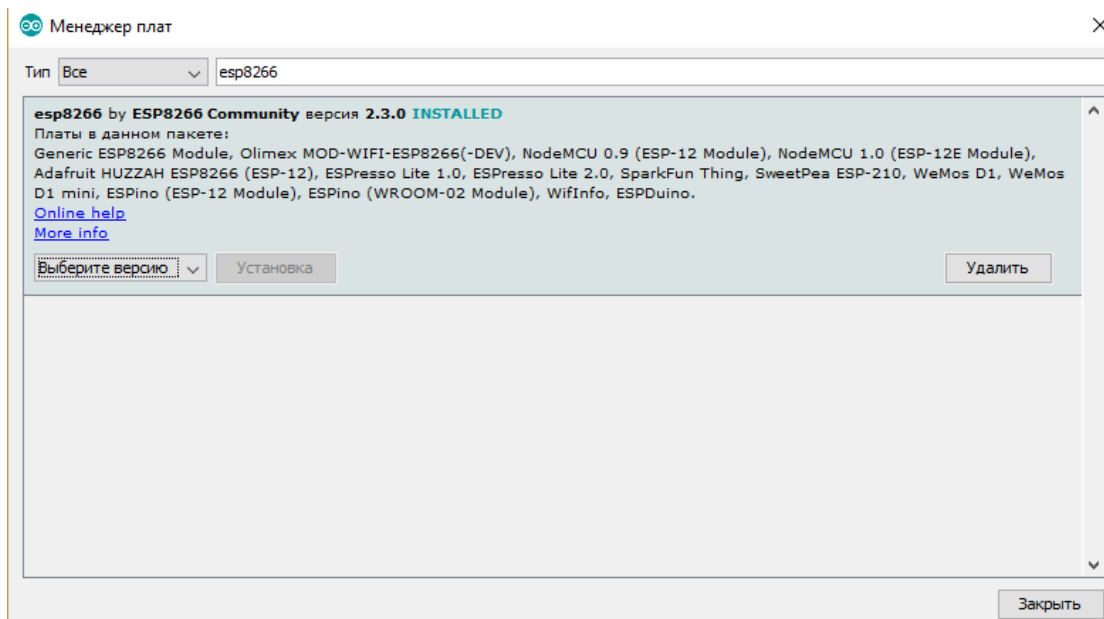


Рисунок 3.11 – Установка модулей для модуля ESP8266

Для изменения температуры от начальной до заданной, необходимо реализовать ПИД-регулятор (листинг 3.1). ПИД регулятор принимает на вход две величины входной сигнал «input» и установку «setpoint» для регулирования выходного сигнала. Управления регулятора устанавливается при помощи коэффициентов K_p , K_i и K_d .

Листинг 3.1 ПИД регулятор

```
#include <PID_v1. h>
#define PIN_INPUT 0
#define PIN_OUTPUT 3

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
void setup()
{
  Input = analogRead(PIN_INPUT);
  Setpoint = 22;
  myPID.SetMode(AUTOMATIC);
}
void loop()
{
  Input = analogRead(PIN_INPUT);
  myPID.Compute();
  analogWrite(PIN_OUTPUT, Output);
}
```

Для подключения платы к сети интернет была реализована функция поиска точки доступа Wi-Fi.

Для работы с датчиками температуры необходима библиотека «OneWire» и «DallasTemperature», для работы с несколькими датчиками сразу.

Для начало необходимо узнать адрес датчика на шине, и подключен ли датчик корректно (листинг 3.2).

Листинг 3.2 Инициализации адреса датчика

```
if ( !ds.search(addr) ) {
    Serial.println("No more addresses.");
    Serial.println();
    ds.reset_search();
    delay(250);
    return;}

```

Функция «ds.search(addr)» отвечает за индексирование датчика. Если устройство найдено, то в массив «addr» будет добавлен адрес датчика.

Затем датчику отправляется команда для считывания температуры и отправки ее в регистр (листинг 3.3).

Листинг 3.3 Считывание температуры с датчика

```
ds.reset();
ds.select(addr);
ds.write(0x44, 1);

```

Далее отправляется команда на чтение данных из регистра и считывание ответа в массив по циклу (листинг 3.4).

Листинг 3.4 Чтение данных из регистра

```
present = ds.reset();
ds.select(addr);
ds.write(0xBE);
Serial.print(" Data = ");
Serial.print(present, HEX);
Serial.print(" ");
for ( i = 0; i < 9; i++) {
    data[i] = ds.read();
    Serial.print(data[i], HEX);
    Serial.print(" ");
}

```

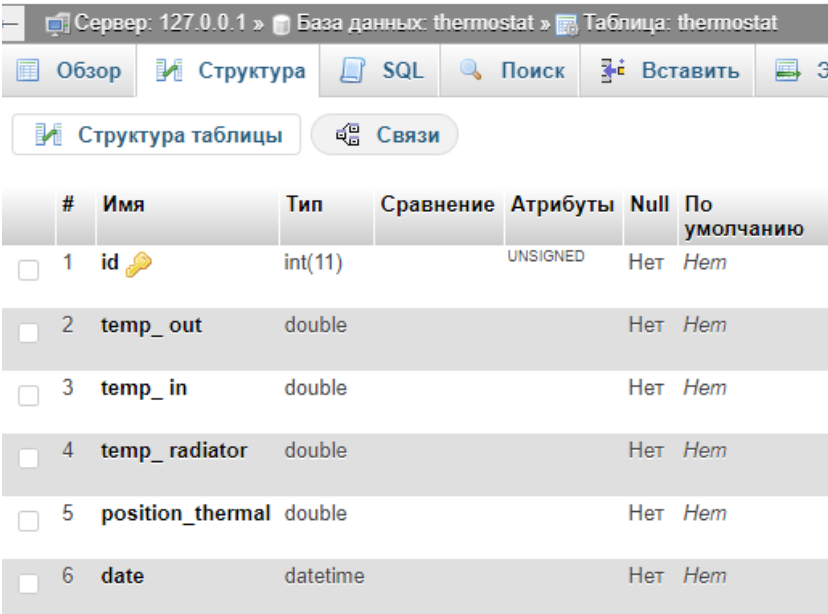
3.5 Реализация программной части

Для реализации программной части была спроектирована и реализована база данных.

Выделим основную сущность необходимую для дальнейшего проектирования структуры базы данных: термостат (содержит данные с датчиков, а также текущую дату и время значений).

На основе этой сущности спроектируем таблицу базы данных «thermostat» со следующими полями (рисунок 3.12):

- id (int) – идентификатор устройства;
- temp_out (double) – температура наружного воздуха;
- temp_in (double) – температура в помещении;
- temp_radiator (double) – температура радиатора отопления;
- position_thermal (double) – положение термоголовки;
- date (datetime) – текущая дата показаний;



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1 id	int(11)		UNSIGNED	Нет	Нет
<input type="checkbox"/>	2 temp_out	double			Нет	Нет
<input type="checkbox"/>	3 temp_in	double			Нет	Нет
<input type="checkbox"/>	4 temp_radiator	double			Нет	Нет
<input type="checkbox"/>	5 position_thermal	double			Нет	Нет
<input type="checkbox"/>	6 date	datetime			Нет	Нет

Рисунок 3.12 – Таблица «thermostat» базы данных

В аппаратной части был реализован ПИД регулятор, который контролирует температуру в помещении на указанное значение.

ПИД-регулятор годами показывает, что он решает задачу поддержания необходимой температуры, но хотелось бы достичь большого комфорта в помещении. Для этого была поставлена задача поддержания комфортной температуры в помещении по расписанию.

Для построения алгоритма необходимо определять время нагрева воздуха в помещении от некой начальной температуры до заданной температуры к заданному моменту времени. Также необходимо рассчитывать время нагрева от начальной температуры до необходимой.

Для решения задачи была реализована нейронная сеть, которая позволяет вычислять время нагрева в помещении от начальной температуры до необходимой температуры к определенному моменту времени.

Алгоритм для управления температурным режимом в помещении был выстроен на базе двух нейронных сетей: многослойный персептрон (MLP) и рекуррентной нейронной сети с долгой краткосрочной памятью (LSTM).

Для получения данных с сервера была реализована функция «get_data» представленная в приложении А. В функции определяются данные, передаваемые с датчиков в базу данных.

Для решения поставленной задачи была реализована тестовая нейронная сеть вида MLP, представленная в листинге 3.5.

Листинг 3.5 Тестовая нейронная сеть вида MLP

```
def create_my_mlp(load = True, num=0):
    if load:
        json_file = open('./models/model.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        model = model_from_json(loaded_model_json)
        model.load_weights("./models/model.h5")
        print("Loaded model from disk")
    else:
        model = Sequential()
        model.add(Dense(256, input_shape = (num,)))
        model.add(BatchNormalization())
        model.add(LeakyReLU())
        model.add(Dense(1))
    model.summary()
    return model
```

Сеть состоит из 256 полно связанных нейронов, на выходе из которых применяется пакетная нормализация данных «BatchNormalization» и функция активация «LeakyReLU». Далее так как необходимо получить одно единственное значение время нагрева или охлаждения на выходе нейронной сети используется один нейрон, который дает значение.

Точность предсказания для нейронной сети вычисляется пропорционально средней абсолютной ошибке по формуле 3.1. Точность прогноза в процентах равна $100\% - MAPE$.

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|Z(t) - \hat{Z}(t)|}{Z(t)} \cdot 100\% \quad (3.1)$$

где $Z(t)$ – фактическое значение временного ряда,

$\hat{Z}(t)$ – прогнозное значение временного ряда.

Данная нейронная сеть по данным исследования показала точность всего 50%, поэтому было принято решение модифицировать нейронную сеть.

Для этого была реализована нейронная сеть MLP с нормализацией (регуляризацией) и разделением сети на два слоя по 64 и 16 полно связанных нейронов (листинг 3.6).

Листинг 3.6 Тестовая нейронная сеть вида MLP с нормализацией

```
def create_best_mlp(load = True, num=0):
    if load:
        json_file = open('./models/best_mlp.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        model = model_from_json(loaded_model_json)
        model.load_weights("./models/best_mlp.h5")
        print("Loaded model from disk")
    else:
        model = Sequential()
        model.add(Dense(64, input_shape=(num, ),
            activity_regularizer=regularizers.l2(0.01)))
        model.add(BatchNormalization())
        model.add(LeakyReLU())
        model.add(Dense(16,
            activity_regularizer=regularizers.l2(0.01)))
```

```

        model.add(BatchNormalization())
        model.add(LeakyReLU())
        model.add(Dense(1))
        model.add(Activation('linear'))
model.summary()
return model

```

В нейронную сеть добавлен процесс регуляризации «regularizers» для ограничения весов нейронной сети, чтобы не было большого разброса в значениях для исключения эффекта переобучения.

Данная нейронная сеть по результатам исследования показала точность выходных данных 65%.

Для сравнения нескольких моделей алгоритма была реализована тестовая функция нейронной сети вида LSTM, представленная в листинге 3.7.

Листинг 3.7 Тестовая нейронная сеть вида LSTM

```

def create_my_lstm(load = False, num=0, prop=1):
    if load:
        json_file = open('./models/model.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        model = model_from_json(loaded_model_json)
        model.load_weights("./models/model.h5")
        print("Loaded model from disk")
    else:
        model = Sequential()
        model.add(LSTM(6, input_shape = (num, prop),
            return_sequences = False))
        model.add(Dense(24))
        ##model.add(Activation('relu'))
    model.summary()
    return model

```

Модель «Sequential» представляет собой линейный стек слоев. Последующие слои добавляются с помощью метода «.add()» с опцией «return_sequence=False», по которому образуется единичный выходной вектор.

Функции визуализации и обучения нейронной сети представлены в приложении А.

На рисунке 3.13 представлен пример работы нейросети LSTM, на котором видно среднеквадратичную ошибку «val_loss», которая изначально высока, однако с каждой эпохой обучения нейронной сети она уменьшается.

```

Train on 427 samples, validate on 107 samples
Epoch 1/30
427/427 [=====] - 3s 7ms/step - loss: 200.7186 - val_loss: 243.7071
Epoch 2/30
427/427 [=====] - 0s 89us/step - loss: 130.2738 - val_loss: 172.7188
Epoch 3/30
427/427 [=====] - 0s 97us/step - loss: 86.9109 - val_loss: 123.9527
Epoch 4/30
427/427 [=====] - 0s 97us/step - loss: 58.5707 - val_loss: 91.3879
Epoch 5/30
427/427 [=====] - 0s 98us/step - loss: 41.0577 - val_loss: 68.7715
Epoch 6/30
427/427 [=====] - 0s 97us/step - loss: 30.9307 - val_loss: 53.7192
Epoch 7/30
427/427 [=====] - 0s 98us/step - loss: 25.0882 - val_loss: 44.5493
Epoch 8/30
427/427 [=====] - 0s 97us/step - loss: 22.1689 - val_loss: 38.9246
Epoch 9/30
427/427 [=====] - 0s 98us/step - loss: 20.8058 - val_loss: 35.9207
Epoch 10/30
427/427 [=====] - 0s 99us/step - loss: 20.3621 - val_loss: 34.2123
Epoch 11/30
427/427 [=====] - 0s 101us/step - loss: 20.1909 - val_loss: 33.4018
Epoch 12/30
427/427 [=====] - 0s 94us/step - loss: 20.1781 - val_loss: 32.6290
Epoch 13/30
427/427 [=====] - 0s 98us/step - loss: 20.1422 - val_loss: 32.8353
Epoch 14/30
427/427 [=====] - 0s 91us/step - loss: 20.1181 - val_loss: 32.6664
Epoch 15/30
427/427 [=====] - 0s 98us/step - loss: 20.0899 - val_loss: 33.2220

```

Рисунок 3.13 – Пример работы нейросети LSTM

На рисунках 3.14 – 3.16 представлены соотношения реальных данных и предсказанных нами нейронной сетью в архитектуре LSTM. На графике по горизонтали указаны данные предсказания времени в нормальной форме в диапазоне от -0,15 до 0,15, по вертикали указаны временные периоды измерения.

На рисунке 3.14 представлена первая эпоха с гиперпараметрами подобранными случайным образом.

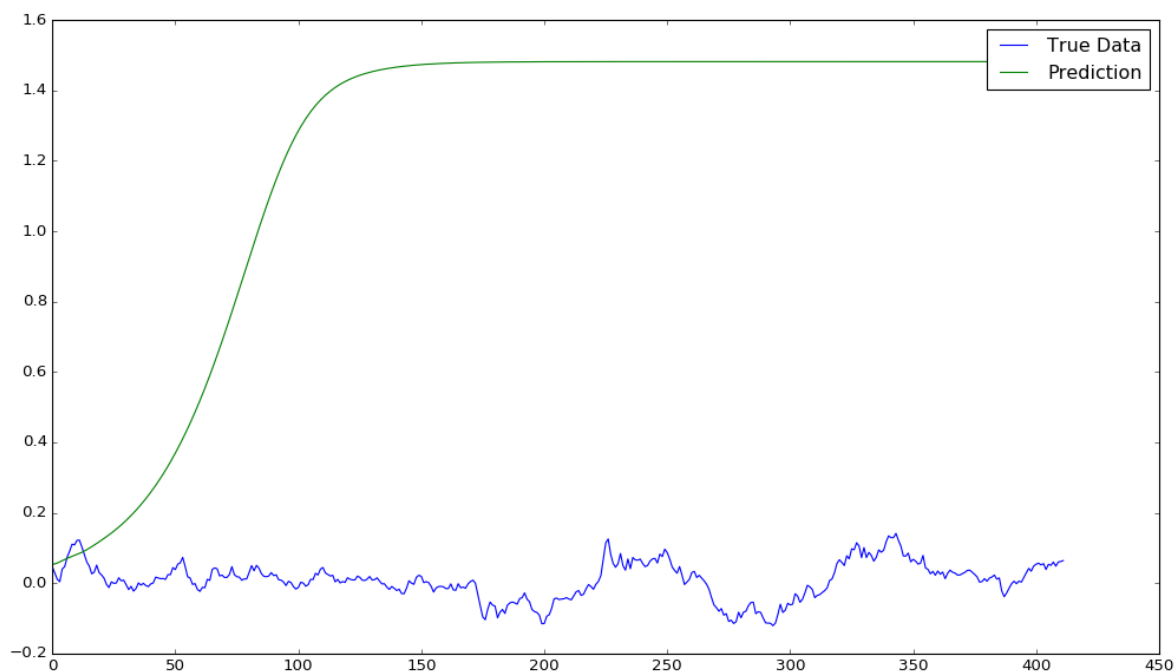


Рисунок 3.14 – Первая проба обучения

На рисунке 3.15 представлена 100 эпоха обучения с гиперпараметрами подобранными случайным образом.



Рисунок 3.15 – Вторая проба обучения

На рисунке 3.18 представлена 500 эпоха с гиперпараметрами подобранными экспериментальным путем.

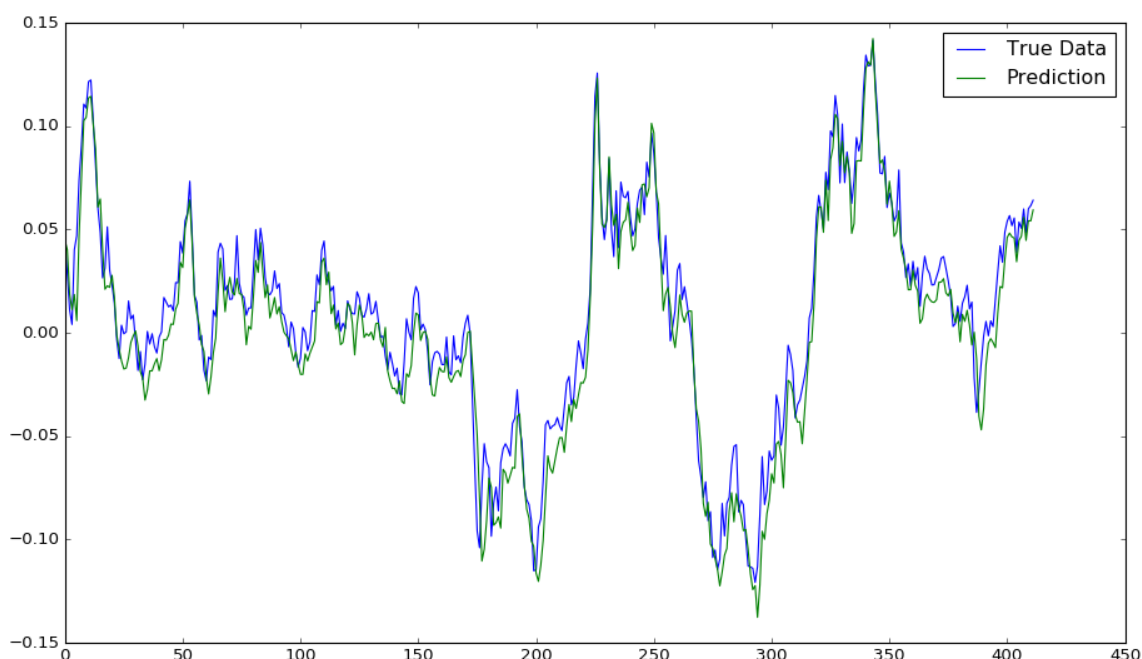


Рисунок 3.16 – Третья проба

Данная нейронная сеть по результатам исследования показала лучший результат, точность предсказания по экспериментальным данным составила 78%.

Из рассмотренных ранее нейронных сетей наилучший результат показала рекуррентная нейронная сеть с долгой краткосрочной памятью. Однако в работе также был рассмотрен алгоритм на основе аналитического решения уравнения теплопроводности, который показал похожий результат вычисления по экспериментальным данным 75%.

Для интеллектуальной системы управления температурой в помещении может быть использован, как алгоритм на основе аналитического решения уравнения теплопроводности, так и алгоритм, основанный на базе нейронной сети. Алгоритм, основанный на аналитическом решении, может быть надёжным в постоянном поддержании параметров, однако нейронная сеть с каждой последующей эпохой обучения может дать результаты с более высоким процентом предсказания или же перейти в стадию переобучения.

Для однозначного выбора алгоритма необходим длительный срок для экспериментов. Также желательно рассмотреть температурные режимы в помещениях других зданий.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы был проведен анализ предметной области, построена иерархическая структура работ, рассмотрены факторы, определяющие климатические условия в помещении и методика расчета теплотерь. Были исследованы современные способы регулирования температуры в помещении и рассмотрены два реализованных алгоритма для управления температурным режимом в помещении, основанные на математическом анализе и на базе нечеткой логике.

Помимо этого была построена архитектура проекта и определены инструменты реализации.

Для интеллектуальной системы управления температурным режимом в помещении был разработан алгоритм на базе двух нейронных сетей. А также была реализована аппаратная и программная части интеллектуальной системы.

Аппаратная часть была реализована с использованием платы NodeMCU со встроенным модулем ESP8266, и подключенных к ней интегральных датчиков температуры DS18B20. Устройство программировалось в среде Arduino IDE.

Программная часть была реализована в виде веб-сервиса и подключенной к ней базе данных. Веб-сервис был реализован при использовании фреймворк «Flask». База данных была реализована в среде MySQL. В качестве веб-сервера для развертывания веб-сервиса был выбран Nginx.

Таким образом, цель работы, а именно разработка интеллектуальной системы управления температурой в помещении – была достигнута, поставленные задачи – выполнены.

Результаты работы могут применяться в различных помещениях с установленными приборами отопления, кондиционирования и вентиляции для поддержания оптимальных условий микроклимата при минимальных энергетических затратах.

Дальнейшие направления работы будут направлены на применение аппаратно-программного комплекса для регулирования температуры в масштабах зданий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 30494-2011. Здания жилые и общественные. Параметры микроклимата в помещениях – [Электронный ресурс] – Режим доступа: <http://docs.cntd.ru/document/gost-30494-2011> (дата последнего обращения 01-10-2020).
2. Ершов В.В. Оптимальное управление температурным режимом в офисных и жилых зданиях – [Электронный ресурс]. – Режим доступа: <http://wila.ru/4/304/article32544/> (дата последнего обращения 15-06-2020).
3. Заварзин Б.Б., Рюмин Р.В., Чукарев А.Г. Методика расчета теплотерь для помещений – [Электронный ресурс]. – Режим доступа: <https://moluch.ru/archive/177/46071/> (дата последнего обращения 22.11.2020).
4. Нейро-нечеткая система контроля температуры воздуха в помещении с использованием Fuzzy controller – [Электронный ресурс] – Режим доступа: <https://topuch.ru/kursovaya-rabota-nejro-nechetkaya-sistema-kontrolya-temperatur/index.html> (дата последнего обращения 01-09-2020).
5. Петров В.Ю., Рудашевская Е.А. Технология «Интернет вещей» как перспективная современная информационная технология – [Электронный ресурс] – Режим доступа: <http://fundamental-research.ru/ru/article/view?id=41775> (дата последнего обращения 01-12-2020).
6. Тяхтий Ю.А., Гоман В.В. Обзор интеллектуальных термостатов для управления микроклиматом помещений – [Электронный ресурс] – Режим доступа: <http://hdl.handle.net/10995/87554> дата последнего обращения 01-12-2020).
7. Устройство и принцип работы терморегулятора – [Электронный ресурс]. – Режим доступа: <https://www.tproekt.com/ustrojstvo-i-princip-raboty-termoregulatora/> (дата последнего обращения 22.11.2020).
8. Ameena Saad al-Sumaiti, Mohammed Hassan Ahmed & Magdy M. A. Salama Smart Home Activities: A Literature Review, Electric Power Components and Systems, 42:3-4, 294-305, (2014) DOI: 10.1080/15325008.2013.832439.

9. Dae-Man, H., Jae-Hyun, L., “Design and implementation of smart home energy management systems based on Zigbee,” IEEE Trans. Consum. Electron., Vol. 56, No. 3, pp. 1417-1425, 2010.
10. Development of the temperature prediction algorithm for smart thermostat – [Электронный ресурс] – Режим доступа: <http://hdl.handle.net/10995/94223> (дата последнего обращения 10-12-2020)
11. DS18B20 – датчик температуры с интерфейсом 1-Wire – [Электронный ресурс] – Режим доступа: <http://mypractic.ru/ds18b20-datchik-temperature-s-interfejsom-1-wire-opisanie-na-russkom-yazyke.html> (дата последнего обращения 01-12-2020).
12. Erol-Kantarci, M., and Mouftah, H. T., “Wireless sensor networks for cost-efficient residential energy management in the smart grid,” IEEE Trans. Smart Grid, Vol. 2, No. 2, pp. 314-325, June 2011.
13. Forsythe G E, Malcolm M A and Moler C B 1977 Computer Methods for Mathematical Computations (Prentice Hall Professional Technical Reference).
14. Ha, D. L., Zamai, S. P. E., and Jacomino, M., “A home automation system to improve household energy control,” 12th IFAC Symposium on Information Control Problems in Manufacturing, Saint Etienne, France, 17–19 May 2006.
15. Honeywell vs. Nest – [Электронный ресурс] – Режим доступа: <https://blog.iridi.com/ru/honeywell-vs-nest-bitva-za-umnyj-termostat/> (дата последнего обращения 01-12-2020).
16. Honeywell vs. Nest: the battle for the smart thermostat – [Электронный ресурс] – Режим доступа: <https://www.predictingourfuture.com/12-honeywell-vs-nest-the-battle-for-the-smart-thermostat/> (дата последнего обращения 01-06-2020)
17. Nest Learning Thermostat v.3. – [Электронный ресурс] – Режим доступа: <https://smart-home.market/> (дата последнего обращения 01-06-2020)
18. S. De, T. Elsaleh, P. Barnaghi, S. Meissner, "An Internet of Things Platform for Real-World and Digital Objects", Journal of Scalable Computing: Practice and Experience, vol 13, no.1, 2012.

19. Siemens REA23 manual [Электронный ресурс] – Режим доступа: <https://www.downloads.siemens.com/download-center/Download.aspx?pos=download&fct=getasset&id1=A6V10075730> (дата последнего обращения 01-06-2020).
20. T9 Smart Thermostat – [Электронный ресурс] – Режим доступа: <https://t9.honeywellhome.com/> (дата последнего обращения 01-06-2020)
21. The best smart thermostats of 2019 – [Электронный ресурс] – Режим доступа: <https://www.cnet.com/news/top-smart-thermostats-of-2019-ecobee-smartthermostat-nest/> (дата последнего обращения 01-12-2019).
22. Williams, E. D., Matthews, H. S., “Scoping the potential of monitoring and control technologies to reduce energy use in homes,” Proceedings of the 2007 IEEE International Symposium on Electronics & the Environment, pp. 239–244, Orlando, FL, 710 May 2007.

ПРИЛОЖЕНИЕ А

Листинги разработанной программы

Листинг функции для получения данных с сервера

```
def get_data(corr=False, norm=False):
    df = pd.read_csv('./data/testradiator3.csv', index_col='Id')
    Y = df['Room_Temp'].as_matrix()
    df.drop(['Room_Temp'], axis=1)
    weeks = []
    months = []
    hours = []
    X_train = []
    Y_train = []
    X_test = []
    Y_test = []
    for item in df['Date']:
        if item:
            items = item.split(' ')
            date = items[0].split('-')
            t = items[1].split(':')
            h = t[0]
            m = date[1]
            d = date[2]
            y = date[0]
            dt = datetime.datetime.strptime(y+m+d, '%Y%m%d')
            w = dt.isoweekday()
            weeks.append(w)
            months.append(int(m))
            hours.append(int(h))
    del df['Date']
    del df['Door_State']
    del df['Window_State']
    del df['Open_Carrier']
    df['Hours'] = hours
    df['Months'] = months
    df['Weeks'] = weeks
    X = df.as_matrix()
    if norm:
        X = preprocessing.normalize(X)
    print('ALL:', X.shape, Y.shape)
    test = int(X.shape[0]*0.7)
    X_train = X[:test]
    Y_train = Y[:test]
    X_test = X[test:]
    Y_test = Y[test:]
    print('TRAIN: ', X_train.shape, Y_train.shape)
    print('TEST: ', X_test.shape, Y_test.shape)
    num = X.shape[1]
    if corr:
        sns.pairplot(df[["Room_Temp", "External_Temp"]], diag_kind="kde")
        sns.plt.show()
```

```
return (X_train, Y_train, X_test, Y_test, num)
```

Листинг функции визуализации и обучения нейронной сети

```
def visual_history(history):
    plt.figure()
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='best')
    plt.show()
    plt.figure()
    plt.plot(history.history['acc'])
    plt.plot(history.history['val_acc'])
    plt.title('model accuracy')
    plt.ylabel('acc')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='best')
    plt.show()

def main():
    X_train, Y_train, X_test, Y_test, num = get_data(corr=False,
norm=False)
    load = False
    name = 'model'
    model = create_my_mlp(load, num)
    model = create_best_mlp(load, num)
    Compile model
    model.compile(loss='mse', optimizer='adam',
metrics=['accuracy'])
    reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.9,
patience=5, min_lr=0.000001, verbose=1)
    erl_stop = EarlyStopping(monitor='val_loss', patience=100)
    if load != True:
        history = model.fit(X_train, Y_train, epochs=1000,
verbose=1, batch_size=32, validation_data=(X_test, Y_test),
callbacks=[erl_stop])
        visual_history(history)
        save_model(model, name)
    else:
        Y_predict = model.predict(X_test)
        #show the inputs and predicted outputs
        y = []
        for i in range(len(X_test)):
            print("Real=%s, Predicted=%s" % (Y_test[i],
Y_predict[i][0]))
            y.append(i)
        plt.figure()
        plt.plot(y, Y_test, 'r', label="Real")
        plt.plot(y, Y_predict, 'b', label="Predicted")
```

```
plt.legend(loc="upper left")
plt.title('Eq model')
plt.ylabel('value')
plt.xlabel('time')
plt.show()

score = model.evaluate(X_test, Y_test, batch_size=32)
print('test loss, test acc:', score)
make a prediction
```