

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт Радиоэлектроники и информационных технологий
Кафедра Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

РОП по направлению 27.04.04

(подпись) В.Г. Лисиенко
« _____ » _____ 2022 г.
(Ф.И.О.)

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Учебный интерактивный тренажер моделирования системы управления охлаждения газа в компрессорных установках

Научный руководитель:

к.т.н., доцент, доцент департамента ИТиА
ИРИТ-РтФ УрФУ

ученая степень, ученое звание, должность

О. Г. Трофимова

(подпись)

Нормоконтролер:

В.В. Телицин

(подпись)

Студент группы РИМ-201202

Н. Н. Таланкин

(подпись)

Екатеринбург
2022

РЕФЕРАТ

Диссертация 138 с., 1 кн., 148 рис., 2 табл., 40 источн., 3 прил.

СТАБИЛИЗАЦИЯ ГАЗА, ТЕМПЕРАТУРА ГАЗА, МАГИСТРАЛЬНЫЕ ГАЗОПРОВОДЫ, ИНТЕРАКТИВНЫЙ ТРЕНАЖЕР, МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ИДЕНТИФИКАЦИЯ, ОПТИМИЗАЦИЯ, ПИД-РЕГУЛЯТОР, НЕЧЕТКИЙ РЕГУЛЯТОР, ГРАФИЧЕСКИЙ ИНТЕРФЕЙС.

Объектом исследования являются компрессорные станции и аппараты воздушного охлаждения магистральных газопроводов.

Цель исследования – разработка учебного интерактивный тренажер методического комплекса моделирования системы охлаждения газа в компрессорных установках.

Разработана математическая модель объекта управления с помощью пакета Simulink Matlab, проведена идентификация и оптимизация многоконтурной модели стабилизации температуры газа, разработаны два типа регуляторов: пропорционально-интегрально-дифференциальный и нечеткий лингвистический по алгоритму Мамдани.

В учебных целях создан интерактивный тренажер в виде графического приложения стабилизации температуры газа в компрессорных установках для разработанных моделей систем управления с помощью модуля AppDesigner Toolbox.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 Технологический процесс охлаждения газа в компрессорных установках. Технологический комплекс охлаждения газа.....	8
2 Описание существующих аналогов систем моделирования технологических процессов	11
2.1 Matlab.....	11
2.2 Simintech.....	12
2.3 Wolfram Mathematica	13
2.4 MathCAD	14
3 Новизна решения магистерской диссертации	17
4 Характеристика объекта управления и математическое описание для математической модели поддержания стабилизации температуры при охлаждении газа в компрессорных установках.....	18
4.1 Характеристика технологического процесса, математическое описание	18
4.2 Представление блоков модели ОУ из математического описания через Simulink в виде математических функций	22
4.3 Результаты	23
5 Идентификация модели СУ с помощью пакета Identification System Toolbox	24
6 Настройка ПИД - регулятора модели СУ	37
7 Устойчивость и адекватность системы ОУ, оптимизация параметров системы.....	41
7.1 Устойчивость системы.....	41

7.2 Адекватность системы	43
8 Описание методов регулирования стабилизации температуры на основе уравнения состояния идеального газа.....	52
8.1 Описание технологических параметров с помощью уравнения состояния идеального газа (уравнения Клапейрона – Менделеева)	52
8.2 Постановка задачи проектирования для нечеткого регулятора	53
9 Проектирование нечеткого лингвистического регулятора по Мамдани средствами пакета Simulink и модуля Fuzzy Logic Designer системы Matlab	55
10 Математическое описание коэффициента теплопроводности	68
11 Стохастический эксперимент	72
12 Однофакторный дисперсионный анализ модели	78
13 Анализ и тестирование программного кода модели Simulink с помощью встроенных компонентов языка Си Matlab.....	84
13.1 Описание системы Embeeded Coder среды MATLAB	83
13.2 Ключевые особенности	83
13.3 Получение программного кода модели Simulink на языке Си	84
13.4 Проверка исходного кода модели Simulink модулем PolySpace Code Verifier	91
14 Проектирование графического интерфейса пользователя с помощью модуля AppDesigner	94
14.1 Описание файлов приложения	94
14.2 Описание функционала приложения.....	98
15 Тестирование приложения.....	100
ЗАКЛЮЧЕНИЕ.....	103
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	105

ПРИЛОЖЕНИЕ А. Листинг программного кода модели Simulink на языке Си.....	105
ПРИЛОЖЕНИЕ Б. Листинг программного кода приложения в системе AppDesigner.....	125
ПРИЛОЖЕНИЕ В. Данные дисперсионного анализа.....	133

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей пояснительной записке к выпускной квалификационной работе применяются следующие сокращения и обозначения.

АВГ – аппарат воздушного охлаждения горизонтального типа

АВО – аппарат воздушного охлаждения

АЧХ – амплитудно-частотная характеристика

КС – компрессорная станция

ЛАЧХ – логарифмическая амплитудно-частотная характеристика

НЛР – нечеткий лингвистический регулятор

ОС – операционная система

ОУ – объект управления

ПИД – пропорционально-интегрально-дифференциальный

СУ – система управления

УГО – установка охлаждения газа

УУ – управляющее устройство

ФЧХ – фазочастотная характеристика

GUI – Graphics User Interface (графический интерфейс пользователя)

ВВЕДЕНИЕ

В современной науке и технике моделирование физических процессов является одной из важнейших составляющих проведения эмпирических исследований. Трудно переоценить значимость рациональной и правильно построенной математической модели для систем управления сложными объектами и процессами.

Нефтегазовая отрасль является одной из основополагающих для Российской Федерации, поэтому, для студентов и магистрантов технических направлений соответствующего профиля очень важно понимать суть термодинамических, теплотехнических и теплофизических процессов, происходящих в процессе охлаждения, обработки и транспортировке газа на компрессорных станциях и магистральных газопроводах.

Разработанный учебный интерактивный тренажер необходим для получения практических знаний основных принципов работы аппаратов воздушного охлаждения и компрессорных станций, проведения вычислительных экспериментов при вариации входных параметров, например, в таких дисциплинах как «Метрология, сертификация и стандартизация», «Теория и методы управления экспериментами», «Моделирование систем управления».

Прикладная значимость данного исследования – интеграция учебного интерактивного тренажера со стендами-эмуляторами стабилизации температуры газа с использованием программируемых логических контроллеров.

1 Технологический процесс охлаждения газа в компрессорных установках. Технологический комплекс охлаждения газа

Поддержание стабильной температуры газа на протяжении всего технологического процесса является одной из главных задач производства. Для подачи в магистральный газопровод, после компремирования (сжатия) газ необходимо охладить до определенной температуры. С этой задачей отлично справляются аппараты воздушного охлаждения [1].

Аппараты воздушного охлаждения (АВО) газа предназначены для охлаждения жидких и парообразных сред в технологических процессах нефтеперерабатывающей, нефтехимической и смежной промышленности, конденсации смесей углеводородов на станциях в системе магистрального транспорта природного газа, для охлаждения природного газа на компрессорных станциях магистральных газопроводов.

АВО входят в состав установок охлаждения газа (УГО), где и происходит технологический процесс снижения температуры газа. Обычно в состав УГО входит, как минимум, 12 секций АВО с двумя электроприводными вентиляторами в каждой секции.

Аппарат воздушного охлаждения представляют собой сложное устройство, состоящий из секций, которые размещены на опорной металлоконструкции, коллекторов ввода и вывода охлаждаемого продукта, вентиляторов и диффузоров. Секция АВО состоит из двух камер (сварных или разъемных) прямоугольной формы с перфорированными трубными досками и крышками (для разъемной камеры) или задними стенками (для сварной камеры) и трубного пучка из оребренных труб. Секции расположены вертикально, горизонтально, наклонно в виде шатра или зигзагообразно.

АВО работает следующим образом: на опорных металлоконструкциях закреплены трубчатые теплообменные секции. По трубам теплообменной секции пропускают транспортируемый газ, а через межтрубное пространство теплообменной секции с помощью вентиляторов, проводимых во вращение

от электромоторов, прокачивают наружный воздух. За счет теплообмена между нагретым при компремировании (сжатии) газом, движущимся в трубах и наружным воздухом, движущимся по межтрубному пространству, и происходит охлаждение технологического газа на КС.

АВО имеют более значительные габариты и металлоемкость по сравнению с аппаратами, охлаждаемыми водой, но затраты на АВО значительно меньше. Кроме того, при использовании АВО исключается опасность загрязнения накипью со стороны охлаждающей среды. Результаты эксплуатации АВО свидетельствуют о возможности их использования при температуре окружающего воздуха от -55 до $+55$ °С.

Опыт эксплуатации АВО на КС показывает, что снижение температуры газа в этих аппаратах можно осуществить примерно на значение порядка $15-25$ °С. Одновременно опыт эксплуатации указывает на необходимость и экономическую целесообразность наиболее полного использования установок охлаждения газа на КС в годовом цикле эксплуатации, за исключением тех месяцев года с весьма низкими температурами наружного воздуха, когда включение всех аппаратов на предыдущей КС приводит к охлаждению транспортируемого газа до температуры, которая может привести к выпадению гидратов. Обычно это относится к зимнему времени года.

При проектировании компрессорной станции количество аппаратов воздушного охлаждения выбирается в соответствии с отраслевыми нормами ОНТП51-1-85. На основании этих норм температура технологического газа на выходе из АВО должна быть не выше $15-20$ °С средней температуры наружного воздуха.

Уменьшение температуры технологического газа, поступающего в газопровод после его охлаждения в АВО, приводит к уменьшению средней температуры газа на линейном участке трубопровода и, как следствие, к снижению температуры и увеличению давления газа на входе в последующую КС. Это, в свою очередь, приводит к уменьшению степени

сжатия на последующей станции (при сохранении давления на выходе из нее) и энергозатрат на сжатие газа по станции.

Следует также отметить, что аппараты воздушного охлаждения газа являются экологически чистыми устройствами для охлаждения газа, не требуют расхода воды, относительно просты в эксплуатации. В эксплуатации применяются следующие типы АВО газа: 2АВГ-75, фирм «Нуово Пиньоне» и «Крезолуар».

В настоящее время установки охлаждения транспортируемого газа являются одним из основных видов технологического оборудования КС.

Энергопотребление АВО газа составляет до 70 % общего потребления электроэнергии компрессорных станций с газотурбинными приводами нагнетателей. Основной идеей разработок, направленных на уменьшение энергопотребления АВО газа, является уменьшение интенсивности обдува теплообменных секций. Мощность привода вентиляторов растет, пропорционально 3-й степени расхода прогоняемого воздуха, а охлаждающий эффект (количество теплоты, отводимой от газа в теплообменнике) растет существенно медленнее (не более чем 0,8-я степень расхода воздуха).

2 Описание существующих аналогов систем моделирования технологических процессов

2.1 Matlab

MATLAB – пакет прикладных программ для решения задач технических вычислений. Пакет используют более миллиона инженерных и научных работников, он работает на большинстве современных операционных систем, включая Linux, Mac OS, Solaris (начиная с версии R2010b поддержка Solaris прекращена) и Windows [2].

Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования.

Программы, написанные на MATLAB, бывают двух типов – функции и скрипты. Функции имеют входные и выходные аргументы, а также собственное рабочее пространство для хранения промежуточных результатов вычислений и переменных. Скрипты же используют общее рабочее пространство. Как скрипты, так и функции сохраняются в виде текстовых файлов и компилируются в машинный код динамически. Существует также возможность сохранять так называемые pre-parsed программы – функции и скрипты, обработанные в вид, удобный для машинного исполнения. В общем случае такие программы выполняются быстрее обычных, особенно если функция содержит команды построения графиков.

Интерфейс программы Matlab представлен на рисунке 2.1:

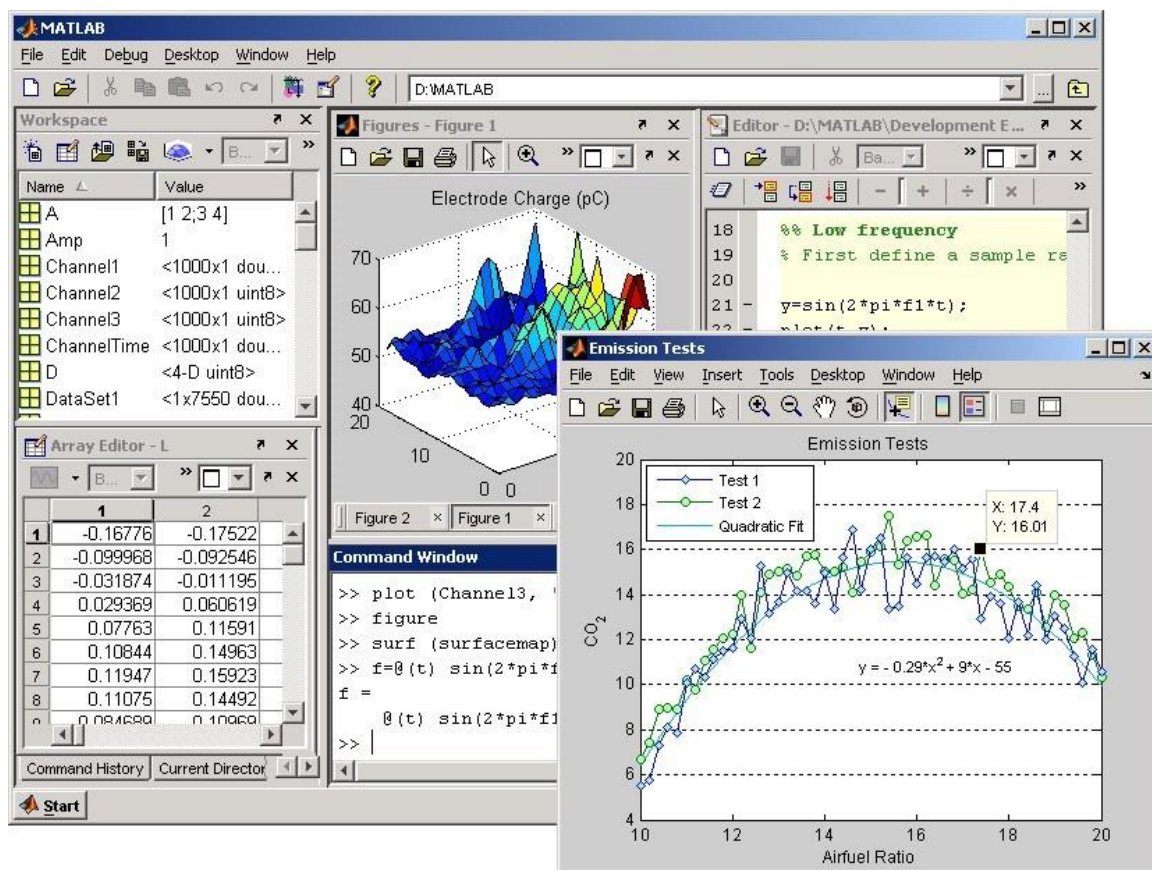


Рисунок 2.1 – Скриншот программы Matlab

2.2 Simintech

SimInTech (Simulation In Technic) – среда динамического моделирования технических систем, предназначенная для расчётной проверки работы систем управления сложными техническими объектами. SimInTech осуществляет моделирование технологических процессов, протекающих в различных объектах, с одновременным моделированием системы управления, и позволяет повысить качество проектирования систем управления за счет проверки принимаемых решений на любой стадии проекта [3].

SimInTech предназначен для детального исследования и анализа нестационарных процессов в ядерных и тепловых энергоустановках, в системах автоматического управления, в следящих приводах и роботах, и в любых технических системах, описание динамики которых может быть представлено в виде системы дифференциально-алгебраических уравнений и/или реализовано методами структурного моделирования. Основными

направлениями использования SimInTech являются создание моделей, проектирование алгоритмов управления, их отладка на модели объекта, генерация исходного кода на языке Си для программируемых контроллеров.

Скриншот программы SiminTech представлен на рисунке 2.2:

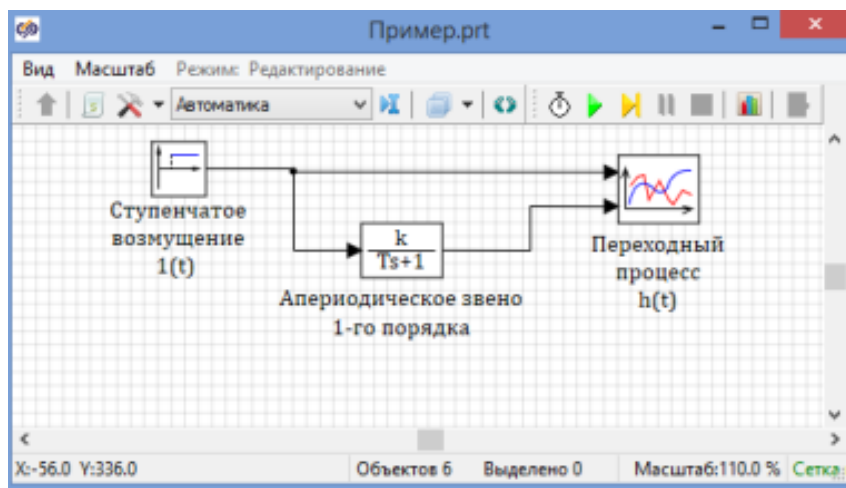


Рисунок 2.2 – Скриншот программы SininTech

2.3 Wolfram Mathematica

Mathematica – система компьютерной алгебры (обычно называется Математика, программный пакет Математика), широко используемая в научных, инженерных, математических и компьютерных областях [4].

Система также осуществляет численные расчёты: определяет значения функций (в том числе специальных)) с произвольной точностью, осуществляет полиномиальную интерполяцию функции от произвольного числа аргументов по набору известных значений, рассчитывает вероятности.

Теоретико-числовые возможности – определение простого числа по его порядковому номеру, определение количества простых чисел, не превосходящих данное; дискретное преобразование Фурье; разложение числа на простые множители, нахождение НОД и НОК.

Также в систему заложены линейно-алгебраические возможности – работа с матрицами (сложение, умножение, нахождение обратной матрицы, умножение на вектор, вычисление экспоненты, взятие определителя), поиск собственных значений и собственных векторов.

Система результаты представляет как в алфавитно-цифровой форме, так и в виде графиков. В частности, реализовано построение графиков функций, в том числе параметрических кривых и поверхностей; построение геометрических фигур (ломаных, кругов, прямоугольников и других); построение и манипулирование графами. Кроме того, реализовано воспроизведение звука, график которого задаётся аналитической функцией или набором точек.

Скриншот интерфейса программы Wolfram Mathematica представлен на рисунке 2.3:

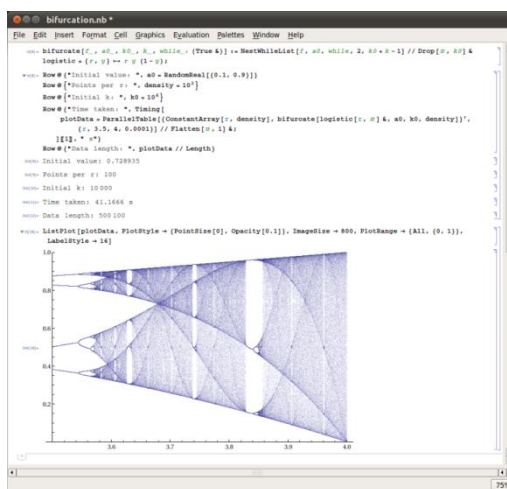


Рисунок 2.3 – Скриншот программы Wolfram Mathematica

2.4 MathCAD

Mathcad – система компьютерной алгебры из класса систем автоматизированного проектирования, ориентированная на подготовку интерактивных документов с вычислениями и визуальным сопровождением, отличается лёгкостью использования и применения для коллективной работы [5].

Mathcad имеет интуитивный и простой для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов.

Некоторые из математических возможностей Mathcad (версии до 13.1 включительно) основаны на подмножестве системы компьютерной алгебры

Maple (MKM, Maple Kernel Mathsoft). Начиная с 14 версии– использует символьное ядро MuPAD.

Работа осуществляется в пределах рабочего листа, на котором уравнения и выражения отображаются графически, в противовес текстовой записи в языках программирования. При создании документов-приложений используется принцип WYSIWYG (What You See Is What You Get – «что видишь, то и получаешь»).

Несмотря на то, что эта программа, в основном, ориентирована на пользователей, не являющихся программистами, Mathcad также используется в сложных проектах, чтобы визуализировать результаты математического моделирования путём использования распределённых вычислений и традиционных языков программирования. Также Mathcad часто используется в крупных инженерных проектах, где большое значение имеет трассируемость и соответствие стандартам.

Скриншот интерфейса программы MathCAD представлен на рисунке 2.4:

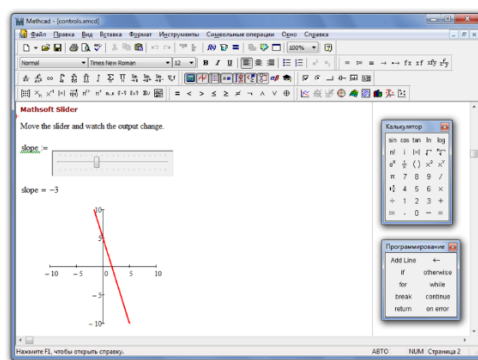


Рисунок 2.4 – Интерфейс программы MathCAD

В дальнейшем, при разработке интерактивного тренажера моделирования для построения алгоритмической структуры, графиков переходных процессов, анализа устойчивости системы и прочего применяется среда моделирования Matlab R2020a, т. к.:

- имеются навыки работы с данной средой моделирования, в том числе с одним из ее основных пакетов – Simulink;

– в Matlab имеется возможность реализации графического интерфейса пользователя (GUI), что является одним из необходимых требований для предоставления законченного программного продукта для защиты магистерской диссертации [6];

– Matlab позволяет быстро получить и проанализировать графики передаточных функций переходных процессов, измерить их физические величины в любой точке и контуре системы, дополнительно подать на систему любые задающие, управляющие и возмущающие воздействия, получить графики АЧХ, ФЧХ, ЛАЧХ, годографов, узнать запас по амплитуде и фазе для проверки системы на устойчивость, а также осуществить квантование процесса, если это необходимо [7].

3 Новизна решения магистерской диссертации

Разрабатываемый комплекс имеет следующие основные отличия:

- системный подход и декомпозиция. Учебный интерактивный тренажер состоит из нескольких основных частей, которые разбиты на более мелкие структуры (модули), т.е. тренажер является совокупностью взаимосвязанных между собой систем. Обучающийся может изучать те структуры системы, которые ему необходимы или интересны, а не рассматривать всю систему целиком, что обеспечит минимальный порог вхождения при работе с тренажером-эмулятором;
- графическое приложение и система подсказок. Практически всегда, системы моделирования позволяют построить качественные модели и получить достоверные результаты, но не обеспечивают пользователя исчерпывающей информацией, почему при выборе одного параметра изменяется другой на определенное значение, либо, почему при появлении единичного ступенчатого воздействия (сигнала ошибки) график переходного процесса начинает изменяться с определенного интервала измеряемой величины. Учебный интерактивный тренажер направлен не просто на демонстрацию возможностей программы и получения произвольных данных, а в первую очередь на когнитивный процесс обучения студентов;
- динамическая интерпретация. Во многих системах моделирования технологических процессов пользователю при изменении одного параметра перед запуском компиляции приходится изменять значение других параметров в целях предотвращения ошибки, т.е. вручную вводить все изменения перед запуском процесса моделирования. Учебный интерактивный тренажер в режиме реального времени динамически создает компоненты, необходимые для корректной работы при условии, что пользователь ввел их неверно, а также система сообщает пользователю, почему и какие данные изменены и зачем это необходимо.

4 Характеристика объекта управления и математическое описание для математической модели поддержания стабилизации температуры при охлаждении газа в компрессорных установках

4.1 Характеристика технологического процесса, математическое описание

Под объектом управления (ОУ) будем понимать устройство, которое осуществляет технологический процесс с помощью организованной в нем системы управления. Таким устройством в данном случае является компрессорная станция (КС) [8].

Под управляющим устройством (УУ) будем понимать устройство, воздействующее на объект управления для достижения самого процесса регулирования (управления) ОУ. Таким устройством является аппарат воздушного охлаждения газа (АВО) [8].

Температура газа при его компримировании (технологическом процессе сжатия газа) не должна превышать 90°C , а при температуре выше 110°C обязательно должно происходить автоматическое отключение КС для того, чтобы обезопасить оборудование.

Кроме того, выявлено, что температуру газа на выходе АВО после сжатия можно (и необходимо) снижать примерно на $15^{\circ}\text{C} - 25^{\circ}\text{C}$ и она должна быть не выше $15^{\circ}\text{C} - 20^{\circ}\text{C}$ средней температуры наружного воздуха на выходе (а она равна $40^{\circ}\text{C} - 50^{\circ}\text{C}$) [9]-[10]-[11].

Другими словами, нормальная технологическая температура газа на выходе АВО должна находиться в диапазоне $55^{\circ}\text{C} - 70^{\circ}\text{C}$ [12]-[13]-[14].

Для описания структуры технологического процесса необходимо дать некоторые определения:

Газ на линейном участке трубопровода – поток газозвдушной смеси, которая движется согласно линейному закону по трубам магистрального газопровода от входа –ого участка к его выходу.

Температура технологического газа, поступающего в газопровод – значение температуры газа, которое получено на выходе АВО (после охлаждения в АВО) и поступило в участок магистрального газопровода для дальнейшего транспорта.

Охлаждение отходящего газа воздухом – охлаждение воздухом отработанного в процессе эксплуатации газа (рабочего тела).

Параметры работы вентилятора – технологические значения, которыми характеризуется вентиляторный модуль АВО. К основным относятся производительность ($L, \text{м}^3/\text{ч}$) – это объем воздуха, который подает вентилятор в единицу времени, а также давление ($P, \text{Па}$) – это энергия, которая сообщается единицей объема воздуха, проходящего через вентиляторный модуль.

Так как модель системы управления (СУ) стабилизации температуры в газокompрессорных станциях построена на модели стабилизации температуры дымовых газов (технологически модели очень схожи с собой), то необходимо дать еще несколько определений, касаемо модели стабилизации температуры дымовых газов:

Температура дыма – мера средней кинетической энергии молекул дымового газа (без учета воздуха).

Температура дымовоздушной смеси – мера средней кинетической энергии молекул дымового газа и воздуха в совокупности.

Температура газа в фильтре – мера средней кинетической энергии молекул газа при поступлении в импульсный очистительный фильтр (иными словами, температура газа при стабилизации и его переработке в очистительном фильтре).

Рассмотрим технологический процесс охлаждения газа в АВО в КС.

На металлоконструкциях АВО закрепляются так называемые ТТС (трубчатые теплообменные секции, которые также называются теплообменником АВО). По этим трубным пучкам АВО поступает технологический транспортируемый газ с определенной температурой, а

через пространство между трубными пучками с помощью вентиляторных модулей (которые соответственно работают от электродвигателей), перекачивается наружный воздух. В результате этого, за счет конвективного теплообмена (теплообмена между потоками вещества), происходит охлаждение технологического газа на КС и его дальнейший транспорт по линейным участкам магистрального газопровода [14].

Здесь следует отметить, что у АВО с возобновляемыми (природоподобными источниками работы) в качестве рабочего тела охлаждения может перекачиваться не воздух, а отходящий (отработанный газ), который может без потерь производительности нагреваться в процессе работы и охлаждаться бесконечное количество раз (то есть АВО с природоподобной работой имеют итеративный характер использования рабочего тела без применения дополнительной энергии) [15].

Аналитический способ построения математической модели основан на анализе энергетического баланса в объекте (рисунок 4.1). Примем следующие обозначения:

θ_r – температура теплоносителя [$^{\circ}\text{C}$];

Cr – теплоемкость теплоносителя [J/K];

QH – поток тепла от нагревателя [W];

QF – поток тепла от других источников в системе (двигатель насоса и др.) [W];

QL – отдача тепла потребителям [W];

θ_A – Температура окружающей среды [$^{\circ}\text{C}$].

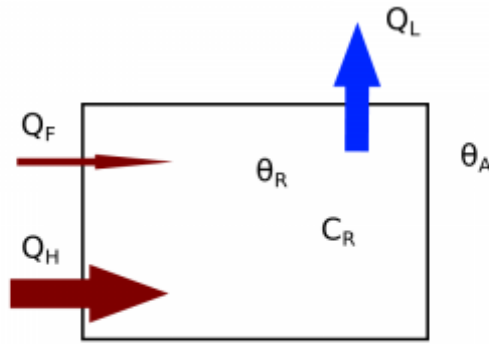


Рисунок 4.1 – Иллюстрация тепловых (энергетических) потоков
Энергетический баланс записывается в символьном виде (4.1.1):

$$QH + QF - QL = CRd\theta_R dt \quad (4.1.1)$$

Потери тепла системой QL (4.1.2):

$$QL = K(\theta_R - \theta_A) \quad (4.1.2)$$

где K – коэффициент теплопередачи, выражающий величину потребления тепла и других теплопотерь системы. Величина $QF \ll QH$, поэтому примем его равным нулю [16].

Так как коэффициент теплопотерь системы за исключением подсистемы теплопотребления является постоянным, то представим коэффициент K , как сумму коэффициентов теплопередачи котла и трубопровода KS и коэффициента теплопередачи подсистемы потребления тепла $KC(t)$, который является непостоянным и неизмеримым. Таким образом имеем следующую модель потерь тепла в установившемся режиме (4.1.3):

$$QL = (KS + KC(t))(\theta_R - \theta_A) \quad (4.1.3)$$

Коэффициент KS и диапазон изменения коэффициента $KC(\theta)$ определяются экспериментально.

Динамическая модель объекта примет следующий вид (4.1.4):

$$\frac{d\theta_R}{dt} = \frac{(QH + QF - (KS + KC(t))(\theta_R - \theta_A))}{Cr} \quad (4.1.4)$$

Анализ баланса энергий позволяет выдвинуть гипотезу о классе и структуре модели в форме передаточной функции (4.1.5):

$$\frac{K}{T_1 s} + \frac{1}{T_2 s} + 1 \quad (4.1.5)$$

Вторая постоянная времени отражает тепловую инерцию нагревателя [17].

4.2 Представление блоков модели ОУ из математического описания через Simulink в виде математических функций

Полученная модель Simulink представлена на рисунке 4.2

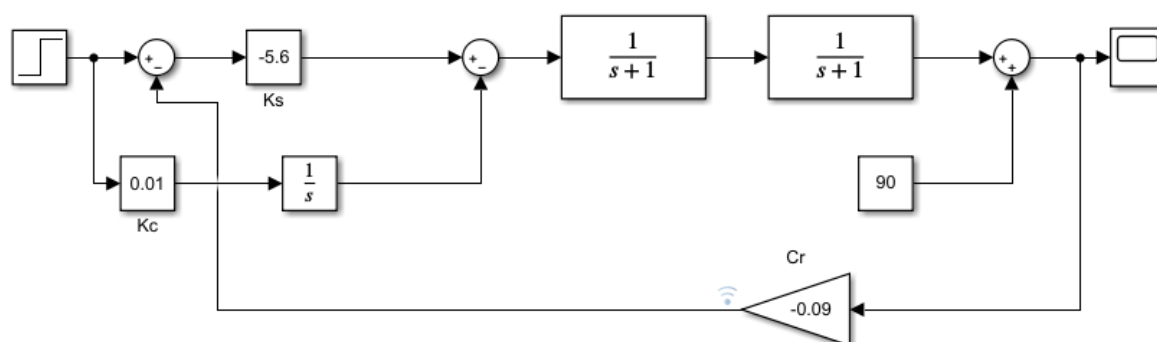


Рисунок 4.2 – Полученная модель стабилизации температуры при охлаждении газа

Следует отметить, что морфологически модель включает в себя следующие основные структурные блоки:

- Constant (значение 90, температура газа на выходе из компрессора, начальная точка отсчета);
- Slider Gain (коэффициенты KS, KC из математической модели, значения –5.6 и 0.01);
- Gain (коэффициент Cr из математической модели, значение –0.09);
- Sum (сумматоры с отрицательной и положительной обратной связью для блоков);
- Step (ступенчатое воздействие на входе модели);
- Transfer Fcn (передаточные функции с $Tp = 1$ из математической модели);

- Integrator (интегральная составляющая дифференциального уравнения математической модели);
- Scope (осциллограф для анализа переходного процесса);

4.3 Результаты

В результате проектирования модели ОУ получен аperiodический переходный процесс, представленный на рисунке 4.3:

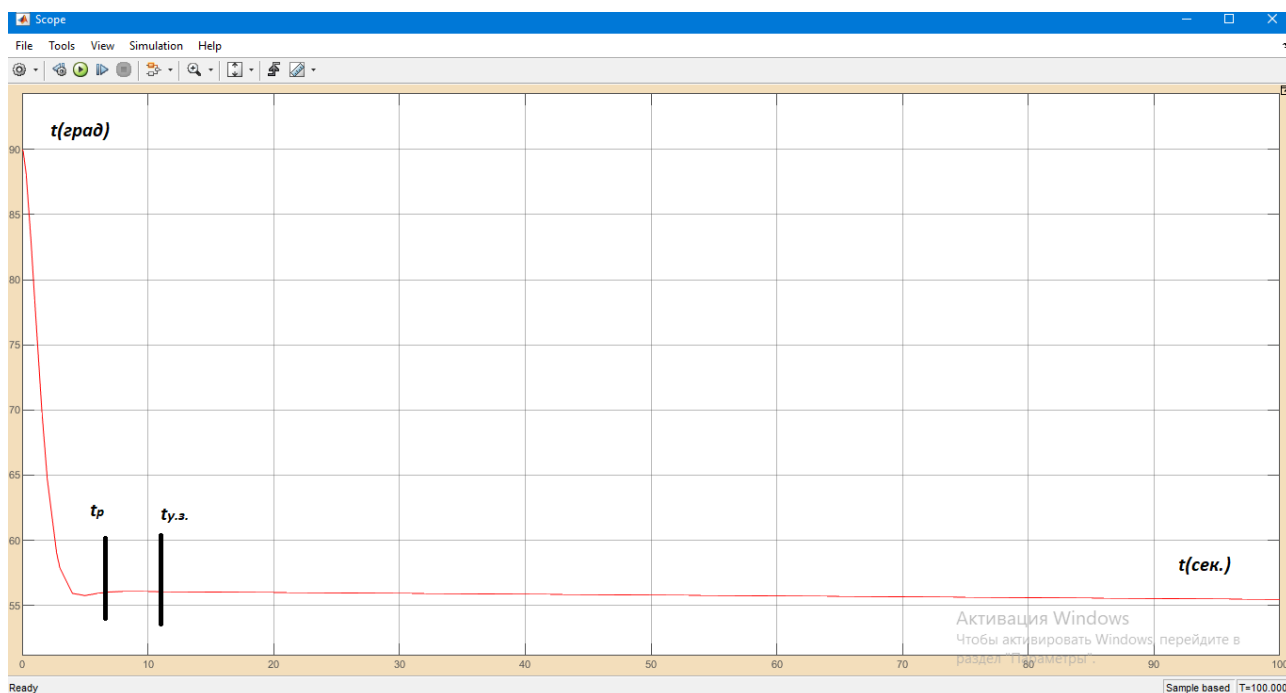


Рисунок 4.3 – Итоговый график переходного процесса

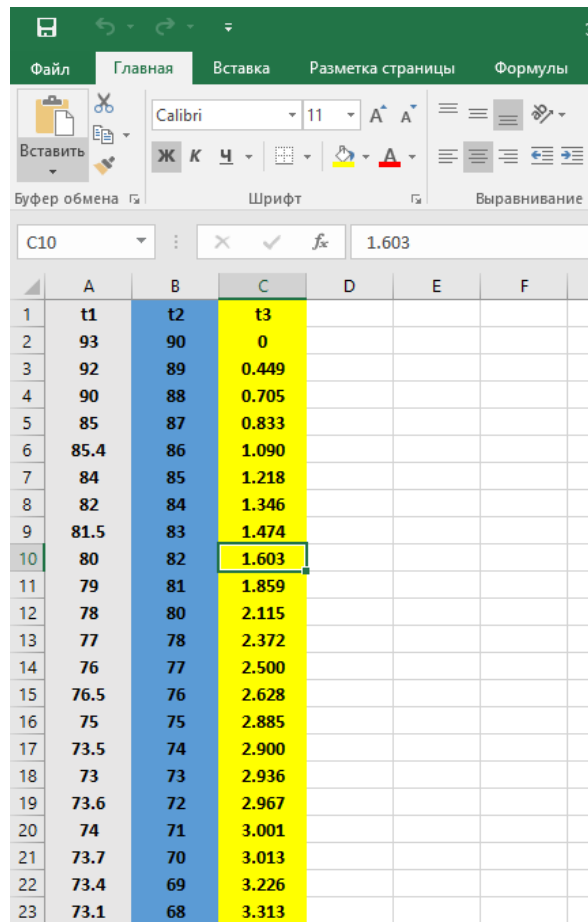
Как можно видеть из рисунка 4.3, значение стабилизации технологической температуры газа на выходе АВО составило примерно 55,5 °С, что является недопустимым значением для реального технологического процесса. При этом время регулирования составило примерно 6 с ($t_p = 6$ с), а время установившегося значения ($t_{y.з.}$) порядка 10 с.

Для технологического охлаждения газа оптимальным значением является диапазон температур 65–69 °С. В связи с этим, необходимо провести идентификацию и оптимизацию системы для получения необходимого переходного процесса [18].

5 Идентификация модели СУ с помощью пакета Identification System Toolbox

Для проведения идентификации будем использовать экспериментальные данные и полученные данные переходного процесса, которые представлены на рисунке 4.3.

Набор экспериментальных данных представлен на рисунке 5.1:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	t1	t2	t3			
2	93	90	0			
3	92	89	0.449			
4	90	88	0.705			
5	85	87	0.833			
6	85.4	86	1.090			
7	84	85	1.218			
8	82	84	1.346			
9	81.5	83	1.474			
10	80	82	1.603			
11	79	81	1.859			
12	78	80	2.115			
13	77	78	2.372			
14	76	77	2.500			
15	76.5	76	2.628			
16	75	75	2.885			
17	73.5	74	2.900			
18	73	73	2.936			
19	73.6	72	2.967			
20	74	71	3.001			
21	73.7	70	3.013			
22	73.4	69	3.226			
23	73.1	68	3.313			

Рисунок 5.1 – Экспериментальные данные

Здесь столбец t_1 – входное воздействие, столбец t_2 – данные на выходе, t_3 – время переходного процесса.

Далее, нам необходимо идентифицировать эти данные с помощью пакета System Identification Toolbox [19].

Для этого, последовательно введем значения t_1 , t_2 , t_3 в окно Command Window среды Matlab. Нужно ввести «t1=», нажать Enter, скопировать столбец данных t_1 из файла «Data.xls» и ввести «;». На рисунках 5.2, 5.3, 5.4

показано как должны быть введены данные для первого, второго и третьего столбца:

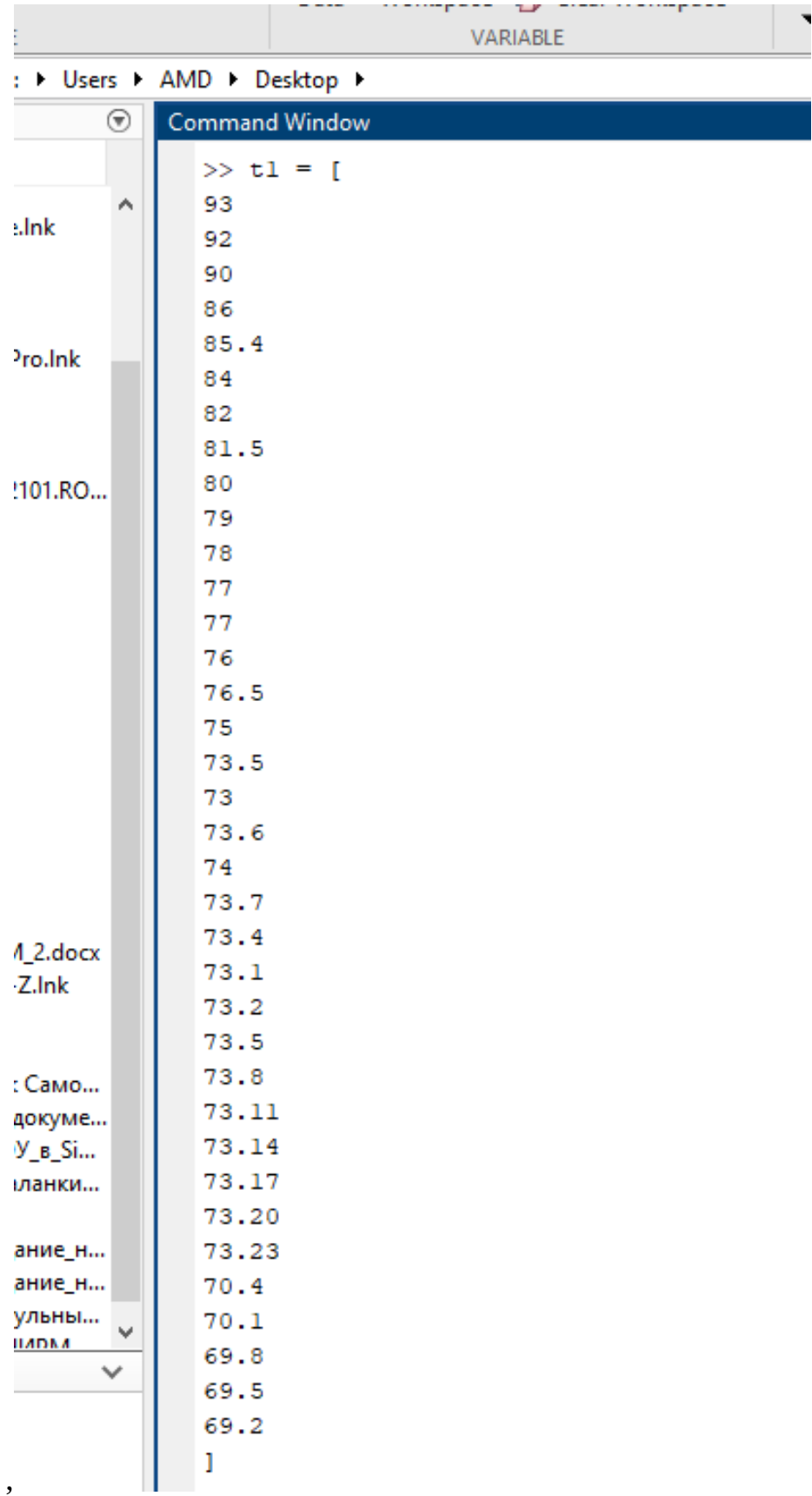


Рисунок 5.2 – Ввод данных для первого столбца

```
>> t2 = [  
90  
89  
88  
87  
86  
85  
84  
83  
82  
81  
80  
79  
78  
77  
76  
75  
74  
73  
72  
71  
70  
69  
68  
67  
66  
65  
64  
63  
62  
61  
60  
59  
58  
57  
56  
55  
];
```

Рисунок 5.3 – Ввод данных для второго столбца

```
>> t3 = [  
0  
0.449  
0.705  
0.833  
1.090  
1.218  
1.346  
1.474  
1.603  
1.859  
2.115  
2.244  
2.372  
2.500  
2.628  
2.885  
2.900  
2.936  
2.967  
3.001  
3.013  
3.226  
3.313  
3.514  
3.756  
3.789  
3.890  
4.089  
4.167  
4.289  
4.551  
6.136  
6.897  
7.597  
7.985  
8.141  
];
```

Рисунок 5.4 – Ввод данных для третьего столбца

Как видно из рисунка 5.5, введенные данные отобразились в окне Workspace:

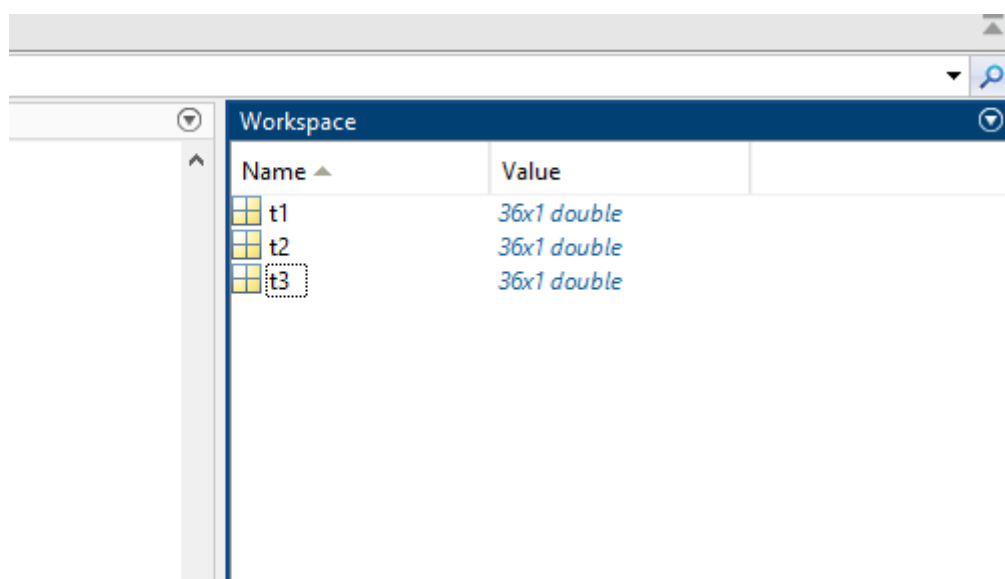


Рисунок 5.5 – Отображение данных в окне Workspace

Теперь нам необходимо построить экспериментальные кривые по полученным данным, для этого нужно ввести в Command Window набор команд, представленных на рисунке 5.6:

```
] ;  
>> hold all  
>> plot(t3,t1)  
>> plot(t3,t2)
```

Рисунок 5.6 – Набор команд

Hold all –строит поле для отображения графика, plot(t3,t1) – строит график входных данных, plot(t3,t2) – строит график выходных данных.

В результате введенных команд, должны отобразиться такие же графики, как на рисунке 5.7:

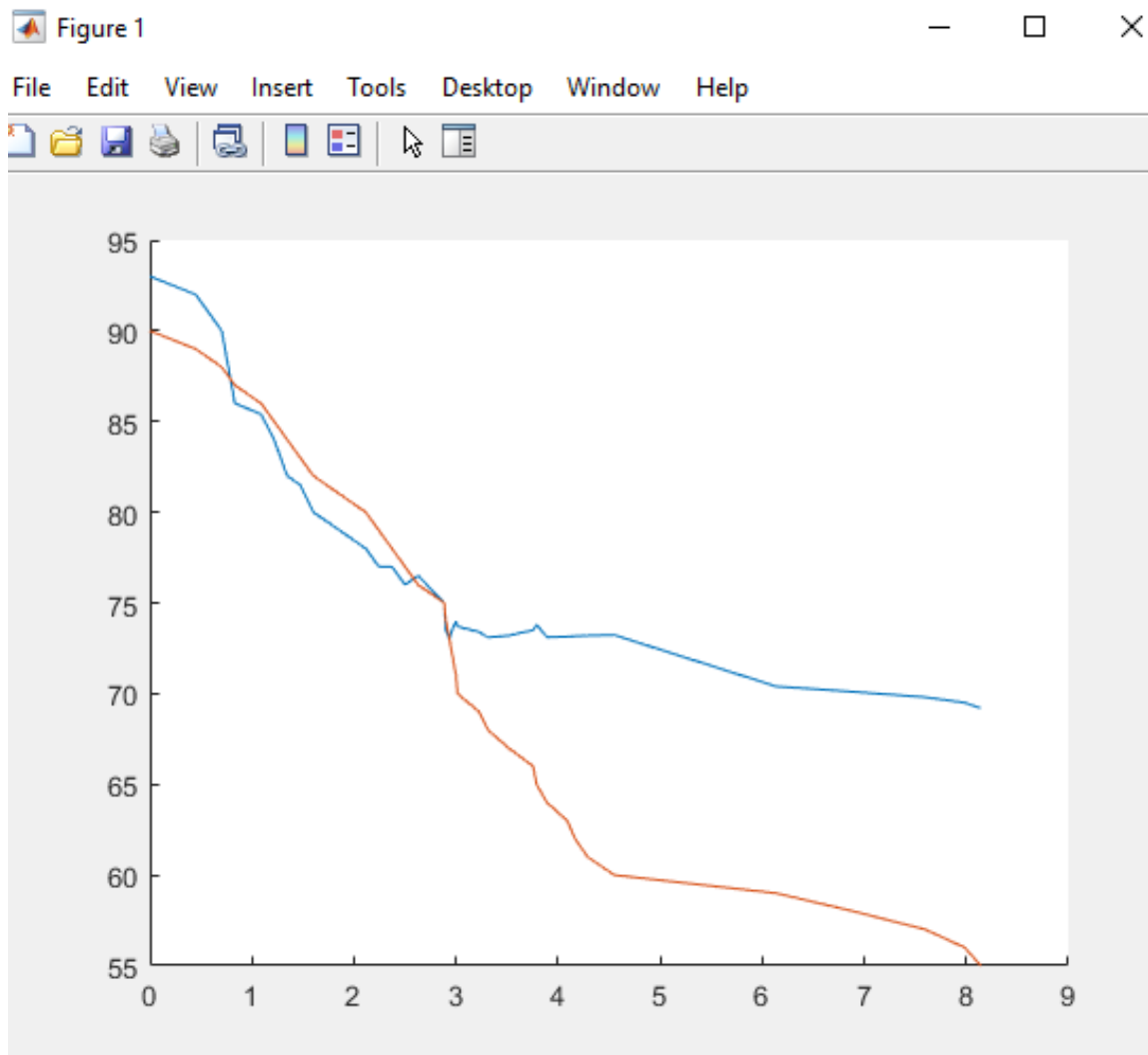


Рисунок 5.7 – Полученные графики

График, который отображен синим цветом, является графиком входных данных, а график, который отображен красным цветом – график выходных данных СУ.

Для использования программой System Identification Toolbox, в окно Command Window нужно ввести команду, которая представлена на рисунке 5.8:

```
>> ident
```

Рисунок 5.8 – Команда для запуска Identification System Toolbox

Откроется диалоговое окно, представленное на рисунке 5.9:

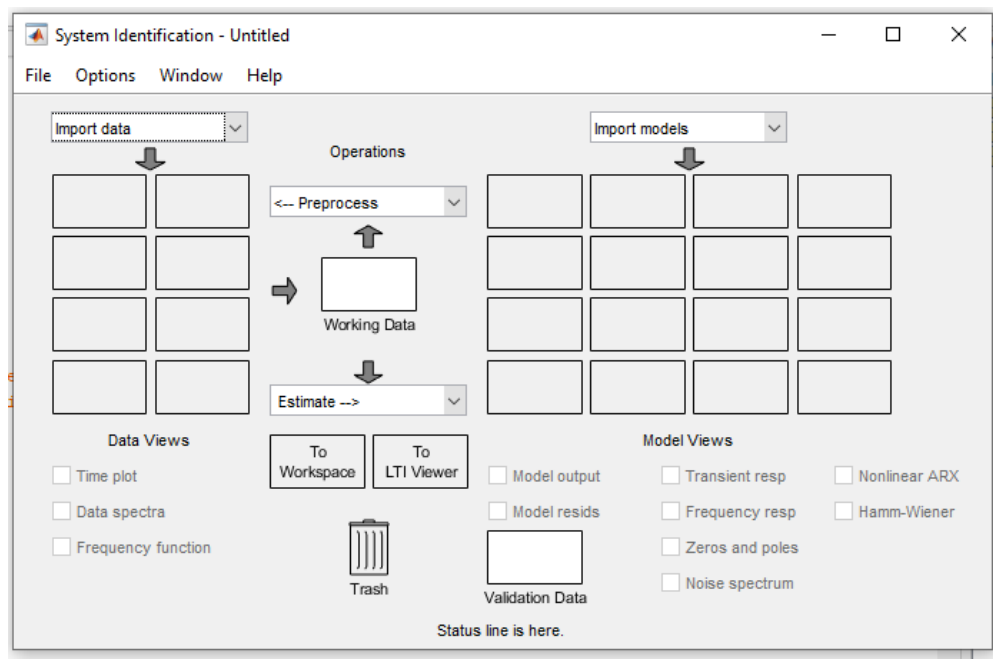


Рисунок 5.9 – Окно Identification System Toolbox

Произведем настройку нашей системы, для этого необходимо ввести наши данные. Чтобы это сделать, нужно выбрать опцию Time Domain Data (рисунок 5.10):

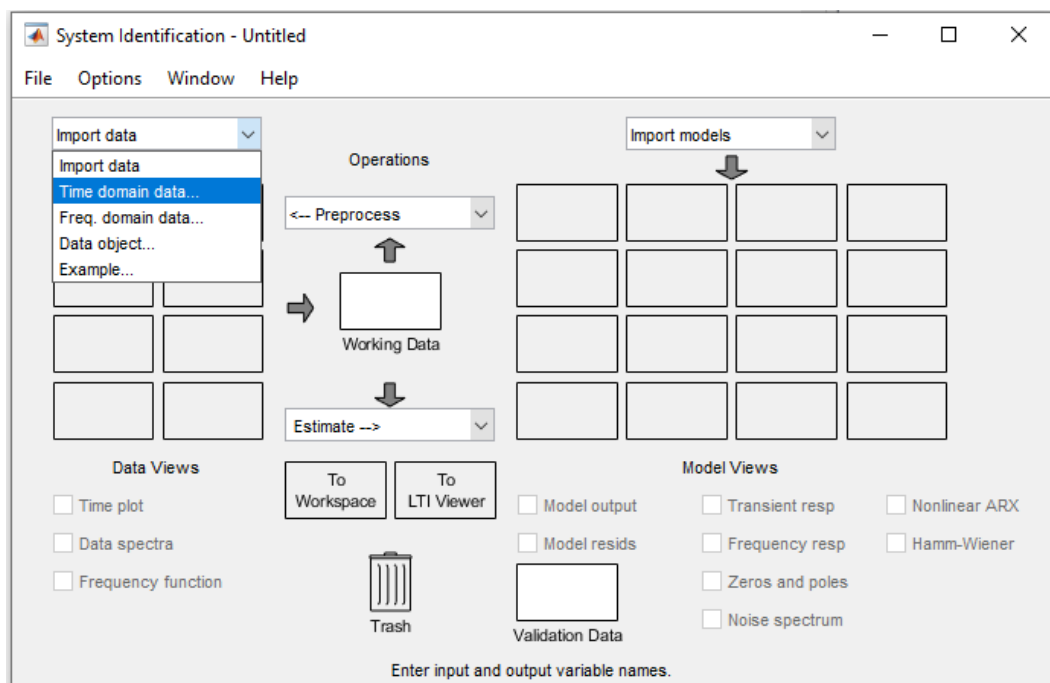


Рисунок 5.10 – Опция Time Domain Data

Затем необходимо ввести переменные входа (Input), выхода (Output) и интервала времени (Sample Time) как на рисунке 5.11. Кроме того, необходимо указать начало отсчета времени (Starting Time = 0). Далее, необходимо нажать Import.

The screenshot shows the 'Import Data' dialog box with the following settings:

- Data Format for Signals:** Time-Domain Signals
- Workspace Variable:** Input: t1, Output: t2
- Data Information:** Data name: mydata, Starting time: 0, Sample time: 0.150
- Buttons:** Import, Reset, Close, Help

Рисунок 5.11 – Задание параметров

Отобразится диалоговое окно, в котором отображен тип полученной зависимости и дополнительные настройки, которыми мы будем пользоваться далее (рисунок 5.12):

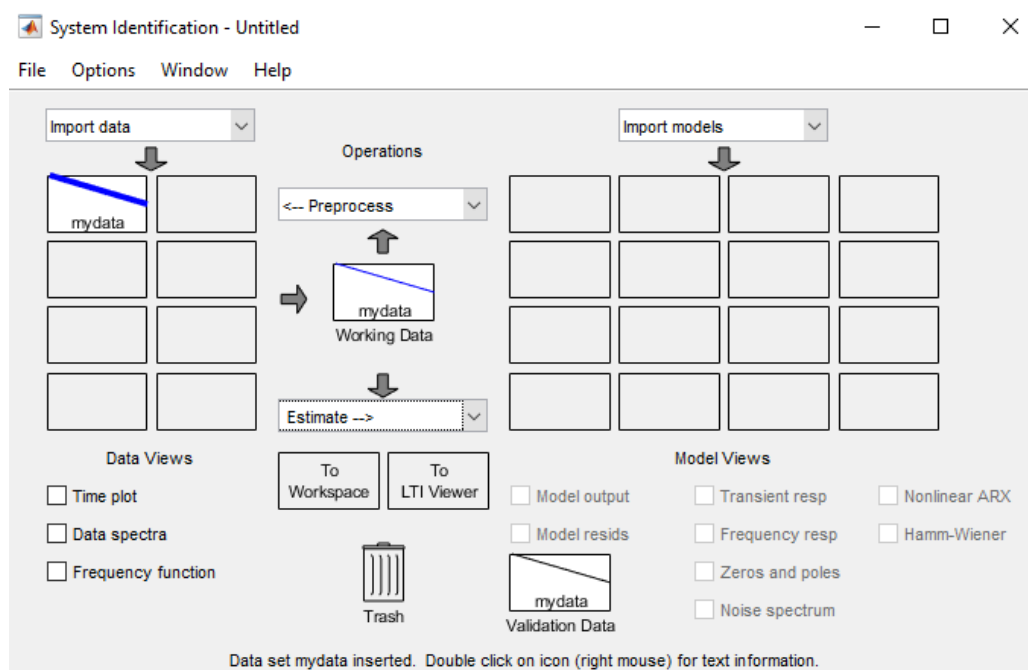


Рисунок 5.12 – Окно настроек

Для настройки идентификации выберем функцию Process Models (рисунок 5.13):

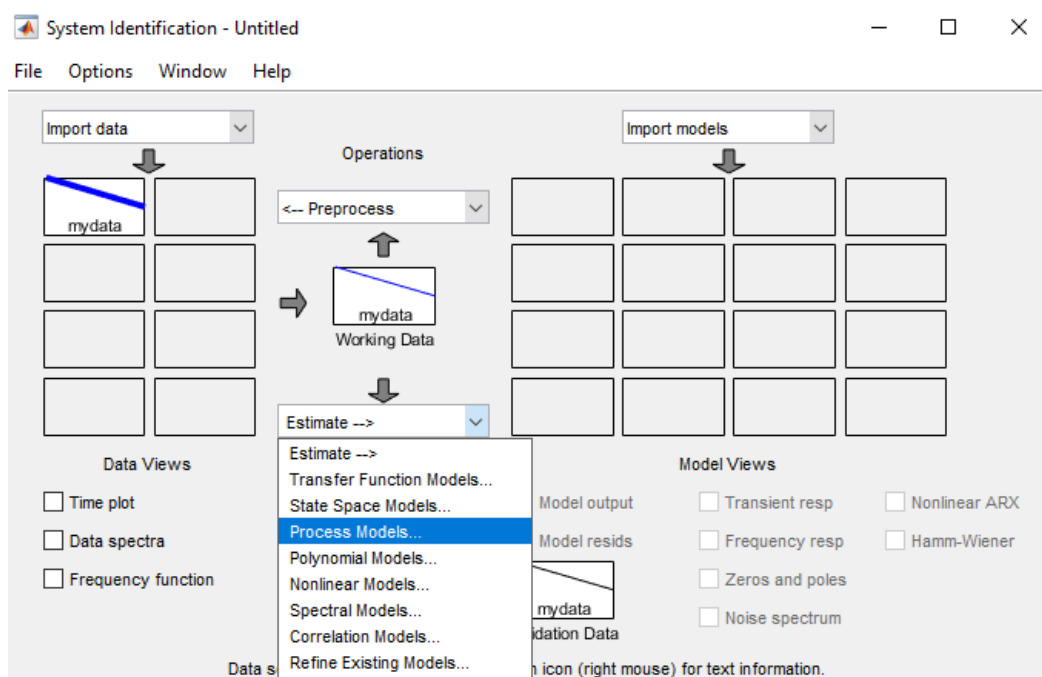


Рисунок 5.13 – Выбор функции

Для начала в качестве идентификации модели выберем точность апериодического звена 1-ого порядка и нажать Estimate (рисунок 5.14):

Process Models

Transfer Function

$$\frac{K \exp(-T_d s)}{(1 + T_p1 s)}$$

Poles

1 All real

☐ Zero

☒ Delay

☐ Integrator

Par	Known	Value	Initial Guess	Bounds
K	☐		Auto	[-Inf Inf]
Tp1	☐		Auto	[0 Inf]
Tp2	☐	0	0	[0 Inf]
Tp3	☐	0	0	[0 Inf]
Tz	☐	0	0	[-Inf Inf]
Td	☐		Auto	[0 4.5]

Initial Guess

☒ Auto-selected

☐ From existing model:

☐ User-defined

Disturbance Model: None

Initial condition: Auto

Focus: Simulation

Covariance: Estimate

☐ Display progress

Name:

Рисунок 5.14 – Начало идентификации

Отобразятся новые графики, представленные на рисунке 5.15. Здесь черным цветом система идентифицировала выходной набор данных, который мы вводили вручную, а синим цветом обозначен переходный процесс, который рассчитывался на основе математической модели и является более ТОЧНЫМ.

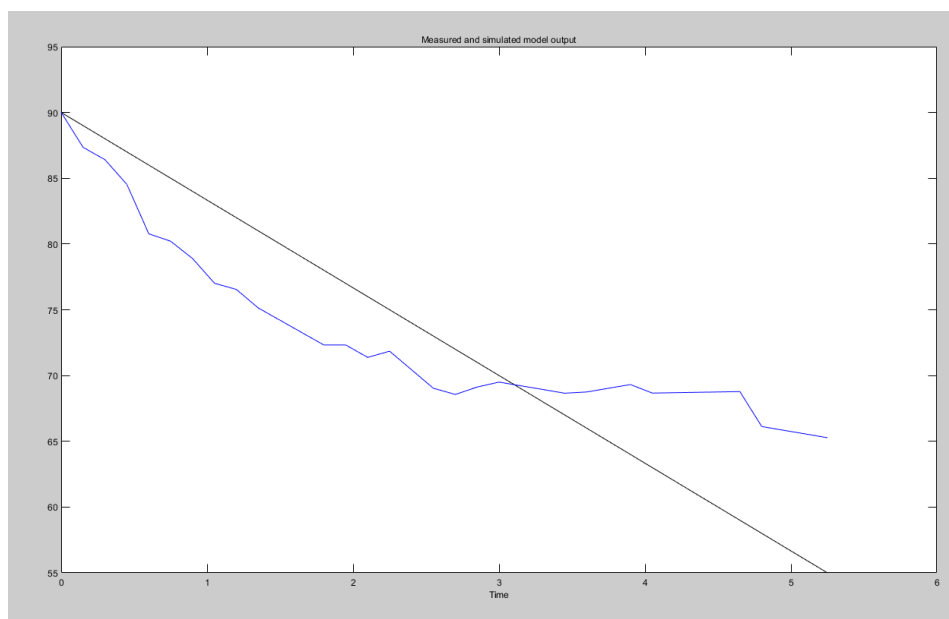


Рисунок 5.15 – Идентификация системы на основе аperiodического процесса 1-ого порядка.

При нажатии на полученный график в окне Model Views мы можем посмотреть значения полученной передаточной функции системы (рисунок 5.16):

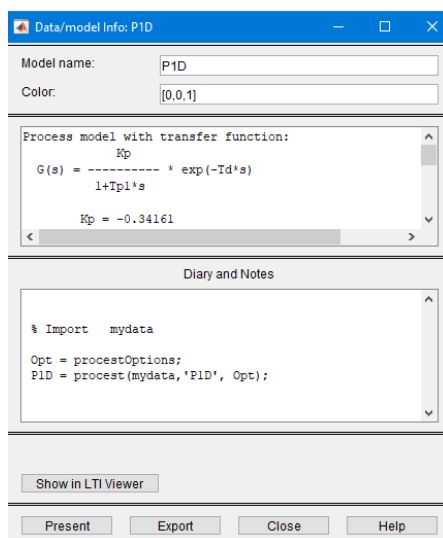


Рисунок 5.16 – Значения функции идентифицированной модели

Попробуем увеличить точность идентификации, для этого вернемся в Process Models и в окне Poles выберем значение «2», то есть будем проводить идентификацию с помощью апериодического звена 2-ого порядка, так как оно может дать большую точность при построении. После выбора значения, нажимаем Estimate (рисунок 5.17):

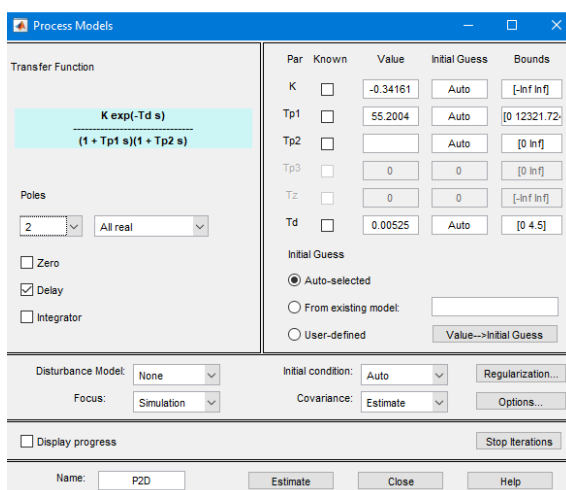


Рисунок 5.17 – Выбор значений

Как видно из рисунка 5.18, расчетная модель показала более высокую точность построения на интервале времени 0-0.3 секунды.

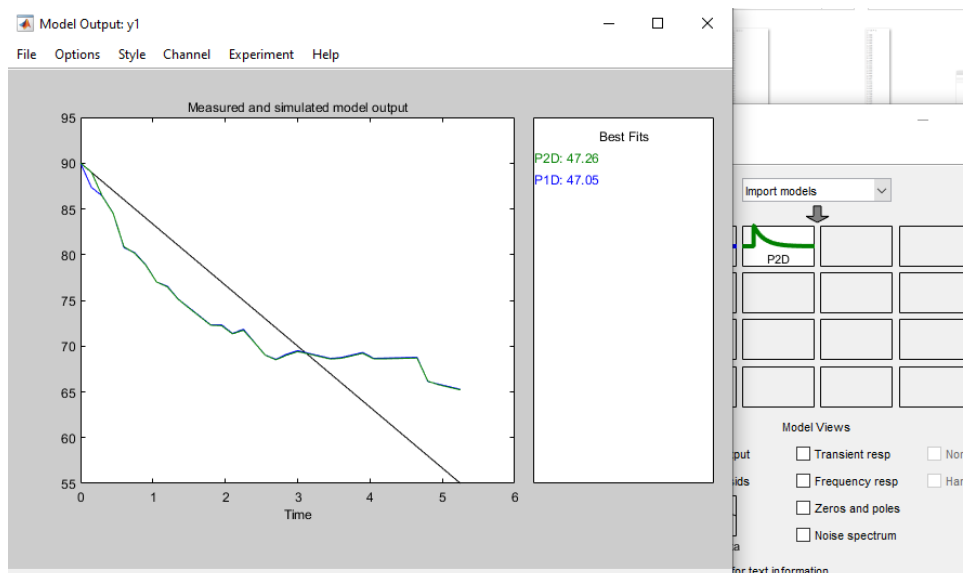


Рисунок 5.18 – Полученные значения

Аналогично, в окне Data/model info можем посмотреть получившуюся передаточную функцию, описывающую модель. Она состоит из апериодического звена второго порядка с транспортным запаздыванием. Полученные значения на рисунке 5.19 будем использовать при расчете настроек регулятора, анализу устойчивости модели, а также ее оптимизации в целях установления соответствия адекватности полученных данных.

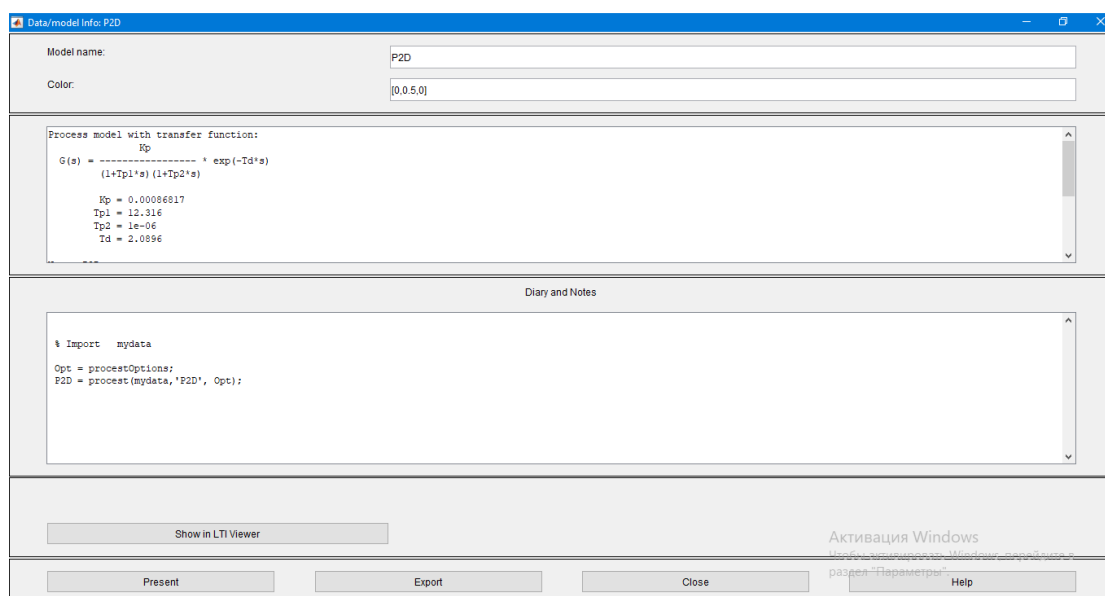


Рисунок 5.19 – Итоговая модель

Полученные значения передаточной функции занесем в блоки TransferFcn 1,2 модели стабилизации температуры, а также добавим блок TransportDelay с установленной задержкой по времени так, как показано на рисунке 5.20.

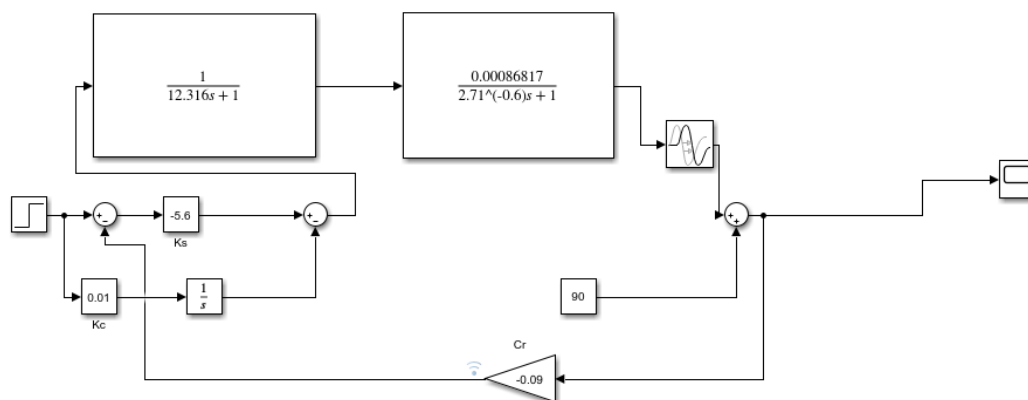


Рисунок 5.20 – Модель системы после идентификации

6 Настройка ПИД - регулятора модели СУ

Для проектирования регулирующего органа СУ [20] необходимо из библиотеки компонентов Simulink добавить два новых блока: PID(s) и Gain, внешний вид представлен на рисунках 6.1 и 6.2:



Рисунок 6.1 – Изображение блока PID(s)



Рисунок 6.2 – Изображение блока PID(s)

В данном случае, у ПИД-регулятора блок Gain с коэффициентом 0,075 отвечает за коэффициент передачи тепла от теплообменного аппарата к стенке трубы.

Далее, необходимо изменить структуру модели СУ так, как показано на рисунке 6.3:

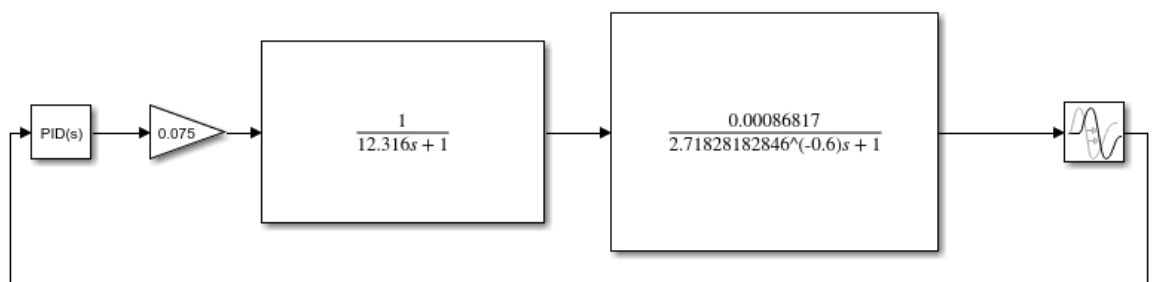


Рисунок 6.3 – ПИД-регулятор СУ

Итоговая модель СУ должна быть изменена так, как показано на рисунке 6.4:

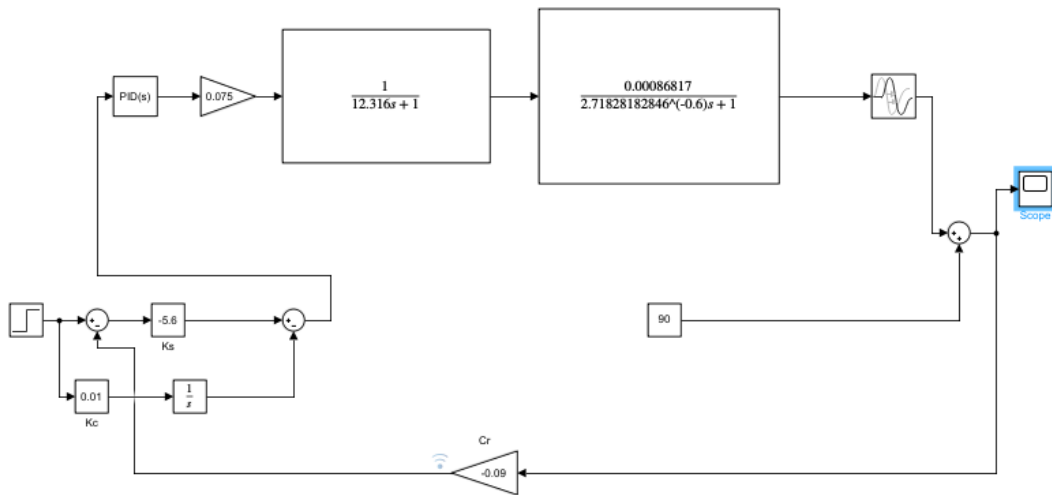


Рисунок 6.4– Итоговая модель СУ

Для настройки ПИД-регулятора необходимо открыть блок PID(s), откроется диалоговое окно, представленное на рисунке 6.5:

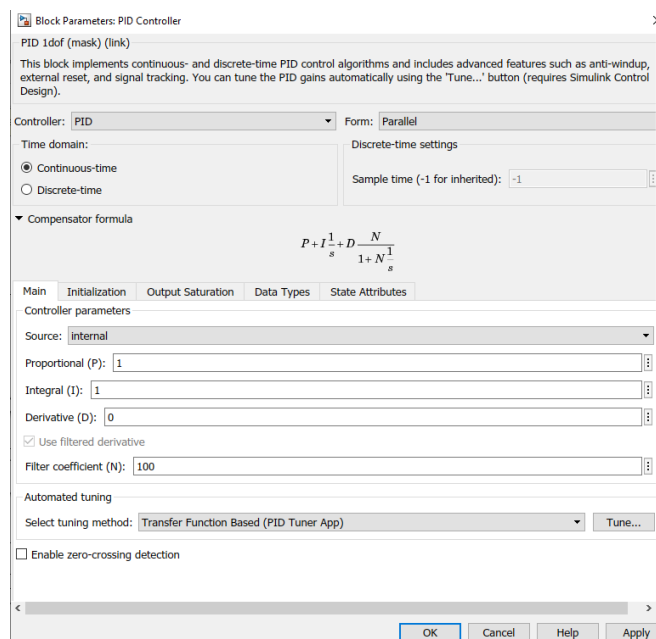


Рисунок 6.5 – Окно настроек ПИД – регулятора

С помощью опции Tune (рисунок 6.6) получены оптимальные числовые значения для передаточной функции, которые представлены на рисунке 6.8. После настройки значений в модуле Tune необходимо нажать кнопку Update Blocks (рисунок 6.7):

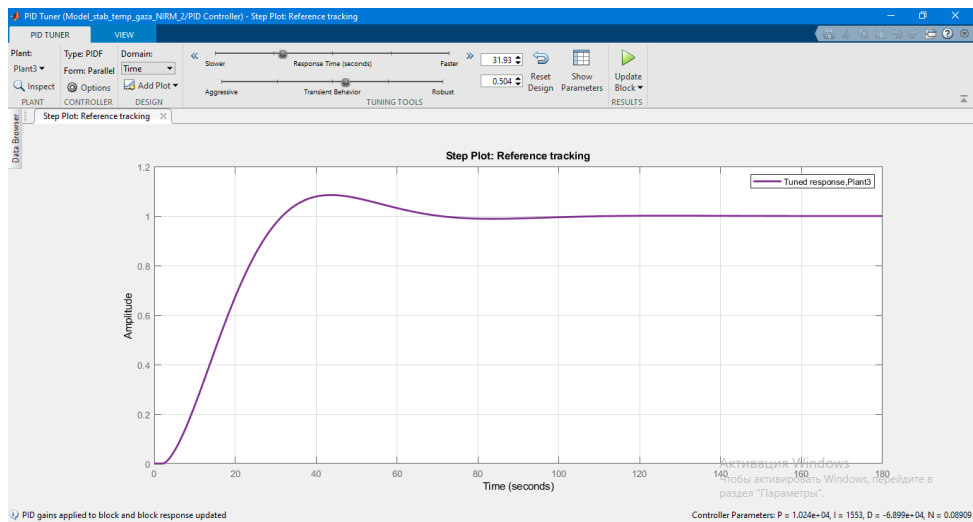


Рисунок 6.6 – Окно настроек функции Tune

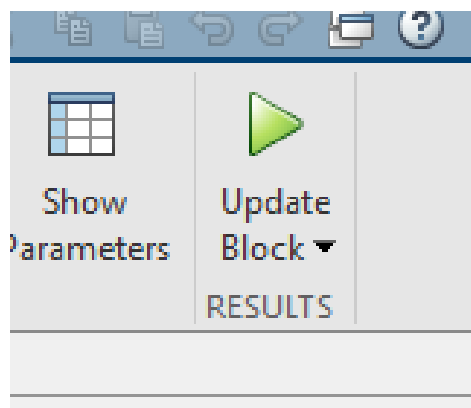


Рисунок 6.7 – Запуск настроек регулятора

Рисунок 6.8 – Оптимальные значения настроек регулятора

В результате, после повторной компиляции модели получен новый переходный процесс стабилизации температуры, представленный на рисунке 6.9.

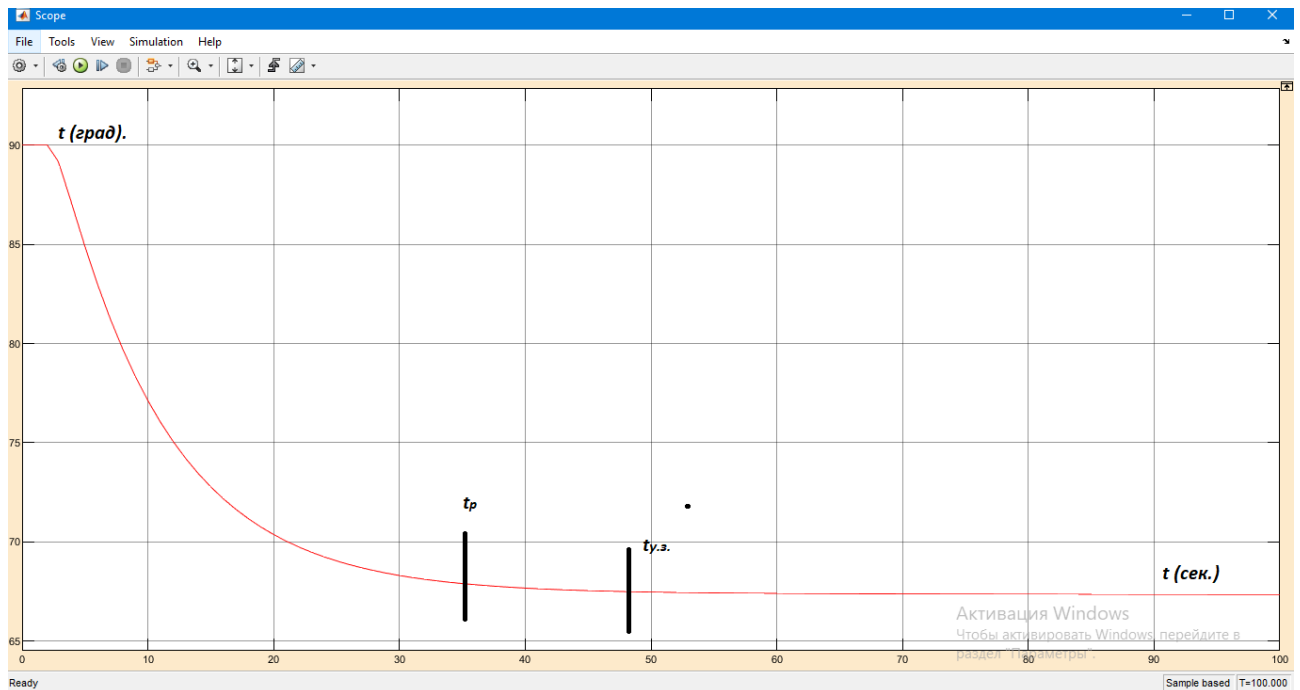


Рисунок 6.9 – Переходный процесс после изменения настроек регулятора

Как можно заметить, температура стабилизации на этот раз удовлетворяет заданному диапазону значений и составила примерно 67°C

Но время регулирования и установившегося значения при этом составило порядка 24 и 42 секунд.

Для того, чтобы достичь оптимального времени переходного процесса (5 – 10 с) проведем оценку адекватности, устойчивости системы и оптимизируем ее с помощью пакета System Optimization Toolbox.

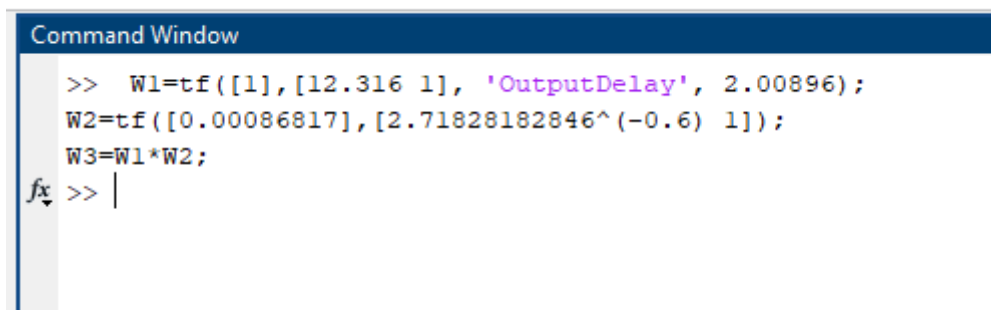
7 Устойчивость и адекватность системы ОУ, оптимизация параметров системы

7.1 Устойчивость системы

Оценку устойчивости системы будем производить с помощью критерия Найквиста [21].

Критерий Найквиста позволяет определить устойчивость системы с отрицательной обратной связью (так называемой замкнутой системы) по амплитудно-фазовой характеристике разомкнутой системы. Это означает, что для устойчивости замкнутой системы необходимо, чтобы амплитудно-фазовая характеристика не охватывала начало координат. Для этого на практике принято брать точку с координатами $(-1, j0)$, тогда система считается устойчивой.

Для этого воспользуемся встроенными операторами языка Matlab, необходимо ввести значения передаточной функции, полученные в разделе 2 так, как показано на рисунке 7.1



```
Command Window
>> W1=tf([1],[12.316 1], 'OutputDelay', 2.00896);
W2=tf([0.00086817],[2.71828182846^(-0.6) 1]);
W3=W1*W2;
fx >> |
```

Рисунок 7.1 – Ввод значений

Первая строка осуществляет ввод передаточной функции Transfer Fcn 1 (раздел 5) с транспортным запаздыванием, данные о ней заносятся в переменную $W1$. Аналогично во второй строке инициализируется значение для Transfer Fcn, которая заносится в переменную $W2$. В переменной $W3$ выполнено произведение функций $W1$ и $W2$ (так как у нас в разделе 2 выбрана модель апериодического звена 2-ого порядка и представлена в виде двух последовательно соединенных апериодических звеньев 1-ого порядка.)

Чтобы построить годограф Найквиста, необходимо ввести функцию, представленную на рисунке 7.2:

```
nyquist(W3)
```

Рисунок 7.2 – Функция построения годографа Найквиста

После ввода функции и нажатия клавиши Enter отобразится график, представленный на рисунке 7.3:

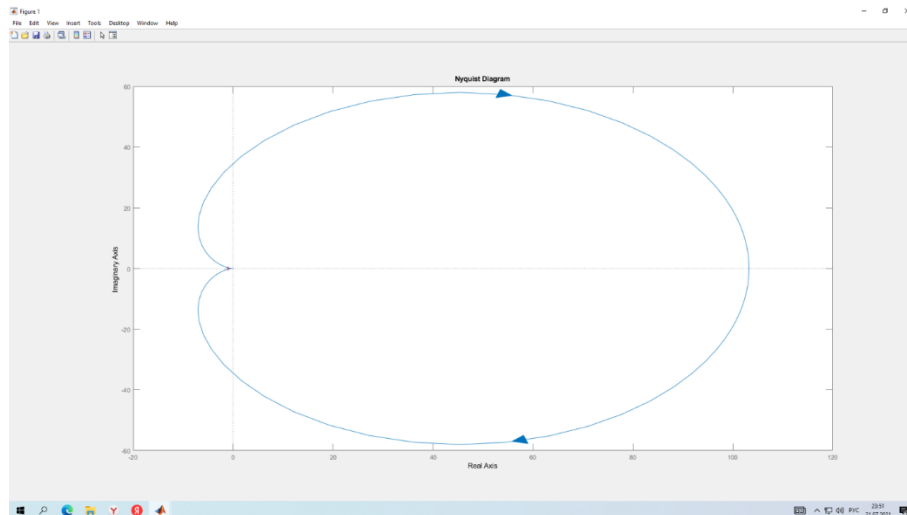


Рисунок 7.3 – Полученный годограф Найквиста

Как мы видим из рисунка 7.4, годограф не охватывает точку с координатами $(-1, j0)$, следовательно система является устойчивой [22].

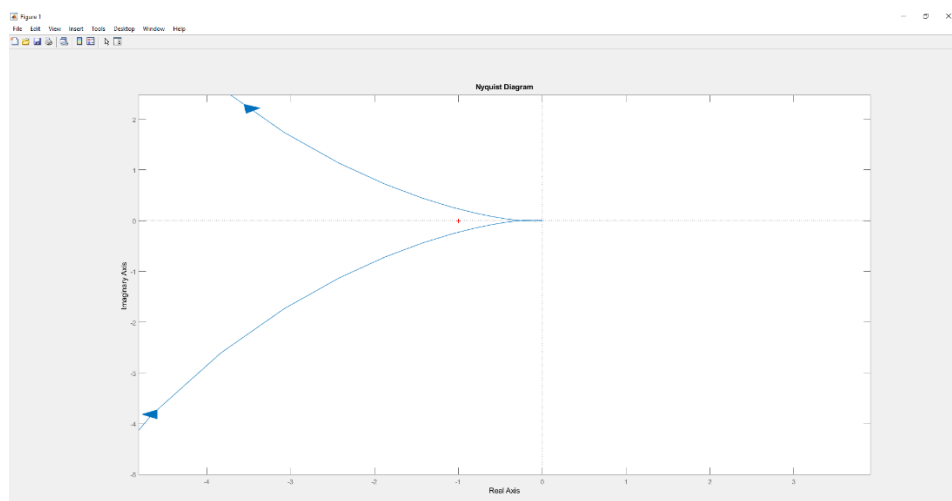


Рисунок 7.4 – Точка охвата годографа Найквиста

Для нахождения полюсов системы необходимо ввести функцию, представленную на рисунке 7.5:

```
--  
>> pole(W3)
```

Рисунок 7.5 – Функция для нахождения полюсов модели СУ

Полученные значения полюсов СУ с помощью функции `pole` [23] представлены на рисунке 7.6:

```
ans =  
  
-1.8221  
-0.0812
```

Рисунок 7.6 – Значения полюсов модели СУ

7.2 Адекватность системы управления

Основная цель моделирования – определение назначения проектируемой модели и установление тех свойств оригинального объекта, которые должны отражаться в разрабатываемой СУ.

Под адекватностью модели будем понимать совпадение этих самых свойств моделируемой модели по отношению к реальному исследуемому объекту.

Для того, чтобы удостовериться в том, что модель СУ является адекватной и ее найденные в разделах 5-6 значения верны, необходимо провести оптимизацию модели с целью установления оптимальных значений коэффициентов уравнений СУ [24, 25].

Для этого, воспользуемся инструментами Optimization Toolbox среды Simulink.

Во-первых, необходимо заменить значения полученной в разделе 5 передаточной функции на коэффициенты, представленные на рисунке 7.7:

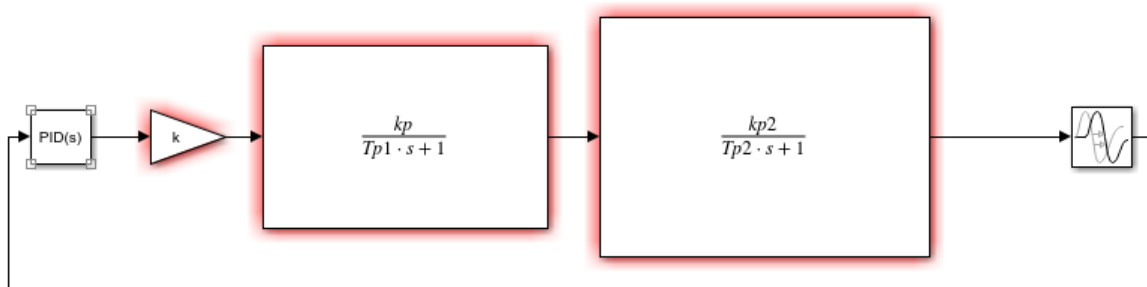


Рисунок 7.7 – Замена значений

После этого, необходимо указать значения этих коэффициентов в окне Command Window (значения коэффициентов берутся из полученной модели, раздел 5), на рисунке 7.8 можно также увидеть необходимые для ввода параметры системы:

```

Command Window

>> k=0.075;
>> Tp1=12.316;
>> Tp2=1*exp(-0.6);
>> kp=1;
>> kp2=0.0086817;
fx >> |

```

Рисунок 7.8 – Необходимые значения коэффициентов

После ввода значений, нам необходимо добавить в нашу модель СУ новый блок типа Signal Constraint, для этого воспользуемся кнопкой Library Browser (рисунок 7.9):

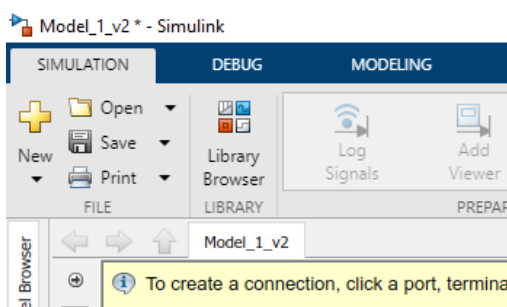


Рисунок 7.9 – Кнопка доступа к библиотеке компонентов Simulink

Далее, необходимо ввести в меню поиска «Signal Constraint» и выбрать блок «Check Step Response Characteristics»[26] (рисунок 7.10):

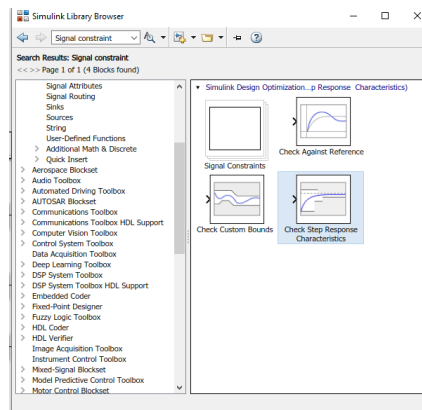


Рисунок 7.10 – Выбор необходимого блока

После этого, нужно выбранный блок подключить к выходу системы и открыть его для настройки параметров. Необходимые для настройки значения показаны на рисунке 7.11:

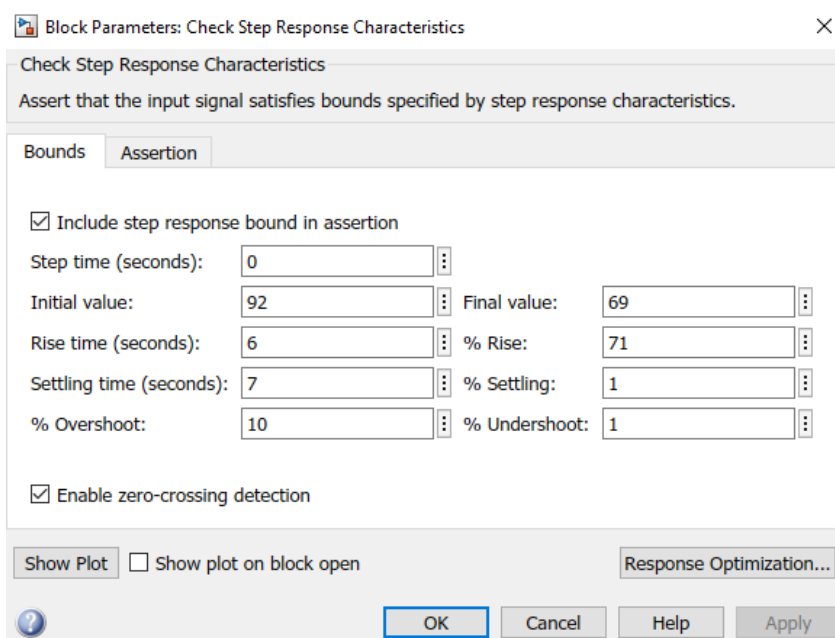


Рисунок 7.11 – Необходимые параметры для настроек оптимизации

Здесь нас интересуют следующие значения: Step Time – шаг модульного времени (начало отсчета), Initial Value – начальное значение переходного процесса, Rise Time – время нарастания переходного сигнала, Setting Time – время регулирования переходного процесса, %Overshoot – перерегулирование, Final Value – значение в установившемся режиме, % Rise – амплитуда нарастания сигнала.

После ввода всех значений, необходимо нажать кнопку «Response Optimization» и дождаться появления диалогового окна, которое представлено на рисунке 7.12:

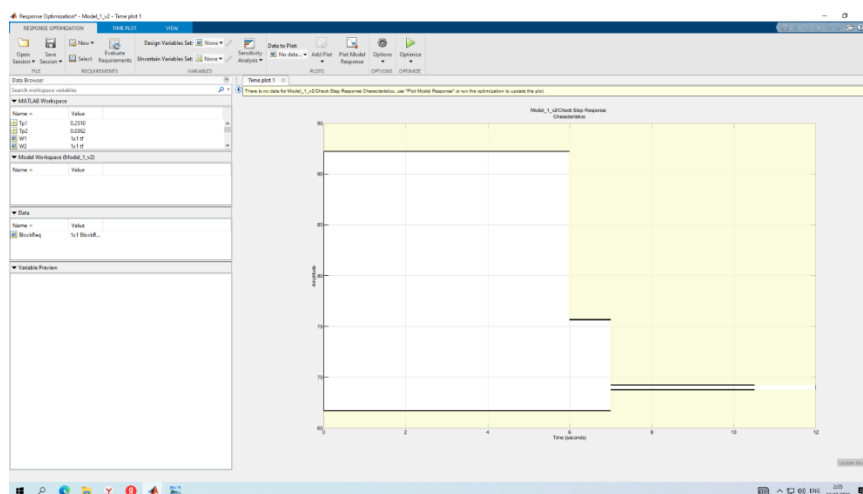


Рисунок 7.12 – Диалоговое окно Response Optimization

Как можно видеть, построен ступенчатый график переходного процесса по нашим введенным значениям. Для того, чтобы график для оптимизации был максимально точным, необходимо нажать кнопку «Evaluate Requirements» на панели инструментов (рисунок 7.13):

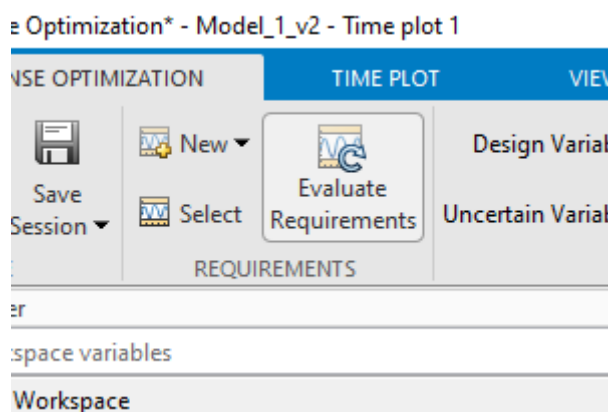


Рисунок 7.13 – Уточнение графика для оптимизации

После нажатия данной кнопки, наше переходный процесс изменится на тот, что получен в Разделе 3 при настройке параметров регулятора (рисунок 7.14):

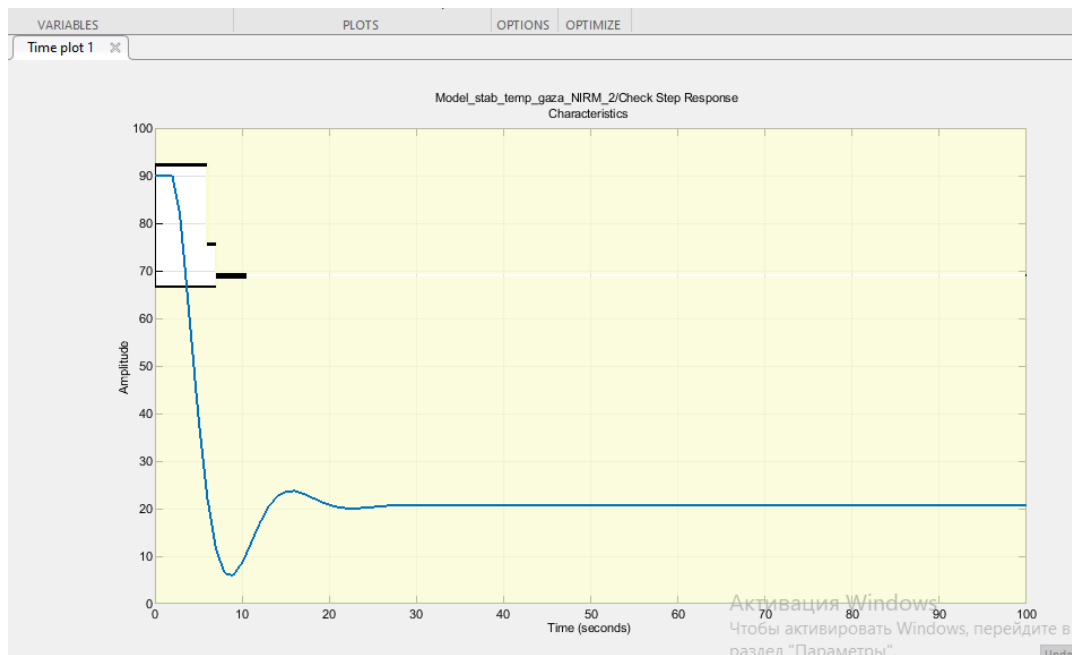


Рисунок 7.14 – Текущий переходный процесс

После этого, нам необходимо инициализировать введенные ранее коэффициенты для оптимизации, для этого нажимаем кнопку «Create New Set of Design Variables» раздела «Design Variables Set» (рисунок 7.15):

Response Optimization* - Model_1_v2 - Time plot 1

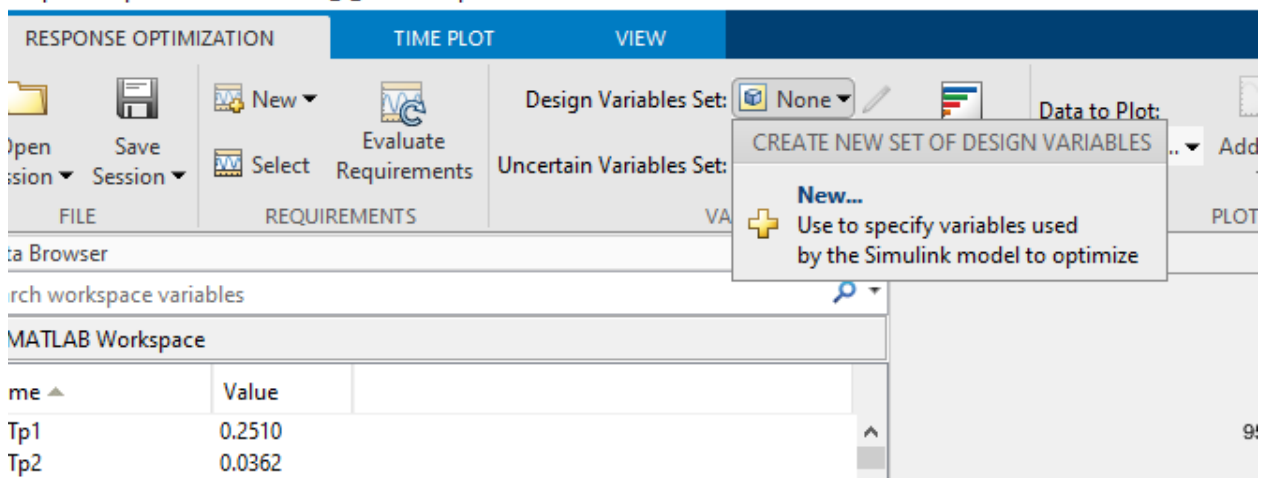


Рисунок 7.15 – Инициализация коэффициентов

Чтобы выбрать наши значения для оптимизации, необходимо выделить конкретное значение и нажать на , таким же образом перенести все остальные значения коэффициентов (должно получиться так, как на рисунке 7.16):

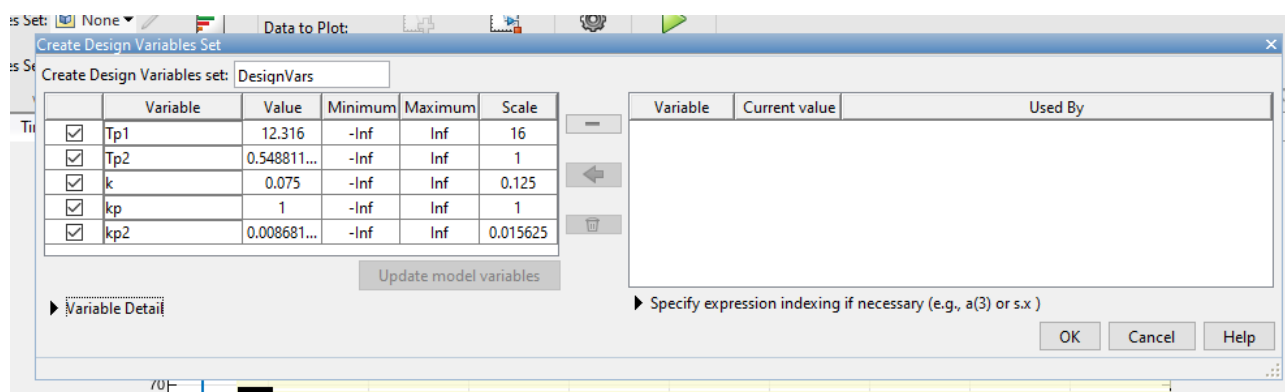


Рисунок 7.16 – Выбор коэффициентов

После выбора всех значений, необходимо нажать кнопку «Optimize» (рисунок 7.17) для начала процесса оптимизации:

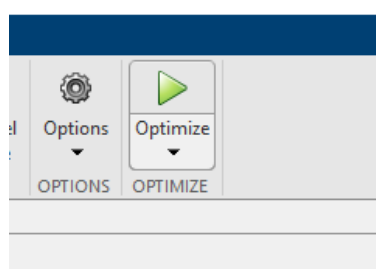


Рисунок 7.17 – Запуск процесса оптимизации

После открытия диалогового окна (рисунок 7.18) начнется процесс оптимизации системы.

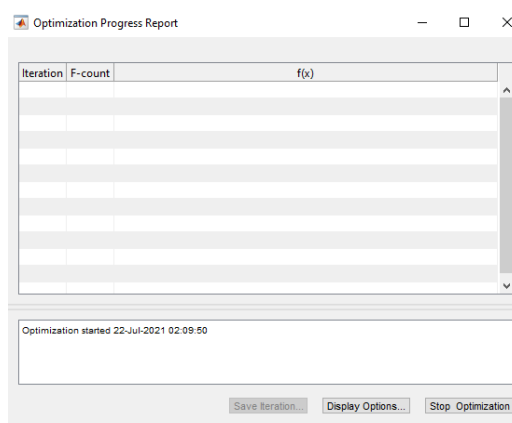


Рисунок 7.18 – Начало процесса оптимизации

Здесь следует отметить, что в зависимости от количества полученных итераций (проходов цикла) процесс может затянуться на достаточно продолжительное время, поэтому важно дождаться его окончания.

Параллельно процессу оптимизации, можно наблюдать изменения построения переходного процесса в окне Response Optimization (рисунок 7.19):

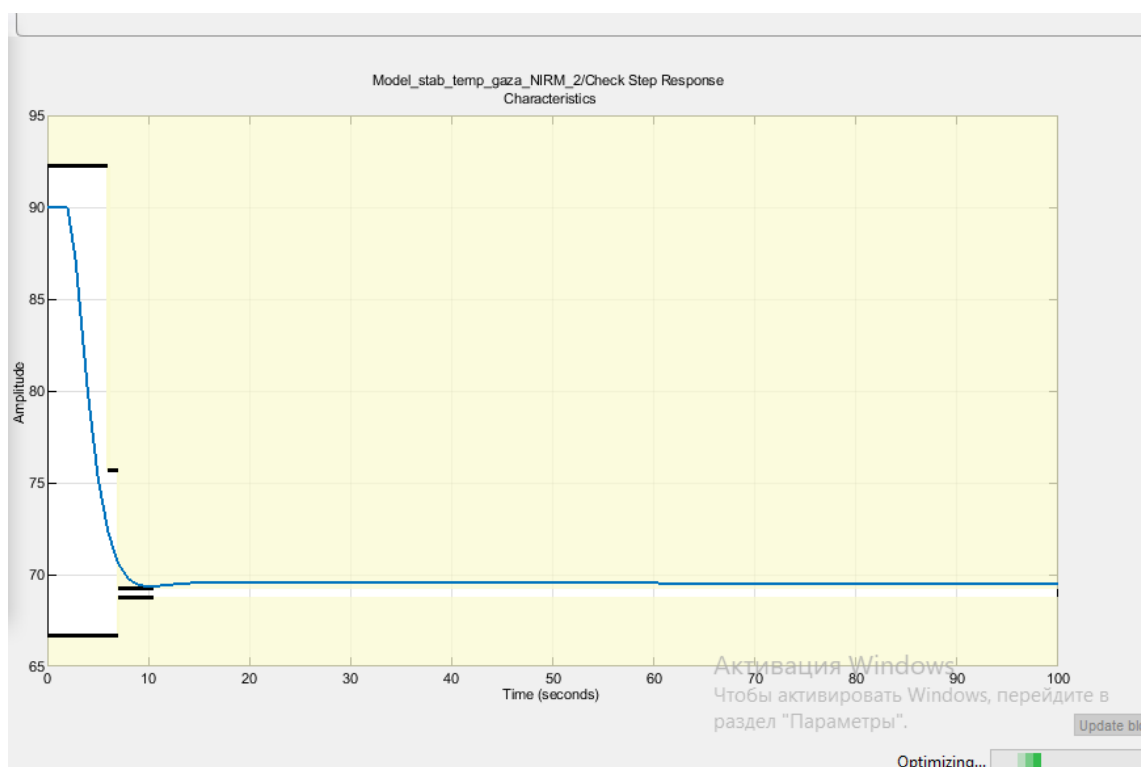


Рисунок 7.19 – Оптимизация модели

После завершения процесса оптимизации, необходимо нажать кнопку «Save Iteration» (рисунок 7.20):

Optimization Progress Report			
Iteration	F-count	Check Step Response Characteristics (Upper)	
3	60	275.3329	
4	78	38.6857	
5	89	1.3688	
6	116	1.3685	
7	147	1.3683	
8	178	1.3681	
9	209	1.3679	
10	239	1.3680	
11	269	1.3680	
12	300	1.3679	
13	331	1.3677	
14	377	1.3677	

Optimization started 21-Apr-2022 11:57:54

Optimization failed to converge, 21-Apr-2022 12:05:11
The optimizer encountered 17 errors during the optimization. Details of the errors have been written to 'EvalErrors' in the Design Optimization workspace.

Save Iteration... Display Options... Optimize

Рисунок 7.20 – Сохранение полученных значений

Кроме того, можно наблюдать изменение всех остальных значений в окне Workspace (рисунок 7.21):

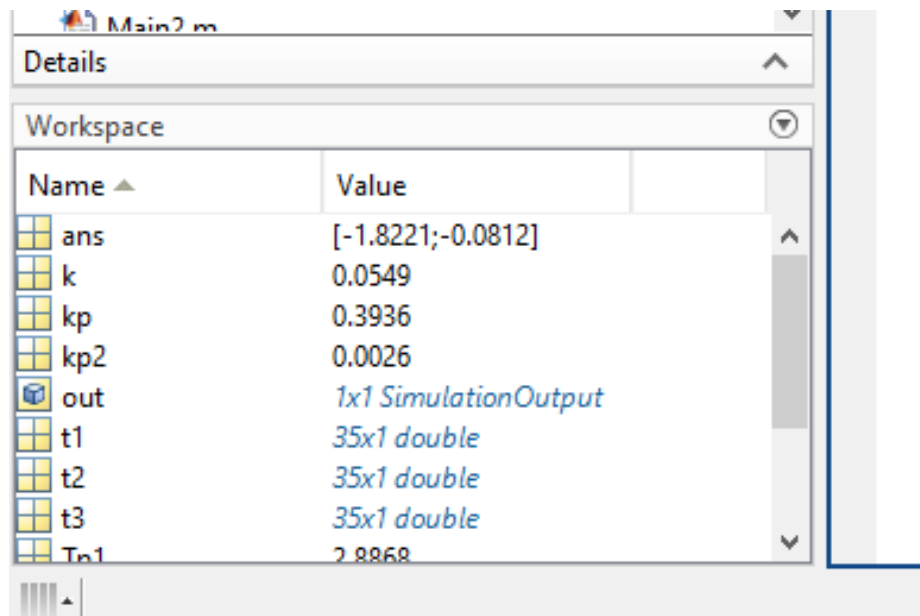


Рисунок 7.21 – Изменение параметров после оптимизации

После этого, необходимо перезапустить компиляцию модели кнопкой «Run» (рисунок 7.22) и получить новый переходный процесс после оптимизации (рисунок 4.23):

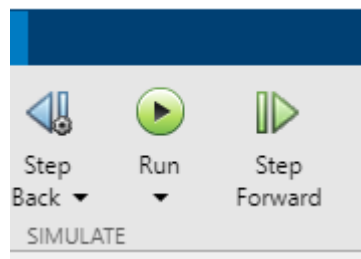


Рисунок 7.22 – Кнопка компиляции и запуска модели

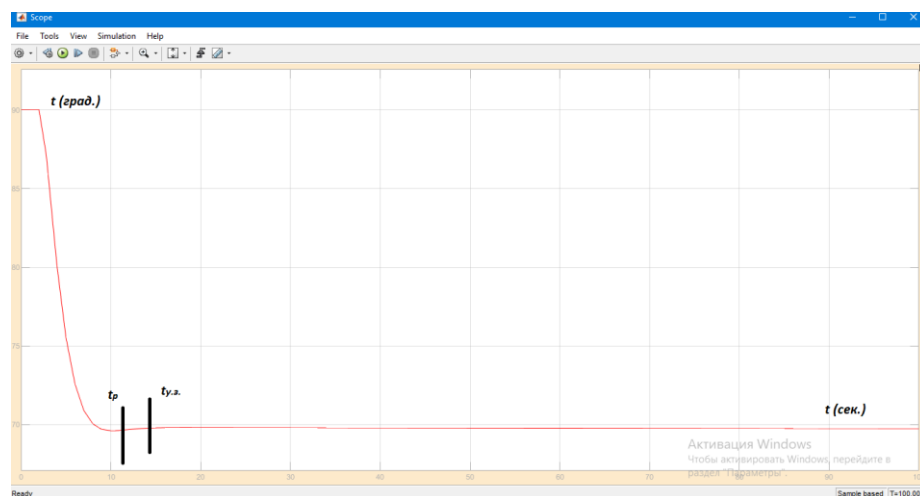


Рисунок 7.23 – Переходный процесс после оптимизации

Как можно заметить, температура стабилизации составила примерно 69°C (удовлетворяет оптимальному диапазону температурных значений).

Время регулирования и установившегося значения при этом сократилось до 9 и 10 секунд, что также удовлетворяет значениям по времени стабилизации реального ОУ.

Таким образом, полученный переходный процесс показал, что модель можно считать адекватной [27].

8 Описание методов регулирования стабилизации температуры на основе уравнения состояния идеального газа

8.1 Описание технологических параметров с помощью уравнения состояния идеального газа (уравнения Клапейрона – Менделеева)

Уравнение Клапейрона – Менделеева [28] является аналитической зависимостью, которая устанавливает зависимость между давлением, молярным объемом и абсолютной температурой идеального газа. Оно имеет вид (8.1):

$$pV = \frac{mRT}{\mu} \quad (8.1)$$

где:

p – давление, Па;

V – объем газа, М³;

$\frac{m}{\mu}$ – количество вещества в молях;

R – универсальная газовая постоянная Менделеева, $8,314 \frac{\text{Дж}}{\text{Моль} \cdot \text{К}}$;

T – термодинамическая температура (функция состояния), К.

В случае постоянной массы газа уравнение можно записать в видах (8.2,8.3):

$$\frac{pV}{T} = \frac{mR}{\mu} \quad (8.2)$$

$$\frac{pV}{T} = \text{const} \quad (8.3)$$

Уравнение (8.3) называют объединенным газовым законом. Из него получаются законы Бойля-Мариотта, Шарля и Гей-Люссака (адиабатный [29] и полиотропный [30] процесс не имеет значения при постановке текущей задачи, поэтому в данном случае не рассматривается).

Таким образом, из объединенного газового закона можно вывести три основных изопроцесса (8.4,8.5,8.6):

$$T = \text{const} \Rightarrow PV = \text{const} \quad (8.4)$$

$$p = \text{const} \Rightarrow \frac{V}{T} = \text{const} \quad (8.5)$$

$$V = const \Rightarrow \frac{p}{T} = const \quad (8.6)$$

Уравнение (8.4) – изотермический процесс (закон Бойля – Мариотта), уравнение (8.5) – изобарный процесс (закон Гей – Люссака), уравнение (8.6) – изохорный процесс (закон Шарля).

В форме (8.7) закон удобен для перевода газа из одного состояния в другое:

$$\frac{P_1 V_1}{T_1} = \frac{P_2 V_2}{T_2} \quad (8.7)$$

8.2 Постановка задачи проектирования для нечеткого регулятора

Как известно, компримирование газа является процессом повышения давления газа с помощью компрессорной станции при транспортировке на магистральных газопроводах.

Очень важно здесь отметить, что при данной технологической операции происходит изохорное увеличение давления. В результате изохорного увеличения давления по первому закону термодинамики происходит увеличение температуры, потому что поступающее от нагревателя тепло на увеличение внутренней энергии, без совершения работы.

Задачей системы регулирования является снижение и стабилизация температуры после выхода из компрессора с помощью аппаратов воздушного охлаждения газа (АВО) до диапазона значений 55–70 °С.

Для этого система регулирования на выходе из компрессора должна произвести изохорное уменьшение давления и стабилизировать температуру до установленного диапазона.

В рамках работы установлено, что оптимальной температурой стабилизации разработанной модели Simulink можно считать диапазон значений 64–69 °С, поэтому необходимо проектировать нечеткий регулятор таким образом, чтобы установившееся значение удовлетворяло данному диапазону.

Так как управлять на входе необходимо давлением, то нужно определиться с числовым диапазоном давления нагнетания в системе сразу после выхода из компрессора, технологически оно может варьироваться в диапазоне 20 – 24.7 МПа, этим диапазоном значений и будем оперировать в дальнейшем.

9 Проектирование нечеткого лингвистического регулятора по Мамдани средствами пакета Simulink и модуля Fuzzy Logic Designer системы Matlab

Рассмотрим технологию проектирования нечеткого регулятора в системе Matlab.

Для начала необходимо открыть ранее созданную модель системы управления стабилизации и создать модель нечеткого регулятора, которая представлена на рисунке 9.1:

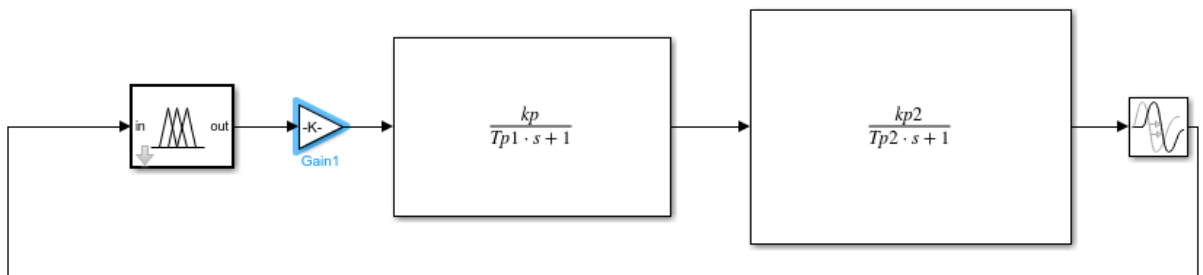


Рисунок 9.1 – Модель нечеткого регулятора

Блок Fuzzy Logic Controller (рисунок 9.2) необходим для управления набором нечетких переменных, дефаззификации данных (преобразования нечетких значений в четкие).

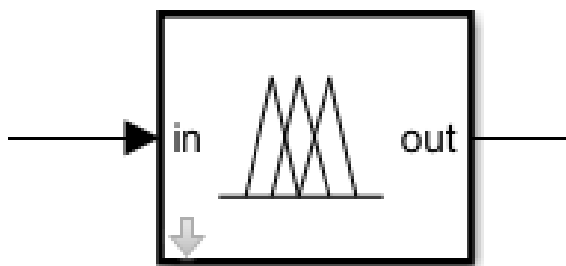


Рисунок 9.2 – Блок Fuzzy Logic Controller

Итоговая модель с нечетким лингвистическим регулятором (НЛР) представлена на рисунке 9.3:

Следует отметить, что замена коэффициентов передаточных функций и блока Gain необходима для проведения оптимизации системы.

Далее, перейдем к непосредственной работе с нечетким регулятором. Для инициализации работы с нечеткими переменными существует модуль Fuzzy Logic Designer. Чтобы его открыть, нужно в окне Command Window среды Matlab ввести команду «fuzzy» [31] и нажать Enter (рисунок 9.4).

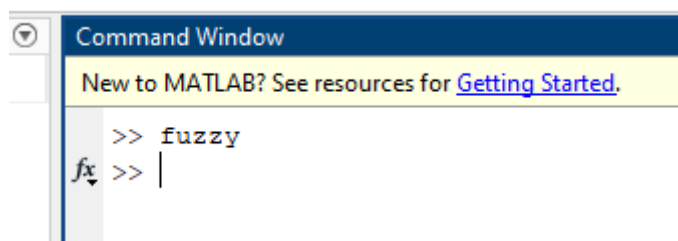


Рисунок 9.4 – Ввод команды

После этого откроется диалоговое окно Fuzzy Logic Designer, представленное на рисунке 9.5:

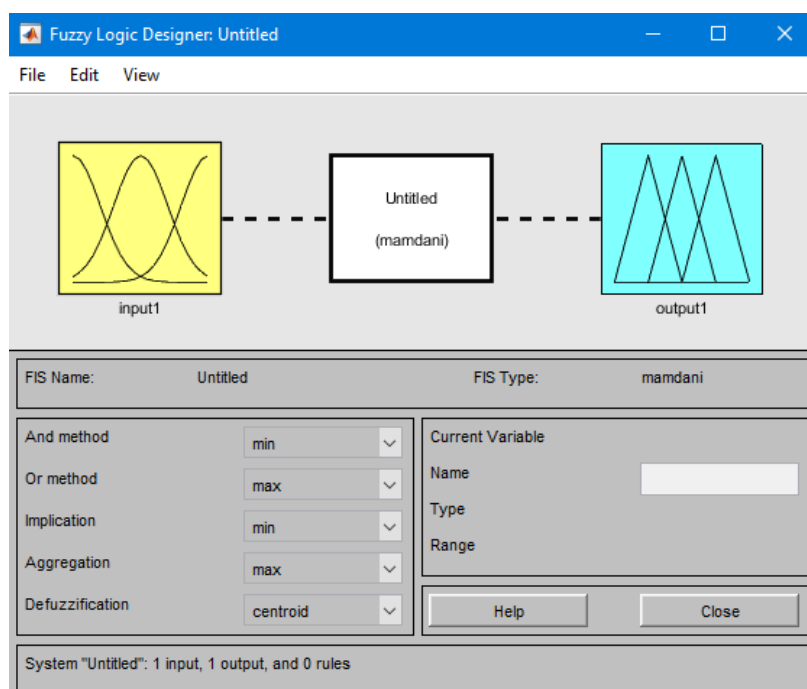


Рисунок 9.5 – Окно модуля Fuzzy Logic Designer

Чтобы создать нечеткий лингвистический регулятор по Мамдани, нужно перейти в меню File и выбрать New FIS→ Mamdani (рисунок 9.6):

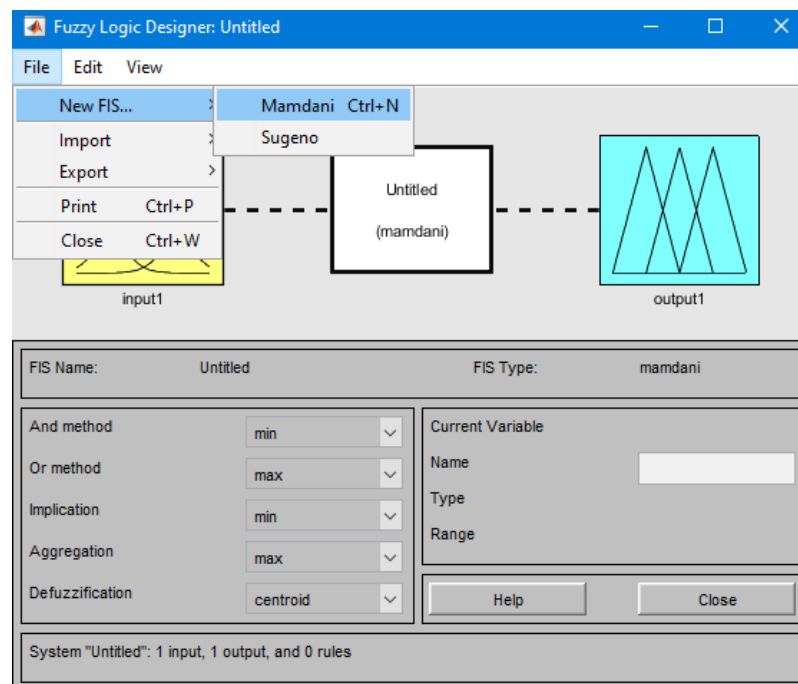


Рисунок 9.6– Выбор типа нечеткого регулятора

После создания регулятора по Мамдани [32] перейдем в область входных данных *input*, для этого необходимо дважды нажать левую кнопку мыши на этот блок (рисунок 9.7).

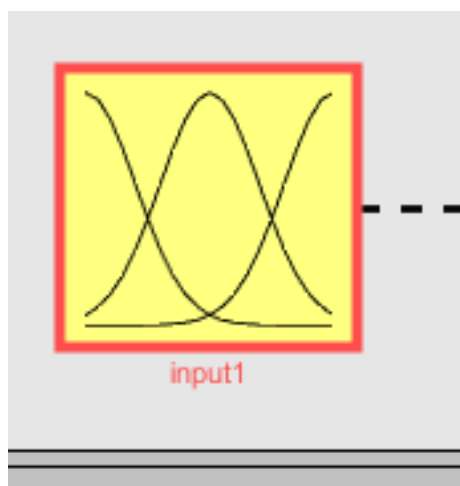


Рисунок 9.7 – Блок входных переменных

В открывшемся окне переименуем область входных данных так, как представлено на рисунке 9.8:

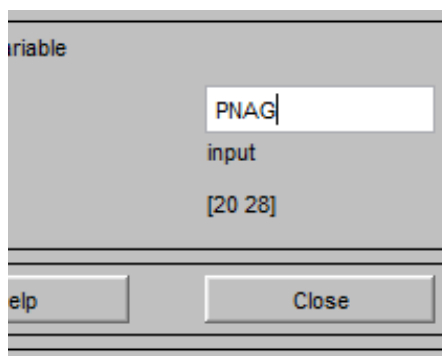


Рисунок 9.8 – Переименование области входных переменных

Далее необходимо задать настраиваемые диапазоны для терм. Терма – это формализация нечеткого множества с помощью функции принадлежности. Набор таких множеств представлен в области Membership function plots (рисунок 9.9). В разделе Range указываем значение диапазона, которое представлено на рисунке 9.9:

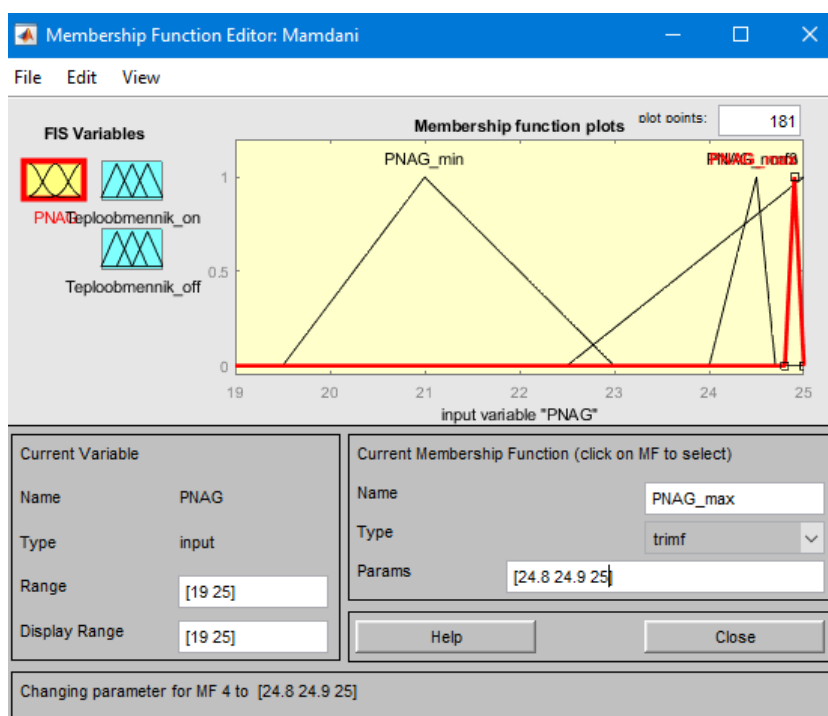


Рисунок 9.9 – Окно Membership Function Editor

Для того, чтобы выделить терму, нужно нажать на ее границу левой кнопкой мыши, тогда оно выделится красным цветом, как представлено на рисунке 9.10:

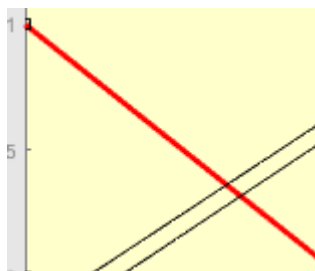


Рисунок 9.10 – Выбор термы

Выделим первую границу термы, переименуем переменную (пункт Name) и зададим в окне Params значения границ (рисунок 9.11):

Current Membership Function (click on MF to select)

Name: PNAG_max

Type: trimf

Params: [24.8 24.9 25]

Help Close

Рисунок 9.11 – Настройка первой термы

Аналогичную выделим вторую и третью границы терм и настроим их, как показано на рисунках 9.12, 9.13:

Current Membership Function (click on MF to select)

Name: PNAG_min

Type: trimf

Params: [19.5 21 23]

Help Close

Рисунок 9.12 – Настройка второй термы

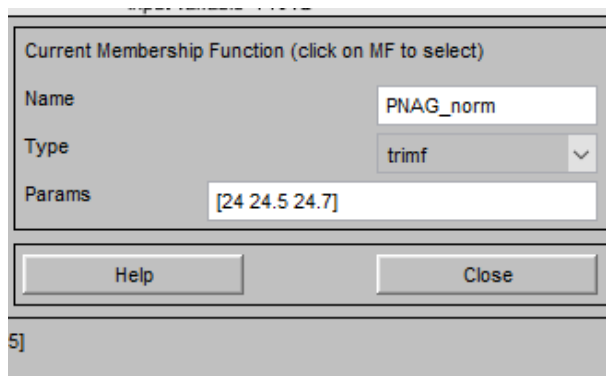


Рисунок 9.13 – Настройка третьей термы

После настройки входных переменных перейдем к настройке выходных данных (output), для начала переименуем данную область так, как показано на рисунке 9.14:

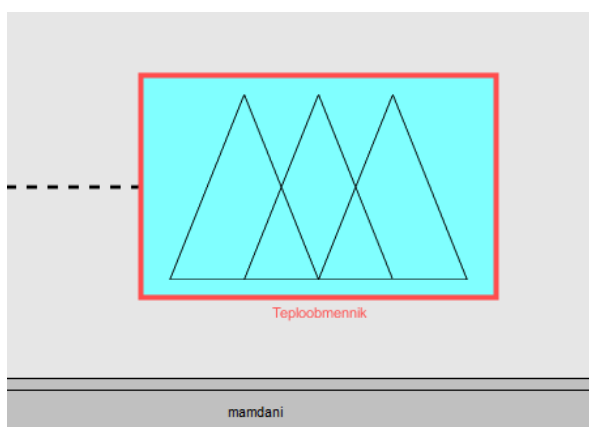


Рисунок 9.14– Переименование области выходных данных

Настроим диапазон значений (Range) так, как показано на рисунках 9.15-9.18:

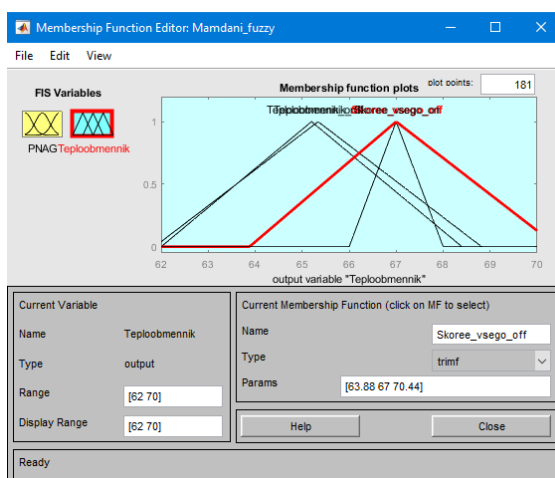


Рисунок 9.15 – Настройка диапазона выходных значений

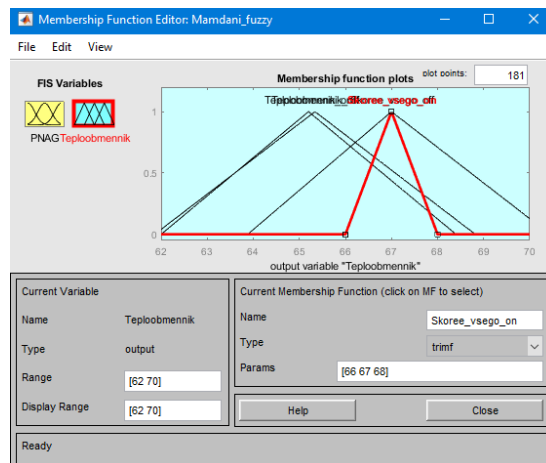


Рисунок 9.16– Настройка диапазона выходных значений

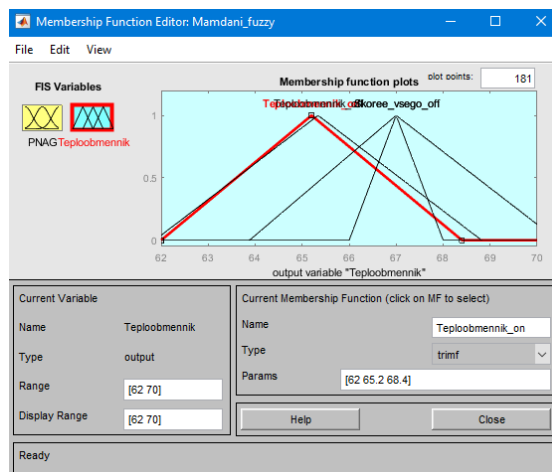


Рисунок 9.17– Настройка диапазона выходных значений

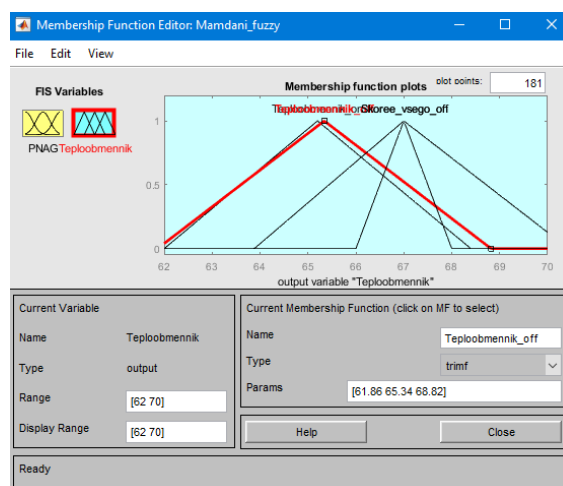


Рисунок 9.18– Настройка диапазона выходных значений

После этого, создадим номенклатуру правил (описание связи между входными и выходными значениями), так как без них нечеткая система работать не будет. Чтобы открыть опцию настройки правил, нужно перейти в меню Edit → Rules (рисунок 9.19).

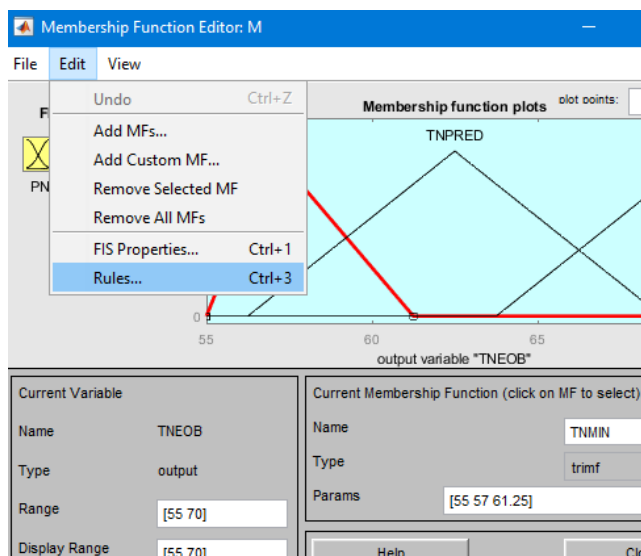


Рисунок 9.19 – Переход в меню настройки правил

Далее, создадим перечень необходимых для работы правил [32], который представлен на Рисунке 9.20:

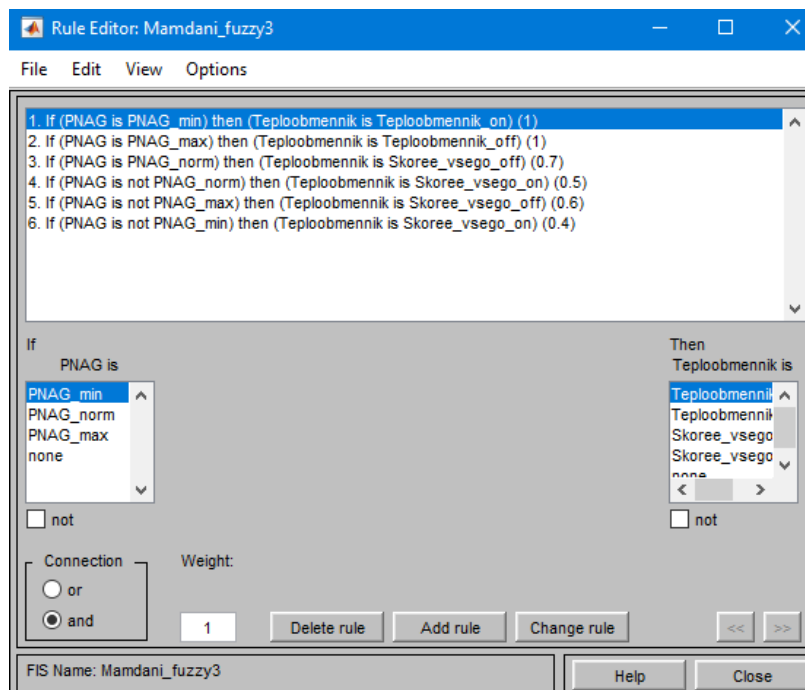


Рисунок 9.20 – Перечень необходимых правил для диапазона входных и выходных переменных

Сведем правила в таблицу 9.1:

Таблица 9.1 – Правила для терм (с пояснениями)

Правило	Пояснение
If (PNAG is PNAGmin) then (Teplobmennik isTeploobmennik_on)	Когда давление нагнетания минимальное, то необходимо включить электродвигатель для работы теплообменного аппарата
If (PNAG is PNAGmax) then (Teplobmennik isTeploobmennik_off)	Когда давление нагнетания максимальное, то электродвигатель теплообменного аппарата должен быть выключен
If (PNAG is PNAGnorm) then (Teplobmennik is Skoree_vsego_off)	Когда давление нагнетания нормальное, то электродвигатель теплообменника, скорее всего, выключен (с вероятностью 70 %)
If (PNAG is not PNAGnorm) then (Teplobmennik is Skoree_vsego_on)	Когда давление нагнетания не является нормальным, то электродвигатель теплообменника, скорее всего, включен (с вероятностью 50 %)
If (PNAG is not PNAGmax) then (Teplobmennik is Skoree_vsego_off)	Когда давление нагнетания не является максимальным, то электродвигатель теплообменника, скорее всего, выключен (с вероятностью 60 %)
If (PNAG is not PNAGmin) then (Teplobmennik is Skoree_vsego_on)	Когда давление нагнетания не является минимальным, то электродвигатель теплообменника, скорее всего, включен (с вероятностью 40 %)

Если перейти в меню View → Rules (рисунок 9.21), то можно увидеть графическое отображение настроенных правил. Здесь можно заметить, например, что при давлении нагнетания равном 22 МПа выходная переменная Teploobmennik (температура стабилизации) составила 67 °С.

Теперь произведем экспорт полученной модели в файл с расширением .FIS, для этого перейдем в меню File → Export → To File, сохраним файл в корневом каталоге с моделью и назовем его «Mamdani_fuzzy.fis».

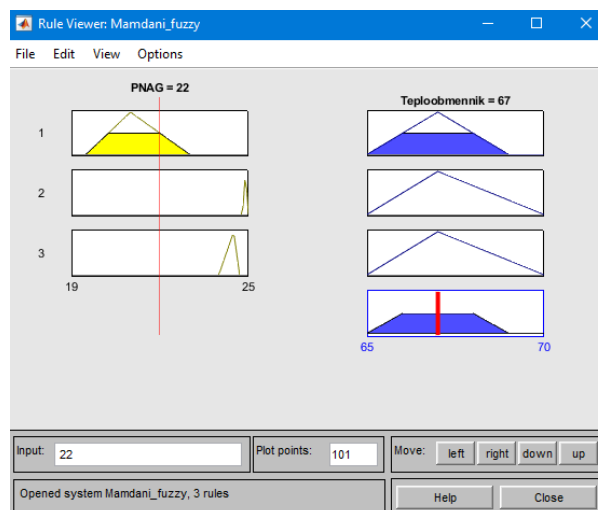


Рисунок 9.21 – Графическое отображение настроенных правил

После экспорта, дважды кликаем левой кнопкой мыши по блоку Fuzzy Logic Controller (рисунок 9.2) и в окне «FIS Name» вписать название сохраненного файла так, как показано на Рисунке 9.22 и нажать «ОК».

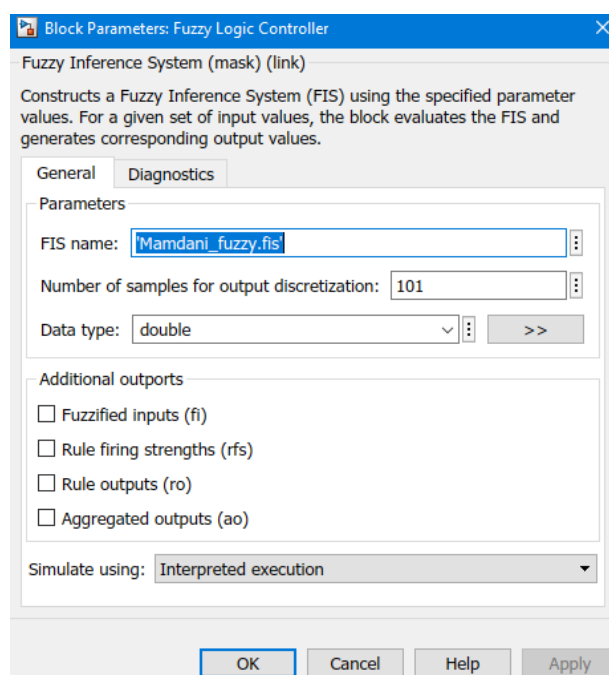


Рисунок 9.22 – Добавление файла в блок Fuzzy Logic Controller

После этого компилируем модель с помощью кнопки «Run» (рисунок 9.23).

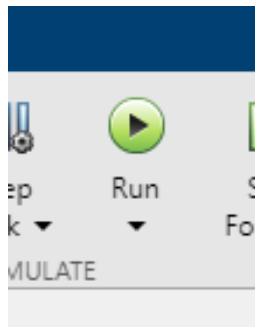


Рисунок 9.23 – Компиляция модели

Теперь должны провести анализ переходного процесса нечеткого регулятора, который получен после оптимизации модели с коэффициентами на рисунке 9.24, для этого необходимо открыть блок Score, дважды кликнув на него левой кнопкой мыши (рисунок 9.25):

```

Файл  Правка  Формат
kf=-316.6483;
Tp1f=1.8760;
Tp2f=1.2472;
kpf=0.3925;
kp2f=0.0025;

```

Рисунок 9.24 – Оптимизированные коэффициенты для НЛР

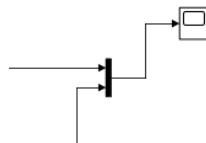


Рисунок 9.25– Подключение моделей ПИД-регулятора и НЛР через Score с помощью блока MUX (мультиплексор)

Получим два переходных процесса, представленных на рисунке 9.26.



Рисунок 9.26 – Переходные процессы стабилизации температуры

Как можно заметить, время регулирования и установившегося значения переходного процесса стабилизации температуры при работе нечеткого регулятора составили порядка 10 и 12 секунд, а температура стабилизации приблизительно равна 68 градусов, что удовлетворяет требуемому условию задачи.

10 Математическое описание коэффициента теплопроводности

Температура стенки трубы магистрального газопровода равна температуре транспортируемого газа (обусловлено, что газ движется по магистральному газопроводу в горизонтальном направлении).

Так как аппараты воздушного охлаждения должны контактировать со стенками теплообменника и трубами (трубчатый пучок АВО, по которым движется газ), то необходимо каким-то образом установить зависимость того, какое количество теплоты принимается (или отдается) теплообменником [33]-[34]-[35].

Для таких случаев при расчетах обычно принимают так называемый коэффициент теплоотдачи, который можно рассчитать по формуле (10.1):

$$\alpha = \frac{NuL}{d_3} \quad (10.1)$$

где:

α – коэффициент теплоотдачи;

L – коэффициент теплопроводности устройства;

Nu – критерий подобия Нуссельта (безразмерный коэффициент подобия);

d_3 – диаметр магистрального трубопровода.

Теперь рассмотрим нашу модель системы стабилизацию температуры, представленную на рисунке 10.1.

Для анализа стохастической составляющей и оценки поведения моделей в системы добавлены блоки RandomNumber с возможностью редактирования значений математического ожидания (Mean) и дисперсии (Variance), подключение к модели показано на рисунках 10.2, 10.3.

За коэффициент теплоотдачи здесь отвечают блоки Gain в структуре ПИД-регулятора и НЛР.



температуры охлаждения газа

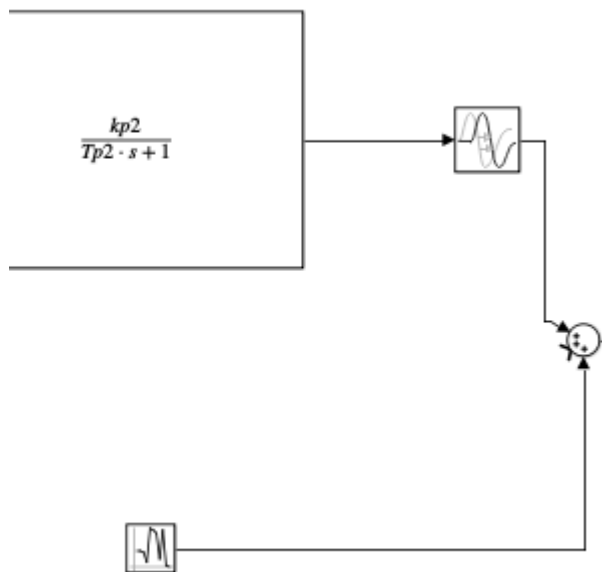


Рисунок 10.2 – Подключение блока RandomNumber к модели с ПИД-регулятором

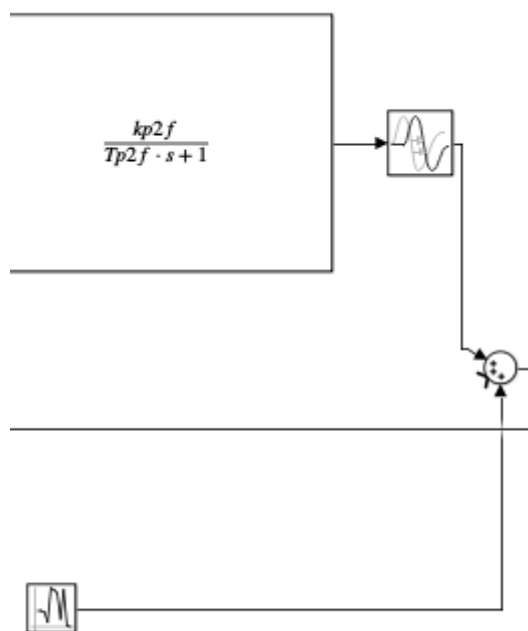


Рисунок 10.3 – Подключение блока RandomNumber к модели с НЛР

На рисунке 10.3 показаны стандартные значения для математического ожидания и дисперсии:

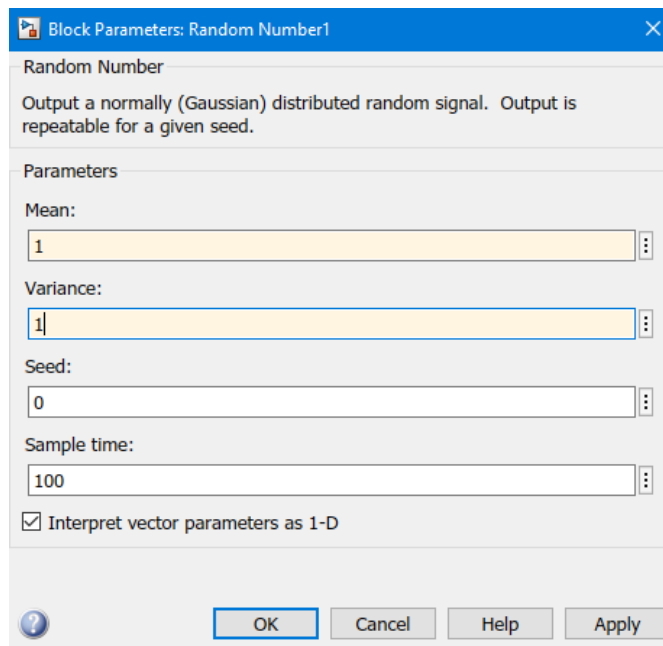


Рисунок 10.3 – Стандартные значения математического ожидания и дисперсии

Следовательно, можно предположить, что при изменении значений математического ожидания и дисперсии необходимо регулировать коэффициент теплоотдачи α таким образом, чтобы оба переходных процесса по регуляторам смогли обеспечить заданные условия по стабилизации температуры.

11 Стохастический эксперимент

Для начала в таблице 11.1 зададим значения для математического ожидания и дисперсии (блоки Random Number (рисунок 10.3)).

Таблица 11.1 – Значения математического ожидания и дисперсии [36] для стохастического эксперимента

Математическое ожидание (Mean), μ	Дисперсия (Variance), δ^2
90	1
90	2
90	3
90	4
90	5
90	6
90	7
90	8
90	9
90	10

Зададим в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 1$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.1.

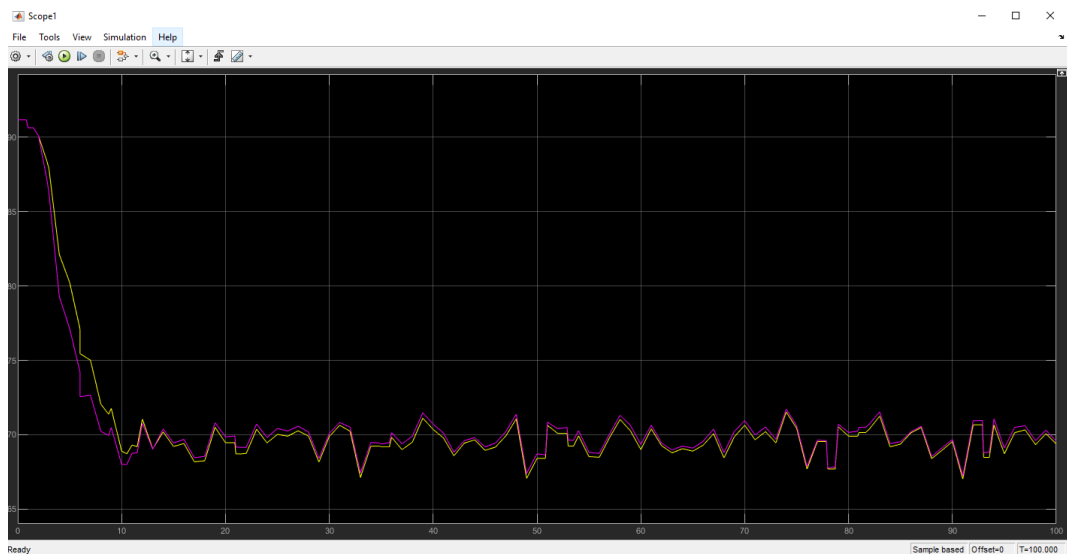


Рисунок 11.1 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 1$

Красным цветом показан переходный процесс модели СУ с оптимизированным ПИД-регулятором, а желтым – переходный процесс модели СУ с нечетким регулятором.

Результаты эксперимента представлены в приложении В.

Время регулирования и установившегося значения для СУ ПИД-регулятором составили 9 и 10 с., для модели СУ с НЛР – 10 и 12 с.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,70$, $\delta^2 = 0,96$, для СУ с нечетким регулятором – $\mu = 69,48$, $\delta^2 = 0,84$.

Теперь в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 2$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.2.



Рисунок 11.2 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 2$

Время регулирования и установившегося значения для СУ с ПИД-регулятором составили 10 и 11 секунд, для модели СУ с НЛР – 12 и 13 с.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,71$, $\delta^2 = 2,03$, для СУ с нечетким регулятором – $\mu = 69,49$, $\delta^2 = 1,84$.

Теперь в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 3$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.3.



Рисунок 11.3 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 3$

Время регулирования и установившегося значения для СУ с ПИД-регулятором составили 10 и 11 секунд, для модели СУ с НЛР – 12 и 13 с.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,3$, $\delta^2 = 2,77$, для СУ с нечетким регулятор – $\mu = 69,42$, $\delta^2 = 2,77$.

Зададим в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 4$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.4.



Рисунок 11.4 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 4$

Время регулирования и установившегося значения для СУ с ПИД-регулятором составили 10 и 11 секунд, для модели СУ с НЛР – 11 и 12 с.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,6$, $\delta^2 = 4,23$, для СУ с нечетким регулятором – $\mu = 69,39$, $\delta^2 = 3,89$.

Зададим в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 5$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.5.



Рисунок 11.5 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 5$

Время регулирования и установившегося значения для СУ с ПИД-регулятором составили 9 и 10 секунд, для модели СУ с НЛР – 10 и 12 секунд.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,63$, $\delta^2 = 5,47$, для СУ с нечетким регулятором – $\mu = 69,43$, $\delta^2 = 5,08$.

Зададим в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 6$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.6.



Рисунок 11.6 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 6$

Время регулирования и установившегося значения для СУ с ПИД-регулятором составили 9 и 10 секунд, для модели СУ с НЛР – 10,5 и 13 с.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,65$, $\delta^2 = 6,79$, для СУ с нечетким регулятором – $\mu = 69,45$, $\delta^2 = 6,36$.

Зададим в блоках Random Number значения для $\mu = 90$ и $\delta^2 = 10$, произведем компиляцию модели и получим переходные процессы, представленные на рисунке 11.7.



Рисунок 11.7 – Полученные переходные процессы для $\mu = 90$, $\delta^2 = 10$

Время регулирования и установившегося значения для СУ ПИД-регулятором составили 9 и 11 секунд, для модели СУ с НЛР – 10 и 12 секунд.

Среднее значения температуры для СУ с ПИД-регулятором $\mu = 69,58$, $\delta^2 = 10,86$, для СУ с нечетким регулятором – $\mu = 69,39$, $\delta^2 = 10,28$.

Исходя из анализа влияния случайной составляющей, можно сделать вывод, что все зависимости от значений математического ожидания и дисперсии в системе, модели СУ с ПИД-регулятором и НЛР качественные характеристики переходных процессов остаются в допустимых пределах, в связи с этим нет необходимости в проведении повторной оптимизации коэффициента теплопроводности моделей, либо коэффициентов передаточных функций.

12 Однофакторный дисперсионный анализ модели

Однофакторный дисперсионный анализ проводится с целью принятия нулевой гипотезы: имеется ли существенное влияние изменения входной температуры на выходную температуру модели СУ АВО.

Однофакторный дисперсионный анализ проводится с помощью встроенного модуля «Анализ данных» пакета MS Excel 2016 [37].

Для начала, необходимо выполнить «выгрузку» данных из модели Simulink в Excel. С целью этого, в модель Simulink добавлен блок To File (рисунок 12.1):

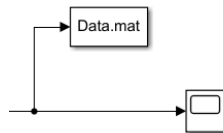


Рисунок 12.1 – Блок To File

Этот блок осуществляет «выгрузку» экспериментальных данных из модели в расширении .mat. Для того, чтобы переименовать полученный файл, необходимо открыть данный блок и в поле «File name» вписать предпочитаемое название файла (рисунок 12.2):

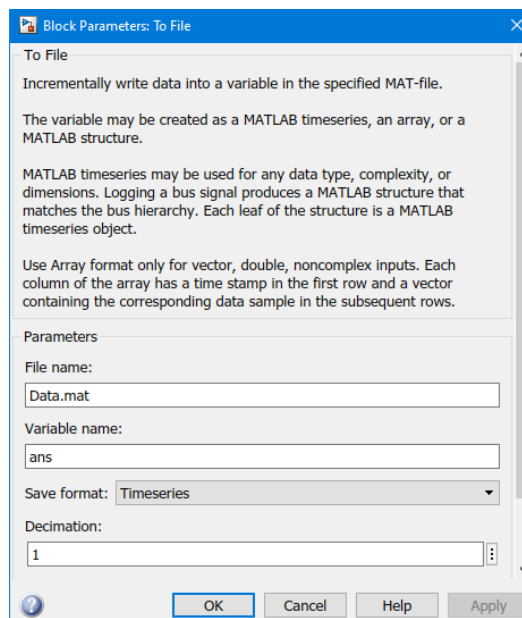


Рисунок 12.2 – Указание названия файла

Затем, в окне Workspace среды Matlab необходимо найти данный файл в «дереве каталогов», щелкнуть по нему правой кнопкой мыши и выбрать опцию «Open Selection» (рисунок 12.3):

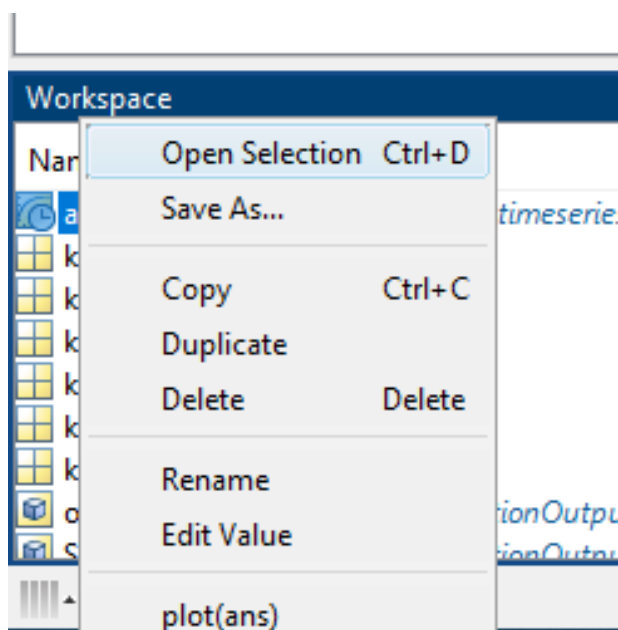


Рисунок 12.3 – Выбор опции

После этого, откроется диалоговое окно, представленное на рисунке 12.4, здесь необходимо скопировать все данные в буфер обмена.

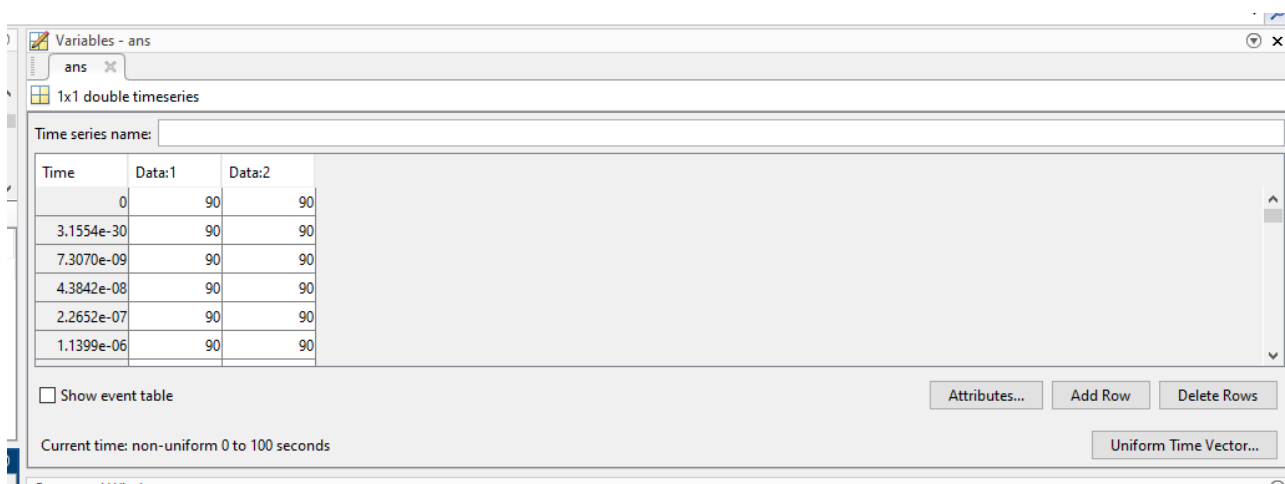


Рисунок 12.4 – Данные модели

Затем необходимо изменить значения математического ожидания и дисперсии в блоке Random Number и проделать операцию на рисунках 12.3-12.4, таким образом необходимо получить 20 столбцов с температурой газа и скопировать в MS Excel так, как показано на рисунке 12.5:

Все результаты эксперимента представлены в приложении В.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	5	F	P		4				3				2		
2	0	92,60492	92,60492		0	92,32991	92,32991		0	92,01776	92,01776		0	91,64749	91,64749
3	3,16E-30	92,60492	92,60492		3,16E-30	92,32991	92,32991		3,16E-30	92,01776	92,01776		3,16E-30	91,64749	91,64749
4	7,10E-09	92,60492	92,60492		7,12E-09	92,32991	92,32991		7,15E-09	92,01776	92,01776		7,18E-09	91,64749	91,64749
5	4,26E-08	92,60492	92,60492		4,27E-08	92,32991	92,32991		4,29E-08	92,01776	92,01776		4,31E-08	91,64749	91,64749
6	2,20E-07	92,60492	92,60492		2,21E-07	92,32991	92,32991		2,22E-07	92,01776	92,01776		2,22E-07	91,64749	91,64749
7	1,11E-06	92,60492	92,60492		1,11E-06	92,32991	92,32991		1,11E-06	92,01776	92,01776		1,12E-06	91,64749	91,64749
8	5,55E-06	92,60492	92,60492		5,56E-06	92,32991	92,32991		5,58E-06	92,01776	92,01776		5,60E-06	91,64749	91,64749
9	2,77E-05	92,60492	92,60492		2,78E-05	92,32991	92,32991		2,79E-05	92,01776	92,01776		2,80E-05	91,64749	91,64749
10	0,000139	92,60492	92,60492		0,000139	92,32991	92,32991		0,00014	92,01776	92,01776		0,00014	91,64749	91,64749
11	0,000693	92,60492	92,60492		0,000696	92,32991	92,32991		0,000698	92,01776	92,01776		0,000701	91,64749	91,64749
12	0,003467	92,60492	92,60492		0,003478	92,32991	92,32991		0,00349	92,01776	92,01776		0,003504	91,64749	91,64749
13	0,017337	92,60492	92,60492		0,017389	92,32991	92,32991		0,017448	92,01776	92,01776		0,017519	91,64749	91,64749
14	0,086687	92,60492	92,60492		0,086946	92,32991	92,32991		0,08724	92,01776	92,01776		0,087593	91,64749	91,64749
15	0,246303	92,60492	92,60492		0,246742	92,32991	92,32991		0,247243	92,01776	92,01776		0,247841	91,64749	91,64749
16	0,486174	92,60492	92,60492		0,486788	92,32991	92,32991		0,487488	92,01776	92,01776		0,488325	91,64749	91,64749
17	0,807997	92,60492	92,60492		0,808799	92,32991	92,32991		0,809714	92,01776	92,01776		0,810806	91,64749	91,64749
18	1	91,40165	91,40165		1	91,25368	91,25368		1	91,08572	91,08572		1	90,88648	90,88648
19	1,520275	91,40165	91,40165		1,520535	91,25368	91,25368		1,520833	91,08572	91,08572		1,521188	90,88648	90,88648
20	2	90,16788	90,16788		2	90,15016	90,15016		2	90,13004	90,13004		2	90,10618	90,10618
21	2,827383	88,41103	87,08814		2,825196	88,40109	87,09233		2,822747	88,38971	87,0968		2,819886	88,37605	87,10174
22	3	88,32869	86,63941		3	88,24576	86,56875		3	88,15163	86,48855		3	88,03998	86,39341
23	3	88,32869	86,63941		3	88,24576	86,56875		3	88,15163	86,48855		3	88,03998	86,39341
24	3,863087	84,25631	81,41895		3,874022	84,1209	81,29822		3,886267	83,96801	81,16234		3,900569	83,78772	81,00272
25	4	81,25664	78,3096		4	81,42106	78,49854		4	81,60768	78,713		4	81,82906	78,96738
26	4,684563	78,27968	75,10807		4,629888	78,6776	75,54451		4,568663	79,12585	76,03648		4,497154	79,65299	76,61547
27	5	82,32384	79,09526		5	81,92355	78,71922		5	81,4692	78,29239		5	80,93026	77,7861
28	5,774812	79,82612	76,75822		5,793312	79,37153	76,33105		6	78,28515	75,32228		6	77,74538	74,81104
29	6	79,14112	76,13303		6	78,7402	75,75329		6	75,44964	72,48677		6	75,4302	72,49586

Рисунок 12.5 – Данные модели

После этого, необходимо в MS Excel перейти в меню «Данные» и выбрать пункт «Анализ данных» (рисунок 12.6):

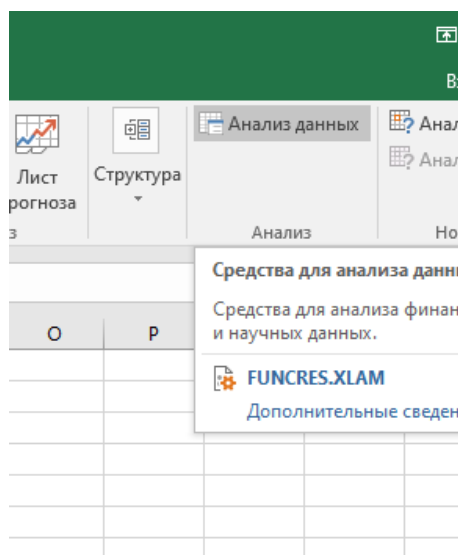


Рисунок 12.6 – Выбор опции

В открывшемся диалоговом окне необходимо выбрать инструмент анализа «Однофакторный дисперсионный анализ» (рисунок 12.7):

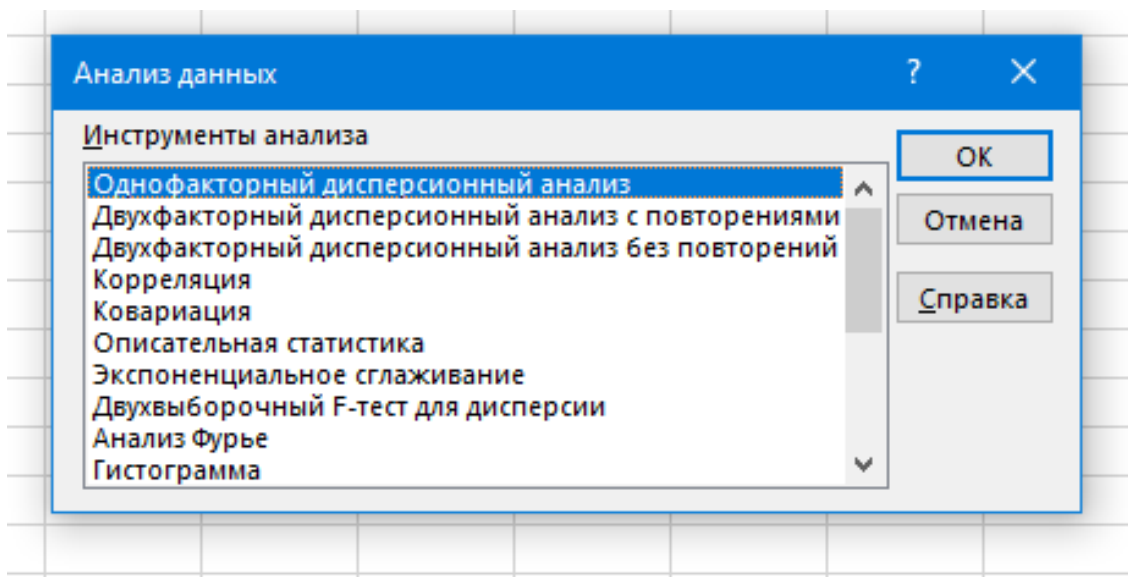


Рисунок 12.7 – Выбор инструмента анализа

Далее, необходимо выделить диапазон 1 и 2 столбца (для ПИД-регулятора) и нажать ОК (рисунок 12.8):

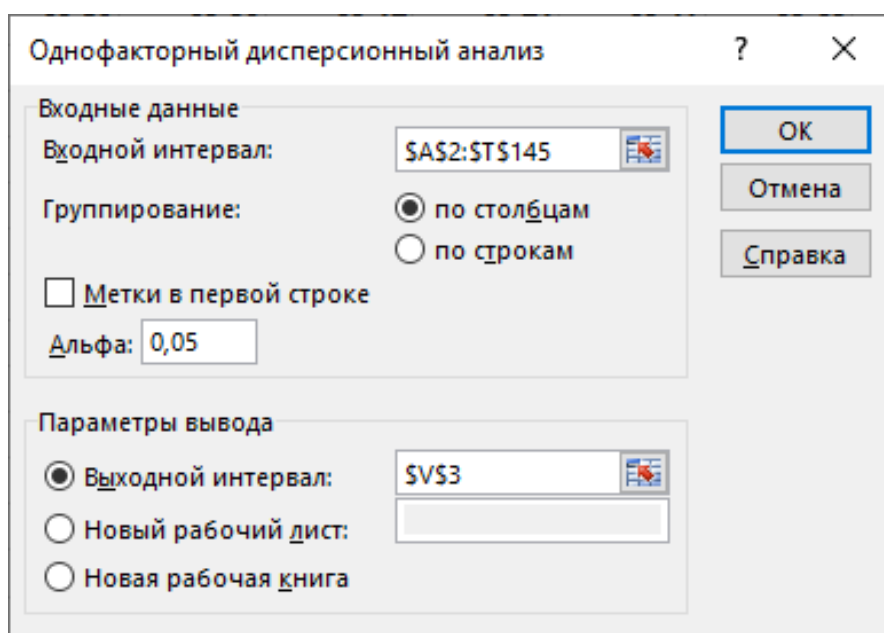


Рисунок 12.8 – Выбор диапазонов значений

В результате, на новом рабочем листе получим таблицу, которая представлена на рисунке 12.9:

	A	B	C	D	E	F	G
1	Однофакторный дисперсионный анализ						
2							
3	ИТОГИ						
4	<i>Группы</i>	<i>Счет</i>	<i>Сумма</i>	<i>Среднее</i>	<i>Дисперсия</i>		
5	Столбец 1	130	9026,226	69,43251	5,079882		
6	Столбец 2	130	9052,025	69,63096	5,466645		
7	Столбец 3	132	9159,748	69,39203	3,89979		
8	Столбец 4	132	9187,306	69,6008	4,230948		
9	Столбец 5	134	9302,879	69,42447	2,768621		
10	Столбец 6	134	9330,291	69,62904	3,019538		
11	Столбец 7	124	8617,007	69,49199	1,841565		
12	Столбец 8	124	8643,612	69,70655	2,028953		
13	Столбец 9	108	7504,639	69,4874	0,844083		
14	Столбец 10	108	7527,713	69,70104	0,965875		
15	Столбец 11	132	9168,143	69,45563	6,352997		
16	Столбец 12	132	9193,922	69,65093	6,789379		
17	Столбец 13	141	9772,66	69,30965	7,09927		
18	Столбец 14	141	9800,229	69,50517	7,555927		
19	Столбец 15	141	9773,934	69,31868	8,229426		
20	Столбец 16	141	9801,781	69,51618	8,738749		
21	Столбец 17	144	9994,431	69,40577	9,16668		
22	Столбец 18	144	10021,91	69,59662	9,715092		
23	Столбец 19	143	9923,454	69,39478	10,28292		
24	Столбец 20	143	9950,279	69,58237	10,85771		
25							
26							
27	Дисперсионный анализ						
28	<i>Источник вариации</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-Значение</i>	<i>F критическое</i>
29	Между группами	36,43676	19	1,917724	0,321895	0,997605037	1,590454562
30	Внутри групп	15716,18	2638	5,957612			
31							
32	Итого	15752,62	2657				

Рисунок 12.9 – Таблица однофакторного дисперсионного анализа

Как видно из рисунка 12.9, стандартное значение критерия Фишера меньше его критического значения, следовательно, нулевая гипотеза принимается, т.е. отсутствует влияние изменения температуры от случайной составляющей.

13 Анализ и тестирование программного кода модели Simulink с помощью встроенных компонентов языка Си среды Matlab

13.1 Описание системы Embeeded Coder среды MATLAB

Embedded Coder [38] генерирует читаемый, компактный и быстрый C/C++ код для встраиваемых процессоров, используемых в массовом производстве. Он расширяет возможности MATLAB Coder и Simulink Coder специфическими оптимизациями для точного управления сгенерированными функциями, файлами и данными. Эти оптимизации повышают эффективность кода и облегчают интеграцию с унаследованным кодом, типами данных и параметрами калибровки. Embedded coder позволяет использовать инструменты разработки производителя процессора для создания исполняемого файла для развертывания «под ключ» во встроенной системе или отладочном комплекте.

Embedded Coder поддерживает такие стандарты программного обеспечения и кодирования, как AUTOSAR, MISRA C и ASAP2. В процессе генерации исходного кода генерируются отчеты о трассируемости, документация по коду. Embedded Coder обеспечивает автоматическую проверку программного обеспечения для обеспечения процесса разработки программного обеспечения согласно DO-178 (KT-178), IEC 61508 (ГОСТ Р МЭК 61508) и ISO 26262 (ГОСТ Р ИСО 26262). Код Embedded Coder является переносимым и может быть скомпилирован и выполнен на любом процессоре. Кроме того, Embedded Coder предлагает пакеты поддержки оборудования с расширенной оптимизацией сгенерированного кода для конкретной аппаратной платформы.

13.2 Ключевые особенности

К ключевым особенностям Embeeded Coder можно отнести:

- оптимизации и настройки кода, расширяющие MATLAB Coder и Simulink Coder;

- определение класса, типа и псевдонима хранилища с использованием словарей данных;
- многоскоростное, многозадачное и многоядерное выполнение кода с RTOS или без;
- проверка кода, включая тестирование SIL и PIL, пользовательские комментарии и отчеты кода с отслеживанием моделей в соответствии с кодом и требованиями;
- поддержка стандартов, включая ASAP2, AUTOSAR, DO-178, IEC 61508, ISO 26262 и MISRA C (с Simulink);
- расширенные возможности оптимизации кода и драйверов устройств для конкретного оборудования, включая ARM, Intel, NXP, STMicroelectronics и Texas Instruments.

13.3 Получение программного кода модели Simulink на языке Си

Для начала необходимо открыть в среде Simulink итоговую модель стабилизации температуры, модель представлена на рисунке 13.1.

Далее, необходимо перейти не вкладку «C Code», на которой расположена кнопка Build с выпадающим меню (рисунок 13.2).

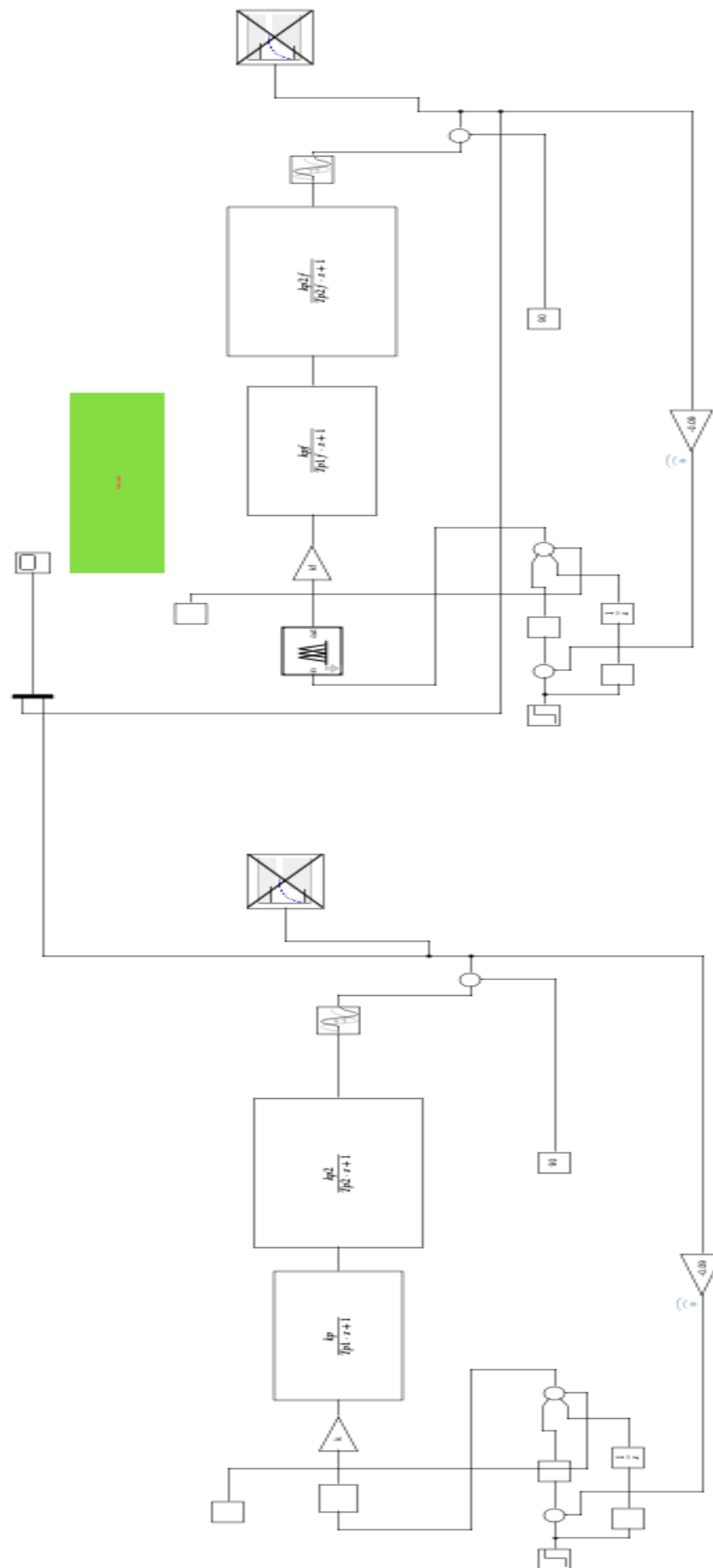


Рисунок 13.1 - Итоговая модель стабилизации температуры

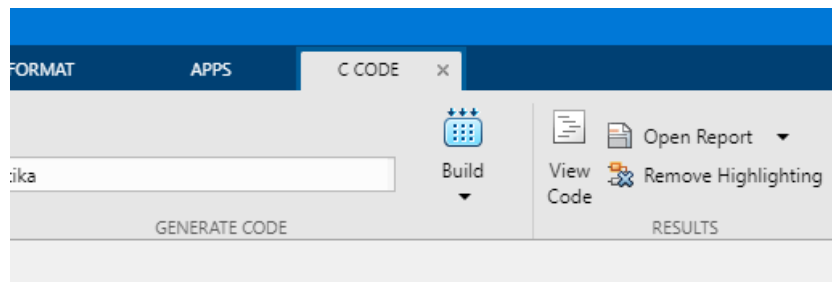


Рисунок 13.2 – Элементы меню C CODE

После нажатия левой кнопкой мыши по вкладке Build, необходимо выбрать и нажать на пункт Build (рисунок 13.3):

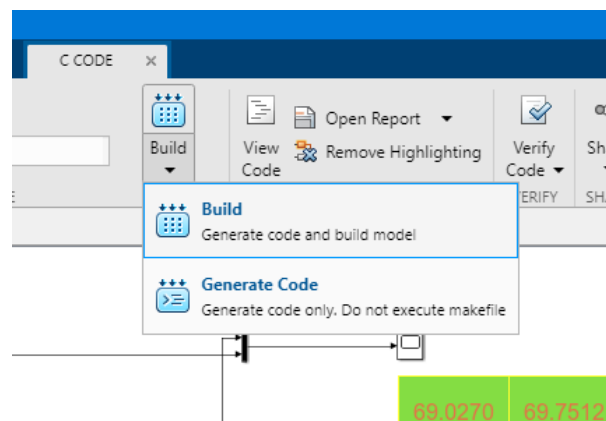


Рисунок 13.3 – Выпадающее меню Build

После этого, появится диалоговое окно диагностики, представленное на рисунке 13.4:

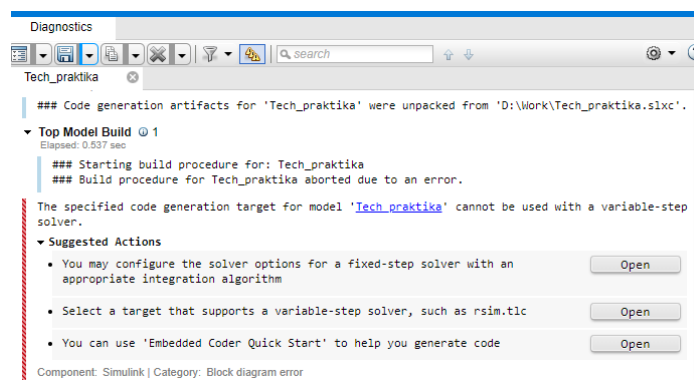


Рисунок 13.4 – Окно Diagnostics меню Build

Затем необходимо нажать кнопку Open для открытия блока «Embedded Coder Quick Start» (рисунок 13.5):

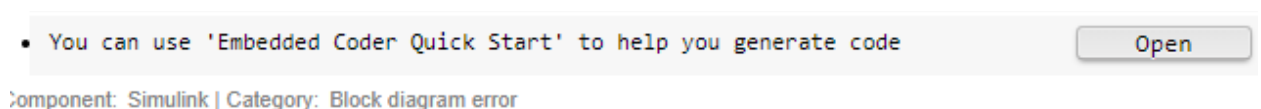


Рисунок 13.5 – Открытие блока

На рисунках 13.6–13.14 – представлен процесс настройки блока «Embeeded Coder Quick Start», после каждой настройки для продолжения необходимо нажимать кнопку Next:

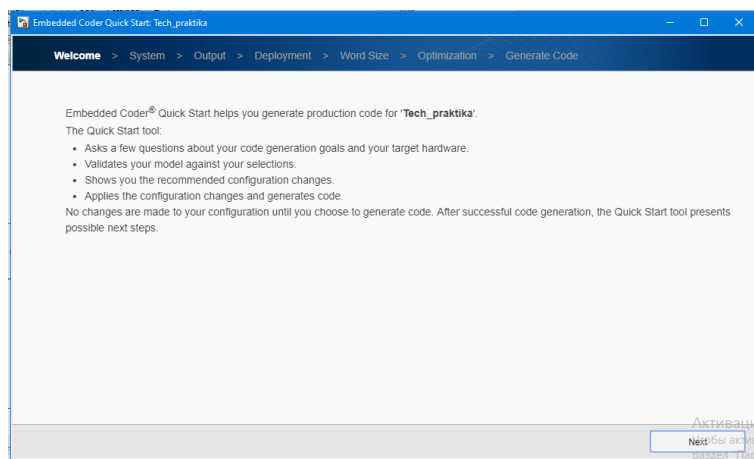


Рисунок 13.6 – Окно Welcome

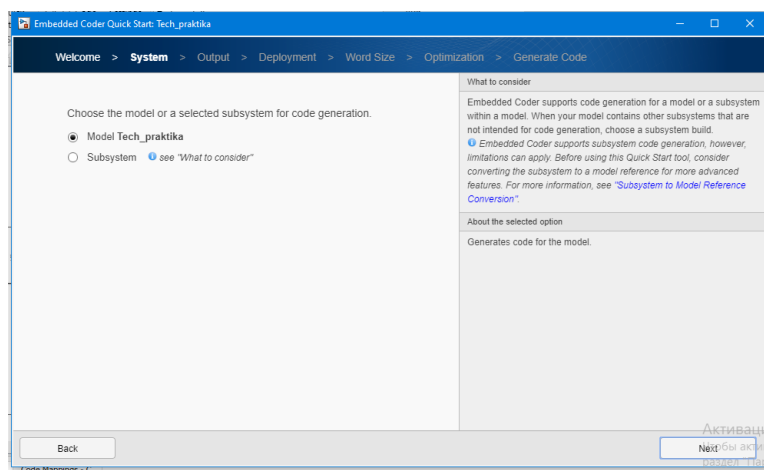


Рисунок 13.7 – Настройка окна System

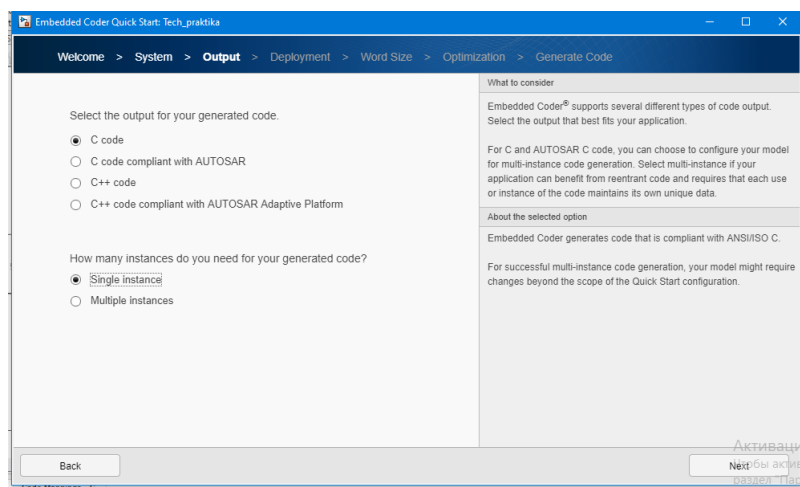


Рисунок 13.8 – Настройка окна Output

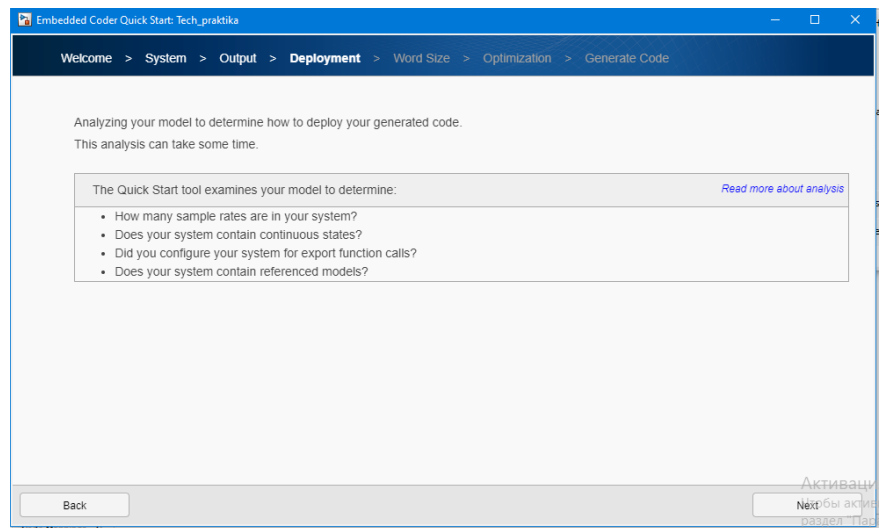


Рисунок 13.9 – Окно Deployment

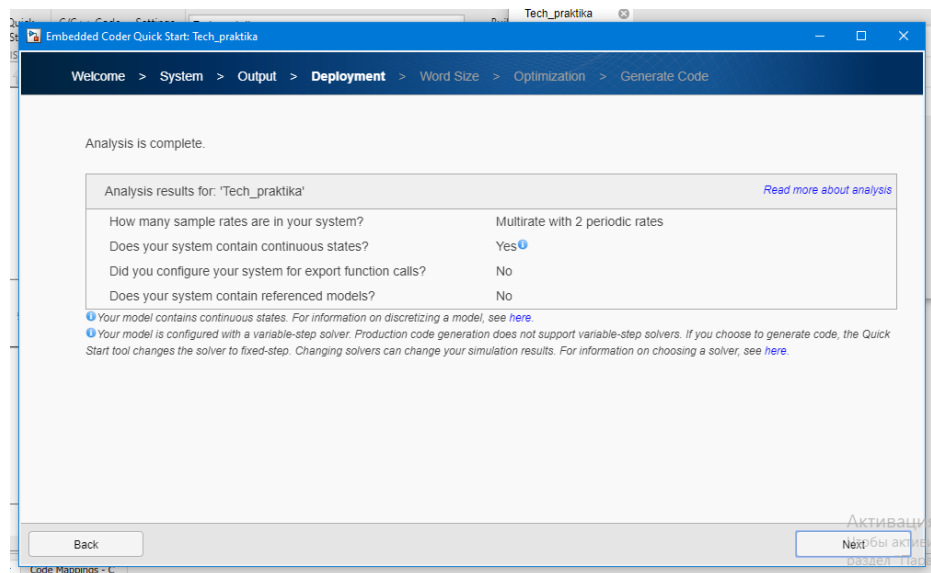


Рисунок 13.10 – Окно Deployment (продолжение настройки)

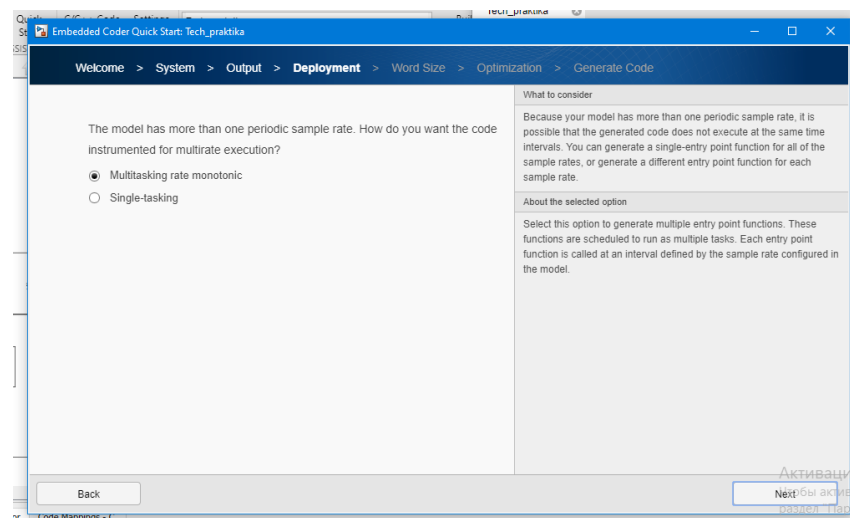


Рисунок 13.11 – Окно Deployment (продолжение настройки)

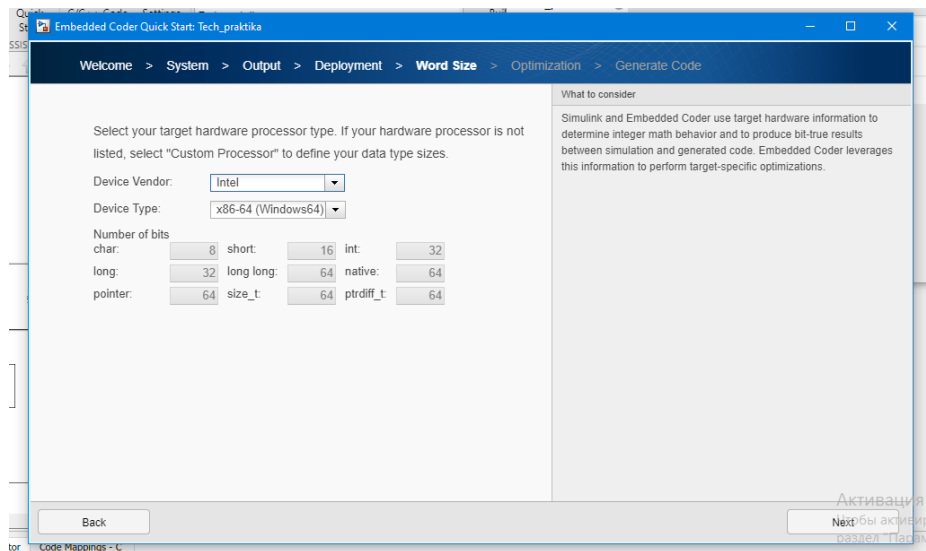


Рисунок 13.12 – Настройка окна Word Size

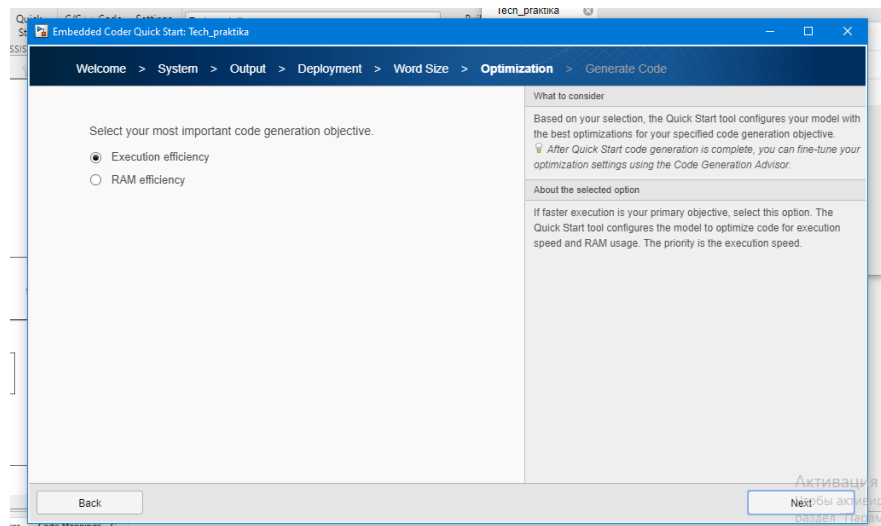


Рисунок 13.13 – Настройка окна Optimization

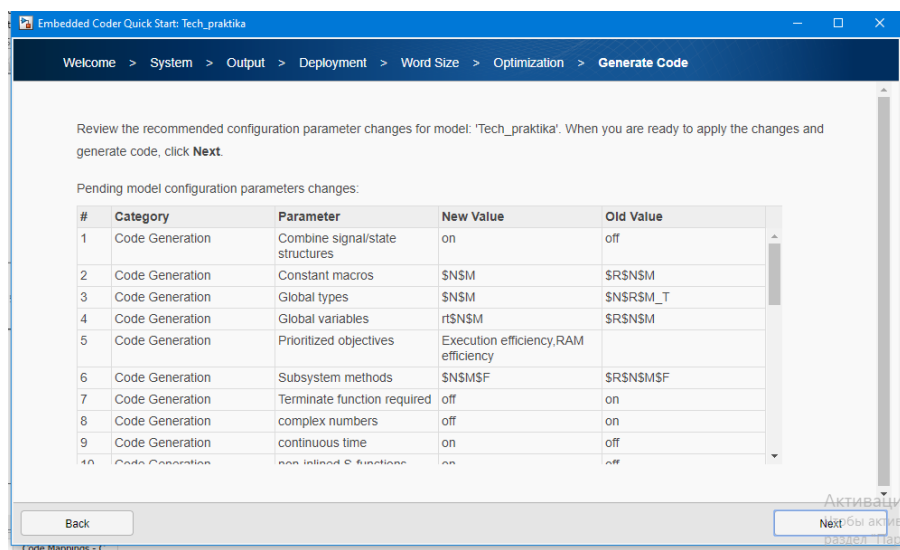


Рисунок 13.14 – Настройка окна Generate Code

После настройке в окне Code Generation Complete необходимо нажать Finish (рисунок 13.15):

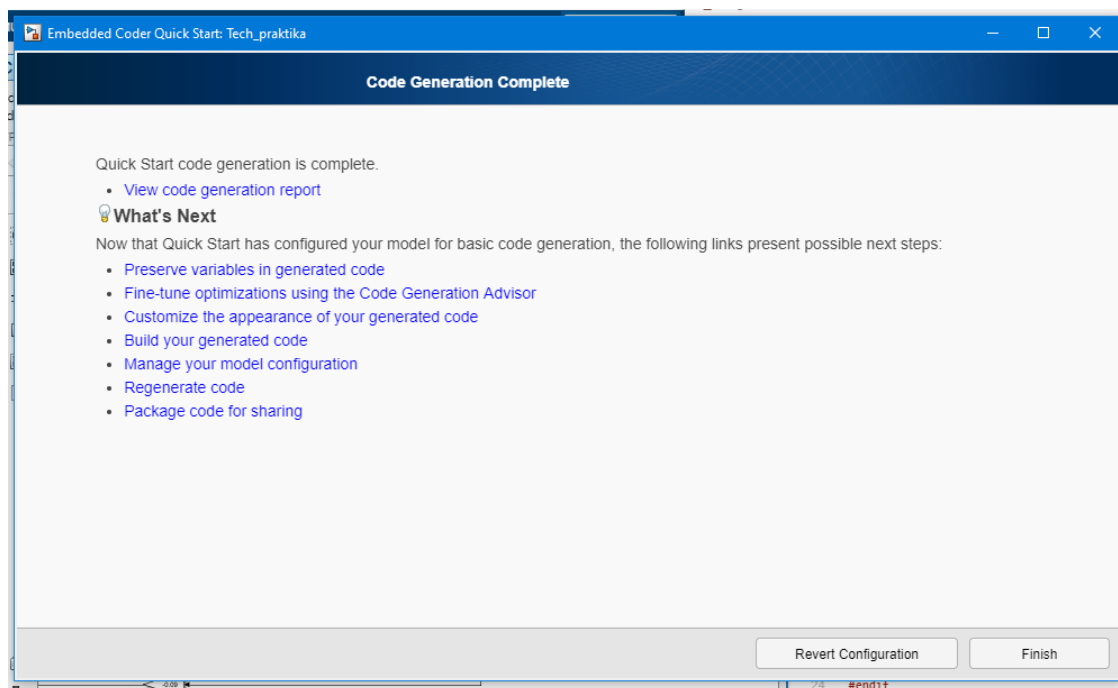


Рисунок 13.15 – Окно Code Generation Complete

После этого начнется процесс компиляции кода модели Simulink в исходный код языка Си, процесс компиляции можно отслеживать в окне Diagnostic Viewer (рисунок 13.16):

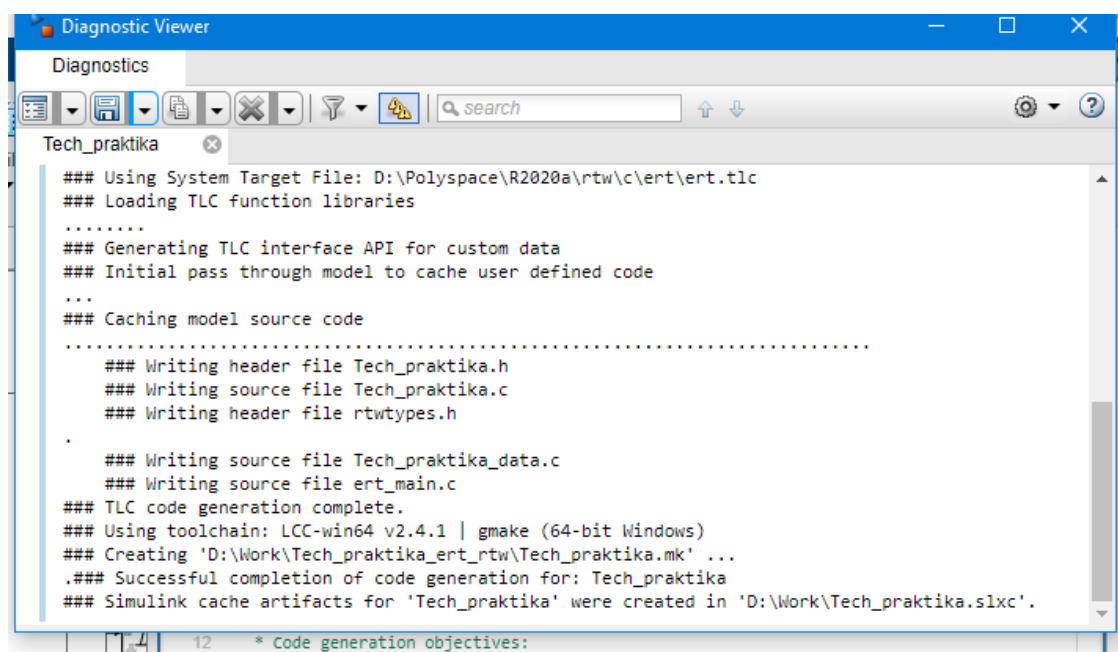


Рисунок 13.16 – Окно Diagnostic Viewer

Исходный код и все файлы будут сохранены в рабочий каталог среды Matlab (рисунок 13.17):

```

529
530 for (i = 0; i < nXc; i++) {
531     x[i] = y[i] + (f0[i]*hb[0] + f1[i]*hb[1] + f2[i]*hb[2]);
532 }
533
534 rtsiSetT(si, tnew);
535 rtsiSetSimTimeStep(si, MAJOR_TIME_STEP);
536 }
537
538 /* Function for MATLAB Function: '<S3>/Evaluate Rule Antecedents' */
539 static real_T trimf(real_T x, const real_T params[3])
540 {
541     real_T y;
542     y = 0.0;
543     if ((params[0] != params[1]) && (params[0] < x) && (x < params[1])) {
544         y = 1.0 / (params[1] - params[0]) * (x - params[0]);
545     }
546     if ((params[1] != params[2]) && (params[1] < x) && (x < params[2])) {
547         y = 1.0 / (params[2] - params[1]) * (params[2] - x);
548     }
549 }
550
551 if (x == params[1]) {
552     y = 1.0;
553 }

```

Рисунок 13.17 – Фрагмент полученного кода на языке Си

13.4 Проверка исходного кода модели Simulink модулем PolySpace Code Verifier

Для начала в окне Embeeded Coder необходимо открыть меню Verify Code [39] (рисунок 13.18):

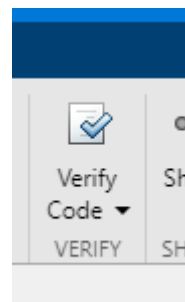


Рисунок 13.18 – Меню Verify Code

После этого необходимо выбрать компонент PolySpace Code Verifier (рисунок 13.19):

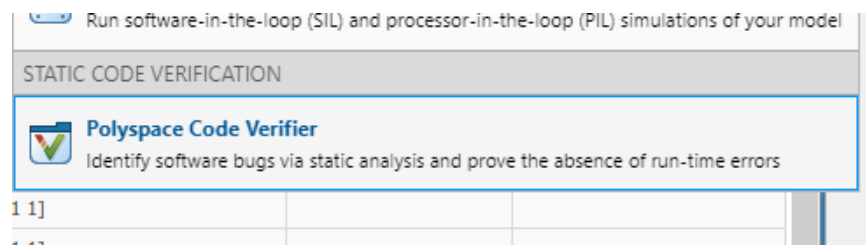


Рисунок 13.19 – Выбор компонента

После этого необходимо в открывшемся блоке нажать кнопку Run Analysis (рисунок 13.20)

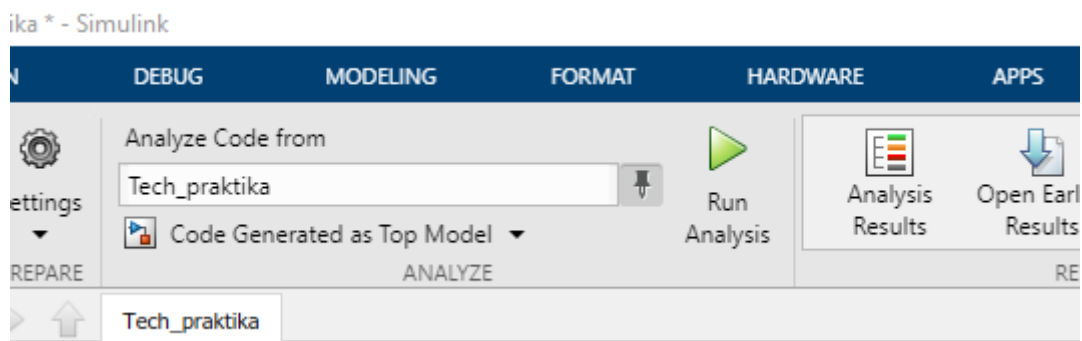


Рисунок 13.20 – Выбор действия

После этого, необходимо дождаться окончания процесса проверки. Так как в нашем случае в окне Command Window не появилось никаких ошибок в процессе анализа, то исходный код модели Simulink скомпилирован верно.

13.5 Проверка исходного кода модели Simulink модулем Analysis Results

Блок Analysis Results находится в меню Polyspace и открывается одним кликом мыши (рисунок 13.21):

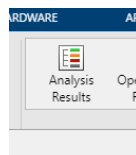


Рисунок 13.21 – Блок Analysis Results

После выбора блока необходимо произвести расчет с помощью кнопки Run Code Prover (рисунок 13.22):

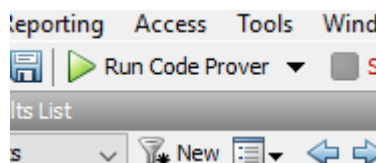


Рисунок 13.22 – Выбор кнопки

Начнется процесс выполнения расчетов анализа данных, представленный на рисунке 13.23:

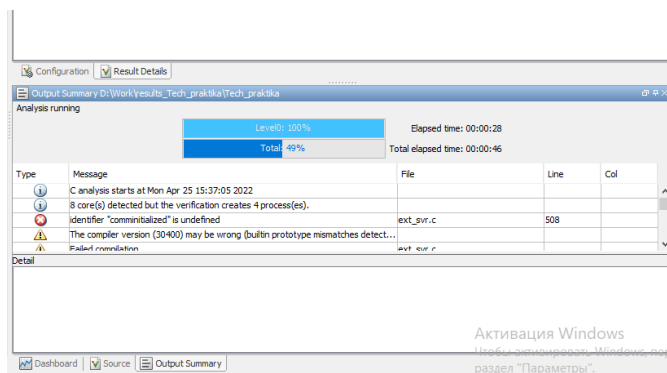


Рисунок 13.23 – Процесс выполнения анализа данных

После выполнения анализа данных получим диаграммы и таблицы, представленные на рисунке 13.24:

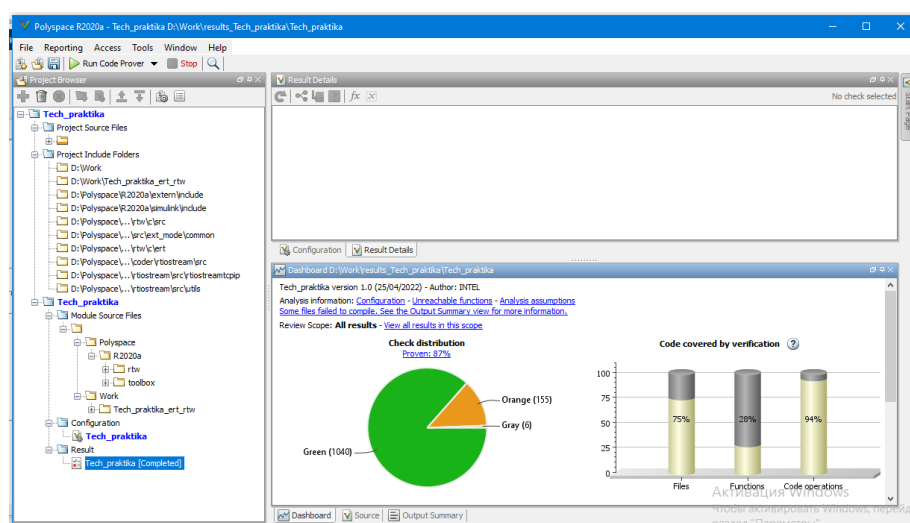


Рисунок 13.24 – Диаграммы после анализа данных

После выполнения данного блока можно заметить, что доказательство верификации кода составило 87%, потеря данных в 13% возникла в некоторых скомпилированных файлах, что привело к некоторым ошибкам в процессе компиляции, но основной заголовочный (.h) файл и файл с исходным кодом (.c) скомпилированы без ошибок.

Листинг кода на языке Си представлен в приложении А.

14 Проектирование графического интерфейса пользователя с помощью модуля AppDesigner

14.1 Описание файлов приложения

Структура основных файлов приложения представлена на рисунке 14.1:

MainApp	14.05.2022 16:27	Папка с файлами	
coefficients	16.05.2022 1:48	Microsoft Access ...	1 КБ
Data.xlsx	16.05.2022 20:10	Лист Microsoft Ex...	18 КБ
Koef.m	14.05.2022 17:30	Файл "M"	1 КБ
Koef_fuzzy.m	24.04.2022 19:25	Файл "M"	1 КБ
logo.png	14.05.2022 17:46	Файл "PNG"	46 КБ
MainApp.mlapp	16.05.2022 1:45	Файл "MLAPP"	66 КБ
MainApp.prj	14.05.2022 17:47	Файл "PRJ"	7 КБ
MainApp_exported.m	16.05.2022 17:40	Файл "M"	20 КБ
Mamdani_fuzzy.fis	16.05.2022 23:27	Файл "FIS"	1 КБ
Preddiplomnaya_praktika.slx	16.05.2022 23:38	Файл "SLX"	47 КБ
Screenshot_1.png	15.05.2022 17:09	Файл "PNG"	29 КБ
splash.png	16.05.2022 17:21	Файл "PNG"	201 КБ

Рисунок 14.1 – Структура файлов программы

Рассмотрим основные из них:

- Папка MainApp содержит файлы для установки и запуска приложения без инсталляции, и также файлы для быстрого тестирования;
- Файл coefficients.mat сохраняет введенные значения коэффициентов в самом приложении для дальнейшей работы;
- Файл Data.slx – данные модели Simulink;
- Файлы Koef.m и Koef_fuzzy.m – коэффициенты передаточных функций модели Simulink;
- Logo.png – пиктограмма приложения (иконка);
- MainApp.mlapp – файл проектирования диалоговой формы, содержит исходный код приложения;
- MainApp.prj – сборщик приложения (линковщик), осуществляет компиляцию файлов;
- Mamdani_fuzzy.fis – файл для работы нечеткого лингвистического регулятора;
- Preddiplomnaya_praktika.slx – модель Simulink;
- Splash.png – splash-скрин приложения (заставка перед запуском).

При открытии файла MainApp.mlapp в окне Workspace можно наблюдать графический интерфейс модуля AppDesigner Matlab [40] (рисунок 14.2):

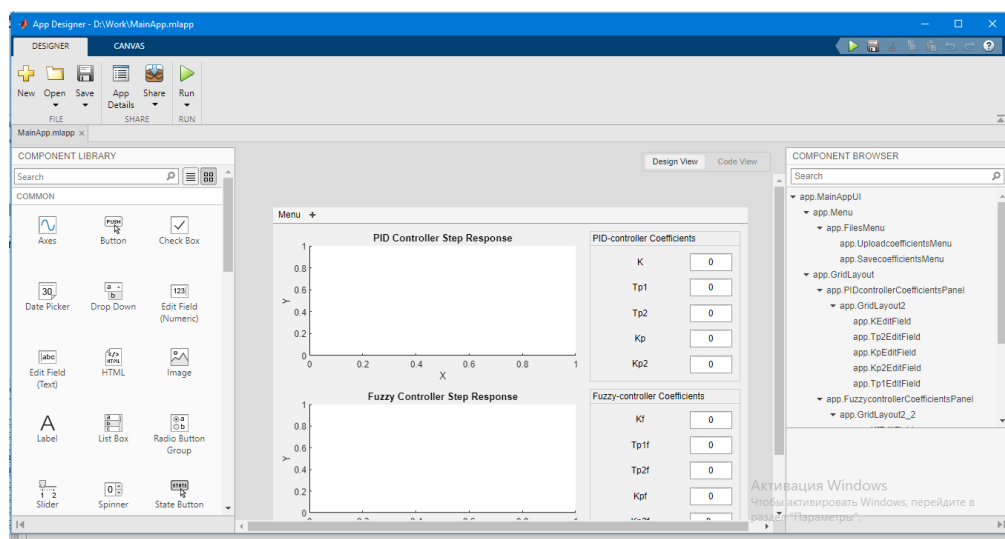


Рисунок 14.2 – Графический интерфейс и функционал AppDesigner

При открытии файла MainApp.prj наблюдаем меню со всеми основными файлами, которые необходимы для работы приложения (рисунок 14.3):

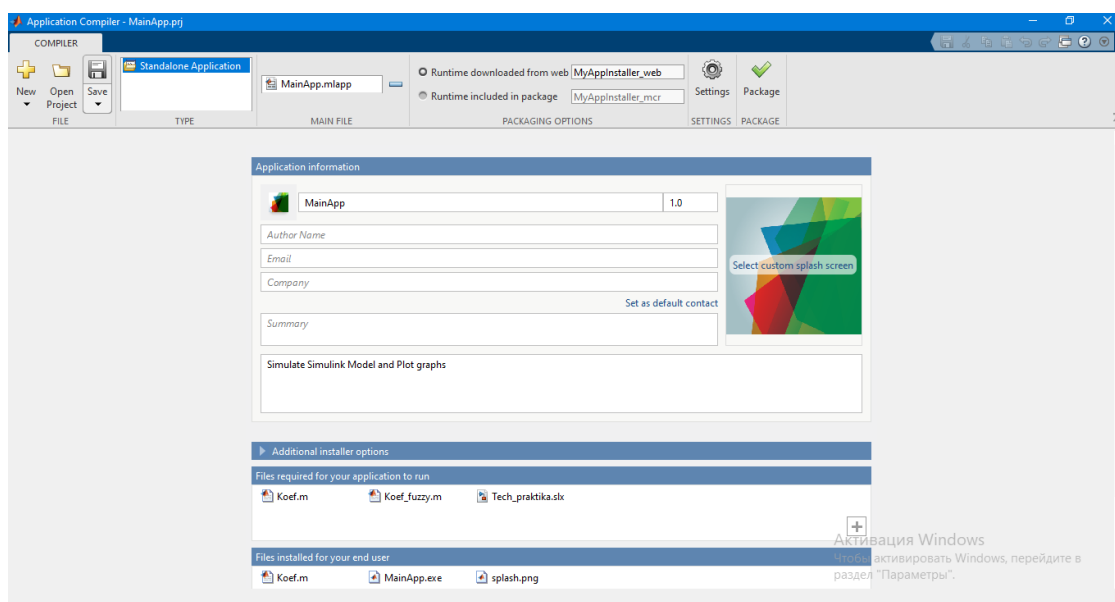


Рисунок 14.3 – Меню линковщика приложения

Splash-скрин приложения представлен на рисунке 14.4:



Рисунок 14.4 – Splash – экран приложения

Для запуска приложения удобнее всего перейти в папку MainApp и далее в папку for_redistribution_files_only, и в ней запустить файл MainApp.exe (рисунок 14.5):

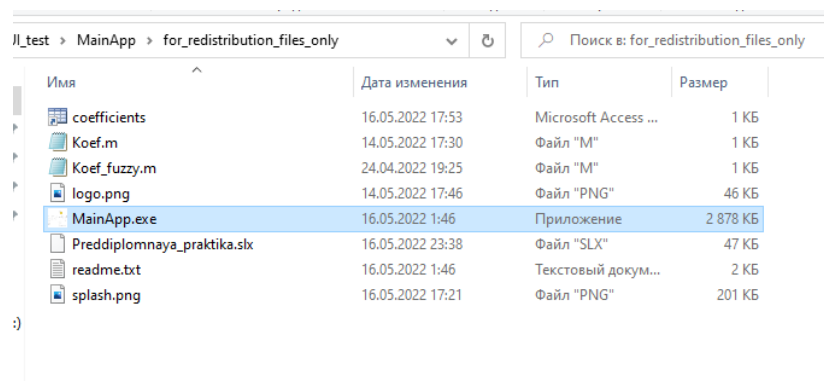


Рисунок 14.5 – Запуск приложения

14.2 Описание функционала приложения

После запуска приложения откроется диалоговая форма, представленная на рисунке 14.6:

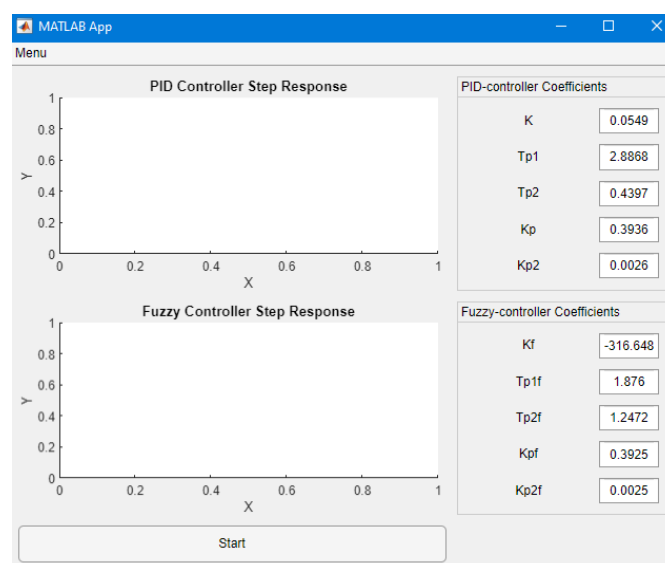


Рисунок 14.6 – Диалоговая форма приложения

Графики переходных процессов отображаются для ПИД и fuzzy – регуляторов отдельно в блоках PID Controller Step Response и Fuzzy Controller Step Response. Данные графики можно сохранять в расширении .png, конвертировать в растровый и векторный формат (рисунок 14.7):

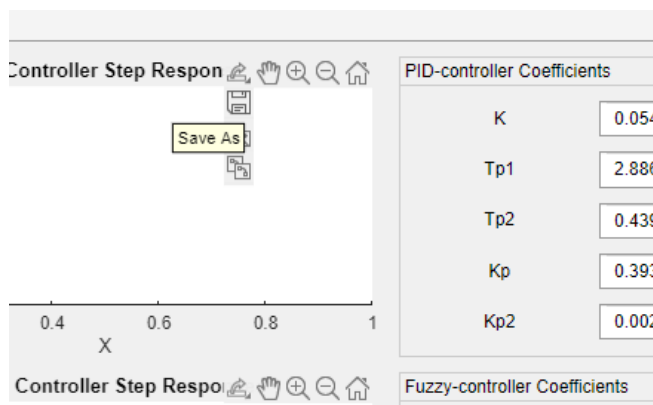


Рисунок 14.7 – Управление графиками

Графики можно перетаскивать внутри рабочей области (рисунок 14.8):

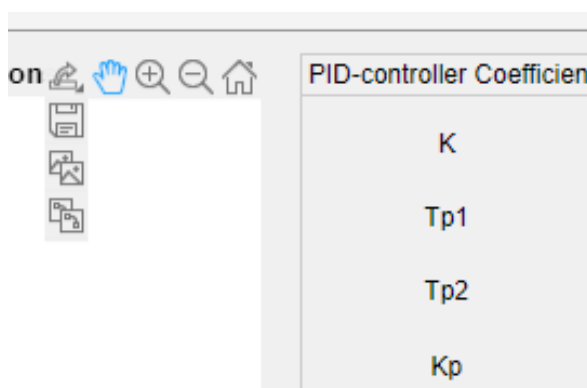


Рисунок 14.8 – Управление графиками

Можно увеличивать (рисунок 14.9) или уменьшать (рисунок 14.10) масштаб:



Рисунок 14.9 – Функция увеличения масштаба

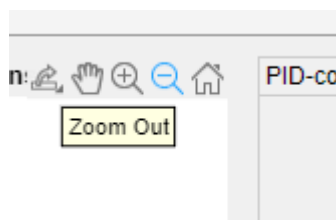


Рисунок 14.10 – Функция уменьшения масштаба

Кнопка Home возвращает все настройки графиков в первоначальные положения (рисунок 14.11):

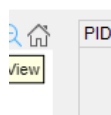


Рисунок 14.12 – Кнопка Home

Графический интерфейс загружает коэффициенты для моделей из m-файлов автоматически (рисунок 14.13):

 A screenshot of a software interface showing a form for loading coefficients for PID and Fuzzy controllers. The form has two sections: 'PID-controller Coefficients' and 'Fuzzy-controller Coefficients'. Each section contains five input fields with numerical values.

PID-controller Coefficients	
K	0.0549
Tr1	2.8868
Tr2	0.4397
Kp	0.3936
Kp2	0.0026

Fuzzy-controller Coefficients	
Kf	-316.648
Tr1f	1.876
Tr2f	1.2472
Kpf	0.3925
Kp2f	0.0025

Рисунок 14.14 – Загрузка коэффициентов в приложение

Также в приложении есть Рорип-меню (контекстное меню), в котором содержатся две функции: Upload coefficients (загружает коэффициенты из .mat-файла), Save coefficients (сохраняет коэффициенты, которые введены в программу вручную, в .mat – файл, рисунок 14.15):

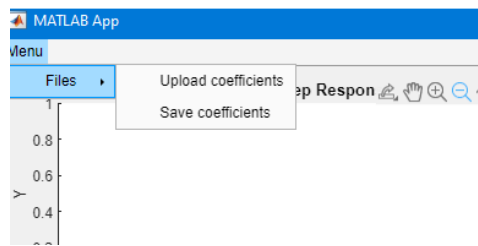


Рисунок 14.16 - Функционал контекстного меню

Листинг программного модуля представлен в приложении Б.

15 Тестирование приложения

Для начала работы с приложением, необходимо запустить симуляцию переходных процессов по модели, для этого в приложении необходимо нажать кнопку Start (рисунок 15.1):

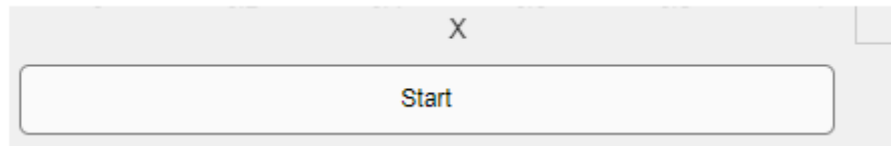


Рисунок 15.1 – Кнопка старта симуляции

В результате приложение отобразит графики переходных процессов, которые получены в модели Simulink на предыдущих этапах реализации (рисунок 15.2):

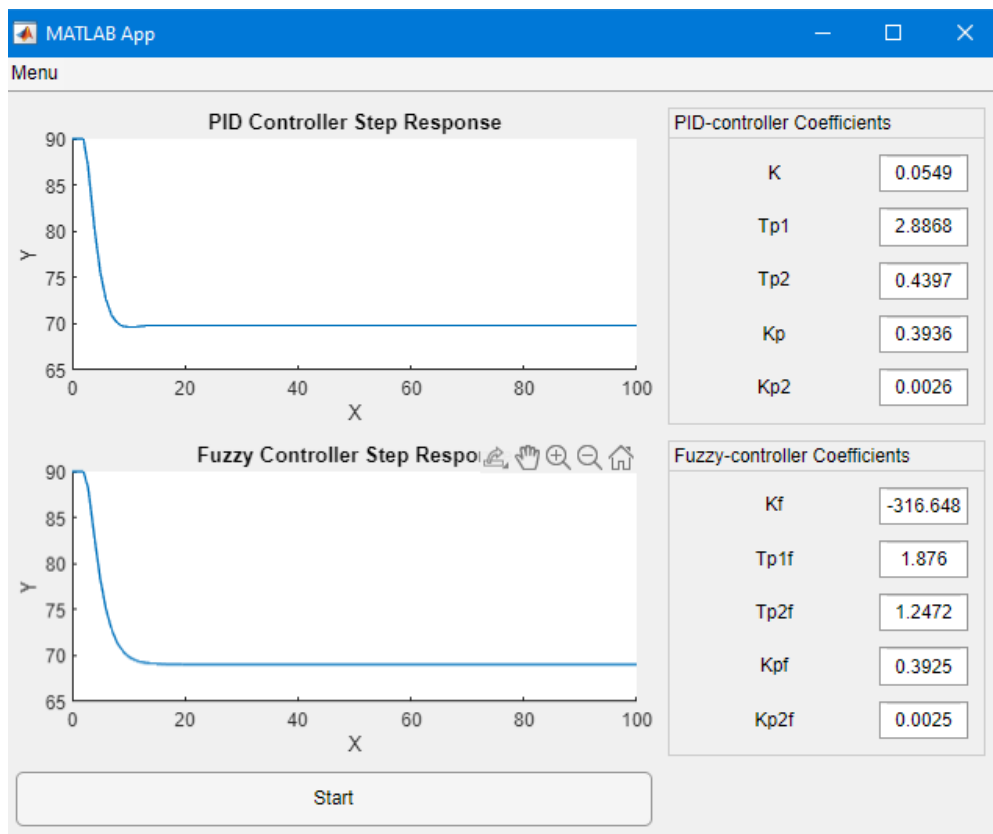


Рисунок 15.2 – Отображение графиков переходных процессов

Например, полученный график по ПИД-регулятору можно сохранить в расширении .png как растровое изображение, и открыть его любой программой, способной работать с данным расширением файла (рисунок 15.3):

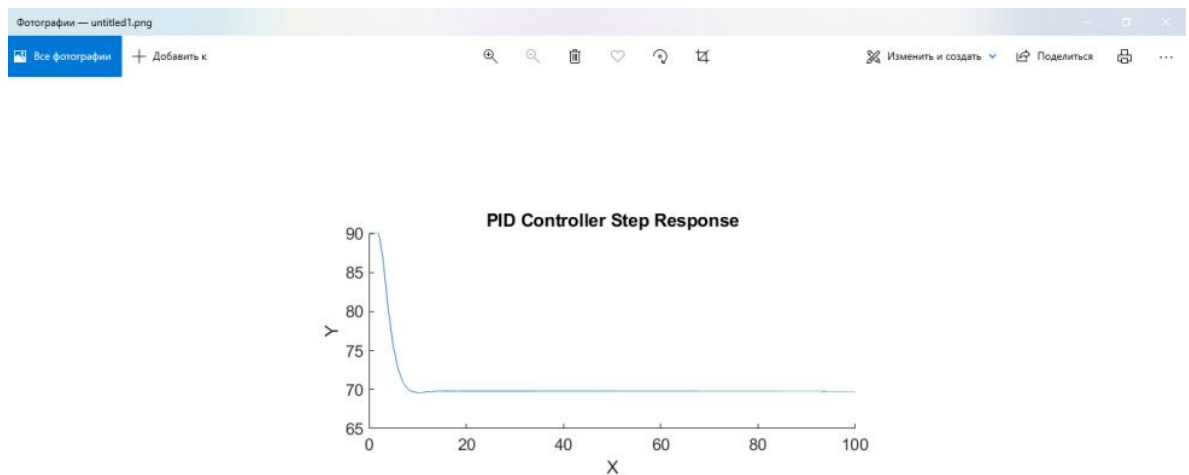


Рисунок 15.3 – Сохраненный график

Этот же график можно увеличить в масштабе (рисунок 15.4):

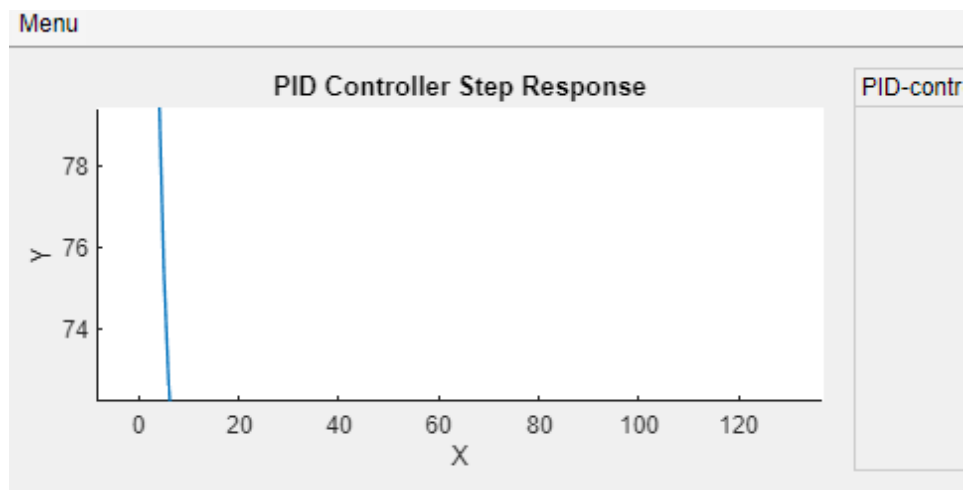


Рисунок 15.4 – График в увеличенном масштабе

Попробуем изменить значение коэффициента K для ПИД-регулятора и оценить существенные изменения, которые данный коэффициент оказывает на переходный процесс. Для этого введем значения коэффициента K , например, $K=1$ (рисунок 15.5):

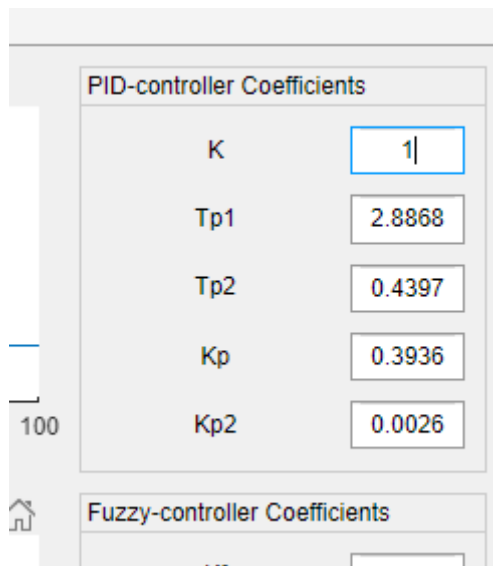


Рисунок 15.5 – Изменение коэффициента

Далее, необходимо нажать кнопку Start и дождаться выполнения симуляции. Как видно из полученного графика, переходный процесс теперь нас совершенно не удовлетворяет, и система говорит о том, что нужно изменить данные (рисунок 15.6):

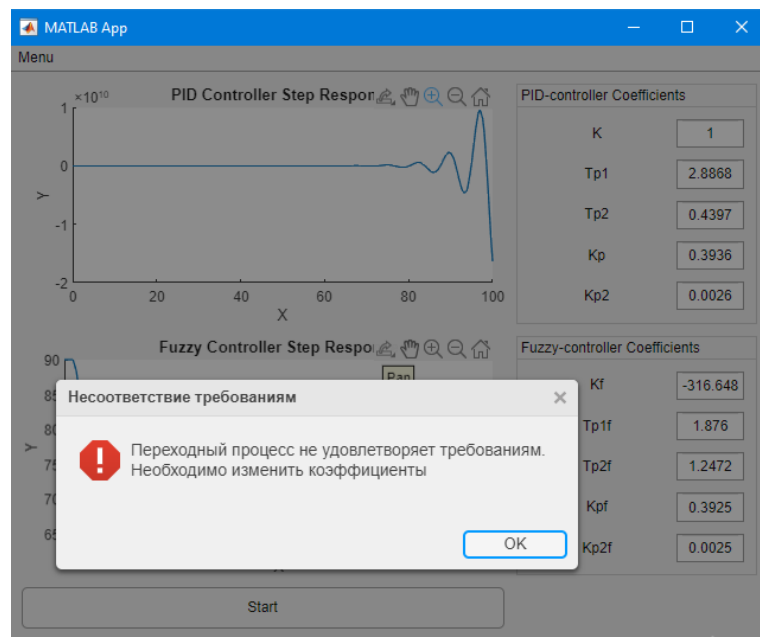


Рисунок 15.6 – Диалоговая подсказка

Проверим, сохраняются ли новые значения коэффициентов в .mat-файле. Для этого, перейдем в контекстное меню (рисунок 14.2.8) и нажмем кнопку Save Coefficients. Данные сохраняются в каталоге с приложением, из которого запущен исполняемый файл (рисунок 15.7):

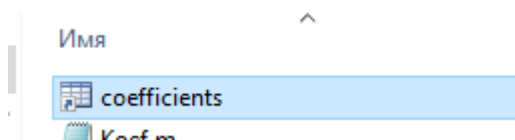


Рисунок 15.7 – Файл с новыми значениями коэффициентов

Открыв файл `coefficients.mat` в среде Matlab увидим, что значение коэффициента «K» действительно изменено на единицу и сохранено (рисунок 15.8):

```
struct with fields:
    k: 1
    kp: 0.3936
    kp2: 0.0026
    Tp1: 2.8868
    Tp2: 0.4397
    kf: -316.6483
    kpf: 0.3925
    kp2f: 0.0025
    Tp1f: 1.8760
    Tp2f: 1.2472
```

~ |

Рисунок 15.8 – Проверка изменения и сохранения коэффициентов

ЗАКЛЮЧЕНИЕ

Целью исследования в настоящей магистерской диссертации являлась разработка цифровой модели стабилизации температуры газа в компрессорных установках в среде Simulink и создание на ее базе учебного интерактивного обучающего тренажера в образовательных и научных целях.

В ходе изучения технологического процесса регулирования газа на компрессорной станции, детально проанализирован объект управления (аппарат воздушного охлаждения), разработано математическое описание переходного процесса охлаждения газа до заданного диапазона значений.

Проведена идентификация апериодической передаточной функции второго порядка (основное уравнение модели), оптимизированы коэффициенты ПИД-регулятора.

Для сравнения полученных результатов построен нечеткий лингвистический регулятор, с оптимальными характеристиками качества системы: перерегулирование, время регулирования и время установившегося переходного процесса.

Проведена проверка компьютерной модели на адекватность и устойчивость с помощью встроенных функций языка Matlab.

В учебных целях создан интерактивный тренажер в виде графического приложения стабилизации температуры газа в компрессорных установках для разработанных моделей систем управления с помощью модуля AppDesigner Toolbox. Автономный графический интерфейс с системой подсказок дополняет и расширяет функционал модели Simulink.

В дальнейшем планируется интегрировать данную цифровую модель со стендом стабилизации температуры с помощью программируемого логического контроллера фирмы PLC NEXT CONTACT (Phoenix AXC F2152) и реализовать полнофункциональный программно-аппаратный комплекс.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Описание технологического процесса воздушного охлаждения газа [Электронный ресурс]. – URL: https://studopedia.net/3_74742_opisanie-tehnologicheskogo-protssessa-vozdushnogo-ohlazhdeniya-gaza.html (дата обращения 2022-05-22).
2. Среда разработки MATLAB [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/MATLAB> (дата обращения 2022-05-24).
3. Среда разработки SiminTech [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Simintech> (дата обращения 2022-05-24).
4. Среда разработки Mathematica [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Mathematica> (дата обращения 2022-05-24).
5. Среда разработки MathCAD [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Mathcad> (дата обращения 2022-05-24).
6. GUI Matlab [Электронный ресурс]. – URL: <https://www.mathworks.com/discovery/matlab-gui.html> (дата обращения 2022-05-25).
7. Васильев В.В., Симак А.М., Рыбникова А.М. Математическое и компьютерное моделирование процессов и систем в среде Matlab/Simulink: учеб. пособие. Киев.: Изд-во НАН Украины, 2008. – 91 с.
8. Принципы моделирования процессов в техносфере [Электронный ресурс]. – URL: <https://portal.tpu.ru/SHARED/o/OKENO/uch/SAiMP/лекция%202.pdf> (дата обращения 2022-05-26).
9. Тепловой и гидравлический расчет теплообменной аппаратуры воздушного охлаждения [Электронный ресурс]. – URL: https://kostmash.ru/assets/_i_gidravlicheskiy_raschet_teploobmennogo_apparata_2009-1.pdf (дата обращения 2022-05-26).
10. Леонтьев А.П., Беев Э.А.. Расчет аппаратов воздушного охлаждения: учеб. пособие. – Тюмень: ТюмГНГУ, 2000. – 74 с.

11. Баутин С.П., Обухов А.Г., Математическое моделирование разрушительных атмосферных вихрей. Новосибирск, изд – во «Наука», 2012. – 152 с.
12. Баутин С.П., Крутова И.Ю., Обухов А.Г., Баутин К.В., Разрушительные атмосферные вихри. Екатеринбург, УрГУПС, 2013. – 215 с.
13. Леонтьев А.П., Беев Э.А.. Расчет аппаратов воздушного охлаждения: учеб. пособие. – Тюмень: ТюмГНГУ, 2000. – 74 с.
14. А.А. Сидягин, В.М. Косырев. Расчет и построение аппаратов воздушного охлаждения: учеб. пособие для студентов вузов. – Н. Новгород: Нижегород. гос. техн. ун-т им. Р.Е. Алексеева, 2009 – 150 с.
15. Ю.М. Бродов, К.Э. Аронсон, А.Ю. Рябчиков, М.А. Ниренштейн; под общ. ред. Ю.М. Бродова. Справочник по тепловым устройствам паротурбинных установок. – М.: Издательский дом МЭИ, 2016. – 300 с.
16. Зысин Л. В., Калютник А. А. Нагревобменное оборудование: учеб. пособие. СПб. : Изд-во Политехн. ун-та, 2010. – 230 с.
17. Разработка системы управления тепловым объектом [Электронный ресурс]. – URL: <https://digiratory.ru/361> (дата обращения 2022-05-27).
18. И.А. Мутугуллина. Устройство и расчет аппаратов воздушного охлаждения (АВО): учеб. пособие. - Бугульма:, 2017. – 80 с.
19. Введение в идентификацию при помощи Матлаб [Электронный ресурс]. – URL: <https://www.mathworks.com/videos/introduction-to-system-identification-97056.html> (дата обращения 2022-05-27).
20. ПИД-регулятор [Электронный ресурс]. – URL: <https://docs.exponenta.ru/simulink/slref/pidcontroller.html> (дата обращения 2022-05-27).
21. Критерий Найквиста [Электронный ресурс]. – URL: https://lms.kgeu.ru/pluginfile.php?file=%2F3644%2Fmod_resource%2Fcontent%2F0%2FЛекция%2012_Понятие%20об%20устойчивости.%20Частотные%20кр

итерии%20устойчивости.%20Критерий%20Найквиста.pdf (дата обращения 2022-05-27).

22. Критерий устойчивости Найквиста [Электронный ресурс]. – URL: https://studme.org/246765/tehnika/kriteriy_ustoychivosti_naykvista (дата обращения 2022-05-27).

23. Полюса динамической системы [Электронный ресурс]. – URL: https://translated.turbopages.org/proxy_u/en-ru.ru.3a840fe3-629a6e4a-0d2d20a6-74722d776562/https/www.mathworks.com/help/ident/ref/lti.pole.html (дата обращения 2022-05-27).

24. Власов, К.П. Теория автоматического управления. Основные положения. Примеры расчета: учеб. пособие / К.П. Власов. – Харьков: Гуман. Центр, 2013. – 544 с.

25. Власов, К.П. Теория автоматического управления. Основные положения. Программы расчета / К.П. Власов. – М.: Гуманитарный Центр, 2013. – 544 с.

26. Границы переходного процесса [Электронный ресурс]. – URL: <https://docs.exponenta.ru/slido/ref/checkstepresponsecharacteristics.html> (дата обращения 2022-05-28).

27. Адекватность модели, имитационное моделирование систем [Электронный ресурс]. – URL: https://studme.org/163940/informatika/adekvatnost_modeli (дата обращения 2022-05-28).

28. Глаголев, К.В. Физическая термодинамика: учеб. пособие / К.В. Глаголев, А.Н. Морозов. – М.: МГТУ, 2007. – 272 с.

29. Иванов, А.Е. Механика. Молекулярная физика и термодинамика: Учебник / А.Е. Иванов, С.А. Иванов. – М.: КноРус, 2016. – 320 с.

30. Иванов, А.Е. Молекулярная физика и термодинамика. том 1 / А.Е. Иванов. – М.: Русайнс, 2018. – 272 с.

31. Fuzzy Logic Toolbox [Электронный ресурс]. – URL: <https://www.mathworks.com/products/fuzzy-logic.html> (дата обращения 2022-05-28).
32. Нечеткие выводы [Электронный ресурс]. – URL: <https://neuronus.com/theory/fl/308-nechetkie-vyvody.html> (дата обращения 2022-05-28).
33. Квасников, И.А. Термодинамика и статистическая физика. Т. 4: Квантовая статистика / И.А. Квасников. – М.: КомКнига, 2014. – 352 с.
34. Квасников, И.А. Термодинамика и статистическая физика. Т.2: Теория равновесных систем: Статистическая физика / И.А. Квасников. – М.: УРСС, 2016. – 432 с.
35. Квасников, И.А. Термодинамика и статистическая физика: Теория равновесных систем: Статистическая физика / И.А. Квасников. – М.: УРСС, 2015. – 436 с.
36. Горобец, Г.С. Теория вероятностей, математическая статистика и элементы случайных процессов / Г.С. Горобец. – М.: КД Либроком, 2013. – 232 с.
37. Анализ данных в Excel [Электронный ресурс]. – URL: <https://support.microsoft.com/ru-ru/office/анализ-данных-в-excel-3223aab8-f543-4fda-85ed-76bb0295ffc4> (дата обращения 2022-05-29).
38. Экспонента. Описание модуля Embeeded Coder [Электронный ресурс]. – URL: <https://exponenta.ru/embedded-coder> (дата обращения 2022-06-01).
39. Code Verification [Электронный ресурс]. – URL: https://www.mathworks.com/help/coder/codeverification.html?s_tid=CRUX_lftnav (дата обращения 2022-06-01).
40. App Designer Matlab [Электронный ресурс]. – URL: https://www.mathworks.com/help/coder/codeverification.html?s_tid=CRUX_lftnav (дата обращения 2022-06-01).

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программного кода модели Simulink на языке Си

```
/*
 * File: Tech_praktika.c
 *
 * Code generated for Simulink model 'Tech_praktika'.
 *
 * Model version          : 1.12
 * Simulink Coder version  : 9.3 (R2020a) 18-Nov-2019
 * C/C++ source code generated on : Mon Apr 25 15:23:53 2022
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives:
 *   1. Execution efficiency
 *   2. RAM efficiency
 * Validation result: Not run
 */

#include "Tech_praktika.h"
#include "Tech_praktika_dt.h"
#define NumBitsPerChar      8U

/* Private macros used by the generated code to access rtModel */
#ifndef rtmIsMajorTimeStep
#define rtmIsMajorTimeStep(rtm)    (((rtm)->Timing.simTimeStep) == MAJOR_TIME_STEP)
#endif

#ifndef rtmIsMinorTimeStep
#define rtmIsMinorTimeStep(rtm)    (((rtm)->Timing.simTimeStep) == MINOR_TIME_STEP)
#endif

#ifndef rtmSetTFinal
#define rtmSetTFinal(rtm, val)    ((rtm)->Timing.tFinal = (val))
#endif

#ifndef rtmSetTPtr
#define rtmSetTPtr(rtm, val)      ((rtm)->Timing.t = (val))
#endif

/* Block signals (default storage) */
B rtB;

/* Continuous states */
X rtX;

/* Block states (default storage) */
DW rtDW;

/* Real-time model */
RT_MODEL rtM_;
RT_MODEL *const rtM = &rtM_;
extern real_T rt_urand_Upu32_Yd_f_pw_snf(uint32_T *u);
extern real_T rt_nrand_Upu32_Yd_f_pw_snf(uint32_T *u);
real_T rt_TDelayInterpolate(
    real_T tMinusDelay,          /* tMinusDelay = currentSimTime - delay */
    real_T tStart,
    real_T *tBuf,
    real_T *uBuf,
    int_T bufSz,
    int_T *lastIdx,
    int_T oldestIdx,
    int_T newIdx,
    real_T initOutput,
    boolean_T discrete,
    boolean_T minorStepAndTAtLastMajorOutput)
;

/* private model entry point functions */
extern void Tech_praktika_derivatives(void);
```

```

/* Forward declaration for local functions */
static real_T trimf(real_T x, const real_T params[3]);
static void trimf_i(const real_T x[101], const real_T params[3], real_T y[101]);
extern real_T rtInf;
extern real_T rtMinusInf;
extern real_T rtNaN;
extern real32_T rtInfF;
extern real32_T rtMinusInfF;
extern real32_T rtNaNF;
static void rt_InitInfAndNaN(size_t realSize);
static boolean_T rtIsInf(real_T value);
static boolean_T rtIsInfF(real32_T value);
static boolean_T rtIsNaN(real_T value);
static boolean_T rtIsNaNF(real32_T value);
typedef struct {
    struct {
        uint32_T wordH;
        uint32_T wordL;
    } words;
} BigEndianIEEEDouble;

typedef struct {
    struct {
        uint32_T wordL;
        uint32_T wordH;
    } words;
} LittleEndianIEEEDouble;

typedef struct {
    union {
        real32_T wordLreal;
        uint32_T wordLuint;
    } wordL;
} IEEESingle;

real_T rtInf;
real_T rtMinusInf;
real_T rtNaN;
real32_T rtInfF;
real32_T rtMinusInfF;
real32_T rtNaNF;
static real_T rtGetInf(void);
static real32_T rtGetInfF(void);
static real_T rtGetMinusInf(void);
static real32_T rtGetMinusInfF(void);
static real_T rtGetNaN(void);
static real32_T rtGetNaNF(void);

/*
 * Initialize the rtInf, rtMinusInf, and rtNaN needed by the
 * generated code. NaN is initialized as non-signaling. Assumes IEEE.
 */
static void rt_InitInfAndNaN(size_t realSize)
{
    (void) (realSize);
    rtNaN = rtGetNaN();
    rtNaNF = rtGetNaNF();
    rtInf = rtGetInf();
    rtInfF = rtGetInfF();
    rtMinusInf = rtGetMinusInf();
    rtMinusInfF = rtGetMinusInfF();
}

/* Test if value is infinite */
static boolean_T rtIsInf(real_T value)
{
    return (boolean_T)((value==rtInf || value==rtMinusInf) ? 1U : 0U);
}

/* Test if single-precision value is infinite */
static boolean_T rtIsInfF(real32_T value)
{
    return (boolean_T)(((value)==rtInfF || (value)==rtMinusInfF) ? 1U : 0U);
}

/* Test if value is not a number */
static boolean_T rtIsNaN(real_T value)
{
    boolean_T result = (boolean_T) 0;

```

```

size_t bitsPerReal = sizeof(real_T) * (NumBitsPerChar);
if (bitsPerReal == 32U) {
    result = rtIsNaNF((real32_T)value);
} else {
    union {
        LittleEndianIEEEDouble bitVal;
        real_T fltVal;
    } tmpVal;

    tmpVal.fltVal = value;
    result = (boolean_T)((tmpVal.bitVal.words.wordH & 0x7FF00000) == 0x7FF00000 &&
        (tmpVal.bitVal.words.wordH & 0x000FFFFF) != 0 ||
        (tmpVal.bitVal.words.wordL != 0));
}

return result;
}

/* Test if single-precision value is not a number */
static boolean_T rtIsNaNF(real32_T value)
{
    IEEESingle tmp;
    tmp.wordL.wordLreal = value;
    return (boolean_T)((tmp.wordL.wordLuint & 0x7F800000) == 0x7F800000 &&
        (tmp.wordL.wordLuint & 0x007FFFFF) != 0);
}

/*
 * Initialize rtInf needed by the generated code.
 * Inf is initialized as non-signaling. Assumes IEEE.
 */
static real_T rtGetInf(void)
{
    size_t bitsPerReal = sizeof(real_T) * (NumBitsPerChar);
    real_T inf = 0.0;
    if (bitsPerReal == 32U) {
        inf = rtGetInfF();
    } else {
        union {
            LittleEndianIEEEDouble bitVal;
            real_T fltVal;
        } tmpVal;

        tmpVal.bitVal.words.wordH = 0x7FF00000U;
        tmpVal.bitVal.words.wordL = 0x00000000U;
        inf = tmpVal.fltVal;
    }

    return inf;
}

/*
 * Initialize rtInfF needed by the generated code.
 * Inf is initialized as non-signaling. Assumes IEEE.
 */
static real32_T rtGetInfF(void)
{
    IEEESingle infF;
    infF.wordL.wordLuint = 0x7F800000U;
    return infF.wordL.wordLreal;
}

/*
 * Initialize rtMinusInf needed by the generated code.
 * Inf is initialized as non-signaling. Assumes IEEE.
 */
static real_T rtGetMinusInf(void)
{
    size_t bitsPerReal = sizeof(real_T) * (NumBitsPerChar);
    real_T minf = 0.0;
    if (bitsPerReal == 32U) {
        minf = rtGetMinusInfF();
    } else {
        union {
            LittleEndianIEEEDouble bitVal;
            real_T fltVal;
        } tmpVal;

        tmpVal.bitVal.words.wordH = 0xFFF00000U;

```

```

    tmpVal.bitVal.words.wordL = 0x00000000U;
    minf = tmpVal.flrVal;
}

return minf;
}

/*
 * Initialize rtMinusInfF needed by the generated code.
 * Inf is initialized as non-signaling. Assumes IEEE.
 */
static real32_T rtGetMinusInfF(void)
{
    IEEEFloat minf;
    minf.wordL.wordLuint = 0xFF800000U;
    return minf.wordL.wordLreal;
}

/*
 * Initialize rtNaN needed by the generated code.
 * NaN is initialized as non-signaling. Assumes IEEE.
 */
static real_T rtGetNaN(void)
{
    size_t bitsPerReal = sizeof(real_T) * (NumBitsPerChar);
    real_T nan = 0.0;
    if (bitsPerReal == 32U) {
        nan = rtGetNaNF();
    } else {
        union {
            LittleEndianIEEEDouble bitVal;
            real_T flrVal;
        } tmpVal;

        tmpVal.bitVal.words.wordH = 0xFFF80000U;
        tmpVal.bitVal.words.wordL = 0x00000000U;
        nan = tmpVal.flrVal;
    }

    return nan;
}

/*
 * Initialize rtNaNF needed by the generated code.
 * NaN is initialized as non-signaling. Assumes IEEE.
 */
static real32_T rtGetNaNF(void)
{
    IEEEFloat nanF = { { 0 } };

    nanF.wordL.wordLuint = 0xFFC00000U;
    return nanF.wordL.wordLreal;
}

/*
 * Time delay interpolation routine
 *
 * The linear interpolation is performed using the formula:
 *
 * 
$$u(t) = \frac{(t_2 - t_{\text{MinusDelay}})}{(t_2 - t_1)} * u_1 + \frac{(t_{\text{MinusDelay}} - t_1)}{(t_2 - t_1)} * u_2$$

 */
real_T rt_TDelayInterpolate(
    real_T tMinusDelay, /* tMinusDelay = currentSimTime - delay */
    real_T tStart,
    real_T *tBuf,
    real_T *uBuf,
    int_T bufSz,
    int_T *lastIdx,
    int_T oldestIdx,
    int_T newIdx,
    real_T initOutput,
    boolean_T discrete,
    boolean_T minorStepAndTAtLastMajorOutput)
{
    int_T i;
    real_T yout, t1, t2, u1, u2;

```

```

/*
 * If there is only one data point in the buffer, this data point must be
 * the t= 0 and tMinusDelay > t0, it ask for something unknown. The best
 * guess if initial output as well
 */
if ((newIdx == 0) && (oldestIdx == 0) && (tMinusDelay > tStart))
    return initOutput;

/*
 * If tMinusDelay is less than zero, should output initial value
 */
if (tMinusDelay <= tStart)
    return initOutput;

/* For fixed buffer extrapolation:
 * if tMinusDelay is small than the time at oldestIdx, if discrete, output
 * tailptr value, else use tailptr and tailptr+1 value to extrapolate
 * It is also for fixed buffer. Note: The same condition can happen for transport delay block where
 * use tStart and t[tail] other than using t[tail] and t[tail+1].
 * See below
 */
if ((tMinusDelay <= tBuf[oldestIdx]) ) {
    if (discrete) {
        return(uBuf[oldestIdx]);
    } else {
        int_T tempIdx= oldestIdx + 1;
        if (oldestIdx == bufSz-1)
            tempIdx = 0;
        t1= tBuf[oldestIdx];
        t2= tBuf[tempIdx];
        u1= uBuf[oldestIdx];
        u2= uBuf[tempIdx];
        if (t2 == t1) {
            if (tMinusDelay >= t2) {
                yout = u2;
            } else {
                yout = u1;
            }
        } else {
            real_T f1 = (t2-tMinusDelay) / (t2-t1);
            real_T f2 = 1.0 - f1;

            /*
             * Use Lagrange's interpolation formula. Exact outputs at t1, t2.
             */
            yout = f1*u1 + f2*u2;
        }

        return yout;
    }
}

/*
 * When block does not have direct feedthrough, we use the table of
 * values to extrapolate off the end of the table for delays that are less
 * than 0 (less then step size). This is not completely accurate. The
 * chain of events is as follows for a given time t. Major output - look
 * in table. Update - add entry to table. Now, if we call the output at
 * time t again, there is a new entry in the table. For very small delays,
 * this means that we will have a different answer from the previous call
 * to the output fcn at the same time t. The following code prevents this
 * from happening.
 */
if (minorStepAndTAtLastMajorOutput) {
    /* pretend that the new entry has not been added to table */
    if (newIdx != 0) {
        if (*lastIdx == newIdx) {
            (*lastIdx)--;
        }

        newIdx--;
    } else {
        if (*lastIdx == newIdx) {
            *lastIdx = bufSz-1;
        }

        newIdx = bufSz - 1;
    }
}

```

```

i = *lastIdx;
if (tBuf[i] < tMinusDelay) {
    /* Look forward starting at last index */
    while (tBuf[i] < tMinusDelay) {
        /* May occur if the delay is less than step-size - extrapolate */
        if (i == newIdx)
            break;
        i = (i < (bufSz-1)) ? (i+1) : 0; /* move through buffer */
    }
} else {
    /*
     * Look backwards starting at last index which can happen when the
     * delay time increases.
     */
    while (tBuf[i] >= tMinusDelay) {
        /*
         * Due to the entry condition at top of function, we
         * should never hit the end.
         */
        i = (i > 0) ? i-1 : (bufSz-1); /* move through buffer */
    }

    i = (i < (bufSz-1)) ? (i+1) : 0;
}

*lastIdx = i;
if (discrete) {
    /*
     * tempEps = 128 * eps;
     * localEps = max(tempEps, tempEps*fabs(tBuf[i]))/2;
     */
    double tempEps = (DBL_EPSILON) * 128.0;
    double localEps = tempEps * fabs(tBuf[i]);
    if (tempEps > localEps) {
        localEps = tempEps;
    }

    localEps = localEps / 2.0;
    if (tMinusDelay >= (tBuf[i] - localEps)) {
        yout = uBuf[i];
    } else {
        if (i == 0) {
            yout = uBuf[bufSz-1];
        } else {
            yout = uBuf[i-1];
        }
    }
} else {
    if (i == 0) {
        t1 = tBuf[bufSz-1];
        u1 = uBuf[bufSz-1];
    } else {
        t1 = tBuf[i-1];
        u1 = uBuf[i-1];
    }

    t2 = tBuf[i];
    u2 = uBuf[i];
    if (t2 == t1) {
        if (tMinusDelay >= t2) {
            yout = u2;
        } else {
            yout = u1;
        }
    } else {
        real_T f1 = (t2-tMinusDelay) / (t2-t1);
        real_T f2 = 1.0 - f1;

        /*
         * Use Lagrange's interpolation formula. Exact outputs at t1, t2.
         */
        yout = f1*u1 + f2*u2;
    }
}

return(yout);
}

```

```

/*
 * This function updates continuous states using the ODE3 fixed-step
 * solver algorithm
 */
static void rt_ertODEUpdateContinuousStates(RTWSolverInfo *si )
{
    /* Solver Matrices */
    static const real_T rt_ODE3_A[3] = {
        1.0/2.0, 3.0/4.0, 1.0
    };

    static const real_T rt_ODE3_B[3][3] = {
        { 1.0/2.0, 0.0, 0.0 },

        { 0.0, 3.0/4.0, 0.0 },

        { 2.0/9.0, 1.0/3.0, 4.0/9.0 }
    };

    time_T t = rtsiGetT(si);
    time_T tnew = rtsiGetSolverStopTime(si);
    time_T h = rtsiGetStepSize(si);
    real_T *x = rtsiGetContStates(si);
    ODE3_IntgData *id = (ODE3_IntgData *)rtsiGetSolverData(si);
    real_T *y = id->y;
    real_T *f0 = id->f[0];
    real_T *f1 = id->f[1];
    real_T *f2 = id->f[2];
    real_T hB[3];
    int_T i;
    int_T nXc = 8;
    rtsiSetSimTimeStep(si, MINOR_TIME_STEP);

    /* Save the state values at time t in y, we'll use x as ynew. */
    (void) memcpy(y, x,
        (uint_T)nXc*sizeof(real_T));

    /* Assumes that rtsiSetT and ModelOutputs are up-to-date */
    /* f0 = f(t,y) */
    rtsiSetdX(si, f0);
    Tech_praktika_derivatives();

    /* f(:,2) = feval(odefile, t + hA(1), y + f*hB(:,1), args(:)(*)) */
    hB[0] = h * rt_ODE3_B[0][0];
    for (i = 0; i < nXc; i++) {
        x[i] = y[i] + (f0[i]*hB[0]);
    }

    rtsiSetT(si, t + h*rt_ODE3_A[0]);
    rtsiSetdX(si, f1);
    Tech_praktika_step0();
    Tech_praktika_derivatives();

    /* f(:,3) = feval(odefile, t + hA(2), y + f*hB(:,2), args(:)(*)) */
    for (i = 0; i <= 1; i++) {
        hB[i] = h * rt_ODE3_B[1][i];
    }

    for (i = 0; i < nXc; i++) {
        x[i] = y[i] + (f0[i]*hB[0] + f1[i]*hB[1]);
    }

    rtsiSetT(si, t + h*rt_ODE3_A[1]);
    rtsiSetdX(si, f2);
    Tech_praktika_step0();
    Tech_praktika_derivatives();

    /* tnew = t + hA(3);
    ynew = y + f*hB(:,3); */
    for (i = 0; i <= 2; i++) {
        hB[i] = h * rt_ODE3_B[2][i];
    }

    for (i = 0; i < nXc; i++) {
        x[i] = y[i] + (f0[i]*hB[0] + f1[i]*hB[1] + f2[i]*hB[2]);
    }

    rtsiSetT(si, tnew);
    rtsiSetSimTimeStep(si, MAJOR_TIME_STEP);

```

```

}

/* Function for MATLAB Function: '<S3>/Evaluate Rule Antecedents' */
static real_T trimf(real_T x, const real_T params[3])
{
    real_T y;
    y = 0.0;
    if ((params[0] != params[1]) && (params[0] < x) && (x < params[1])) {
        y = 1.0 / (params[1] - params[0]) * (x - params[0]);
    }

    if ((params[1] != params[2]) && (params[1] < x) && (x < params[2])) {
        y = 1.0 / (params[2] - params[1]) * (params[2] - x);
    }

    if (x == params[1]) {
        y = 1.0;
    }

    return y;
}

/* Function for MATLAB Function: '<S3>/Evaluate Rule Consequents' */
static void trimf_i(const real_T x[101], const real_T params[3], real_T y[101])
{
    real_T a;
    real_T b;
    real_T c;
    int32_T i;
    a = params[0];
    b = params[1];
    c = params[2];
    for (i = 0; i < 101; i++) {
        y[i] = 0.0;
        if ((a != b) && (a < x[i]) && (x[i] < b)) {
            y[i] = 1.0 / (b - a) * (x[i] - a);
        }

        if ((b != c) && (b < x[i]) && (x[i] < c)) {
            y[i] = 1.0 / (c - b) * (c - x[i]);
        }

        if (x[i] == b) {
            y[i] = 1.0;
        }
    }
}

real_T rt_urand_Upu32_Yd_f_pw_snf(uint32_T *u)
{
    uint32_T lo;
    uint32_T hi;

    /* Uniform random number generator (random number between 0 and 1)

    #define IA    16807          magic multiplier = 7^5
    #define IM    2147483647      modulus = 2^31-1
    #define IQ    127773         IM div IA
    #define IR    2836           IM modulo IA
    #define S     4.656612875245797e-10  reciprocal of 2^31-1
    test = IA * (seed % IQ) - IR * (seed/IQ)
    seed = test < 0 ? (test + IM) : test
    return (seed*S)
    */
    lo = *u % 127773U * 16807U;
    hi = *u / 127773U * 2836U;
    if (lo < hi) {
        *u = 2147483647U - (hi - lo);
    } else {
        *u = lo - hi;
    }

    return (real_T)*u * 4.6566128752457969E-10;
}

real_T rt_rndrand_Upu32_Yd_f_pw_snf(uint32_T *u)
{
    real_T y;
    real_T sr;

```

```

real_T si;

/* Normal (Gaussian) random number generator */
do {
    sr = 2.0 * rt_urand_Upu32_Yd_f_pw_snf(u) - 1.0;
    si = 2.0 * rt_urand_Upu32_Yd_f_pw_snf(u) - 1.0;
    si = sr * sr + si * si;
} while (si > 1.0);

y = sqrt(-2.0 * log(si) / si) * sr;
return y;
}

/* Model step function for TID0 */
void Tech_praktika_step0(void) /* Sample time: [0.0s, 0.0s] */
{
    /* local block i/o variables */
    real_T rtb_TransferFcn2;
    real_T rtb_TransferFcn4;
    real_T rtb_Sum1;
    real_T inputMFCache[3];
    real_T outputMFCache[202];
    real_T area;
    real_T rtb_antecedentOutputs[3];
    real_T rtb_Sum4;
    real_T rtb_aggregatedOutputs[101];
    int32_T i;
    real_T tmp[101];
    real_T tmp_0[101];
    real_T x_idx_0;
    int32_T outputMFCache_tmp;
    static const real_T e[3] = { 19.5, 21.0, 23.0 };

    static const real_T d[3] = { 24.0, 24.5, 24.7 };

    static const real_T c[3] = { 24.8, 24.9, 25.0 };

    static const real_T c_0[3] = { 65.0, 67.0, 70.0 };

    static const int8_T b[3] = { 1, 2, 2 };

    if (rtmIsMajorTimeStep(rtm)) {
        /* set solver stop time */
        rtsiSetSolverStopTime(&rtM->solverInfo, ((rtM->Timing.clockTick0+1)*
            rtM->Timing.stepSize0));

        /* Update the flag to indicate when data transfers from
         * Sample time: [1.0s, 0.0s] to Sample time: [100.0s, 0.0s] */
        (rtM->Timing.RateInteraction.TID1_2)++;
        if ((rtM->Timing.RateInteraction.TID1_2) > 99) {
            rtM->Timing.RateInteraction.TID1_2 = 0;
        }
    } /* end MajorTimeStep */

    /* Update absolute time of base rate at minor time step */
    if (rtmIsMinorTimeStep(rtm)) {
        rtM->Timing.t[0] = rtsiGetT(&rtM->solverInfo);
    }

    /* TransportDelay: '<Root>/Transport Delay1' */
    {
        real_T **uBuffer = (real_T**) &rtDW.TransportDelay1_PWORK.TUbufferPtrs[0];
        real_T **tBuffer = (real_T**) &rtDW.TransportDelay1_PWORK.TUbufferPtrs[1];
        real_T simTime = rtM->Timing.t[0];
        real_T tMinusDelay = simTime - 2.0896;
        rtb_Sum1 = rt_TDelayInterpolate(
            tMinusDelay,
            0.0,
            *tBuffer,
            *uBuffer,
            rtDW.TransportDelay1_IWORK.CircularBufSize,
            &rtDW.TransportDelay1_IWORK.Last,
            rtDW.TransportDelay1_IWORK.Tail,
            rtDW.TransportDelay1_IWORK.Head,
            0.0,
            0,
            0);
    }
}

```

```

/* Sum: '<Root>/Sum2' incorporates:
 * Constant: '<Root>/Constant'
 */
rtB.Sum2 = rtb_Sum1 + 90.0;

/* TransportDelay: '<Root>/Transport Delay2' */
{
    real_T **uBuffer = (real_T**) &rtDW.TransportDelay2_PWORK.TUbufferPtrs[0];
    real_T **tBuffer = (real_T**) &rtDW.TransportDelay2_PWORK.TUbufferPtrs[1];
    real_T simTime = rtM->Timing.t[0];
    real_T tMinusDelay = simTime - 2.0896;
    rtb_Sum1 = rt_TDelayInterpolate(
        tMinusDelay,
        0.0,
        *tBuffer,
        *uBuffer,
        rtDW.TransportDelay2_IWORK.CircularBufSize,
        &rtDW.TransportDelay2_IWORK.Last,
        rtDW.TransportDelay2_IWORK.Tail,
        rtDW.TransportDelay2_IWORK.Head,
        0.0,
        0,
        0);
}

/* Sum: '<Root>/Sum5' incorporates:
 * Constant: '<Root>/Constant1'
 */
rtB.Sum5 = rtb_Sum1 + 90.0;

/* RateTransition generated from: '<Root>/Sum4' */
if (rtmIsMajorTimeStep(rtM)) {
    /* Step: '<Root>/Step1' */
    rtB.Step1 = !(((rtM->Timing.clockTick1) * 1.0) < 100.0);
    if (rtM->Timing.RateInteraction.TID1_2 == 1) {
        rtB.TmpRTBAAtSum4Inport3 = rtDW.TmpRTBAAtSum4Inport3_Buffer0;
    }
}

/* End of RateTransition generated from: '<Root>/Sum4' */

/* Integrator: '<Root>/Integrator1' */
rtb_Sum1 = rtX.Integrator1_CSTATE;

/* Sum: '<Root>/Sum4' incorporates:
 * Gain: '<Root>/Cr1'
 * Gain: '<S7>/Slider Gain'
 * Sum: '<Root>/Sum3'
 */
rtb_Sum4 = ((rtB.Step1 - 0.09 * rtB.Sum5) * -5.6 - rtb_Sum1) +
    rtB.TmpRTBAAtSum4Inport3;

/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller' */
/* MATLAB Function: '<S3>/Evaluate Rule Antecedents' */
inputMFCache[0] = trimf(rtb_Sum4, e);
inputMFCache[1] = trimf(rtb_Sum4, d);
inputMFCache[2] = trimf(rtb_Sum4, c);
if (1.0 > inputMFCache[0]) {
    rtb_Sum4 = inputMFCache[0];
} else {
    rtb_Sum4 = 1.0;
}

area = rtb_Sum4;
rtb_antecedentOutputs[0] = rtb_Sum4;
if (1.0 > inputMFCache[2]) {
    rtb_Sum4 = inputMFCache[2];
} else {
    rtb_Sum4 = 1.0;
}

area += rtb_Sum4;
rtb_antecedentOutputs[1] = rtb_Sum4;
if (1.0 > inputMFCache[1]) {
    rtb_Sum4 = inputMFCache[1];
} else {
    rtb_Sum4 = 1.0;
}
}

```

```

area += rtb_Sum4;
rtb_antecedentOutputs[2] = rtb_Sum4;

/* MATLAB Function: '<S3>/Evaluate Rule Consequents' incorporates:
 * Constant: '<S3>/Output Sample Points'
 */
inputMFCache[0] = 65.0;
inputMFCache[1] = 67.0;
inputMFCache[2] = 69.0;
trimf_i(rConstP.OutputSamplePoints_Value, inputMFCache, tmp);
trimf_j(rConstP.OutputSamplePoints_Value, c_0, tmp_0);
for (i = 0; i < 101; i++) {
    rtb_aggregatedOutputs[i] = 0.0;
    outputMFCache_tmp = i << 1;
    outputMFCache[outputMFCache_tmp] = tmp[i];
    outputMFCache[outputMFCache_tmp + 1] = tmp_0[i];
}

for (i = 0; i < 3; i++) {
    for (outputMFCache_tmp = 0; outputMFCache_tmp < 101; outputMFCache_tmp++) {
        rtb_Sum4 = rtb_aggregatedOutputs[outputMFCache_tmp];
        x_idx_0 = outputMFCache[(outputMFCache_tmp << 1) + b[i] - 1];
        if (x_idx_0 > rtb_antecedentOutputs[i]) {
            x_idx_0 = rtb_antecedentOutputs[i];
        } else {
            if (rtIsNaN(x_idx_0)) {
                x_idx_0 = rtb_antecedentOutputs[i];
            }
        }

        if (rtb_aggregatedOutputs[outputMFCache_tmp] < x_idx_0) {
            rtb_Sum4 = x_idx_0;
        }

        rtb_aggregatedOutputs[outputMFCache_tmp] = rtb_Sum4;
    }
}

/* End of MATLAB Function: '<S3>/Evaluate Rule Consequents' */

/* MATLAB Function: '<S3>/Defuzzify Outputs' incorporates:
 * Constant: '<S3>/Output Sample Points'
 * MATLAB Function: '<S3>/Evaluate Rule Antecedents'
 */
if (area == 0.0) {
    rtb_Sum4 = 67.5;
} else {
    rtb_Sum4 = 0.0;
    area = 0.0;
    for (i = 0; i < 101; i++) {
        area += rtb_aggregatedOutputs[i];
    }

    if (area == 0.0) {
        rtb_Sum4 = 67.5;
    } else {
        for (i = 0; i < 101; i++) {
            rtb_Sum4 += rConstP.OutputSamplePoints_Value[i] *
                rtb_aggregatedOutputs[i];
        }

        rtb_Sum4 *= 1.0 / area;
    }
}

/* Gain: '<Root>/Gain1' incorporates:
 * MATLAB Function: '<S3>/Defuzzify Outputs'
 */
rtB.Gain1 = -316.6483 * rtb_Sum4;

/* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller' */

/* RateTransition generated from: '<Root>/Sum1' */
if (rtmIsMajorTimeStep(rtM)) {
    /* Gain: '<S5>/Slider Gain' */
    rtB.SliderGain = 0.01 * rtB.Step1;

    /* Step: '<Root>/Step' */
    rtB.Step = !(((rtM->Timing.clockTick1) * 1.0) < 100.0);
}

```

```

if (rtM->Timing.RateInteraction.TID1_2 == 1) {
    rtB.TmpRTBAtSum1Inport3 = rtDW.TmpRTBAtSum1Inport3_Buffer0;
}
}

/* End of RateTransition generated from: '<Root>/Sum1' */

/* Integrator: '<Root>/Integrator' */
rtb_Sum1 = rtX.Integrator_CSTATE;

/* Sum: '<Root>/Sum1' incorporates:
 * Gain: '<Root>/Cr'
 * Gain: '<S6>/Slider Gain'
 * Sum: '<Root>/Sum'
 */
rtb_Sum1 = ((rtB.Step - -0.09 * rtB.Sum2) * -5.6 - rtb_Sum1) +
    rtB.TmpRTBAtSum1Inport3;

/* Gain: '<S46>/Filter Coefficient' incorporates:
 * Gain: '<S37>/Derivative Gain'
 * Integrator: '<S38>/Filter'
 * Sum: '<S38>/SumD'
 */
rtB.FilterCoefficient = (-680.5567182967 * rtb_Sum1 - rtX.Filter_CSTATE) *
    0.890872358532341;

/* Gain: '<Root>/Gain' incorporates:
 * Gain: '<S48>/Proportional Gain'
 * Integrator: '<S43>/Integrator'
 * Sum: '<S52>/Sum'
 */
rtB.Gain = ((10200.5292317928 * rtb_Sum1 + rtX.Integrator_CSTATE_j) +
    rtB.FilterCoefficient) * 0.0549;

/* Gain: '<S40>/Integral Gain' */
rtB.IntegralGain = 0.531911734483 * rtb_Sum1;
if (rtmIsMajorTimeStep(rtM)) {
    /* Gain: '<S4>/Slider Gain' */
    rtB.SliderGain_h = 0.01 * rtB.Step;
}

/* TransferFcn: '<Root>/Transfer Fcn1' */
rtB.TransferFcn1 = 0.0;
rtB.TransferFcn1 += 0.20922174840085289 * rtX.TransferFcn1_CSTATE;

/* TransferFcn: '<Root>/Transfer Fcn2' */
rtb_TransferFcn2 = 0.0;
rtb_TransferFcn2 += 0.0020044900577293133 * rtX.TransferFcn2_CSTATE;

/* TransferFcn: '<Root>/Transfer Fcn3' */
rtB.TransferFcn3 = 0.0;
rtB.TransferFcn3 += 0.13634474158237495 * rtX.TransferFcn3_CSTATE;

/* TransferFcn: '<Root>/Transfer Fcn4' */
rtb_TransferFcn4 = 0.0;
rtb_TransferFcn4 += 0.005913122583579713 * rtX.TransferFcn4_CSTATE;
if (rtmIsMajorTimeStep(rtM)) {
    /* Update for TransportDelay: '<Root>/Transport Delay1' */
    {
        real_T **uBuffer = (real_T**) &rtDW.TransportDelay1_PWORK.TUbufferPtrs[0];
        real_T **tBuffer = (real_T**) &rtDW.TransportDelay1_PWORK.TUbufferPtrs[1];
        real_T simTime = rtM->Timing.t[0];
        rtDW.TransportDelay1_IWORK.Head = ((rtDW.TransportDelay1_IWORK.Head <
            (rtDW.TransportDelay1_IWORK.CircularBufSize-1)) ?
            (rtDW.TransportDelay1_IWORK.Head+1) : 0);
        if (rtDW.TransportDelay1_IWORK.Head == rtDW.TransportDelay1_IWORK.Tail) {
            rtDW.TransportDelay1_IWORK.Tail = ((rtDW.TransportDelay1_IWORK.Tail <
                (rtDW.TransportDelay1_IWORK.CircularBufSize-1)) ?
                (rtDW.TransportDelay1_IWORK.Tail+1) : 0);
        }

        (*tBuffer)[rtDW.TransportDelay1_IWORK.Head] = simTime;
        (*uBuffer)[rtDW.TransportDelay1_IWORK.Head] = rtb_TransferFcn4;
    }

    /* Update for TransportDelay: '<Root>/Transport Delay2' */
    {
        real_T **uBuffer = (real_T**) &rtDW.TransportDelay2_PWORK.TUbufferPtrs[0];
        real_T **tBuffer = (real_T**) &rtDW.TransportDelay2_PWORK.TUbufferPtrs[1];
    }
}

```

```

real_T simTime = rtM->Timing.t[0];
rtDW.TransportDelay2_IWORK.Head = ((rtDW.TransportDelay2_IWORK.Head <
    (rtDW.TransportDelay2_IWORK.CircularBufSize-1)) ?
    (rtDW.TransportDelay2_IWORK.Head+1) : 0);
if (rtDW.TransportDelay2_IWORK.Head == rtDW.TransportDelay2_IWORK.Tail) {
    rtDW.TransportDelay2_IWORK.Tail = ((rtDW.TransportDelay2_IWORK.Tail <
        (rtDW.TransportDelay2_IWORK.CircularBufSize-1)) ?
        (rtDW.TransportDelay2_IWORK.Tail+1) : 0);
}

(*tBuffer)[rtDW.TransportDelay2_IWORK.Head] = simTime;
(*uBuffer)[rtDW.TransportDelay2_IWORK.Head] = rtb_TransferFcn2;
}

/* External mode */
rtExtModeUploadCheckTrigger(3);
rtExtModeUpload(1, (real_T)rtM->Timing.t[0]);
} /* end MajorTimeStep */

if (rtmIsMajorTimeStep(rtM)) {
    /* signal main to stop simulation */
    { /* Sample time: [0.0s, 0.0s] */
        if ((rtmGetTFinal(rtM)!=-1) &&
            !((rtmGetTFinal(rtM)-((rtM->Timing.clockTick1) * 1.0)) >
                ((rtM->Timing.clockTick1) * 1.0) * (DBL_EPSILON))) {
            rtmSetErrorStatus(rtM, "Simulation finished");
        }

        if (rtmGetStopRequested(rtM)) {
            rtmSetErrorStatus(rtM, "Simulation finished");
        }
    }

    rt_ertODEUpdateContinuousStates(&rtM->solverInfo);

    /* Update absolute time */
    /* The "clockTick0" counts the number of times the code of this task has
    * been executed. The absolute time is the multiplication of "clockTick0"
    * and "Timing.stepSize0". Size of "clockTick0" ensures timer will not
    * overflow during the application lifespan selected.
    */
    ++rtM->Timing.clockTick0;
    rtM->Timing.t[0] = rtsiGetSolverStopTime(&rtM->solverInfo);

    /* Update absolute time */
    /* The "clockTick1" counts the number of times the code of this task has
    * been executed. The resolution of this integer timer is 1.0, which is the step size
    * of the task. Size of "clockTick1" ensures timer will not overflow during the
    * application lifespan selected.
    */
    rtM->Timing.clockTick1++;
} /* end MajorTimeStep */
}

/* Derivatives for root system: '<Root>' */
void Tech_praktika_derivatives(void)
{
    XDdot * _rtXDdot;
    _rtXDdot = ((XDdot *) rtM->derivs);

    /* Derivatives for Integrator: '<Root>/Integrator1' */
    _rtXDdot->Integrator1_CSTATE = rtB.SliderGain;

    /* Derivatives for Integrator: '<Root>/Integrator' */
    _rtXDdot->Integrator_CSTATE = rtB.SliderGain_h;

    /* Derivatives for Integrator: '<S43>/Integrator' */
    _rtXDdot->Integrator_CSTATE_j = rtB.IntegralGain;

    /* Derivatives for Integrator: '<S38>/Filter' */
    _rtXDdot->Filter_CSTATE = rtB.FilterCoefficient;

    /* Derivatives for TransferFcn: '<Root>/Transfer Fcn1' */
    _rtXDdot->TransferFcn1_CSTATE = 0.0;
    _rtXDdot->TransferFcn1_CSTATE += -0.53304904051172708 * rtX.TransferFcn1_CSTATE;
    _rtXDdot->TransferFcn1_CSTATE += rtB.Gain1;

    /* Derivatives for TransferFcn: '<Root>/Transfer Fcn2' */
    _rtXDdot->TransferFcn2_CSTATE = 0.0;

```

```

_rtXdot->TransferFcn2_CSTATE += -0.80179602309172537 * rtX.TransferFcn2_CSTATE;
_rtXdot->TransferFcn2_CSTATE += rtB.TransferFcn1;

/* Derivatives for TransferFcn: '<Root>/Transfer Fcn3' */
_rtXdot->TransferFcn3_CSTATE = 0.0;
_rtXdot->TransferFcn3_CSTATE += -0.3464043231259526 * rtX.TransferFcn3_CSTATE;
_rtXdot->TransferFcn3_CSTATE += rtB.Gain;

/* Derivatives for TransferFcn: '<Root>/Transfer Fcn4' */
_rtXdot->TransferFcn4_CSTATE = 0.0;
_rtXdot->TransferFcn4_CSTATE += -2.2742779167614282 * rtX.TransferFcn4_CSTATE;
_rtXdot->TransferFcn4_CSTATE += rtB.TransferFcn3;
}

/* Model step function for TID2 */
void Tech_praktika_step2(void) /* Sample time: [100.0s, 0.0s] */
{
    real_T rtb_RandomNumber;

    /* RandomNumber: '<Root>/Random Number' */
    rtb_RandomNumber = rtDW.NextOutput;

    /* RateTransition generated from: '<Root>/Sum1' */
    rtDW.TmpRTBAAtSum1Inport3_Buffer0 = rtb_RandomNumber;

    /* RandomNumber: '<Root>/Random Number1' */
    rtb_RandomNumber = rtDW.NextOutput_e;

    /* RateTransition generated from: '<Root>/Sum4' */
    rtDW.TmpRTBAAtSum4Inport3_Buffer0 = rtb_RandomNumber;

    /* Update for RandomNumber: '<Root>/Random Number' */
    rtDW.NextOutput = rt_nrand_Upu32_Yd_f_pw_snf(&rtDW.RandSeed) * 0.0;

    /* Update for RandomNumber: '<Root>/Random Number1' */
    rtDW.NextOutput_e = rt_nrand_Upu32_Yd_f_pw_snf(&rtDW.RandSeed_a) * 0.0;
    rtExtModeUpload(2, (real_T)((rtM->Timing.clockTick2) * 100.0));

    /* Update absolute time */
    /* The "clockTick2" counts the number of times the code of this task has
    * been executed. The resolution of this integer timer is 100.0, which is the step size
    * of the task. Size of "clockTick2" ensures timer will not overflow during the
    * application lifespan selected.
    */
    rtM->Timing.clockTick2++;
}

/* Model initialize function */
void Tech_praktika_initialize(void)
{
    /* Registration code */

    /* initialize non-finites */
    rt_InitInfAndNaN(sizeof(real_T));

    {
        /* Setup solver object */
        rtsiSetSimTimeStepPtr(&rtM->solverInfo, &rtM->Timing.simTimeStep);
        rtsiSetTPtr(&rtM->solverInfo, &rtmGetTPtr(rtM));
        rtsiSetStepSizePtr(&rtM->solverInfo, &rtM->Timing.stepSize0);
        rtsiSetdXPtr(&rtM->solverInfo, &rtM->derivs);
        rtsiSetContStatesPtr(&rtM->solverInfo, (real_T **) &rtM->contStates);
        rtsiSetNumContStatesPtr(&rtM->solverInfo, &rtM->Sizes.numContStates);
        rtsiSetNumPeriodicContStatesPtr(&rtM->solverInfo,
            &rtM->Sizes.numPeriodicContStates);
        rtsiSetPeriodicContStateIndicesPtr(&rtM->solverInfo,
            &rtM->periodicContStateIndices);
        rtsiSetPeriodicContStateRangesPtr(&rtM->solverInfo,
            &rtM->periodicContStateRanges);
        rtsiSetErrorStatusPtr(&rtM->solverInfo, (&rtmGetErrorStatus(rtM)));
        rtsiSetRTModelPtr(&rtM->solverInfo, rtM);
    }

    rtsiSetSimTimeStep(&rtM->solverInfo, MAJOR_TIME_STEP);
    rtM->intgData.y = rtM->odeY;
    rtM->intgData.f[0] = rtM->odeF[0];
    rtM->intgData.f[1] = rtM->odeF[1];
    rtM->intgData.f[2] = rtM->odeF[2];
    rtM->contStates = ((X *) &rtX);
}

```

```

rtsiSetSolverData(&rtM->solverInfo, (void *)&rtM->intgData);
rtsiSetSolverName(&rtM->solverInfo,"ode3");
rtmSetTPtr(rtM, &rtM->Timing.tArray[0]);
rtmSetTFinal(rtM, 100.0);
rtM->Timing.stepSize0 = 1.0;

/* External mode info */
rtM->Sizes.checksums[0] = (2930134216U);
rtM->Sizes.checksums[1] = (1792573674U);
rtM->Sizes.checksums[2] = (1901650391U);
rtM->Sizes.checksums[3] = (3723107567U);

{
    static const sysRanDType rtAlwaysEnabled = SUBSYS_RAN_BC_ENABLE;
    static RTWExtModelInfo rt_ExtModelInfo;
    static const sysRanDType *systemRan[5];
    rtM->extModelInfo = (&rt_ExtModelInfo);
    rteiSetSubSystemActiveVectorAddresses(&rt_ExtModelInfo, systemRan);
    systemRan[0] = &rtAlwaysEnabled;
    systemRan[1] = &rtAlwaysEnabled;
    systemRan[2] = &rtAlwaysEnabled;
    systemRan[3] = &rtAlwaysEnabled;
    systemRan[4] = &rtAlwaysEnabled;
    rteiSetModelMappingInfoPtr(rtM->extModelInfo, &rtM->SpecialInfo.mappingInfo);
    rteiSetChecksumsPtr(rtM->extModelInfo, rtM->Sizes.checksums);
    rteiSetTPtr(rtM->extModelInfo, rtmGetTPtr(rtM));
}

/* data type transition information */
{
    static DataTypeTransInfo dtInfo;
    (void) memset((char_T *) &dtInfo, 0,
        sizeof(dtInfo));
    rtM->SpecialInfo.mappingInfo = (&dtInfo);
    dtInfo.numDataTypes = 19;
    dtInfo.dataTypeSizes = &rtDataTypeSizes[0];
    dtInfo.dataTypeNames = &rtDataTypeNames[0];

    /* Block I/O transition table */
    dtInfo.BTransTable = &rtBTransTable;
}

/* Start for TransportDelay: '<Root>/Transport Delay1' */
{
    real_T *pBuffer = &rtDW.TransportDelay1_RWORK.TUbufferArea[0];
    rtDW.TransportDelay1_IWORK.Tail = 0;
    rtDW.TransportDelay1_IWORK.Head = 0;
    rtDW.TransportDelay1_IWORK.Last = 0;
    rtDW.TransportDelay1_IWORK.CircularBufSize = 1024;
    pBuffer[0] = 0.0;
    pBuffer[1024] = rtM->Timing.t[0];
    rtDW.TransportDelay1_PWORK.TUbufferPtrs[0] = (void *) &pBuffer[0];
    rtDW.TransportDelay1_PWORK.TUbufferPtrs[1] = (void *) &pBuffer[1024];
}

/* Start for TransportDelay: '<Root>/Transport Delay2' */
{
    real_T *pBuffer = &rtDW.TransportDelay2_RWORK.TUbufferArea[0];
    rtDW.TransportDelay2_IWORK.Tail = 0;
    rtDW.TransportDelay2_IWORK.Head = 0;
    rtDW.TransportDelay2_IWORK.Last = 0;
    rtDW.TransportDelay2_IWORK.CircularBufSize = 1024;
    pBuffer[0] = 0.0;
    pBuffer[1024] = rtM->Timing.t[0];
    rtDW.TransportDelay2_PWORK.TUbufferPtrs[0] = (void *) &pBuffer[0];
    rtDW.TransportDelay2_PWORK.TUbufferPtrs[1] = (void *) &pBuffer[1024];
}

/* InitializeConditions for Integrator: '<Root>/Integrator1' */
rtX.Integrator1_CSTATE = 0.0;

/* InitializeConditions for Integrator: '<Root>/Integrator' */
rtX.Integrator_CSTATE = 0.0;

/* InitializeConditions for Integrator: '<S43>/Integrator' */
rtX.Integrator_CSTATE_j = 0.0;

/* InitializeConditions for Integrator: '<S38>/Filter' */
rtX.Filter_CSTATE = 0.0;

```

```

/* InitializeConditions for TransferFcn: '<Root>/Transfer Fcn1' */
rtX.TransferFcn1_CSTATE = 0.0;

/* InitializeConditions for TransferFcn: '<Root>/Transfer Fcn2' */
rtX.TransferFcn2_CSTATE = 0.0;

/* InitializeConditions for TransferFcn: '<Root>/Transfer Fcn3' */
rtX.TransferFcn3_CSTATE = 0.0;

/* InitializeConditions for TransferFcn: '<Root>/Transfer Fcn4' */
rtX.TransferFcn4_CSTATE = 0.0;

/* InitializeConditions for RandomNumber: '<Root>/Random Number' */
rtDW.RandSeed = 1144108930U;
rtDW.NextOutput = rt_nrand_Upu32_Yd_f_pw_snf(&rtDW.RandSeed) * 0.0;

/* InitializeConditions for RandomNumber: '<Root>/Random Number1' */
rtDW.RandSeed_a = 1144108930U;
rtDW.NextOutput_e = rt_nrand_Upu32_Yd_f_pw_snf(&rtDW.RandSeed_a) * 0.0;
}

/*
 * File trailer for generated code.
 *
 * [EOF]
 */

```

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг программного кода приложения в системе AppDesigner

```
classdef MainApp_exported < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    MainAppUI          matlab.ui.Figure
    Menu                matlab.ui.container.Menu
    FilesMenu           matlab.ui.container.Menu
    UploadcoefficientsMenu  matlab.ui.container.Menu
    SavecoefficientsMenu  matlab.ui.container.Menu
    GridLayout          matlab.ui.container.GridLayout
    PIDcontrollerCoefficientsPanel  matlab.ui.container.Panel
    GridLayout2         matlab.ui.container.GridLayout
    KEditFieldLabel     matlab.ui.control.Label
    KEditField          matlab.ui.control.EditField
    Tp2EditFieldLabel   matlab.ui.control.Label
    Tp2EditField        matlab.ui.control.EditField
    KpEditFieldLabel    matlab.ui.control.Label
    KpEditField         matlab.ui.control.EditField
    Kp2EditFieldLabel  matlab.ui.control.Label
    Kp2EditField        matlab.ui.control.EditField
    Tp1EditFieldLabel   matlab.ui.control.Label
    Tp1EditField        matlab.ui.control.EditField
    FuzzycontrollerCoefficientsPanel  matlab.ui.container.Panel
    GridLayout2_2       matlab.ui.container.GridLayout
    KfEditFieldLabel    matlab.ui.control.Label
    KfEditField         matlab.ui.control.EditField
    Tp1fEditFieldLabel  matlab.ui.control.Label
    Tp1fEditField       matlab.ui.control.EditField
    Tp2fEditFieldLabel  matlab.ui.control.Label
    Tp2fEditField       matlab.ui.control.EditField
    KpfEditFieldLabel   matlab.ui.control.Label
    KpfEditField        matlab.ui.control.EditField
    Kp2fEditFieldLabel  matlab.ui.control.Label
    Kp2fEditField       matlab.ui.control.EditField
    PIDUIAxes           matlab.ui.control.UIAxes
    FuzzyUIAxes         matlab.ui.control.UIAxes
    StartButton         matlab.ui.control.Button
end

methods (Access = public)

function loadFromFile(app)
    % Load coefficients from files and add them to EditFields
    run('Kcoef.m'); % PID-controller
    run('Kcoef_fuzzy.m'); % Fuzzy-controller
    evalin('base', 'Kcoef');
    evalin('base', 'Kcoef_fuzzy');
    % Insert variables for Fuzzy-Controller in EditFields
    app.KEditField.Value = num2str(k);
    app.Tp2EditField.Value = num2str(Tp2);
    app.Tp1EditField.Value = num2str(Tp1);
    app.KpEditField.Value = num2str(kp);
    app.Kp2EditField.Value = num2str(kp2);
    % Insert variables for PID-Controller in EditFields
```

```

app.KfEditField.Value = num2str(kf);
app.Tp2fEditField.Value = num2str(Tp2f);
app.Tp1fEditField.Value = num2str(Tp1f);
app.KpfEditField.Value = num2str(kpf);
app.Kp2fEditField.Value = num2str(kp2f);

ylabel(app.PIDUIAxes, 'Temperature C°');
ylabel(app.FuzzyUIAxes, 'Temperature C°');
xlabel(app.PIDUIAxes, 'Time, sec');
xlabel(app.FuzzyUIAxes, 'Time, sec');
end
end

methods (Access = private)

function toggleUIC(app, state, label)
    % toggle the state of the uicontrols in the app
    uics = findall(app.MainAppUI);
    for idx = 1:length(uics)
        uic = uics(idx);
        if isprop(uic,'Enable')
            uic.Enable = state;
        end
    end
    % update the simulate button text
    app.StartButton.Text = label;
    drawnow; % flush the ui events queue
end % toggleUIC
end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
    app.KEditField.Value = num2str(0.0549);
    app.Tp2EditField.Value = num2str(0.4397);
    app.Tp1EditField.Value = num2str(2.8868);
    app.KpEditField.Value = num2str(0.3936);
    app.Kp2EditField.Value = num2str(0.0026);
    % Insert variables for PID-Controller in EditFields
    app.KfEditField.Value = num2str(-316.6483);
    app.Tp2fEditField.Value = num2str(1.2472);
    app.Tp1fEditField.Value = num2str(1.8760);
    app.KpfEditField.Value = num2str(0.3925);
    app.Kp2fEditField.Value = num2str(0.0025);
end

% Callback function
function LoadfromfilesButtonPushed(app, event)
    app.loadFromFile;
end

% Button pushed function: StartButton
function StartButtonPushed(app, event)
    % Assign variables for simulink model
    app.toggleUIC('off', 'Simulating...');
    %      rmdir('slprj', 's');
    %      delete Tech_praktika.slxc
    mdl = 'Tech_praktika';
    simout = Simulink.SimulationInput(mdl);
    simout = simout.setVariable('k', app.KEditField.Value);

```

```

simout = simout.setVariable('Tp1', app.Tp1EditField.Value);
simout = simout.setVariable('Tp2', app.Tp2EditField.Value);
simout = simout.setVariable('kp', app.KpEditField.Value);
simout = simout.setVariable('kp2', app.Kp2EditField.Value);

simout = simout.setVariable('kp2f', app.Kp2fEditField.Value);
simout = simout.setVariable('kf', app.KfEditField.Value);
simout = simout.setVariable('Tp1f', app.Tp1fEditField.Value);
simout = simout.setVariable('Tp2f', app.Tp2fEditField.Value);
simout = simout.setVariable('kpf', app.KpfEditField.Value);
simout = simulink.compiler.configureForDeployment(simout);

w = warning('off','all');
simOut = sim(simout);
warning(w);
s1 = stepinfo(simOut.y.signals(1).values, simOut.y.time);
s2 = stepinfo(simOut.y.signals(2).values, simOut.y.time);
plot(app.PIDUIAxes, simOut.y.time, simOut.y.signals(1).values);
plot(app.FuzzyUIAxes, simOut.y.time, simOut.y.signals(2).values);

if s1.SettlingTime > 15 || s2.SettlingTime > 15 || s1.SettlingMin < 65 || s2.SettlingMin < 65 ||
s1.SettlingMax > 75 || s2.SettlingMax > 75
    uialert(app.MainAppUI,'яяяяяяяяяя яяяяяя яя яяяяяяяяяяяя яяяяяяяяяя. яяяяяяяяяя
яяяяяяяяяя яяяяяяяяяяяя','яяяяяяяяяяяя яяяяяяяяяя');
end
app.toggleUIC('on', 'Start');
end

% Menu selected function: UploadcoefficientsMenu
function UploadcoefficientsMenuSelected(app, event)
[baseFileName, folder] = uigetfile('*.mat');
fullFileName = fullfile(folder, baseFileName);
if exist(fullFileName, 'file')
    % Normal situation - they picked an existing file.
    storedStructure = load(fullFileName);
    % Now do something with storedStructure, like extract fields into new variables or whatever you
    want.

    if sum(isfield(storedStructure, {'Tp1', 'Tp1f', 'Tp2', 'Tp2f', 'k', 'kf', 'kp', 'kp2', 'kp2f', 'kpf'})) == 10
        app.KEditField.Value = num2str(storedStructure.k);
        app.Tp2EditField.Value = num2str(storedStructure.Tp2);
        app.Tp1EditField.Value = num2str(storedStructure.Tp1);
        app.KpEditField.Value = num2str(storedStructure.kp);
        app.Kp2EditField.Value = num2str(storedStructure.kp2);
        % Insert variables for PID-Controller in EditFields
        app.KfEditField.Value = num2str(storedStructure.kf);
        app.Tp2fEditField.Value = num2str(storedStructure.Tp2f);
        app.Tp1fEditField.Value = num2str(storedStructure.Tp1f);
        app.KpfEditField.Value = num2str(storedStructure.kpf);
        app.Kp2fEditField.Value = num2str(storedStructure.kp2f);
    end
else
    % Error: Would only get here if they typed in a name of a non-existent file
    % instead of picking one from the folder.
    warningMessage = sprintf('Warning: mat file does not exist:\n%s', fullFileName);
    uiwait(errordlg(warningMessage));
    return;
end
end

% Menu selected function: SavecoefficientsMenu
function SavecoefficientsMenuSelected(app, event)
k = str2double(app.KEditField.Value);
kp = str2double(app.KpEditField.Value);

```

```

        kp2 = str2double(app.Kp2EditField.Value);
        Tp1 = str2double(app.Tp1EditField.Value);
        Tp2 = str2double(app.Tp2EditField.Value);
        kf = str2double(app.KfEditField.Value);
        kpf = str2double(app.KpfEditField.Value);
        kp2f = str2double(app.Kp2fEditField.Value);
        Tp1f = str2double(app.Tp1fEditField.Value);
        Tp2f = str2double(app.Tp2fEditField.Value);

        save('coefficients.mat', 'k', 'kp', 'kp2', 'Tp1', 'Tp2', 'kf', 'kpf', 'kp2f', 'Tp1f', 'Tp2f')
    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create MainAppUI and hide until all components are created
    app.MainAppUI = uifigure('Visible', 'off');
    app.MainAppUI.Position = [100 100 640 480];
    app.MainAppUI.Name = 'MATLAB App';

    % Create Menu
    app.Menu = uimenu(app.MainAppUI);
    app.Menu.Text = 'Menu';

    % Create FilesMenu
    app.FilesMenu = uimenu(app.Menu);
    app.FilesMenu.Text = 'Files';

    % Create UploadcoefficientsMenu
    app.UploadcoefficientsMenu = uimenu(app.FilesMenu);
    app.UploadcoefficientsMenu.MenuSelectedFcn = createCallbackFcn(app,
@UploadcoefficientsMenuSelected, true);
    app.UploadcoefficientsMenu.Text = 'Upload coefficients';

    % Create SavecoefficientsMenu
    app.SavecoefficientsMenu = uimenu(app.FilesMenu);
    app.SavecoefficientsMenu.MenuSelectedFcn = createCallbackFcn(app,
@SavecoefficientsMenuSelected, true);
    app.SavecoefficientsMenu.Text = 'Save coefficients';

    % Create GridLayout
    app.GridLayout = uigridlayout(app.MainAppUI);
    app.GridLayout.ColumnWidth = {'2x', '1x'};
    app.GridLayout.RowHeight = {'3x', '3x', 35};

    % Create PIDcontrollerCoefficientsPanel
    app.PIDcontrollerCoefficientsPanel = uipanel(app.GridLayout);
    app.PIDcontrollerCoefficientsPanel.Title = 'PID-controller Coefficients';
    app.PIDcontrollerCoefficientsPanel.Layout.Row = 1;
    app.PIDcontrollerCoefficientsPanel.Layout.Column = 2;

    % Create GridLayout2
    app.GridLayout2 = uigridlayout(app.PIDcontrollerCoefficientsPanel);
    app.GridLayout2.ColumnWidth = {'2x', '1x'};
    app.GridLayout2.RowHeight = {'1x', '1x', '1x', '1x', '1x'};

    % Create KEditFieldLabel
    app.KEditFieldLabel = uilabel(app.GridLayout2);
    app.KEditFieldLabel.HorizontalAlignment = 'center';

```

```

app.KEditFieldLabel.Layout.Row = 1;
app.KEditFieldLabel.Layout.Column = 1;
app.KEditFieldLabel.Text = 'K';

% Create KEditField
app.KEditField = uieditfield(app.GridLayout2, 'text');
app.KEditField.HorizontalAlignment = 'center';
app.KEditField.Layout.Row = 1;
app.KEditField.Layout.Column = 2;
app.KEditField.Value = '0';

% Create Tp2EditFieldLabel
app.Tp2EditFieldLabel = uilabel(app.GridLayout2);
app.Tp2EditFieldLabel.HorizontalAlignment = 'center';
app.Tp2EditFieldLabel.Layout.Row = 3;
app.Tp2EditFieldLabel.Layout.Column = 1;
app.Tp2EditFieldLabel.Text = 'Tp2';

% Create Tp2EditField
app.Tp2EditField = uieditfield(app.GridLayout2, 'text');
app.Tp2EditField.HorizontalAlignment = 'center';
app.Tp2EditField.Layout.Row = 3;
app.Tp2EditField.Layout.Column = 2;
app.Tp2EditField.Value = '0';

% Create KpEditFieldLabel
app.KpEditFieldLabel = uilabel(app.GridLayout2);
app.KpEditFieldLabel.HorizontalAlignment = 'center';
app.KpEditFieldLabel.Layout.Row = 4;
app.KpEditFieldLabel.Layout.Column = 1;
app.KpEditFieldLabel.Text = 'Kp';

% Create KpEditField
app.KpEditField = uieditfield(app.GridLayout2, 'text');
app.KpEditField.HorizontalAlignment = 'center';
app.KpEditField.Layout.Row = 4;
app.KpEditField.Layout.Column = 2;
app.KpEditField.Value = '0';

% Create Kp2EditFieldLabel
app.Kp2EditFieldLabel = uilabel(app.GridLayout2);
app.Kp2EditFieldLabel.HorizontalAlignment = 'center';
app.Kp2EditFieldLabel.Layout.Row = 5;
app.Kp2EditFieldLabel.Layout.Column = 1;
app.Kp2EditFieldLabel.Text = 'Kp2';

% Create Kp2EditField
app.Kp2EditField = uieditfield(app.GridLayout2, 'text');
app.Kp2EditField.HorizontalAlignment = 'center';
app.Kp2EditField.Layout.Row = 5;
app.Kp2EditField.Layout.Column = 2;
app.Kp2EditField.Value = '0';

% Create Tp1EditFieldLabel
app.Tp1EditFieldLabel = uilabel(app.GridLayout2);
app.Tp1EditFieldLabel.HorizontalAlignment = 'center';
app.Tp1EditFieldLabel.Layout.Row = 2;
app.Tp1EditFieldLabel.Layout.Column = 1;
app.Tp1EditFieldLabel.Text = 'Tp1';

% Create Tp1EditField
app.Tp1EditField = uieditfield(app.GridLayout2, 'text');
app.Tp1EditField.HorizontalAlignment = 'center';

```

```

app.Tp1EditField.Layout.Row = 2;
app.Tp1EditField.Layout.Column = 2;
app.Tp1EditField.Value = '0';

% Create FuzzycontrollerCoefficientsPanel
app.FuzzycontrollerCoefficientsPanel = uipanel(app.GridLayout);
app.FuzzycontrollerCoefficientsPanel.Title = 'Fuzzy-controller Coefficients';
app.FuzzycontrollerCoefficientsPanel.Layout.Row = 2;
app.FuzzycontrollerCoefficientsPanel.Layout.Column = 2;

% Create GridLayout2_2
app.GridLayout2_2 = uigridlayout(app.FuzzycontrollerCoefficientsPanel);
app.GridLayout2_2.ColumnWidth = {'2x', '1x'};
app.GridLayout2_2.RowHeight = {'1x', '1x', '1x', '1x', '1x'};

% Create KfEditFieldLabel
app.KfEditFieldLabel = uilabel(app.GridLayout2_2);
app.KfEditFieldLabel.HorizontalAlignment = 'center';
app.KfEditFieldLabel.Layout.Row = 1;
app.KfEditFieldLabel.Layout.Column = 1;
app.KfEditFieldLabel.Text = 'Kf';

% Create KfEditField
app.KfEditField = uieditfield(app.GridLayout2_2, 'text');
app.KfEditField.HorizontalAlignment = 'center';
app.KfEditField.Layout.Row = 1;
app.KfEditField.Layout.Column = 2;
app.KfEditField.Value = '0';

% Create Tp1fEditFieldLabel
app.Tp1fEditFieldLabel = uilabel(app.GridLayout2_2);
app.Tp1fEditFieldLabel.HorizontalAlignment = 'center';
app.Tp1fEditFieldLabel.Layout.Row = 2;
app.Tp1fEditFieldLabel.Layout.Column = 1;
app.Tp1fEditFieldLabel.Text = 'Tp1f';

% Create Tp1fEditField
app.Tp1fEditField = uieditfield(app.GridLayout2_2, 'text');
app.Tp1fEditField.HorizontalAlignment = 'center';
app.Tp1fEditField.Layout.Row = 2;
app.Tp1fEditField.Layout.Column = 2;
app.Tp1fEditField.Value = '0';

% Create Tp2fEditFieldLabel
app.Tp2fEditFieldLabel = uilabel(app.GridLayout2_2);
app.Tp2fEditFieldLabel.HorizontalAlignment = 'center';
app.Tp2fEditFieldLabel.Layout.Row = 3;
app.Tp2fEditFieldLabel.Layout.Column = 1;
app.Tp2fEditFieldLabel.Text = 'Tp2f';

% Create Tp2fEditField
app.Tp2fEditField = uieditfield(app.GridLayout2_2, 'text');
app.Tp2fEditField.HorizontalAlignment = 'center';
app.Tp2fEditField.Layout.Row = 3;
app.Tp2fEditField.Layout.Column = 2;
app.Tp2fEditField.Value = '0';

% Create KpfEditFieldLabel
app.KpfEditFieldLabel = uilabel(app.GridLayout2_2);
app.KpfEditFieldLabel.HorizontalAlignment = 'center';
app.KpfEditFieldLabel.Layout.Row = 4;
app.KpfEditFieldLabel.Layout.Column = 1;
app.KpfEditFieldLabel.Text = 'Kpf';

```

```

% Create KpfEditField
app.KpfEditField = uicontrol(app.GridLayout2_2, 'text');
app.KpfEditField.HorizontalAlignment = 'center';
app.KpfEditField.Layout.Row = 4;
app.KpfEditField.Layout.Column = 2;
app.KpfEditField.Value = '0';

% Create Kp2fEditFieldLabel
app.Kp2fEditFieldLabel = uicontrol(app.GridLayout2_2);
app.Kp2fEditFieldLabel.HorizontalAlignment = 'center';
app.Kp2fEditFieldLabel.Layout.Row = 5;
app.Kp2fEditFieldLabel.Layout.Column = 1;
app.Kp2fEditFieldLabel.Text = 'Kp2f';

% Create Kp2fEditField
app.Kp2fEditField = uicontrol(app.GridLayout2_2, 'text');
app.Kp2fEditField.HorizontalAlignment = 'center';
app.Kp2fEditField.Layout.Row = 5;
app.Kp2fEditField.Layout.Column = 2;
app.Kp2fEditField.Value = '0 ' ;

% Create PIDUIAxes
app.PIDUIAxes = uiaxes(app.GridLayout);
title(app.PIDUIAxes, 'PID Controller Step Response')
xlabel(app.PIDUIAxes, 'X')
ylabel(app.PIDUIAxes, 'Y')
app.PIDUIAxes.Layout.Row = 1;
app.PIDUIAxes.Layout.Column = 1;

% Create FuzzyUIAxes
app.FuzzyUIAxes = uiaxes(app.GridLayout);
title(app.FuzzyUIAxes, 'Fuzzy Controller Step Response')
xlabel(app.FuzzyUIAxes, 'X')
ylabel(app.FuzzyUIAxes, 'Y')
app.FuzzyUIAxes.Layout.Row = 2;
app.FuzzyUIAxes.Layout.Column = 1;

% Create StartButton
app.StartButton = uicontrol(app.GridLayout, 'push');
app.StartButton.ButtonPushedFcn = createCallbackFcn(app, @StartButtonPushed, true);
app.StartButton.Layout.Row = 3;
app.StartButton.Layout.Column = 1;
app.StartButton.Text = 'Start';

% Show the figure after all components are created
app.MainAppUI.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = MainApp_exported

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.MainAppUI)

% Execute the startup function

```

```

        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.MainAppUI)
    end
end
end
end

```

ПРИЛОЖЕНИЕ В

(обязательное)

Данные дисперсионного анализа

Таблица В.1 – Температура на выходе АВО для разных значений дисперсий σ^2 входной температуры, $\mu = 90^\circ\text{C}$

$\sigma^2=5$	$\sigma^2=5$	$\sigma^2=4$	$\sigma^2=4$	$\sigma^2=3$	$\sigma^2=3$	$\sigma^2=2$	$\sigma^2=2$	$\sigma^2=1$	$\sigma^2=1$	$\sigma^2=6$	$\sigma^2=6$	$\sigma^2=7$	$\sigma^2=7$	$\sigma^2=8$	$\sigma^2=8$	$\sigma^2=9$	$\sigma^2=9$	$\sigma^2=10$	$\sigma^2=10$
67,11	66,04	67,45	66,42	67,84	66,84	68,30	67,35	68,90	68,01	66,80	65,70	66,52	65,38	66,25	65,09	66,00	64,81	65,77	64,55
66,93	66,01	67,27	66,39	67,66	66,82	68,11	67,32	68,72	68,00	66,62	65,66	66,34	65,34	66,07	65,05	65,83	64,77	65,59	64,51
66,93	66,01	67,27	66,39	67,66	66,82	68,11	67,32	68,72	68,00	66,62	65,66	66,34	65,34	66,07	65,05	65,83	64,77	65,59	64,51
66,93	66,01	67,27	66,39	67,66	66,82	68,11	67,32	68,72	68,00	66,62	65,66	66,34	65,34	66,07	65,05	65,83	64,77	65,59	64,51
68,44	67,68	68,61	67,89	68,79	68,12	69,02	68,40	69,31	68,76	68,29	67,49	68,15	67,32	68,03	67,16	67,90	67,01	67,79	66,87
68,33	67,72	68,49	67,93	68,69	68,15	68,91	68,43	69,21	68,79	68,18	67,53	68,04	67,36	67,91	67,20	67,79	67,05	67,68	66,91
72,59	72,10	72,30	71,84	71,97	71,55	71,57	71,21	71,06	70,76	72,86	72,33	73,10	72,54	73,33	72,73	73,55	72,91	73,75	73,09
72,49	72,28	68,41	68,31	68,58	68,50	68,78	68,73	69,05	69,02	72,76	72,52	73,01	72,73	73,24	72,94	73,45	73,13	73,65	73,31
68,26	68,14	68,35	68,53	68,52	68,70	68,72	68,90	70,19	70,39	68,12	67,99	68,00	67,85	67,88	67,72	67,77	67,60	67,67	67,48
70,91	71,15	70,77	71,00	70,62	70,84	70,43	70,65	69,21	69,44	71,03	71,28	67,94	68,13	67,82	68,01	67,71	67,90	67,61	67,79
70,87	71,14	68,85	69,08	70,58	70,81	70,40	70,63	69,41	69,69	70,99	71,27	71,14	71,40	71,25	71,51	71,35	71,62	71,44	71,71
68,77	69,02	69,28	69,55	68,95	69,18	69,06	69,29	68,18	68,46	68,69	68,95	68,62	68,88	68,55	68,81	68,49	68,75	68,43	68,70
69,24	69,54	66,83	67,10	68,93	69,19	69,36	69,63	68,25	68,55	69,22	69,52	68,60	68,87	68,53	68,80	68,47	68,74	68,41	68,68
66,51	66,79	66,98	67,27	69,31	69,59	67,62	67,90	70,50	70,81	66,23	66,50	69,19	69,47	69,16	69,45	69,14	69,43	69,12	69,41
66,68	66,97	66,98	67,30	67,19	67,47	67,73	68,02	69,47	69,85	66,41	66,70	69,18	69,45	69,15	69,42	69,13	69,39	69,11	69,37
66,68	67,00	71,49	71,81	67,32	67,61	70,91	71,23	69,47	69,92	66,41	66,73	65,96	66,22	65,72	65,97	65,48	65,73	65,27	65,51
71,72	72,04	71,49	71,88	71,22	71,54	69,45	69,87	68,71	69,17	71,93	72,26	66,16	66,44	65,93	66,21	65,71	65,99	65,51	65,78
71,72	72,14	69,43	69,89	69,44	69,88	69,45	69,96	68,71	69,15	71,93	72,37	66,16	66,48	65,93	66,25	65,71	66,03	65,50	65,83
69,41	69,90	67,92	68,53	69,44	69,99	68,38	68,91	68,75	69,15	69,41	69,90	72,12	72,45	72,30	72,64	72,47	72,81	72,63	72,97
67,73	68,38	67,98	68,51	68,13	68,71	68,38	68,85	70,37	70,73	67,56	68,24	72,12	72,59	72,30	72,80	72,47	72,99	72,63	73,12
67,80	68,36	71,23	71,65	68,19	68,67	68,43	68,88	69,45	69,82	67,64	68,22	69,40	69,91	69,39	69,92	69,38	69,93	69,37	69,93
71,44	71,86	69,40	69,83	71,00	71,39	70,73	71,10	70,02	70,42	67,64	68,09	67,40	68,12	69,39	70,10	69,38	70,06	66,99	67,79

Продолжение табл. В.1

69,39	69,83	69,40	69,88	69,41	69,83	69,43	69,82	69,91	70,25	71,62	72,06	67,49	68,09	67,25	68,00	67,12	67,89	66,99	67,68
69,39	69,88	70,54	71,03	69,41	69,87	69,43	69,86	70,27	70,58	71,62	72,06	67,49	67,95	67,25	67,90	67,22	67,85	67,09	67,75
70,66	71,17	70,54	70,95	70,40	70,87	70,24	70,67	69,91	70,18	69,37	69,83	71,79	72,24	67,35	67,97	67,22	67,68	67,09	67,64
70,66	71,07	70,30	70,69	70,40	70,81	70,24	70,64	68,17	68,41	70,77	71,30	71,79	72,24	67,35	67,86	72,10	72,55	72,24	72,71
70,40	70,79	71,02	71,34	70,20	70,57	70,07	70,43	69,89	70,09	70,48	70,91	69,36	69,83	71,95	72,39	72,10	72,56	72,24	72,72
70,40	70,74	70,31	70,56	70,82	71,14	70,58	70,89	70,64	70,83	71,36	71,69	70,87	71,42	71,95	72,41	69,34	69,83	69,33	69,83
71,20	71,52	66,83	67,01	70,20	70,46	70,08	70,34	70,24	70,52	70,49	70,72	70,56	70,99	69,35	69,82	71,05	71,64	71,13	71,73
70,41	70,65	70,26	70,37	67,19	67,39	67,61	67,83	67,13	67,43	66,22	66,38	71,51	71,85	70,96	71,53	71,05	71,53	71,13	71,62
66,51	66,68	70,26	70,35	70,16	70,30	70,04	70,21	69,24	69,48	66,22	66,31	70,57	70,80	70,63	71,08	70,70	71,16	70,77	71,22
70,35	70,44	71,76	71,84	70,16	70,28	70,04	70,19	69,24	69,44	70,43	70,50	65,96	66,11	71,65	71,99	70,70	71,09	70,77	71,16
70,35	70,41	71,76	72,00	71,46	71,57	71,10	71,25	69,19	69,38	72,27	72,30	65,96	66,03	70,64	70,87	71,78	72,12	71,90	72,24
72,03	72,09	70,97	71,22	71,46	71,68	71,10	71,32	69,19	69,48	72,27	72,45	70,50	70,55	65,72	65,85	70,71	70,93	70,78	70,99
71,14	71,39	70,97	71,27	70,77	71,03	70,54	70,81	69,83	70,15	71,30	71,52	72,49	72,50	65,72	65,78	65,49	65,61	65,27	65,39
64,19	64,52	64,76	65,07	65,39	65,71	66,15	66,46	69,00	69,39	63,69	63,99	72,49	72,65	70,57	70,61	65,49	65,53	65,27	65,30
64,19	64,43	64,76	65,00	65,39	65,64	66,15	66,41	69,51	69,91	63,69	63,92	71,44	71,66	70,57	70,57	70,64	70,65	70,70	70,70
68,90	69,09	68,96	69,17	69,04	69,26	69,12	69,36	71,12	71,48	68,84	69,03	63,22	63,52	72,69	72,69	70,64	70,62	70,70	70,66
68,79	68,87	68,86	68,96	68,95	69,08	69,05	69,21	70,36	70,71	68,84	68,93	63,22	63,45	71,57	71,79	72,89	72,86	73,07	73,03
68,79	69,02	68,86	69,11	68,95	69,21	69,05	69,33	69,78	70,12	68,72	68,78	68,79	68,97	71,57	71,90	71,70	71,91	71,81	72,02
70,22	70,53	70,15	70,46	70,06	70,37	69,96	70,27	68,59	68,83	70,29	70,61	68,79	68,86	62,79	63,11	71,70	72,03	71,81	72,15
68,37	68,85	70,15	70,59	70,06	70,47	69,96	70,34	69,44	69,60	68,26	68,79	68,66	68,70	62,79	63,02	62,38	62,71	61,99	62,32
69,51	70,04	68,49	68,95	68,63	69,07	68,79	69,20	69,66	69,81	69,51	70,06	70,35	70,67	68,74	68,90	62,38	62,60	61,99	62,20
73,10	73,56	68,49	68,98	69,51	69,98	69,51	69,95	68,95	69,18	73,45	73,91	68,16	68,71	68,74	68,79	68,69	68,85	68,64	68,79
71,41	71,84	69,51	70,01	72,29	72,71	71,78	72,18	69,17	69,45	71,59	72,04	69,51	70,08	68,60	68,61	68,69	68,73	68,64	68,67
71,41	71,82	72,72	73,16	70,98	71,38	70,71	71,09	69,93	70,21	71,59	72,01	73,76	74,24	70,41	70,73	68,54	68,54	68,49	68,47
70,11	70,51	71,21	71,62	70,98	71,36	70,71	71,07	71,07	71,38	70,17	70,58	73,76	74,23	68,06	68,63	70,46	70,78	70,52	70,84
67,45	67,63	71,21	71,60	69,97	70,35	69,89	70,25	67,07	67,40	67,25	67,42	71,75	72,22	69,50	70,10	67,98	68,56	67,89	68,49

Продолжение табл. В.1

69,35	69,36	70,05	70,44	69,97	70,23	69,89	70,18	68,41	68,72	69,34	69,32	70,22	70,64	74,05	74,55	69,50	70,12	69,50	70,14
69,35	69,34	70,05	70,27	67,91	68,12	68,20	68,42	68,41	68,65	69,34	69,29	67,07	67,25	74,05	74,53	74,33	74,84	74,59	75,11
69,84	69,82	67,66	67,86	69,39	69,46	69,41	69,52	70,63	70,85	69,87	69,83	69,32	69,29	71,91	72,38	74,33	74,82	74,59	75,09
69,84	69,94	67,66	67,73	69,39	69,43	69,72	69,81	70,09	70,40	69,87	69,97	69,32	69,26	70,27	70,70	72,05	72,54	72,19	72,69
68,26	68,41	69,37	69,41	69,77	69,81	68,72	68,92	70,09	70,49	68,15	68,28	69,90	69,83	66,90	67,07	70,31	70,75	70,36	70,81
68,76	69,03	69,81	69,82	69,77	69,87	69,03	69,30	69,24	69,65	68,69	68,95	69,90	69,99	69,31	69,26	66,74	66,91	66,59	66,75
70,44	70,73	68,40	68,55	68,55	68,72	70,10	70,38	69,24	69,63	70,53	70,82	68,04	68,16	69,31	69,22	69,30	69,22	69,29	69,19
72,99	73,35	68,84	69,10	68,93	69,20	71,71	72,04	69,92	70,28	73,33	73,69	68,04	68,27	69,93	69,83	69,30	69,18	69,29	69,15
64,04	64,45	70,34	70,63	70,23	70,52	66,05	66,41	68,53	68,81	63,52	63,94	68,62	68,88	69,93	70,02	69,95	69,84	69,98	69,84
64,04	64,40	72,63	72,98	72,21	72,55	66,05	66,39	68,49	68,75	63,52	63,89	70,61	70,90	67,93	68,05	67,84	67,94	67,75	67,84
67,05	67,40	64,62	65,02	65,28	65,66	67,96	68,28	69,83	70,07	66,82	67,18	73,63	73,99	67,93	68,12	68,50	68,76	68,45	68,71
67,05	67,25	64,62	64,97	65,28	65,62	71,10	71,28	71,03	71,32	66,82	66,99	63,04	63,47	68,56	68,82	70,76	71,05	70,82	71,11
72,02	72,15	67,31	67,66	67,61	67,94	70,33	70,63	70,26	70,62	72,26	72,38	63,04	63,42	70,69	70,98	74,18	74,56	74,44	74,82
72,02	72,27	67,31	67,54	67,61	67,87	70,33	70,76	69,00	69,35	72,26	72,50	66,61	66,97	73,92	74,28	74,18	74,62	74,44	74,88
70,81	71,12	71,76	71,90	71,46	71,62	69,13	69,58	70,40	70,65	70,94	71,25	66,61	66,75	73,92	74,35	62,18	62,62	61,78	62,24
68,90	69,47	71,76	72,02	70,52	70,83	69,13	69,53	69,26	69,44	68,85	69,44	72,48	72,58	62,60	63,03	62,18	62,59	61,78	62,20
70,44	70,92	70,67	70,98	70,52	70,99	70,10	70,48	68,78	68,98	70,53	71,02	72,48	72,73	62,60	63,01	66,22	66,59	66,04	66,41
67,32	67,62	68,97	69,51	69,04	69,54	70,10	70,40	69,07	69,24	70,53	70,84	71,05	71,36	66,40	66,77	66,22	66,34	66,04	66,15
67,23	67,48	70,34	70,80	70,23	70,64	68,13	68,41	68,90	69,10	67,12	67,41	68,79	69,41	66,40	66,54	72,88	72,97	73,06	73,14
67,23	67,44	67,55	67,85	67,82	68,10	68,07	68,32	69,30	69,57	67,12	67,38	70,61	71,12	72,69	72,79	72,88	72,99	73,06	73,17
70,22	70,42	67,47	67,72	67,74	68,00	69,96	70,19	70,07	70,39	67,01	67,26	70,61	70,93	72,69	72,83	71,26	71,57	71,35	71,66
70,22	70,49	67,47	67,68	67,74	67,96	71,65	71,95	68,45	68,80	70,29	70,49	66,92	67,22	71,16	71,47	68,70	69,36	68,65	69,33
72,90	73,21	70,15	70,36	70,06	70,28	70,57	70,97	69,86	70,22	73,22	73,52	66,92	67,19	68,74	69,38	70,75	71,30	70,82	71,38
71,19	71,65	70,15	70,41	70,06	70,31	68,79	69,18	70,64	70,95	73,22	73,67	66,81	67,06	70,68	71,22	70,75	71,09	70,82	71,16
71,19	71,66	72,54	72,85	72,14	72,44	70,76	71,00	69,66	69,99	71,35	71,82	70,35	70,54	70,68	71,01	66,58	66,88	66,42	66,72
68,38	68,84	71,01	71,45	70,81	71,23	69,16	69,29	70,21	70,52	71,35	71,82	73,52	73,82	66,74	67,05	66,58	66,84	66,42	66,68

Продолжение табл. В.1

68,38	68,67	71,01	71,46	70,81	71,23	68,48	68,64	69,46	69,69	68,27	68,73	73,52	73,98	66,74	67,01	66,45	66,69	66,28	66,53
71,49	71,71	68,50	68,94	68,63	69,05	68,88	69,02	71,53	71,73	71,68	71,91	71,49	71,99	66,62	66,87	70,46	70,65	70,52	70,69
68,96	69,00	68,50	68,81	68,63	68,96	68,64	68,82	70,43	70,60	71,68	71,73	71,49	71,98	70,41	70,60	74,06	74,36	74,30	74,60
68,96	69,02	71,28	71,51	71,04	71,28	69,21	69,48	67,70	67,87	68,90	68,93	68,17	68,64	73,80	74,09	74,06	74,54	74,30	74,79
67,89	67,96	69,01	69,08	69,08	69,18	70,31	70,64	69,55	69,65	68,90	68,97	68,17	68,47	73,80	74,27	71,76	72,28	71,88	72,42
67,89	67,96	69,01	69,10	69,08	69,19	68,01	68,39	69,55	69,61	67,73	67,81	71,85	72,08	71,63	72,14	67,99	68,49	67,91	68,42
68,52	68,57	68,06	68,16	68,25	68,37	70,01	70,40	67,70	67,76	68,42	68,48	68,85	68,86	71,63	72,13	72,17	72,39	67,91	68,20
68,14	68,25	68,06	68,15	68,25	68,37	71,11	71,44	67,70	67,84	68,01	68,10	68,85	68,88	68,07	68,57	72,17	72,19	72,31	72,53
69,04	69,30	68,62	68,70	68,74	68,85	69,72	70,08	70,54	70,71	69,00	69,26	67,59	67,63	68,07	68,45	68,77	68,75	68,73	68,69
70,77	71,14	68,28	68,41	68,45	68,59	70,50	70,83	69,91	70,16	70,89	71,27	67,59	67,62	72,01	72,24	67,33	67,37	68,73	68,71
67,13	67,59	69,09	69,35	69,15	69,41	69,44	69,65	69,91	70,24	66,90	67,38	68,33	68,35	68,81	68,80	67,33	67,35	67,21	67,22
70,30	70,77	70,63	70,99	70,48	70,83	72,37	72,54	70,15	70,51	70,37	70,86	68,33	68,40	68,81	68,81	68,18	68,18	68,10	68,08
70,30	70,68	67,38	67,82	67,67	68,08	70,82	70,94	70,15	70,50	70,37	70,77	67,89	67,97	67,46	67,49	68,18	68,22	67,57	67,61
72,04	72,41	70,21	70,66	70,12	70,54	70,82	70,96	70,38	70,72	72,29	72,66	67,89	68,06	67,46	67,47	67,67	67,74	68,85	69,10
72,04	72,44	70,21	70,58	70,12	70,47	66,95	67,09	71,26	71,54	72,29	72,70	68,96	69,21	68,25	68,26	68,88	69,13	71,29	71,70
69,85	70,26	71,78	72,13	71,47	71,81	66,95	67,02	69,19	69,41	69,88	70,30	71,00	71,39	68,25	68,29	71,20	71,60	66,15	66,68
71,08	71,46	71,78	72,16	71,47	71,83	69,56	69,59	69,37	69,53	69,88	70,29	66,70	67,19	67,78	67,84	66,32	66,84	66,15	66,70
69,39	69,60	69,81	70,21	69,77	70,15	66,95	66,92	70,13	70,19	71,23	71,62	66,70	67,20	68,92	69,17	66,32	66,85	70,62	71,18
74,02	74,14	70,92	71,28	70,73	71,08	66,95	67,05	70,49	70,58	69,38	69,58	70,44	70,95	71,10	71,50	70,57	71,10	70,62	71,09
74,02	74,09	69,41	69,62	69,42	69,64	70,96	71,08	68,40	68,54	74,45	74,55	70,44	70,89	66,50	67,01	70,57	71,02	73,09	73,50
71,58	71,62	73,55	73,68	73,01	73,16	70,96	71,15	68,96	69,12	74,45	74,50	72,51	72,89	66,50	67,02	72,91	73,30	69,98	70,44
65,45	65,53	71,36	71,43	71,11	71,21	70,07	70,31	69,55	69,68	71,77	71,80	69,91	70,33	70,51	71,03	69,96	70,40	69,98	70,43
65,45	65,43	71,36	71,44	71,11	71,22	70,41	70,82	67,03	67,23	65,07	65,12	69,91	70,32	70,51	70,96	69,96	70,39	71,73	72,16
69,59	69,51	65,88	65,97	66,37	66,48	70,75	71,14	70,67	70,94	65,07	65,02	71,37	71,77	72,72	73,10	71,62	72,04	71,73	71,98
65,47	65,27	65,88	65,86	66,37	66,38	71,99	72,28	70,67	70,96	69,60	69,49	71,37	71,66	69,93	70,37	71,62	71,88	69,34	69,51
65,47	65,44	69,58	69,52	69,57	69,55	69,06	69,26	68,48	68,77	65,08	64,84	69,37	69,55	69,93	70,36	69,35	69,53	75,89	75,94

Продолжение табл. В.1

71,81	71,85	65,89	65,74	69,57	69,48	69,32	69,43	68,48	68,84	65,08	65,02	74,85	74,93	71,50	71,91	75,57	75,62	75,89	75,87
71,81	72,01	65,89	65,90	66,38	66,28	69,32	69,32	70,66	71,07	72,03	72,05	74,85	74,88	71,50	71,78	75,57	75,57	72,43	72,40
70,39	70,62	71,57	71,63	66,38	66,44	70,38	70,37	68,72	69,10	72,03	72,20	71,96	71,97	69,36	69,54	72,28	72,26	72,43	72,40
70,39	70,80	71,57	71,76	71,29	71,38	70,38	70,38	70,15	70,50	70,48	70,70	64,71	64,75	75,22	75,29	64,07	64,08	63,77	63,77
70,94	71,42	70,30	70,53	71,29	71,49	70,89	70,91	70,33	70,61	71,08	71,58	64,71	64,64	75,22	75,24	64,07	63,92	63,77	63,63
71,47	71,92	70,79	71,25	70,19	70,44	67,93	68,03	69,33	69,61	71,65	72,14	69,60	69,47	72,12	72,12	69,62	69,43	69,62	69,40
73,43	73,74	71,26	71,70	70,62	71,05	68,73	68,86	70,07	70,31	73,80	74,12	64,73	64,45	64,38	64,40	69,62	69,32	69,62	69,24
68,79	68,97	73,02	73,32	71,02	71,44	68,73	68,83	69,38	69,56	68,73	68,90	64,73	64,64	64,38	64,29	64,09	63,74	63,79	63,39
68,79	68,86	68,87	69,06	72,55	72,85	69,57	69,65			68,73	68,76	72,23	72,23	69,61	69,44	64,09	63,97	63,79	63,61
69,20	69,24	69,24	69,30	68,96	69,15	69,57	69,72			69,17	69,19	72,23	72,40	69,61	69,32	72,60	72,57	72,76	72,71
70,89	70,73	69,24	69,15	69,27	69,36	66,00	66,18			69,17	69,09	70,55	70,77	64,40	64,08	72,60	72,80	72,76	72,94
71,70	71,56	70,74	70,62	69,27	69,23	71,15	71,43			71,02	70,82	71,20	71,72	64,40	64,27	70,69	70,94	70,76	71,00
67,02	67,03	71,47	71,39	70,58	70,51	71,15	71,46			71,91	71,74	71,82	72,33	72,42	72,40	71,43	71,98	71,53	72,11
67,02	67,06	67,28	67,32	70,58	70,53	68,06	68,37			66,78	66,77	74,15	74,47	72,42	72,60	72,13	72,67	72,28	72,83
68,28	68,33	68,41	68,48	71,20	71,17	68,06	68,51			66,78	66,80	68,66	68,83	70,63	70,87	74,77	75,10	75,05	75,39
69,60	69,59	68,41	68,44	67,58	67,64	71,14	71,62			68,16	68,19	68,66	68,68	71,32	71,86	68,55	68,70	68,50	68,64
63,96	64,09	69,59	69,61	68,56	68,66	71,14	71,60			69,61	69,58	69,15	69,15	71,98	72,50	68,55	68,54	68,50	68,48
63,96	64,22	64,54	64,70	68,56	68,62	68,40	68,84			63,43	63,54	69,15	69,02	74,47	74,80	69,10	69,06	69,08	69,03
72,10	72,40	64,54	64,82	69,58	69,63	70,41	70,81			63,43	63,69	71,14	70,90	68,61	68,76	69,10	68,89	69,08	68,83
72,10	72,44	71,83	72,13	69,58	69,73	70,41	70,73			72,35	72,65	72,10	71,89	68,61	68,61	71,36	71,05	71,46	71,12
67,22	67,56	71,83	72,16	65,21	65,38	70,67	70,98			72,35	72,69	66,56	66,53	69,12	69,10	72,44	72,18	72,60	72,31
72,09	72,68	67,46	67,79	65,21	65,47	70,67	70,96			67,00	67,35	66,56	66,56	69,12	68,95	72,44	72,35	72,60	72,49
72,09	72,69	71,82	72,37	71,52	71,81	69,26	69,55			67,00	67,54	68,06	68,07	71,25	70,98	66,16	66,09	65,98	65,89
67,75	68,30	71,82	72,37	71,52	71,84	70,31	70,55			72,34	72,95	69,62	69,56	72,28	72,04	66,16	66,12	65,98	65,93
70,93	71,45	67,94	68,46	67,73	68,06					72,34	72,98	62,94	63,05	66,36	66,30	67,86	67,84	67,77	67,74
70,93	71,32	67,94	68,43	71,51	72,02					67,58	68,17	62,94	63,20	66,36	66,34	69,63	69,54	69,64	69,53

Продолжение табл. В.1

71,35	71,69	70,78	71,27	71,51	72,02					71,07	71,59	72,58	72,89	67,95	67,95	62,06	62,15	61,66	61,73
69,12	69,43	70,78	71,20	68,15	68,63					71,07	71,47	72,58	72,93	69,62	69,55	62,06	62,32	61,66	61,91
70,77	71,01	71,15	71,49	68,15	68,61					71,52	71,88	66,80	67,16	62,49	62,58	72,99	73,30	73,17	73,50
69,23	69,33	69,16	69,45	70,61	71,06					69,08	69,40	66,80	67,36	62,49	62,74	72,99	73,35	73,17	73,54
		70,64	70,88	70,93	71,26					70,89	71,13	72,56	73,21	72,79	73,10	66,43	66,81	66,27	66,65
		69,26	69,37	69,21	69,49					70,89	71,00	72,56	73,24	72,79	73,15	66,43	67,03	66,27	66,88
				70,48	70,72							67,43	68,04	66,61	66,98	72,97	73,67	73,16	73,89
				69,29	69,42							67,43	67,97	66,61	67,19	72,97	73,71	73,16	73,92
												71,19	71,73	72,77	73,45	67,15	67,81	67,02	67,70
												71,68	72,05	72,77	73,48	67,15	67,73	67,02	67,62
												71,68	71,98	67,29	67,92	71,42	72,00	71,52	72,12
												69,04	69,35	67,29	67,85	71,97	72,36	72,10	72,50
												69,04	69,29	71,31	71,87	71,97	72,28	72,10	72,41
												71,00	71,23	71,83	72,21	68,98	69,30	68,95	69,27
												69,17	69,25	71,83	72,13	68,98	69,22	71,29	71,51
																71,20	71,42	71,29	71,37
																71,20	71,34	69,11	69,16
																69,13	69,19		