

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный технологический институт
(технический университет)»
СПбГТИ(ТУ)

УГНС	09.00.00	Информатика и вычислительная техника
Направление подготовки	09.03.01	Информатика и вычислительная техника
Направленность (профиль)		Системы автоматизированного проектирования
Форма обучения		очная
Факультет		Информационных технологий и управления
Кафедра		Систем автоматизированного проектирования и управления
Учебная дисциплина		ОПЕРАЦИОННЫЕ СИСТЕМЫ
Курс	III	Группа 494

КУРСОВОЙ ПРОЕКТ

Тема: Создание приложения под ОС MS Windows 10
Задача: Мониторинг контролируемых технологических параметров

Студент	_____	Гусев А.А.
Руководитель, доцент, к.т.н.	_____	Макарук Р.В.
Оценка за курсовой проект	_____	_____

Санкт-Петербург
2021-2022 уч. год

Содержание

ВВЕДЕНИЕ	3
1 Аналитический обзор	4
1.1 Обзор имеющихся на рынке систем мониторинга контролируемых технологических параметров.....	4
1.2 Анализ технологии разработки программного обеспечения в среде RAD (RAD - rapid application development)	6
1.3 Использование Microsoft Windows API (API - application programming interface)	7
1.4 Возможности технологии Active X компании Microsoft	8
1.5 Установка программного продукта в операционной системе и анализ возможностей для переноса приложения на другой компьютер	8
1.6 Обзор и обоснование выбора инструментальных средств разработки программного комплекса.	9
2 Основная часть.....	11
2.1 Определение структурной (иерархической) схемы решения задачи	11
2.2 Определение основных этапов проектирования.....	11
2.3 Анализ ограничений и исключительных ситуаций для алгоритмов	11
2.4 Разработка основных алгоритмов задачи	12
2.5 Разработка архитектуры программы	13
2.6 Разработка дисплейных фрагментов.....	14
2.7 Отладка программного комплекса	19
2.8 Создание справочной системы приложения	19
2.9 Создание инсталляционного пакета.....	21
2.10 Тестирование разработанного программного продукта	21
2.11 Разработка эксплуатационного документа «Руководство системного программиста»	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24

ВВЕДЕНИЕ

В настоящее время все чаще применяется метод планирования активного эксперимента для построения математической модели при выборе контролируемых параметров технологических процессов независимо от их физической природы.

Методология эксперимента позволяет успешно решать наиболее важные для исследователя вопросы: определять количество опытов, которые следует провести, каким образом обрабатывать их результаты, чтобы решить поставленную задачу при минимальном числе опытов.

Характеристикой современного производства является существенное повышение требований к качеству и точности изготовленных деталей. Это в свою очередь требует оснащения металлообрабатывающих станков высокоэффективными управляемыми, специальными информационными системами и другими функциональными устройствами с улучшенными техническими характеристиками.

Мониторинг технологического процесса объединяет в себе диагностику, идентификацию, прогнозирование и управление состоянием технической системой на основе анализа полученной информации, а также принятие решения о введении коррекции [1].

Обеспечение мониторинга контролируемых технологических параметров является неотъемлемой частью разработки системы автоматизированного управления для безопасной эксплуатации сложных технических систем. Целью данного курсового проекта является разработка программного комплекса для мониторинга контролируемых технологических параметров химического процесса с использованием возможностей операционной системы Microsoft Windows 10.

Необходимо ознакомиться с имеющимися на рынке системами мониторинга контролируемых технологических параметров, проанализировать имеющиеся возможности для реализации поставленной цели в Microsoft Windows 10, а также обеспечить возможности для переноса реализуемого приложения на другой компьютер.

1 Аналитический обзор

1.1 Обзор имеющихся на рынке систем мониторинга контролируемых технологических параметров

НТЦ «Комплексные системы» разрабатывают распределенные технологические информационные системы мониторинга, успешно работающие на ряде предприятий энергетики. Среди разработанных систем мониторинга можно выделить следующие.

Система мониторинга технологических процессов (СМТП) ПАО «Юнипро» (рисунок 1).

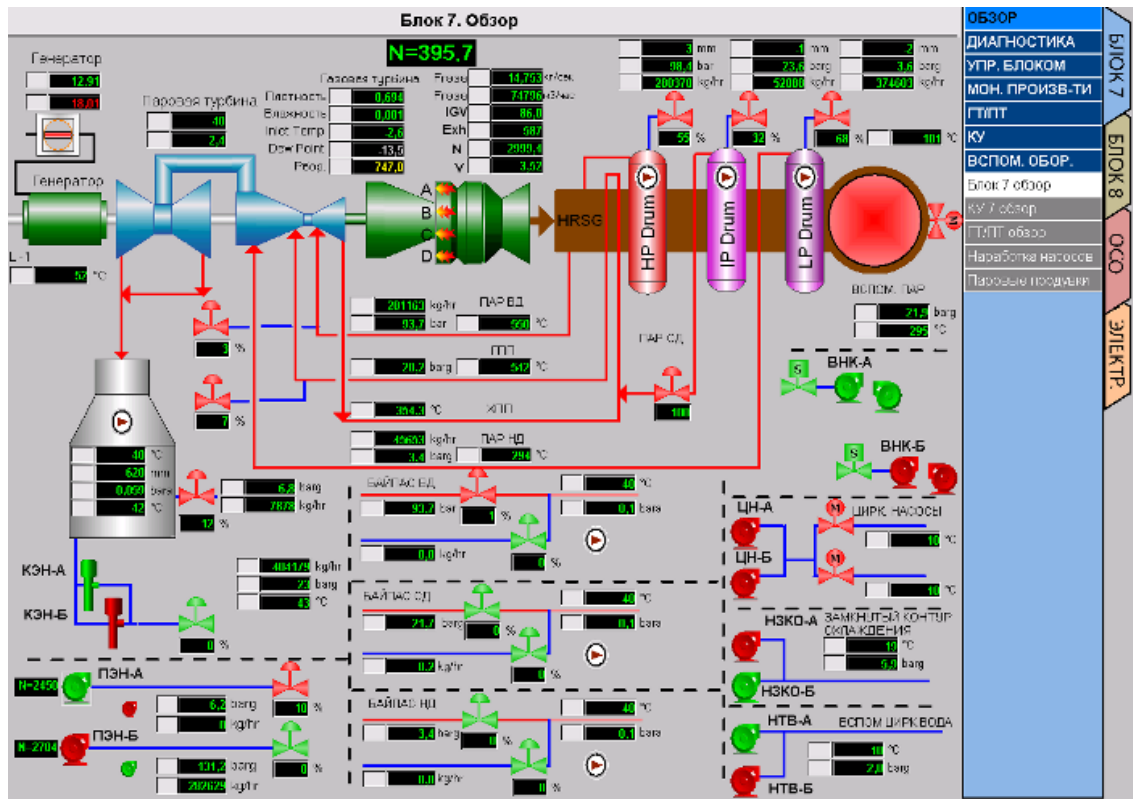


Рисунок 1 — Система мониторинга технологических процессов «Юнипро»

Система мониторинга технологических процессов (СМТП) – система реального времени, выполняет следующие задачи:

- Автоматический сбор информации реального времени (значений параметров технологических процессов) посредством интерфейсов ко всем функционирующим на объектах ПАО «Юнипро» системам АСУТП, телемеханики и учета энергоресурсов, и других информационных систем;
- Долговременное хранение данных в единой базе данных реального времени;
- Обеспечение стандартного, унифицированного доступа к данным для пользователей - посредством клиентских приложений;
- Сбор, хранение и обработка коммерческой информации по рынку электроэнергии в интервалах;

- Обеспечение руководителей и специалистов всех уровней информацией о состоянии технологического оборудования, необходимой для анализа и принятия решений в режиме реального времени [2].

Также примером системы мониторинга контролируемых технологических параметров является распределенная система учета тепла на границах магистральных тепловых сетей г. Тюмень, разработанная также научно-техническим центром «Комплексные системы» (рисунок 2).

Показания ТП					
Показания технологических параметров (общие ТЭЦ-1, ТЭЦ-2)					
Тип данных	Норма расхода, т/ч	Расход, т/ч		Небаланс расходов, т/ч G1-G2	Средняя Т обратки, °C
		G1	G2		
Мгновенные	23 570	23 609	23 060	548	66,31
Среднечасовые	23 570	24 306	23 714	592	66,36

Показания технологических параметров контур ТЭЦ-1					
Тип данных	Норма расхода, т/ч	Расход, т/ч		Небаланс расходов, т/ч G1-G2	Средняя Т обратки, °C
		G1	G2		
Мгновенные	13 483	13 694	13 337	357	67,25
Среднечасовые	13 483	13 701	13 351	350	67,66

Показания технологических параметров контур ТЭЦ-2					
Тип данных	Норма расхода, т/ч	Расход, т/ч		Небаланс расходов, т/ч G1-G2	Средняя Т обратки, °C
		G1	G2		
Мгновенные	10 087	9 915	9 723	192	65,38
Среднечасовые	10 087	10 606	10 364	242	65,05

Рисунок 2 — Распределенная система учета тепла

Данная система реального времени, предназначена для осуществления автоматизированного коммерческого учета количества отданной потребителям тепловой энергии и объема теплоносителя. Система обеспечивает сбор данных с узлов учета, передачу данных на сервер базы данных, производство расчетов, хранение, документирование, вывод информации для отображения и печати, формирование и передачу отчетных данных, используемых в финансовых расчетах с потребителями тепловой энергии.

Автоматизированный учет и регистрация отпуска отпускаемой тепловой энергии организуются с целью:

- Контроля за тепловыми и гидравлическими режимами работы систем теплоснабжения;
- Контроля за рациональным использованием тепловой энергии и теплоносителя;

- Документирования физических параметров теплоносителя;
- Минимизации ручного труда при учете тепловой энергии и теплоносителя;
- Повышения оперативности доступа к информации;
- Предоставления обслуживающему персоналу и руководству оперативной и учетной информации о параметрах теплоносителя и теплопотребления по отдельным потребителям (магистральям);
- Архивации исходных данных и результатов учета и получения возможность их ретроспективного анализа [3].

1.2 Анализ технологии разработки программного обеспечения в среде RAD (RAD - rapid application development)

RAD (от англ. Rapid application development — быстрая разработка приложений) — концепция создания средств разработки программных продуктов, уделяющая особое внимание скорости и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. С конца XX века RAD получила широкое распространение и одобрение. Концепцию RAD также часто связывают с концепцией визуального программирования [4].

Основные принципы RAD:

- Инструментарий должен быть нацелен на минимизацию времени разработки.
- Создание прототипа для уточнения требований заказчика.
- Цикличность разработки: каждая новая версия продукта основывается на оценке результата работы предыдущей версии заказчиком.
- Минимизация времени разработки версии, за счёт переноса уже готовых модулей и добавления функциональности в новую версию.
- Команда разработчиков должна тесно сотрудничать, каждый участник должен быть готов выполнять несколько обязанностей.
- Управление проектом должно минимизировать длительность цикла разработки [4].

Причины популярности RAD вытекают из тех преимуществ, которые обеспечивает эта технология (рисунок 3).

Наиболее существенными из них являются:

- высокая скорость разработки;
- низкая стоимость;
- высокое качество.



Рисунок 3 — Преимущества технологии RAD

Последнее из указанных свойств подразумевает полное выполнение требований заказчика как функциональных, так и нефункциональных, с учетом их возможных изменений в период разработки системы, а также получение качественной документации, обеспечивающей удобство эксплуатации и сопровождения системы. Это означает, что дополнительные затраты на сопровождение сразу после поставки будут значительно меньше. Таким образом, полное время от начала разработки до получения приемлемого продукта при использовании этого метода значительно сокращается [4].

1.3 Использование Microsoft Windows API (API - application programming interface)

Microsoft Windows API — общее наименование набора базовых функций интерфейсов программирования приложений операционных систем семейств Microsoft Windows. Предоставляет прямой способ взаимодействия приложений пользователя с операционной системой Windows. Для создания программ, использующих Windows API, корпорация «Майкрософт» выпускает комплект разработчика программного обеспечения, который называется Platform SDK и содержит документацию, набор библиотек, утилит и других инструментальных средств для разработки [5].

Windows API спроектирован для использования в языке Си для написания прикладных программ, предназначенных для работы под управлением операционной системы MS Windows. Работа через Windows API — это наиболее близкий к операционной системе способ взаимодействия с ней из прикладных программ. Более низкий уровень доступа, необходимый только для драйверов устройств, в текущих версиях Windows предоставляется через Windows Driver Model.

Windows API представляет собой множество функций, структур данных и числовых констант. Все языки программирования, способные вызывать такие функции и оперировать такими типами данных в программах, исполняемых в среде Windows, могут пользоваться этим API [5].

Для облегчения программирования под Windows в компании Microsoft и сторонними разработчиками было предпринято множество попыток создать библиотеки и среды программирования, частично или полностью скрывающие от программиста особенности

Windows API и предоставляющие ту или иную часть его возможностей в более удобном виде [6].

Основными компонентами WinAPI являются:

- WinBase: функции ядра, CreateFile, CreateProcess и др.
- WinUser: функции GUI, CreateWindow, RegisterClass и др.
- WinGDI: графические функции, Ellipse, SelectObject и др.
- Общие элементы управления: стандартные элементы управления, списки и др.

1.4 Возможности технологии Active X компании Microsoft

ActiveX — фреймворк для определения программных компонентов, пригодных к использованию из программ, написанных на разных языках программирования. Программное обеспечение может собираться из одного или более таких компонентов, чтобы использовать их функциональность [7].

Впервые эта технология была внедрена в 1996 году компанией Microsoft как развитие технологий Component Object Model (COM) и на данный момент она широко используется в операционных системах семейства Microsoft Windows.

Множество приложений для Microsoft Windows, включая приложения самой компании Microsoft, такие, как Internet Explorer, Microsoft Office, Microsoft Visual Studio, Windows Media Player, используют управляющие элементы ActiveX, чтобы реализовать набор функциональных возможностей и в дополнение инкапсулировать их собственную функциональность в управляющие элементы ActiveX, чтобы предоставить возможность встраивать данные элементы в другие приложения.

Управляющие элементы ActiveX — это строительные блоки для программ. Примеры включают в себя настраиваемые приложения по сбору данных, просмотру определённых типов файлов и отображения анимации. Создание управляющих элементов ActiveX возможно при помощи любого языка программирования, поддерживающего разработку компонентов Component Object Model (COM) [7].

С помощью ActiveX браузер загружает Windows Media Player, Quicktime и другие приложения, которые могут воспроизводить файлы, внедрённые в веб-страницы. Элементы управления ActiveX активизируются при щелчке по такому объекту на веб-странице, например, .WMV-файл, чтобы загрузить его для отображения в окне браузера Internet Explorer [7].

1.5 Установка программного продукта в операционной системе и анализ возможностей для переноса приложения на другой компьютер

Установка программного обеспечения — процесс установки программного обеспечения на компьютер конечного пользователя. Выполняется особой программой (пакетным менеджером), присутствующей в операционной системе (например, Установщик Windows в Microsoft Windows), или же входящим в состав самого программного обеспечения средством установки [8].

Большинство программ поставляются для распространения в сжатом виде. Для нормальной работы они должны быть распакованы, а необходимые данные правильно размещены на компьютере, учитывая различия между компьютерами и настройками пользователя. В процессе установки выполняются различные тесты на соответствие заданным требованиям, а компьютер необходимым образом настраивается для хранения файлов и данных, необходимых для правильной работы программы.

Установка, включает в себя размещение всех необходимых программе файлов в файловой системе, а также модификацию и создание конфигурационных файлов. Пакетные менеджеры также выполняют при установке контроль зависимостей, проверяя, есть ли в системе необходимые для работы данной программы пакеты, а в случае успешной установки регистрируя новый пакет в списке доступных [8].

Одной из задач данного курсового проекта является обеспечение возможности переноса программного продукта на другие компьютеры. Для выполнения данной задачи необходимо проанализировать возможности современных пакетов программ, обеспечивающих создание установочных файлов, иными словами, упаковку программного комплекса в специальных архив, запустив который пользователь сможет установить программу на своём компьютере. Ниже приведены сравнительная таблица отобранных для анализа пакетов программ:

Таблица 1 — Сравнительная характеристика программ-установщиков [9]

Критерий сравнения	Clickteam Install Creator 2	Actual Installer	Advanced Installer
Пошаговое создание	+	+	+
Мультиязычность	-	+	+
Создание и выпуск обновлений	-	+	+
Использование алгоритмов компрессии	+	+	+
Проверка зависимостей	-	+	+

По результатам сравнения (таблица 2) наиболее подходящими пакетами программ являются Actual Installer и Advanced Installer. В силу того, что для данного проекта не требуются специализированные функции и возможности, предоставляемые Advanced Installer, был выбран пакет программ Actual Installer.

1.6 Обзор и обоснование выбора инструментальных средств разработки программного комплекса.

Реализация поставленной задачи требует разработки программного продукта для мониторинга контролируемых технологических параметров. Для этого было выбрано 3 языка программирования C#, C++ и Python и составлена их сравнительная характеристика, представленная в таблице ниже:

Таблица 2 — Сравнительная характеристика языков программирования

Критерий сравнения	C#	C++	Python
Поддержка ООП	+	+	+
Сетевое программирование	+	+	+
Статическая типизация	+	+	-
Автоматический сбор мусора	+	-	+
Компилируемый язык	+	+	-

По результатам сравнения (таблица 2) и учитывая, что для реализации программного комплекса для обеспечения мониторинга технологических параметров не требуется гибкое распределение памяти и возможности более низкоуровневого программирования, был выбран язык C#, располагающий автоматическим сборщиком мусора и распределением используемой памяти. Также данный язык программирования имеет сравнительно невысокий порог вхождения, достаточно часто используется для реализации задач, похожих на задачи данного курсового проекта, а значит располагает необходимыми инструментами и библиотеками.

В качестве среды разработки были отобраны Visual Studio 2019 и Sublime Text. Sublime Text является простым редактором кода с возможностями автодополнения и выбора используемого языка, но без возможности отладки программ без установки дополнительных расширений.

Для данного курсового проекта в качестве среды разработки была выбрана Visual Studio 2019 от компании Microsoft, так как она имеет максимальную совместимость с выбранным языком программирования, позволяет отлаживать программы, располагает множеством инструментов для быстрой и качественной разработки, а также включает в себя все возможности Sublime Text.

2 Основная часть

2.1 Определение структурной (иерархической) схемы решения задачи

В рамках данного курсового проекта необходимо спроектировать программный комплекс для мониторинга контролируемых технологических параметров проведения процесса химико-технологического производства. Необходимо реализовать требуемые возможности, изображенные в виде иерархической схемы ниже:

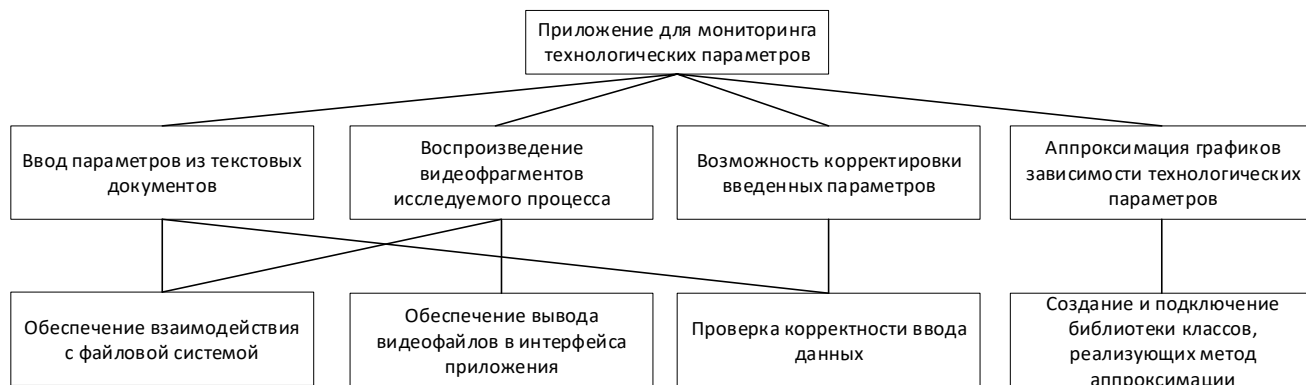


Рисунок 4 — Иерархическая схема решения задачи

2.2 Определение основных этапов проектирования

Среди этапов проектирования приложения выделены следующие:

- Анализ требований и составление иерархической схемы решения задачи;
- Анализ математических методов решения задачи;
- Разработка программного обеспечения;
 - Разработка интерфейсной составляющей проекта;
 - Разработка взаимодействия с файловой системой для обеспечения ввода данных;
 - Создание и подключение библиотеки классов, реализующих математические методы аппроксимации;
 - Обеспечение проверки корректности входных данных;
 - Обработка исключительных ситуаций;
- Тестирование программного продукта;
- Составление технической документации по проекту.

2.3 Анализ ограничений и исключительных ситуаций для алгоритмов

В процессе разработки программного продукта возникал ряд исключительных ситуаций, требующих обработки. Основные ограничения введены для обеспечения корректности ввода исходных данных.

Формат файлов, из которых производится ввод значений технологических параметров — строго текстовый .txt. Внутри читаемого .txt файла для корректной работы

программы должно содержаться 3 строки с десятичными дробями, знак деления целой и дробной части числа — запятая. Числа разделены между собой знаком пробела, строки — знаком возврата каретки (перевода строки). Количество чисел в каждой строке должно быть одинаково, таким образом каждому значению числа из первой строки соответствует значения из второй и третьей строк. Допустимый диапазон значений: 10^{-324} - 10^{308} . Наличие любых символов, отличных от десятичных цифр и знака запятой, приведут к возникновению исключительной ситуации и невозможности ввести параметры из данного файла до устранения ошибки в них. Обеспечение корректности ввода входных параметров позволяет избежать некоторых исключительных ситуаций при работе основного алгоритма программы.

Также для корректной работы программного комплекса необходимо обеспечить минимально допустимый набор входных значений технологических параметров, который должен включать в себя не менее чем 5 пар значений.

2.4 Разработка основных алгоритмов задачи

Основным алгоритмом программы является построение графика зависимости технологических параметров. Программный комплекс обеспечивает возможность построения графика двумя способами:

- Без использования аппроксимации, соединяя точки ломаной линией. Данный способ используется в случае отсутствия шума на графике, т.е. резких изменений технологических параметров в ходе проведения процесса.
- С применением аппроксимации методом наименьших квадратов. Данный способ позволяет построить график функции, усредняющей значения набора технологических параметров, тем самым минимизируя влияние резких изменений значения параметров.

При применении метода наименьших квадратов, имея набор значений некоторой функции, производится поиск коэффициентов полинома (1).

$$a_n * x^n + a_{n-1} * x^{n-1} + \dots + a_1 * x^1 + a_0 * x^0, \quad (1)$$

где n — степень полинома.

Пользователь может изменять степень полинома при помощи управляющих элементов интерфейса для выбора наиболее подходящего отображения главной тенденции.

Для применения в ходе работы программы метода наименьших квадратов необходимо реализовать методы работы с матрицами, а именно:

- Транспонирование;
- Произведение матриц;
- Решение систем линейных алгебраических уравнений;
- Инвертирование матрицы;
- Умножение матрицы на число.

Для обеспечения поставленных требований необходимо создать библиотеку классов, реализующих перечисленные методы работы с матрицами и расчета коэффициентов искомого полинома.

Результатом работы алгоритма является набор коэффициентов полинома, который используется функцией построения графика для расчета значения зависимой переменной и, на основании полученных данных, построения приближенного графика зависимости технологических параметров.

2.5 Разработка архитектуры программы

Программа разбита на 5 классов. Каждый класс отвечает за свой перечень задач, взаимодействуя с методами из других классов для получения необходимых данных. Также была создана библиотека классов, содержащая логику работы метода наименьших квадратов. В таблице ниже представлены классы и их основные методы:

Таблица 3 — Классы и методы приложения

Название класса	Название метода	Назначение
CreateGraphs	CreateGraphs	Конструктор класса, инициализирует поля класса
	MakeUsualGraph	Строить график без использования методов аппроксимации
	SetupBorders	Перенастраивает границы графика при применении аппроксимации
	MakeLSMGraph	Строит график используя метод наименьших квадратов
WorkWithFiles	InputFromFile	Обеспечивает ввод данных из файла и корректность формата данных
LSM	Polynomial	Рассчитывает коэффициенты для полинома функции в методе наименьших квадратов
	getDelta	Метод нахождения среднеквадратичного отклонения
Matrix	Transposition	Транспонирование матрицы
	getMinor	Получение минора матрицы
	Determ	Нахождение детерминанта матрицы
	InverseMatrix	Инвертирование матрицы
MainWindow	FillingTable	Заполнение таблицы входными данными
	SetupGraphs	Установка границ на осях графика
	MakeGraphButton_Click	Построение графика, вызов методов класса CreateGraphs

Также помимо создания перечисленных методов использовано переопределение существующих методов для умножения матрицы на матрицу и на число.

2.6 Разработка дисплейных фрагментов

При первом запуске программного продукта появляется рабочая область, ожидающая ввода исходных данных (рисунок 5).

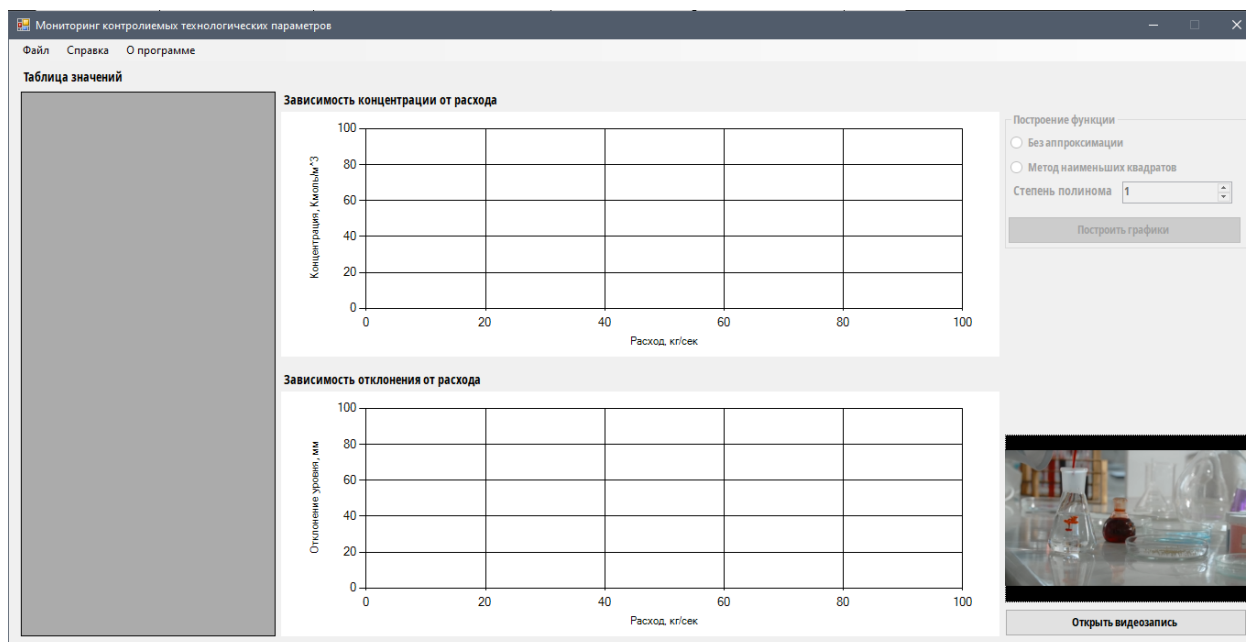


Рисунок 5 — Главное окно приложения при первом запуске

Возможности построения графиков на данном этапе недоступны в силу отсутствия данных для обработки. Пользователю предлагается ввести данные из файла при помощи меню «Файл» (рисунок 6).

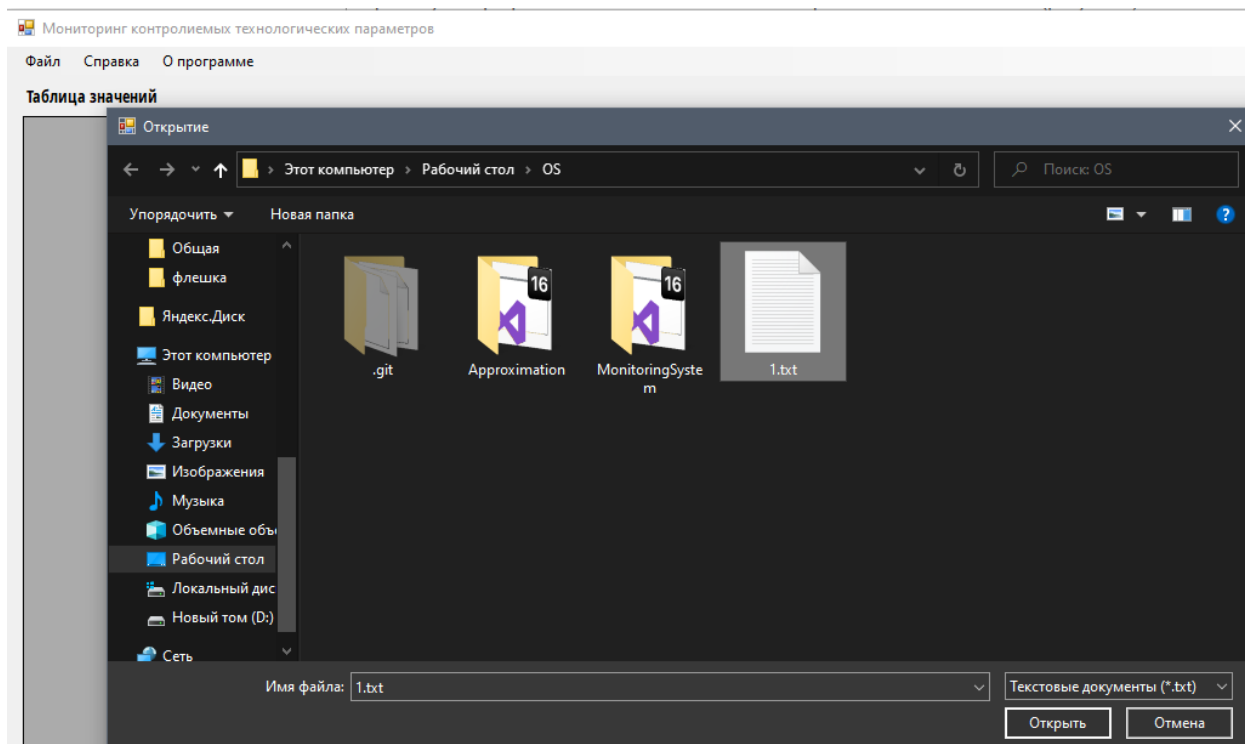


Рисунок 6 — Процесс открытия файла с данными

В случае, если файл содержит различное количество данных в строках, то есть не всем значениям параметра первой строки соответствует значения параметров второй и третьей строк, то пользователь получит соответствующее сообщение об ошибке (рисунок 7).

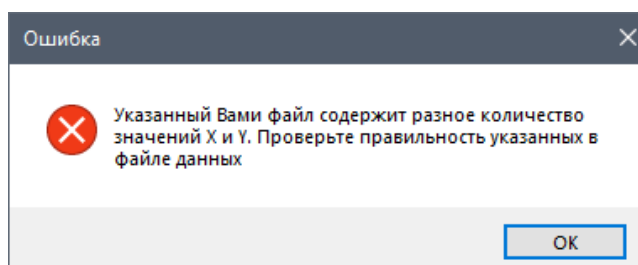


Рисунок 7 — Ошибка ввода данных

В случае некорректности данных внутри указанного файла, пользователь также получит соответствующее сообщение об ошибке (рисунок 8).

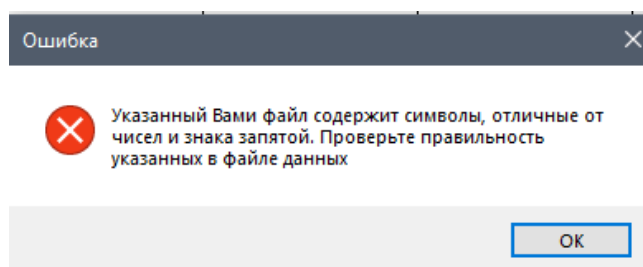
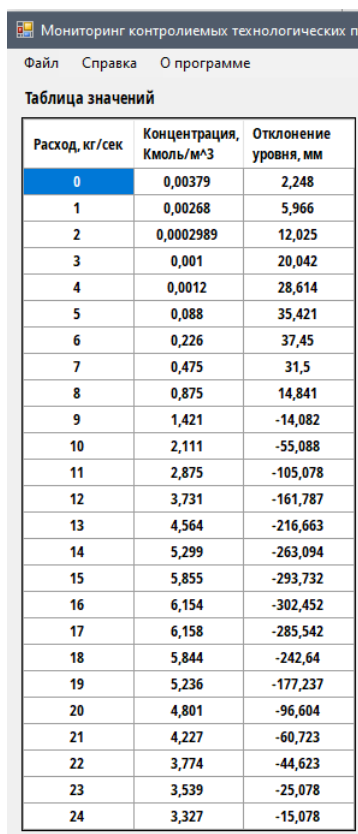


Рисунок 8 — Ошибка ввода некорректных данных

После корректного ввода технологических параметров формируется таблица значений, расположенная в левой части программного интерфейса (рисунок 9). Пользователь может скорректировать значения зависимых переменных (концентрации и отклонения уровня). Изменения данных параметров также проверяется на корректность.



Мониторинг контролируемых технологических параметров

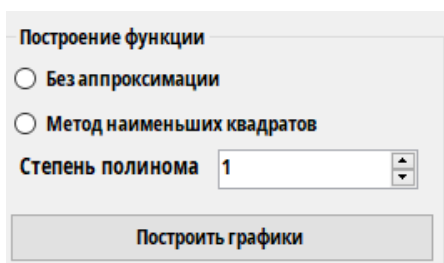
Файл Справка О программе

Таблица значений

Расход, кг/сек	Концентрация, Кмоль/м³З	Отклонение уровня, мм
0	0,00379	2,248
1	0,00268	5,966
2	0,0002989	12,025
3	0,001	20,042
4	0,0012	28,614
5	0,088	35,421
6	0,226	37,45
7	0,475	31,5
8	0,875	14,841
9	1,421	-14,082
10	2,111	-55,088
11	2,875	-105,078
12	3,731	-161,787
13	4,564	-216,663
14	5,299	-263,094
15	5,855	-293,732
16	6,154	-302,452
17	6,158	-285,542
18	5,844	-242,64
19	5,236	-177,237
20	4,801	-96,604
21	4,227	-60,723
22	3,774	-44,623
23	3,539	-25,078
24	3,327	-15,078

Рисунок 9 — Таблица значений технологических параметров

В правой части интерфейса расположены управляющие элементы для построения графика. Пользователю доступен выбор между построением без использования аппроксимации и построением при помощи метода наименьших квадратов, указав степень искомого полинома (рисунок 10). При нажатии кнопки «Построить график» в центральной части интерфейса происходит построение графика по значениям из таблицы, используя настройки, выставленные пользователем (рисунок 11).



Построение функции

☐ Без аппроксимации

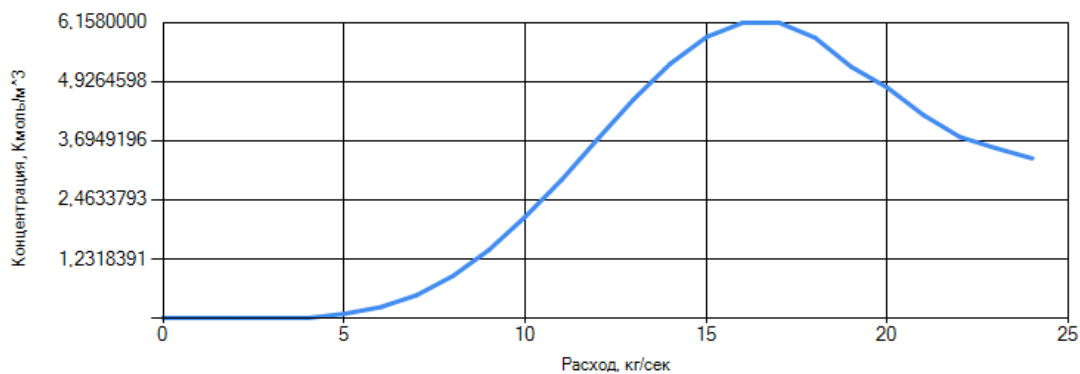
☒ Метод наименьших квадратов

Степень полинома

Построить графики

Рисунок 10 — Варианты построения графиков зависимостей

Зависимость концентрации от расхода



Зависимость отклонения от расхода

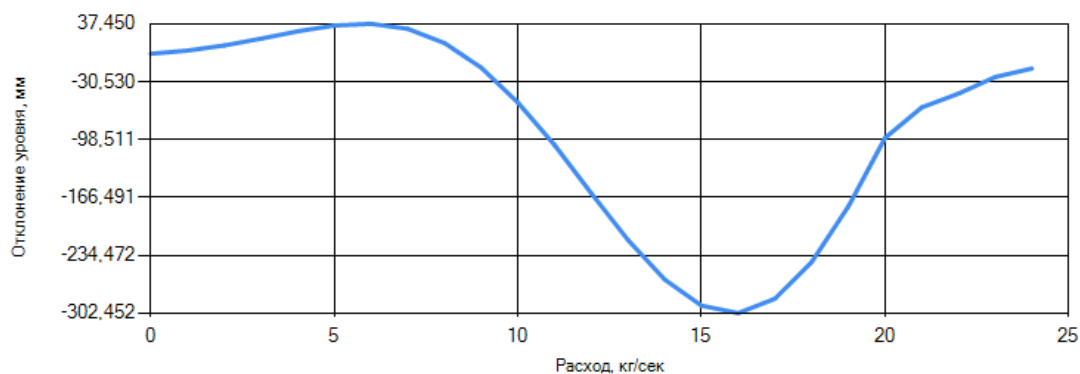
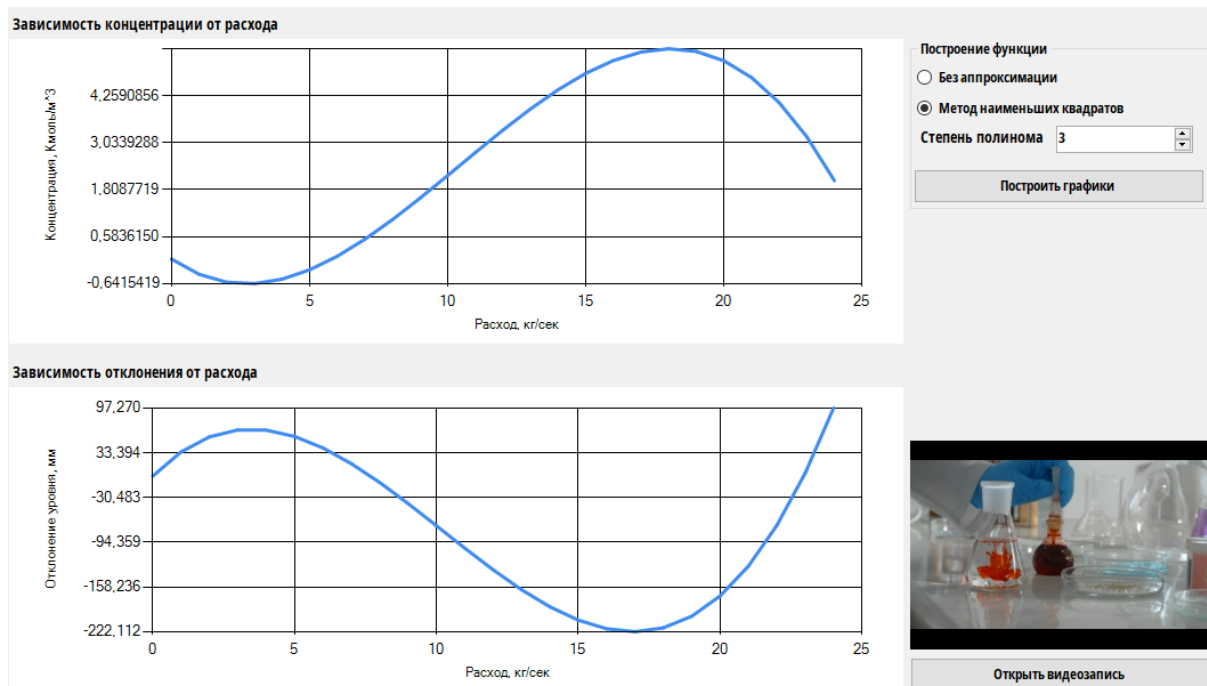


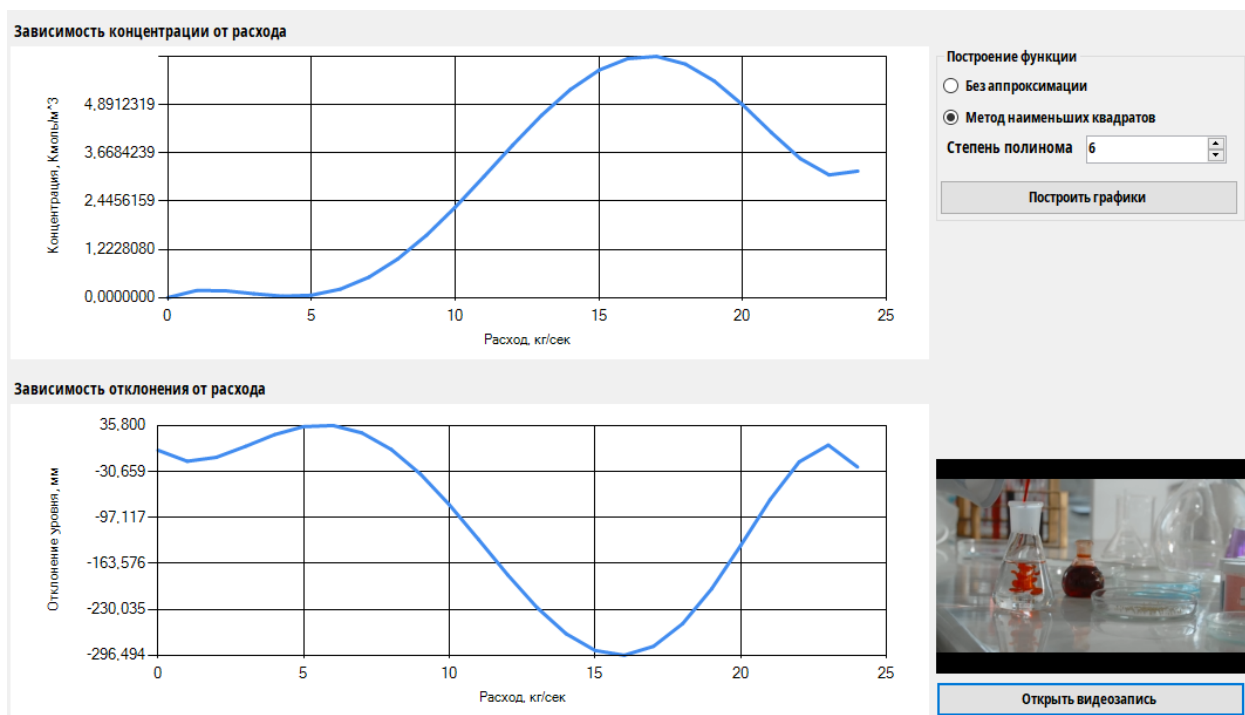
Рисунок 11 — Построение графиков, выполненное без аппроксимации

Настраивая степень полинома, можно получить различную степень аппроксимации графиков зависимостей (рисунок 12, 13).



Степень полинома равна 3.

Рисунок 12 — Построение, используя метод наименьших квадратов



Степень полинома равна 6.

Рисунок 13 — Построение, используя метод наименьших квадратов

В крайней правой части интерфейса воспроизводится видеофрагмент исследуемого технологического процесса (рисунок 14). Пользователь может выбрать другой видеофайл, при помощи кнопки «Открыть видеозапись».

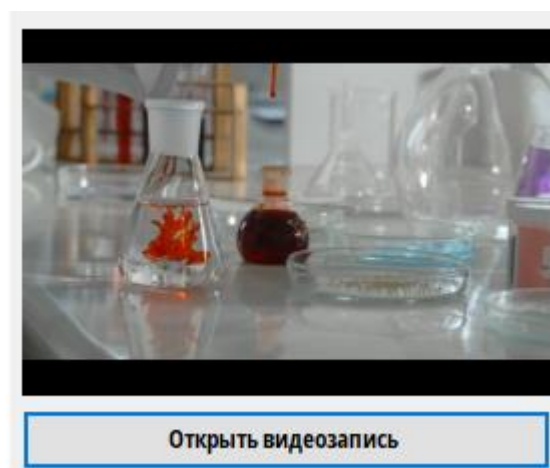


Рисунок 14 — Воспроизведение видеофрагмента

В меню в верхней части окна приложения созданы кнопки «Справка» и «О программе», открывающие информационные окна с краткой справочной информацией и сведениях о программном продукте и его авторе (рисунок 15, 16).

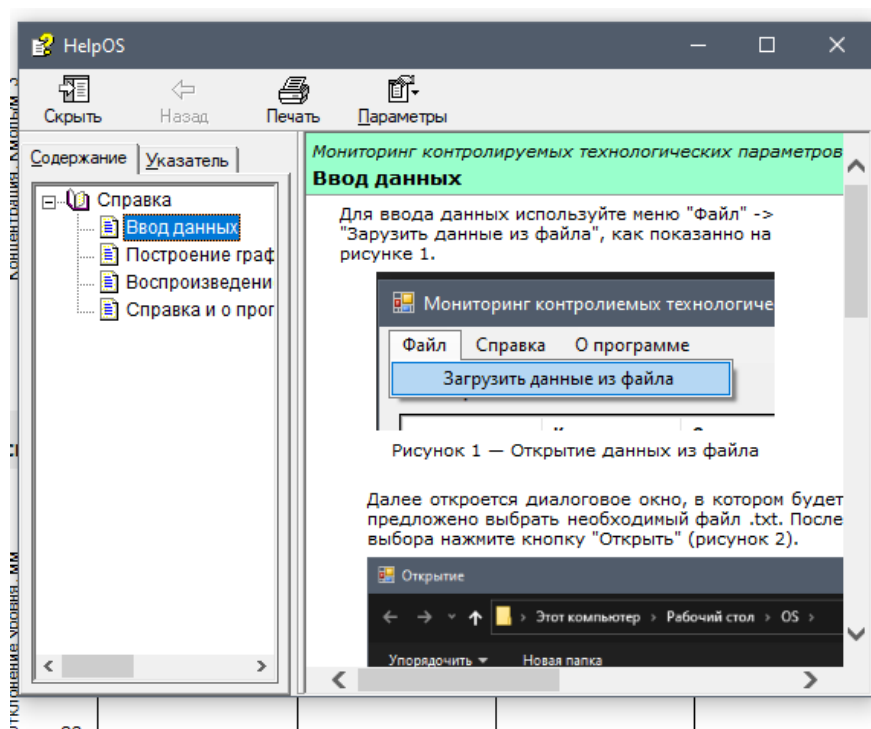


Рисунок 15 - Справочная система приложения

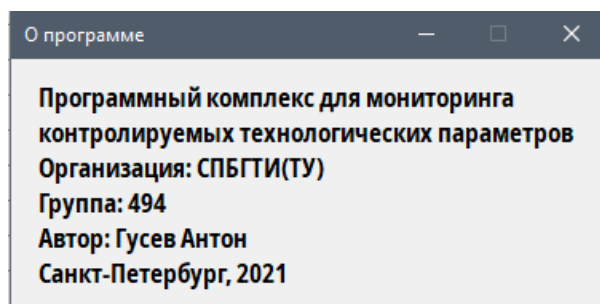


Рисунок 16 — Информационное окно «О программе»

2.7 Отладка программного комплекса

Отладка программного комплекса производилась путем проверки поведения различных частей программного кода при изменяющихся входных параметрах и возмущающем воздействии. В процессе отладки были выявлены, локализованы и устранены следующие ошибки:

- Построение более одного графика в одной системе координат одновременно;
- Ошибки, возникающие при открытии недопустимых файлов с исходными данными;
- Ошибки, возникающие при открытии неподдерживаемого формата видеофайлов;
- Возможность запуска неограниченного количества окон справочной системы;
- Некорректная работа метода наименьших квадратов.

2.8 Создание справочной системы приложения

Для создания справочной системы использовался пакет программ CHM Editor (рисунок 17), позволяющий редактировать файлы с расширением .chm, добавлять

текстовую и графическую информацию на страницы справки, а также структурировать ее содержимое.

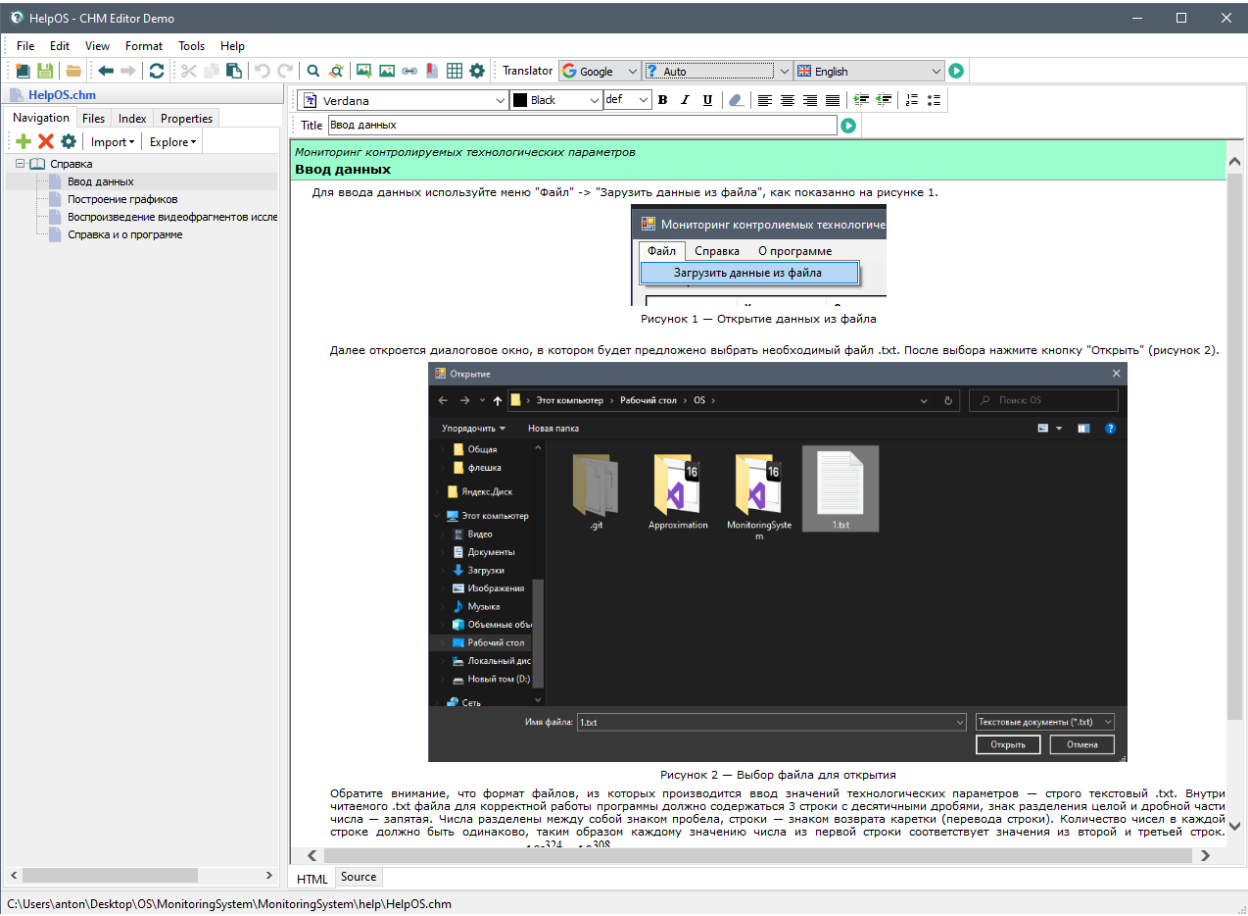


Рисунок 17 — Основное окно программы для создания справочной системы

В процессе создания справки были описаны основных возможности программного продукта и способы взаимодействия с приложением. Шаги и способы работы с программным комплексом для полноты восприятия сопровождаются рисунками (рисунок 18).

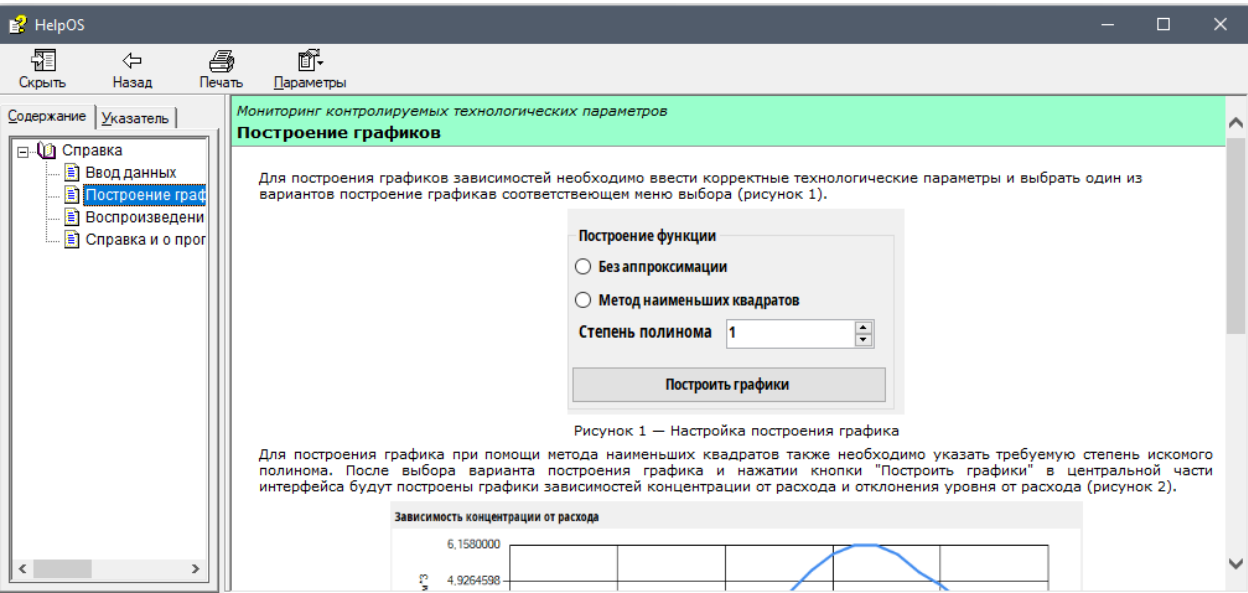


Рисунок 18 — Фрагмент справочной системы

Также в справочной системе присутствует поиск по ключевым словам для более быстрого взаимодействия и поиска необходимой информации.

2.9 Создание инсталляционного пакета

После проверки программы

2.10 Тестирование разработанного программного продукта

Добавить часть про тестирование установки ПО

Тестирование программного комплекса мониторинга контролируемых технологических параметров производилось на тестовых данных, представленных на рисунке ниже:

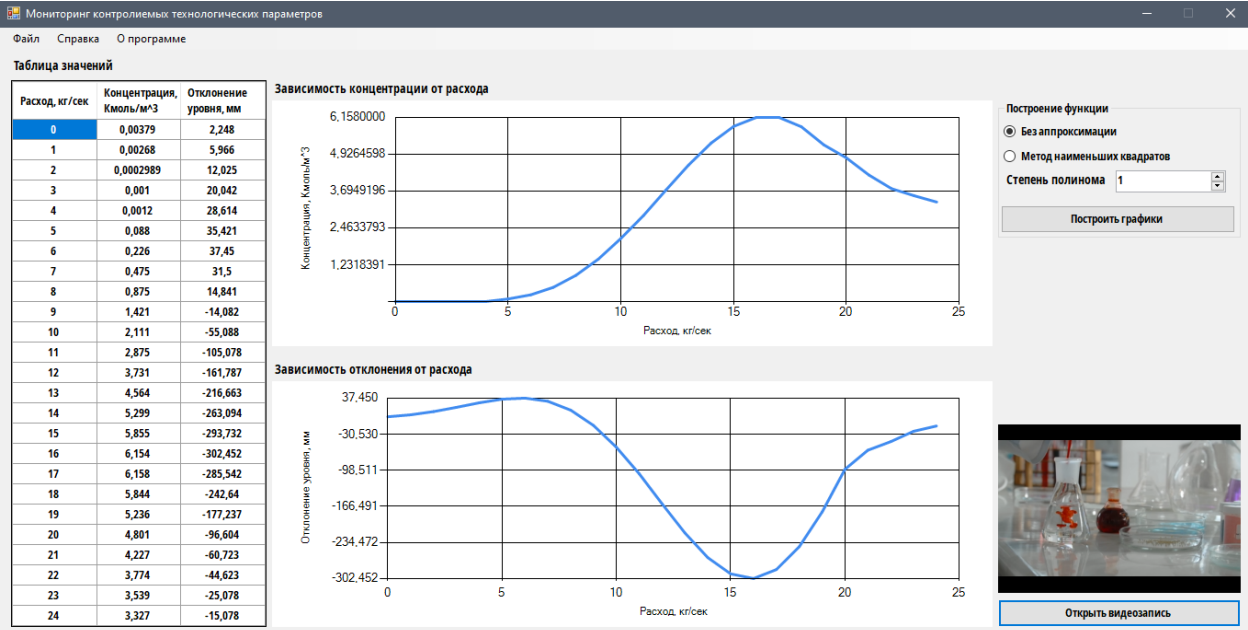
Вариант № 464-04

Опыт № 1		Дата: 5 сентября 2018 г.	
Расход, кг/сек	Выходные параметры		
	Концентрация, Кмоль/м*3	Отклонение уровня, мм	
0	3,7900E-03	2,248	
1	2,6800E-03	5,966	
2	2,9890E-04	12,025	
3	0,001	20,042	
4	0,012	28,614	
5	0,088	35,421	
6	0,226	37,450	
7	0,475	31,500	
8	0,875	14,841	
9	1,421	-14,082	
10	2,111	-55,088	
11	2,875	-105,078	
12	3,731	-161,787	
13	4,564	-216,663	
14	5,299	-263,094	
15	5,855	-293,732	
16	6,154	-302,452	
17	6,158	-285,542	
18	5,844	-242,640	
19	5,236	-177,237	
20	4,801	-96,604	
21	4,227	-60,723	
22	3,774	-44,623	
23	3,539	-25,078	
24	3,327	-15,078	

Рисунок 19 — Таблица тестовых параметров

Данные, представленные на рисунке 19, содержат одинаковое количество переменных во всех столбцах, а также корректные, с точки зрения формата, значения

технологических параметров. В процессе тестирования данные из таблицы подвергались корректировке. Тестирование программного комплекса было успешно пройдено (рисунок 20 - 23). Полученные результаты совпали с ожидаемыми.



Исходные данные не корректировались.

Рисунок 20 — Построение графика без аппроксимации

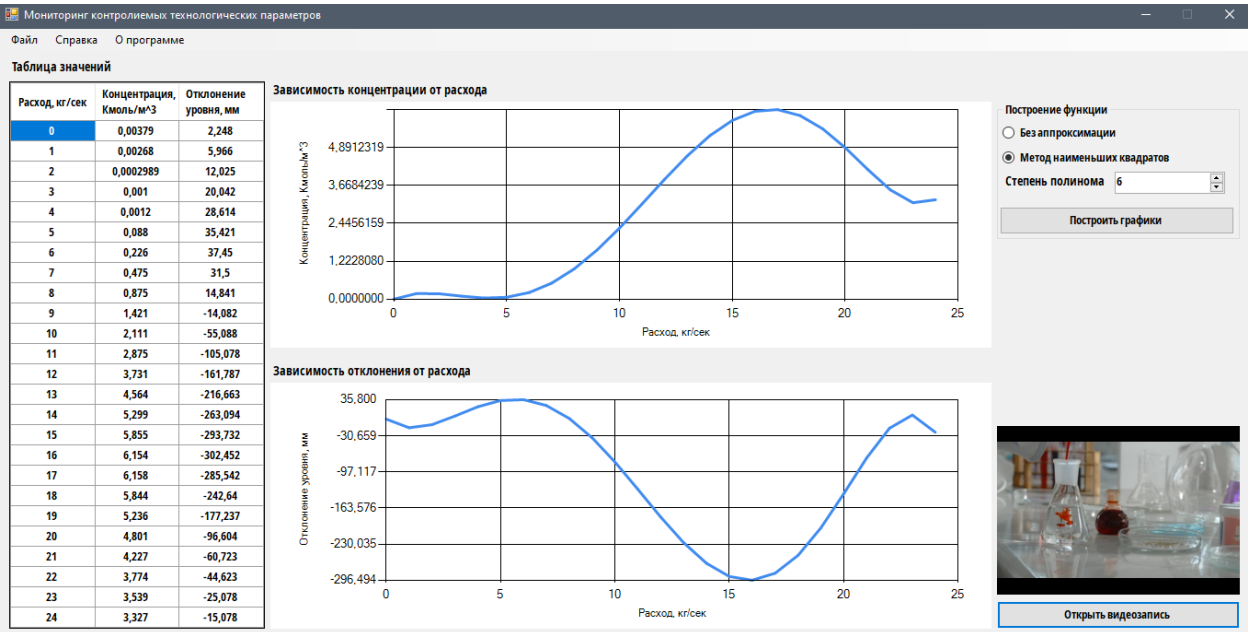


Рисунок 21 — Построение графика используя метод наименьших квадратов



Добавлены шумы, изменены выделенные ячейки.

Рисунок 22 — Построение графика без аппроксимации

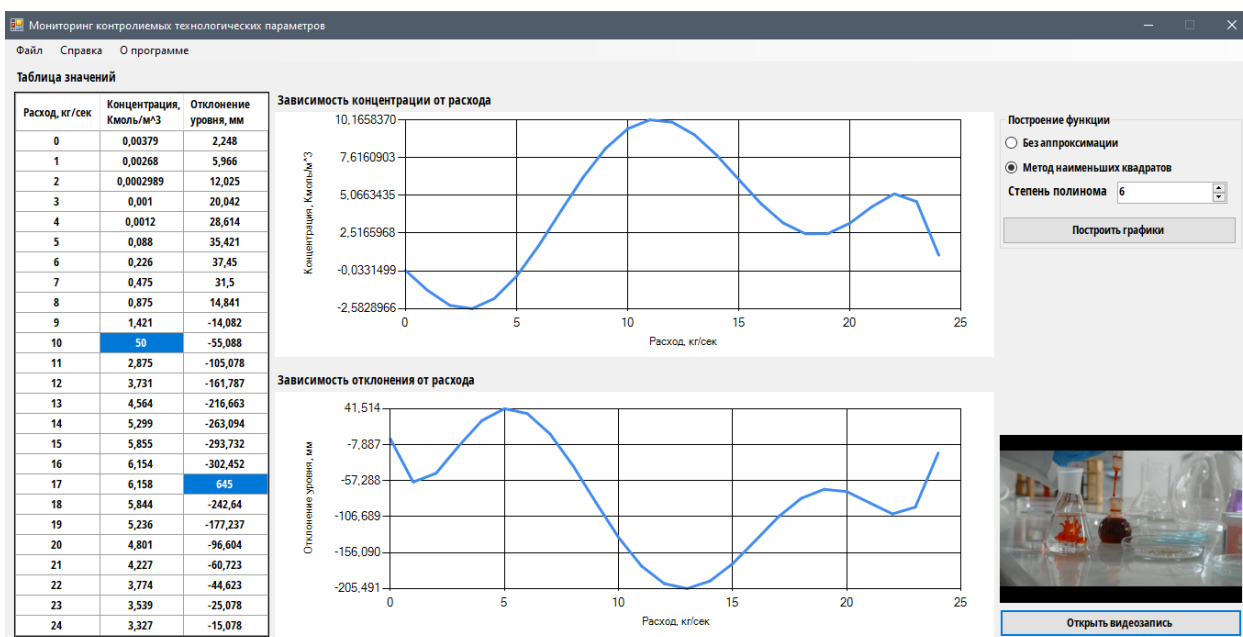


Рисунок 23 — Построение графика используя метод наименьших квадратов

2.11 Разработка эксплуатационного документа «Руководство системного программиста»

Уточнить содержание

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Петраков Ю. В. Теория автоматического управления технологическими системами / Ю. В. Петраков, О. И. Драчев. Москва: Машиностроение, 2009. 336 с.
- 2 НТЦ Комплексные системы: официальный сайт : сайт. — Санкт-Петербург, 2017 — . — URL: <http://www.complexsystems.ru/smtp-eon.html> (дата обращения 24.09.2021)
- 3 НТЦ Комплексные системы: проект системы учета тепла: сайт. — Санкт-Петербург, 2017 — . — URL: <http://www.complexsystems.ru/heat-tumen.html> (дата обращения 24.09.2021)
- 4 Солонин, Е.Б. Современные методики разработки информационных систем: учебное пособие / Е.Б. Солонин. — Екатеринбург: УрФУ, 2015. — 45 с.
- 5 Щупак, Ю.А. Win32 API. Эффективная разработка приложений / Ю.А. Щупак. — Санкт-Петербург: Питер, 2011. — 572 с. — ISBN: 978-5-388-00301-0
- 6 Давыдов, В.Г. Visual C++. Разработка Windows-приложений с помощью MFC и API функций / В.Г. Давыдов. — Санкт-Петербург: БХВ-Петербург, 2008. — 576 с. — ISBN: 978-5-9775-0157-6
- 7 Деннинг, А. ActiveX для профессионалов / А. Деннинг. Пер. с англ. Е. Матвеев. — Санкт-Петербург: Питер, 2010. — 624 с. — ISBN 5-314-00056-3.
- 8 Таненбаум, Э. Современные операционные системы / Э. Таненбаум. Пер. с англ. — Санкт-Петербург: Питер, 2015. — 1120 с. — ISBN: 978-5-4461-1155-8
- 9 Портал IT технологий: информационный портал : сайт — Санкт-Петербург, 2021 — . — URL: <https://www.ixbt.com/soft/installers-2.shtml> (дата обращения 02.10.2021)
- 10 Русинович, М. Внутреннее устройство Microsoft Windows: Основные подсистемы ОС / М. Русинович, Д. Соломон, А. Ионеску. — 6-е изд. — Санкт-Петербург: Питер, 2014. — 672 с.
- 11 Гульятеев, А.К. Help. Разработка справочных систем. Учебный курс / А.К. Гульятеев. Санкт-Петербург: Питер, 2004. — 272 с. — ISBN 5-94723-921-3