

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«__» _____ 2024 г.

Разработка программы для автоматизированного расчета синхронного двигателя
на постоянных магнитах для беспилотных летательных аппаратов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ П.О. Шабуров
«__» _____ 2024 г.

Автор работы,
студент группы КЭ-406
_____ Д.В. Рогов
«__» _____ 2024 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
«__» _____ 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
« ____ » _____ 2024 г.

ЗАДАНИЕ
на выпускную квалификационную работу бакалавра
студенту группы КЭ-406
Рогову Дмитрию Владимировичу
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Разработка программы для автоматизированного расчета синхронного двигателя на постоянных магнитах для беспилотных летательных аппаратов» утверждена приказом по университету от «22» апреля 2024 г. № 764-13/12
2. **Срок сдачи студентом законченной работы:** 01 июня 2024 г.
3. **Исходные данные к работе:** параметрическая модель тягового синхронного электродвигателя с возбуждением от высококоэрцитивных постоянных магнитов.
 - 3.1. Входные данные программы:
 - выходная мощность двигателя;
 - частота вращения;
 - напряжение питания;
 - наружный диаметр машины.

3.2. Выходные данные:

3.2.1. трехмерная модель двигателя в САПР SolidWorks;

3.2.2. таблица с основными параметрами двигателя:

- КПД;
- масса;
- удельная мощность;
- наружный диаметр ротора;
- внутренний диаметр ротора;
- высота спинки ротора;
- высота спинки статора;
- ширина зубца статора;
- длина двигателя.

3.3. Программа должна осуществлять подбор оптимальных характеристик, изменяя следующие параметры двигателя:

- высота спинки статора в пределах от 1 до 10 мм;
- высота спинки ротора в пределах от 1 до 10 мм;
- ширина зубца статора в пределах от 1 до 10 мм;
- высота магнита в пределах от 1 до 10мм;
- длина магнитного провода в пределах от 10 до 45 мм.

3.4. Ограничения на характеристики двигателя:

- индукция в спинке ротора не более 1,5 Тл;
- индукция в зубце статора не более 1,7 Тл;
- индукция в спинке статора не более 1,5 Тл;
- отклонение величины потока в нейтральном сечении магнита от магнитного потока не более 1%.

4. Перечень подлежащих разработке вопросов:

1. Аналитический обзор.
2. Перенос алгоритма расчета в программную среду.
3. Создание макроса для реализации трехмерной модели.

4. Исследовательские испытания.

Дата выдачи задания: 1 декабря 2023 г.

Руководитель работы _____ / П.О. Шабуров /

Студент _____ / Д.В. Рогов /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	20.12.2023	
Перенос алгоритма расчета в программную среду с последующей автоматизацией и оптимизацией алгоритма с возможностью нахождения оптимальной модели по параметрам КПД и массы	03.03.2024	
Создание макроса для реализации трехмерной модели в системе автоматизированного проектирования SolidWorks	24.03.2024	
Проведение исследовательских испытаний	07.04.2024	
Компоновка текста работы и сдача на нормоконтроль	08.04.2024	
Подготовка презентации и доклада	09.04.2024	

Руководитель работы _____ / П.О. Шабуров /

Студент _____ / Д.В. Рогов /

АННОТАЦИЯ

Д.В. Рогов. Разработка программы для автоматизированного расчета синхронного двигателя на постоянных магнитах для беспилотных летательных аппаратов. - Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2024, 43 с., библиогр. список – 29 наим.

В рамках выпускной квалификационной работы производится проектирование и разработка программного обеспечения – программы для расчета характеристик двигателя. Процесс выполнения выпускной квалификационной работы включает в себя решение следующих вопросов: обзор аналогов, обзор средств реализации, проектирование архитектуры, разработка программного решения, а также тестирования программного решения.

Основная цель разрабатываемого приложения – упростить расчет параметрических характеристик для синхронного двигателя и получение линейки двигателей.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1. АНАЛИТИЧЕСКИЙ ОБЗОР.....	9
1.1. АНАЛИТИЧЕСКИЙ ОБЗОР ЛИТЕРАТУРЫ.....	9
1.2. ОБЗОР АНАЛОГОВ	16
2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ.....	19
2.1. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	19
2.2. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ.....	19
2.3. ОБЩЕЕ ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ	20
2.4. ПРОЕКТИРОВАНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА	22
3. РАЗРАБОТКА ПРОГРАММЫ	24
3.1. РАЗРАБОТКА АЛГОРИТМА	24
3.2. ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ	27
3.3. ФАЙЛОВАЯ СТРУКТУРА ПРОЕКТА	28
3.4. РАЗРАБОТКА ТРЕХМЕРНОЙ МОДЕЛИ.....	28
4. ТЕСТИРОВАНИЕ ПРОГРАММЫ.....	31
4.1. ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНОСТИ.....	31
4.2. ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ	33
4.3. ТЕСТИРОВАНИЕ СОВМЕСТИМОСТИ	34
ЗАКЛЮЧЕНИЕ	37
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	38
ПРИЛОЖЕНИЕ А ПЛАН ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЯ.....	41
ПРИЛОЖЕНИЕ Б ЧАСТЬ КОДА РАСЧЕТА	42

ВВЕДЕНИЕ

Синхронные двигатели, как важный элемент многих промышленных установок, имеют ключевое значение для точности и эффективности процессов. Они используются в ситуациях, где необходимо точное соблюдение скорости вращения, что критично для таких отраслей, как энергетика, промышленное производство и медицинское оборудование. При этом точность расчета их параметров напрямую влияет на эффективность и надежность всей системы.

Традиционные методы расчета параметров синхронных двигателей часто оказываются весьма затратными и подверженными ошибкам из-за человеческого фактора. С учетом этого разработка специализированного программного обеспечения для автоматизации этих расчетов является актуальной задачей.

Основной целью данной квалификационной работы является разработка и внедрение программного продукта для автоматизации расчета ключевых параметров синхронных двигателей. Задачи разработки включают себя:

1. Аналитический обзор.
2. Перенос алгоритма расчета в программную среду.
3. Создание макроса для реализации трехмерной модели.
4. Исследовательские испытания.

В результате выполнения выпускной квалификационной работы будет представлена программа, которая позволит значительно ускорить и упростить расчет синхронного электродвигателя.

1. АНАЛИТИЧЕСКИЙ ОБЗОР

1.1. АНАЛИТИЧЕСКИЙ ОБЗОР ЛИТЕРАТУРЫ

Моделирование, как метод познания, используется давно и подразумевает исследование основных закономерностей и особенностей поведения каких-либо процессов, явлений или других реальных объектов с помощью их моделей [1].

Математическое и физическое моделирование составляют два столпа, на которых удерживается вся современная научная и инженерная мысль. Самым важным и наиболее распространенным предназначением моделей является их применение при изучении и прогнозировании поведения сложных процессов и явлений. В идеальном случае модель отражает только главные свойства объекта моделирования и не отражает несущественные свойства. Однако модель, как самостоятельный объект, имеет и свои собственные свойства, которые никак не связаны с моделируемым объектом [2]. Однако использование упрощенной модели, отражающей отдельные черты исследуемого объекта, позволяет яснее увидеть взаимосвязь причин и следствий, входов и выходов, быстрее сделать необходимые выводы, принять правильные решения [3].

Среди целей моделирования можно выделить следующие [4]: понять, как устроен конкретный объект: какова его структура, внутренние связи, основные свойства, законы развития, саморазвития и взаимодействия с окружающим миром; научиться управлять объектом или процессом, определить наилучшие способы управления при заданных целях и критериях; прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействий на объект. Из сказанного выше следует, что моделей без оригиналов не существует.

Оригиналами моделей могут быть [5]:

- объекты (ракета, человек, атом, лицо, галактика, планета, животное, здание, космическое тело);
- процессы (изменение климата, война, спад экономики, инфляция, эпидемия, переговоры, судебный процесс);

- явления природы (землетрясение, цунами, северное сияние, снегопад, град, извержение вулкана, молния, гроза).

Среди форм представления моделей можно выделить следующие [6]:

- инвариантная — запись соотношений модели с помощью;
- традиционного математического языка безотносительно к методу решения уравнений модели;
- аналитическая — запись модели в виде результата аналитического решения исходных уравнений модели;
- алгоритмическая — запись соотношений модели и выбранного численного метода решения в форме алгоритма;
- схемная (графическая) — представление модели на некотором графическом языке (например, язык графов, эквивалентные схемы, диаграммы и т.п.);
- физическая — представление моделей как уменьшенных копий реальных аппаратов и технологических процессов;
- аналоговая — модели, основанные на подобию явлений, имеющих различную физическую природу, но описываемых одинаковыми математическими уравнениями.

Математические модели – это абстрактные конструкции, состоящие из уравнений и неравенств, которые воплощают в себе формализованный язык для описания и анализа реальных систем и явлений. Они обладают способностью упрощать сложные концепции и делать предсказания о будущем поведении объектов и процессов. Основными требованиями, предъявляемыми к математическим моделям, являются требования адекватности, универсальности и экономичности [7].

Натурная модель, физическая модель — это модель-копия объекта, выполненная из другого материала, в другом масштабе, с отсутствием ряда деталей [8]. Они могут быть представлены в форме простых формул, комплексных компьютерных программ или даже физических макетов, которые демонстрируют реальные физические процессы и феномены.

Эти модели находят широкое применение в самых разных областях – от космической физики и климатологии до медицины и экономики. С их помощью можно выявить основные тенденции, предсказать поведение систем при различных условиях, разработать оптимизационные стратегии и создать надежные прогнозы, необходимые для научных исследований, технического развития и планирования.

Таким образом, были рассмотрены основные виды моделей. Данная выпускная квалификационная работа представляет собой реализацию математической и графической модели, так что дальнейший обзор будет посвящен именно этим моделям.

Модели, используемые в задачах математического моделирования, можно условно разделить на три основных класса: аналитическое, имитационное и нечёткое. В первом случае устанавливаются формульные, аналитические зависимости между параметрами системы. Для описания зависимостей разработан ряд алгебраических, дифференциальных, игральных и других уравнений. В аналитических задачах решаются вопросы управления. Для задач с большим количеством факторов, разработаны методы имитационного моделирования [9]. Примерная схема представлена на рисунке 1 [10].



Рисунок 1 – Виды математического моделирования

Математические модели используют уравнения и математические выражения для описания систем или явлений. Они могут быть использованы для представления различных характеристик системы и предсказания их поведения.

Электродвигатели также могут быть представлены различными моделями для их анализа и проектирования.

Модель параметрического электродвигателя. Эта модель использует параметры для описания характеристик электродвигателя, таких как напряжение, ток, скорость вращения и нагрузка. Эта модель позволяет предсказывать работу электродвигателя в зависимости от его параметров и внешних условий. Пример параметрической модели представлен на рисунке 2 и 3.

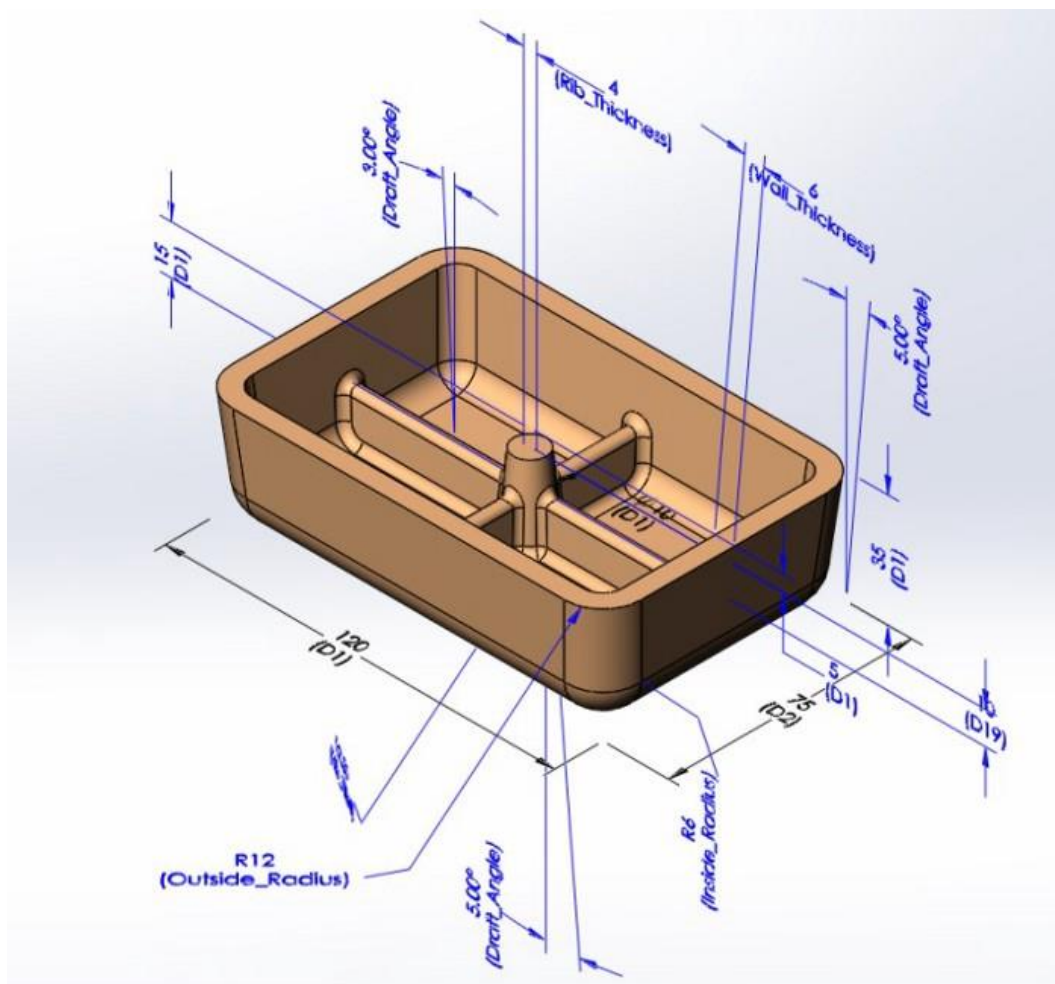


Рисунок 2 – Пример параметрической модели в программе SolidWorks

Name	Value / Equation	Evaluates to	Comments
Global Variables			
Add global variable			
Features			
Add feature suppression			
Dimensions			
D1@Sketch1	120.00mm	120.00mm	
D2@Sketch1	75.00mm	75.00mm	
D1@Base	35.00mm	35.00mm	
Draft_Angle@Base	5deg	5deg	
Wall_Thickness@Shell1	6.00mm	6.00mm	
Inside_Radius@Fillet1	6.00mm	6.00mm	
Outside_Radius@Fillet2	12.00mm	12.00mm	
D1@Rib Plane	15.00mm	15.00mm	
Draft_Angle@Rib	3deg	3deg	
Rib_Thickness@Rib	4.00mm	4.00mm	
D1@Boss Plane	5.00mm	5.00mm	
D1@Sketch4	10.00mm	10.00mm	
Draft_Angle@Cylinder Boss	5deg	5deg	
D19@Cylinder Boss	10.00mm	10.00mm	
Rib_Fillet@Fillet3	2.50mm	2.50mm	

Automatically rebuild Angular equation units: **Radians** Automatic solve order
 Link to external file:

Рисунок 3 – Таблица глобальных переменных и уравнений

Модель конечно-элементного анализа электродвигателя: эта модель использует метод конечных элементов для анализа структурных, тепловых и электромагнитных характеристик электродвигателя. Она позволяет учитывать сложное взаимодействие различных физических процессов внутри двигателя. Пример такой модели представлена на рисунке 4.

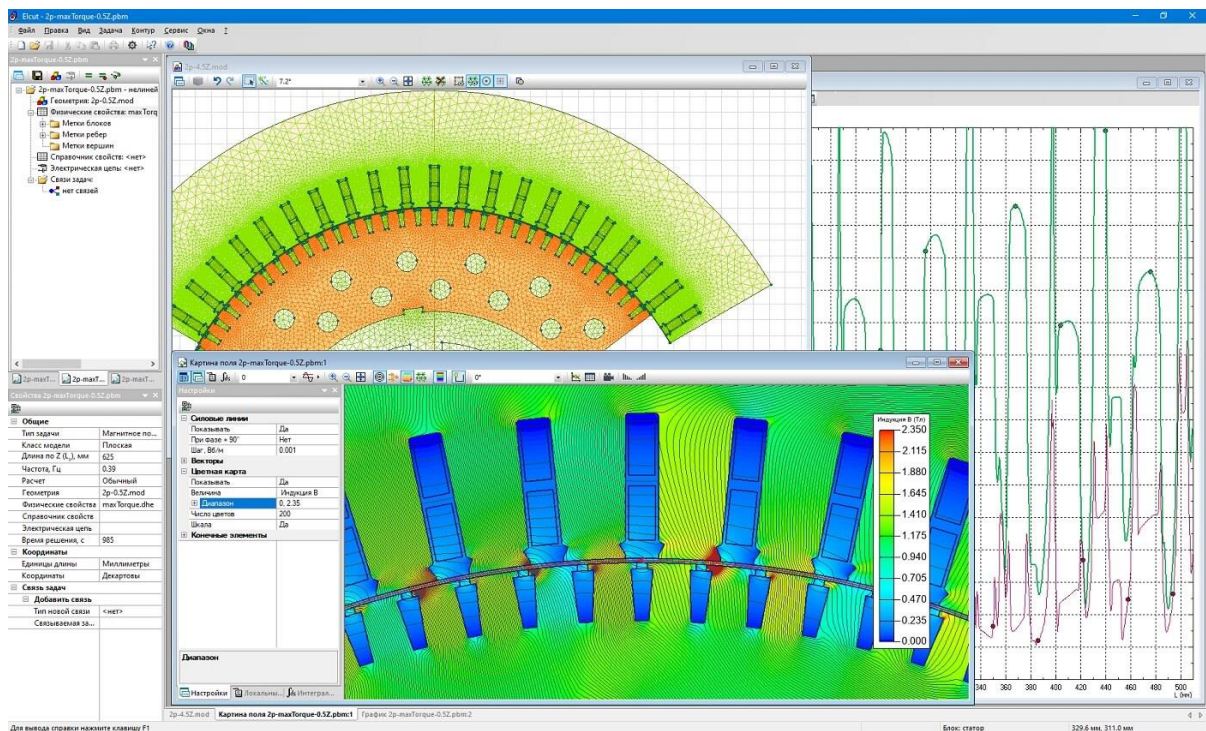


Рисунок 4 – Пример конечно-элементной модели

Модель электромагнитного поля в электродвигателе: эта модель фокусируется на анализе электромагнитного поля внутри электродвигателя, включая распределение магнитных полей, индуктивность и силы, действующие на вращающиеся части.

Для реализации программы в данной ВКР была выбрана параметрическая модель двигателя. Параметрическое моделирование – проектирование, в основе которого лежит использование параметров элементов, являющихся составной частью общей модели, а также соотношения между этими параметрами, определяющие геометрическую форму модели [11]. Существенное отличие параметрического моделирования от обычного двумерного или трехмерного прежде всего в том, что модель задается математическими уравнениями в виде изменяющихся функций [12]. Параметрическая модель же была выбрана так как расчет на основе этой модели легко реализовать на ЭВМ [13], в отличие от расчета энергетическим методом, где требуется постоянное обращение к данным, представленным в виде графиков и номограмм [14]. В тоже время, используемая модель является «быстрой», в отличие от конечноэлементной модели [15], что позволяет рассмотреть множество вариантов конструкции и организовать оптимизацию двигателя. Моделирование электромагнитного поля сложнее параметрической модели и требует больше времени, ресурсов и вычислительных мощностей. Поэтому и была выбрана параметрическая модель двигателя.

Для анализа электромагнитных процессов и проектирования конструкции тяговых электродвигателей требуется иметь первоначальную параметрическую и геометрическую модели. Характерной особенностью современных систем автоматического проектирования является также то, что они позволяют формировать в своих средах параметрические модели [16] проектируемых объектов. В этом случае модель однозначно определяется набором переменных, подлежащих варьированию в процессе проектирования, а некоторые изначально заданные свойства модели при этом остаются неизменными [17]. Однако создание таких моделей может быть трудоемким процессом, который требует много времени, ресурсов и знаний [18-22]. Далее в данном анализе будут приведены способы создания геометрической модели и выбор системы САПР.

Solidworks является одним из ведущих программных продуктов в области трехмерного моделирования. Однако, несмотря на обширный набор инструментов и функций, которые предлагает Solidworks, иногда возникает необходимость в автоматизации некоторых задач. В этом контексте API и макросы Solidworks играют ключевую роль [23-24].

API Solidworks — это COM-программный интерфейс к программному обеспечению Solidworks [25]. Функции в API предоставляют программистам прямой доступ к функциональности Solidworks. API содержит сотни функций, которые можно вызывать из таких языков, как Visual Basic (VB), Visual Basic for Applications (VBA), VB.NET, C++, C#, или файлов макросов Solidworks [26]. Сами макросы представляют собой сценарии, которые позволяют автоматически запускать операции в программном обеспечении Solidworks [27]. Макросы могут быть созданы и запрограммированы вне программного обеспечения Solidworks или в самом обеспечении, когда макрос фиксирует последовательность действий и команд при их выполнении в программном обеспечении Solidworks.

API и макросы Solidworks представляют собой мощные инструменты для автоматизации и упрощения рабочих процессов в Solidworks. Они позволяют пользователям создавать настраиваемые сценарии для выполнения повторяющихся задач, экономя время и усилия. Понимание того, как создавать и использовать API и макросы в Solidworks, является важным навыком для любого пользователя Solidworks [28].

Для построения параметрической модели двигателя была разработана программа для расчета основных характеристик двигателя, таких как внутренние и внешние диаметры ротора, высоты спинок ротора и статора, число пар полюсов ротора, масса ротора и КПД двигателя. Программа позволяет подобрать оптимальные значения параметров для целой линейки двигателей. Однако, в дальнейшем эти параметры требуется преобразовать в 3D модель для дальнейшего ее анализа.

Создание предварительной модели двигателя и ее последующая модификация с помощью конфигураций. Метод удобен тем, что если имеется уже

готовые модели, то не требуется воссоздавать модель заново, а лишь модифицировать модель для использования модификаций. Также есть возможность создать конфигурации для нескольких вариантов. Также конфигурация представляет собой Excel таблицу, которую можно автоматически создать, а затем уже редактировать, что крайне удобно. Также конфигурация имеет полный требуемый функционал для изменения готовой трехмерной модели.

Создание макроса для проектирования детали. Этот подход позволяет полностью контролировать процесс создания детали, включая размеры, количество элементов и даже форму детали. Однако для написания макроса, помимо знания SolidWorks, требуется также знание языка программирования. Несмотря на то, что это может быть сложнее, чем просто создание модели, из-за необходимости написания большого количества кода без возможности быстрого просмотра результата.

В данном случае выбор конфигурации оправдан, потому что на данный момент уже имеются готовые детали, которые можно модифицировать, путем изменения последовательности действий построения и установки нужных размеров. Макросы можно использовать для автоматического импорта конфигураций в модели, что значительно ускорит процесс.

1.2. ОБЗОР АНАЛОГОВ

На данный момент единственным аналогом разрабатываемой программы является программа Eldvig, разрабатываемая компанией ИнтерБалт.

Программа Eldvig предназначена для расчета [29]:

- обмоток трехфазных асинхронных электродвигателей мощностью от 1 до 100 кВт и выше;
- обмоток фазных роторов;
- обмоток якорей коллекторных машин (электродвигателей и генераторов) постоянного тока мощностью до 10 кВт;

- обмоток якорей и обмоток возбуждения микродвигателей постоянного и переменного тока мощностью до 1 кВт.

Также Eldvig имеет следующий функционал:

- проверить правильность расчета существующей обмотки трехфазного асинхронного электродвигателя;
- смоделировать работу трехфазного асинхронного электродвигателя под различной, изменяемой Пользователем, нагрузкой с определением основных параметров - к.п.д., $\cos(\varphi)$, $\sin(\varphi)$, полной, активной и реактивной мощности и т.п.;
- составить смету на материалы на ремонт асинхронных электродвигателей.

На рисунке 5 представлен скриншот интерфейса программы Eldvig.

Рисунок 5 – Интерфейс программы Eldvig

Ниже в таблице 1 представлено сравнение разрабатываемой программой с Eldvig.

Таблица 1 – Сравнение разрабатываемой программы с Eldvig

Характеристика сравнения	Eldvig	Разрабатываемая программа
Кроссплатформенность	Да	Да
Вид приложения	Desktop	Web
Характеристики двигателя	Не все требуемые	Все требуемые
Несколько вариантов двигателя	Расчет только одного варианта	Целая линейка двигателей
Возможность экспорта в Excel	Отсутствует	Присутствует
Возможность создания трехмерной модели	Отсутствует	Присутствует

Выводы по первой главе:

1. Проведен анализ видов различных моделей двигателя.
2. Данная выпускная квалификационная работа представляет собой разработку математической и графической модели. В рамках этого исследования была выбрана параметрическая модель двигателя, основанная на использовании параметров элементов, входящих в общую модель, а также на определении соотношений между этими параметрами, которые определяют геометрическую форму модели.
3. Для создания трехмерной модели будет использоваться API SolidWorks, который удобен для взаимодействия с САПР и обеспечит полный контроль при автоматизированном создании трехмерной модели.
4. Был проведен обзор аналогов с выявлением преимуществ разрабатываемой программы.

2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

2.1. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

В рамках проектирования были выявлены функциональные требования к системе, которые определяют поведение системы во время обработки информации вне зависимости от ее реализации. Для разрабатываемой программной системы выявлены следующие функциональные требования:

1. Ввод параметров, таких как выходная мощность двигателя, частота вращения, напряжения питания, наружный диаметр.
2. Результатом работы программы является таблица с возможностью сортировки по заданным характеристикам.
3. Иметь возможность скачать таблицу с характеристиками для ее дальнейшего открытия в Excel.
4. Иметь возможность преобразовать характеристики в трехмерную модель для программы SolidWorks.

К разрабатываемой системе имеются следующие нефункциональные требования

1. Читаемый интерфейс – высокий контраст, ненагруженность интерфейса, вся информация должна помещаться на одной странице и интерфейс должен быть интуитивно понятным.
2. Наличие серверной части для доступа на множестве компьютеров (не более 10 одновременных пользователей).
3. Высокая производительность (выполнение расчета должно занимать менее 5 секунд для подсчета 1000 параметров).

2.2. ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Для проектирования системы был использован язык графического описания для объектно-ориентированного программирования UML. Была построена модель взаимодействия внешнего актера с системой веб-приложения в виде диаграммы

вариантов использования. В ходе анализа требований к разрабатываемой системе были выявлены следующие варианты использования, показанные на рисунке 6.

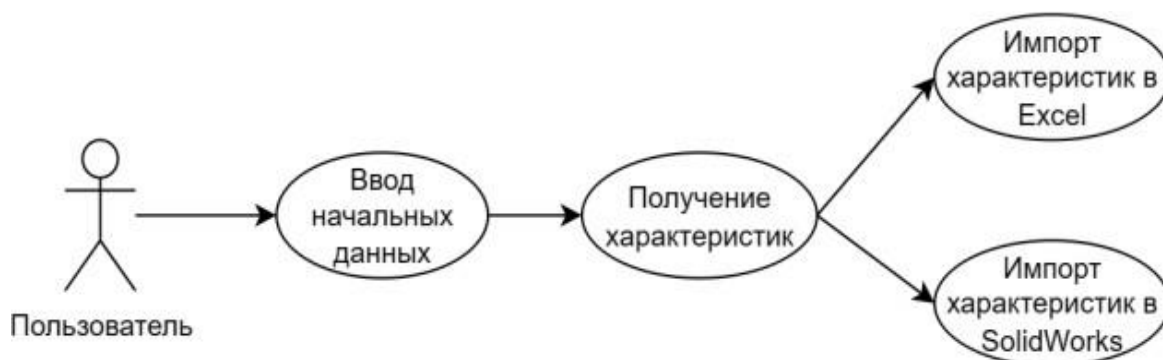


Рисунок 6 – Варианты использования программы

2.3. ОБЩЕЕ ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ

Так как для программы не требуется база данных, то рациональным является использованием двухуровневой архитектуры клиент-сервер.

Двухуровневая архитектура клиент-сервер — это модель распределенной сетевой системы, которая разделяет функции между двумя независимыми приложениями: клиентом и сервером.

Клиент – это приложение пользовательского уровня, которое взаимодействует непосредственно с пользователем, предоставляет пользовательский интерфейс и обрабатывает бизнес-логику на стороне клиента.

Сервер – это приложение, которое работает на центральном хосте (сервере) и обеспечивает сервисы, доступ к данным и функции управления ресурсами, используемые клиентом.

Уровень клиента (Presentation Layer или Client Tier):

1. Пользовательский интерфейс (UI) – обеспечивает ввод и отображение данных пользователем.
2. Прикладная логика – выполняет обработку данных, требуемую пользователем (может включать в себя некоторую часть бизнес-логики).
3. Сетевое взаимодействие – функции для обмена данными с сервером.

Уровень сервера (Data Layer или Server Tier):

- бизнес-логика – обработка данных согласно бизнес-правилам и процедурам. В некоторых случаях этот уровень может отсутствовать на клиенте, и вся логика централизованно обрабатывается на сервере;
- управление ресурсами – контроль за аппаратными и программными ресурсами сервера. Это включает управление транзакциями, многопоточностью, пулами соединений и т. д;
- слой данных – управляет хранением и доступом к данным, чаще всего с использованием СУБД (система управления базами данных).

Основные преимущества двухуровневой архитектуры:

- простота разработки – меньше слоев для управления и взаимодействия, уменьшается сложность системы;
- централизованное управление данными – все данные хранятся на сервере, что упрощает управление и обновление данных;

Схема архитектуры приложения представлена на рисунке 7.



Рисунок 7 – Схема двухуровневой архитектуры приложения

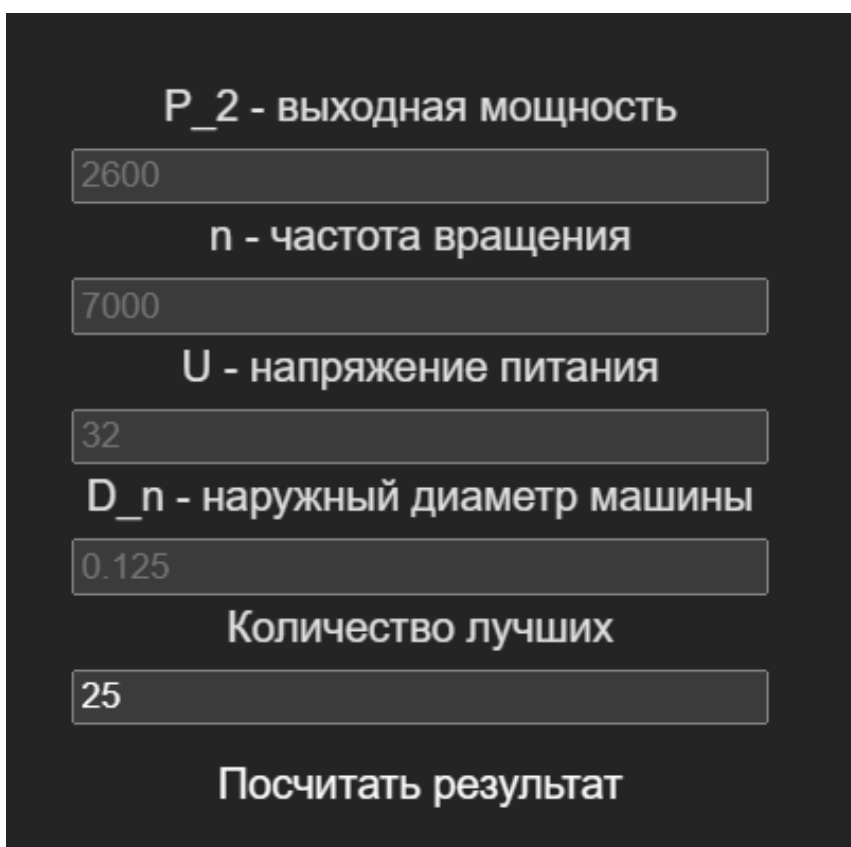
2.4. ПРОЕКТИРОВАНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Интерфейс структурно состоит из двух частей: входные данные и таблица с выходными данными.

Часть с входными данными представляет собой текстовые поля со значениями по умолчанию:

- выходная мощность;
- частота вращения;
- напряжения питания;
- наружный диаметр машины.

Также присутствует нижнее поле, где требуется указать количество наилучших вариантов. Пример интерфейса приведен на рисунке 8.



P₂ - выходная мощность
2600

n - частота вращения
7000

U - напряжение питания
32

D_n - наружный диаметр машины
0.125

Количество лучших
25

Посчитать результат

Рисунок 8 – Пример графического интерфейса

После нажатия на кнопку «Посчитать результат» ниже на этой же странице должна появиться таблица с нужными рассчитанными характеристиками. Также, если таблица слишком велика для страницы, требуется, чтобы она имела полосу

прокрутки, чтобы можно было увидеть все характеристики. Сверху таблицы находятся две кнопки: для скачивания таблицы характеристик в формате Excel и для скачивания макроса для программы SolidWorks. Пример графического интерфейса таблицы приведен на рисунке 9.

Номер	p	z	n	D _a	D _v	l _a	k _c	a _z	h _{s1}	b _{s1}	h _{z1}	d ₁	d ₂
1	8	18	7000	0.1080000	0.0780000	0.0445000	0.9000000	0.0075000	0.0005000	0.0010000	0.0010000	0.0108260	0.0078
2	8	18	7000	0.1080000	0.0780000	0.0440000	0.9000000	0.0075000	0.0005000	0.0010000	0.0010000	0.0108260	0.0078
3	8	18	7000	0.1080000	0.0780000	0.0435000	0.9000000	0.0075000	0.0005000	0.0010000	0.0010000	0.0108260	0.0078
4	8	18	7000	0.1080000	0.0780000	0.0430000	0.9000000	0.0075000	0.0005000	0.0010000	0.0010000	0.0108260	0.0078
5	8	18	7000	0.1080000	0.0780000	0.0425000	0.9000000	0.0075000	0.0005000	0.0010000	0.0010000	0.0108260	0.0078
6	8	18	7000	0.1080000	0.0790000	0.0415000	0.9000000	0.0070000	0.0005000	0.0010000	0.0010000	0.0113260	0.0083
7	8	18	7000	0.1080000	0.0790000	0.0420000	0.9000000	0.0075000	0.0005000	0.0010000	0.0010000	0.0108260	0.0078
8	8	18	7000	0.1090000	0.0800000	0.0410000	0.9000000	0.0070000	0.0005000	0.0010000	0.0010000	0.0115005	0.0083
9	8	18	7000	0.1090000	0.0800000	0.0405000	0.9000000	0.0070000	0.0005000	0.0010000	0.0010000	0.0115005	0.0083
10	8	18	7000	0.1090000	0.0800000	0.0400000	0.9000000	0.0070000	0.0005000	0.0010000	0.0010000	0.0115005	0.0083
11	8	18	7000	0.1090000	0.0800000	0.0395000	0.9000000	0.0070000	0.0005000	0.0010000	0.0010000	0.0115005	0.0083

Рисунок 9 – Пример таблицы с выходными параметрами

Целиком примерная структура всей страницы представлена на рисунке 10.

Рисунок 10 – Структура всей страницы

Выводы по второй главе

Были определены основные функциональные и нефункциональные требования к программе, были обозначены основные варианты использования программы, была описана архитектура клиент-сервер и спроектирован макет главной страницы программы.

3. РАЗРАБОТКА ПРОГРАММЫ

3.1. РАЗРАБОТКА АЛГОРИТМА

Алгоритм на естественном языке:

1. Вначале задаются стартовые значения для следующих переменных:
 - h_m - высота магнита;
 - h_c - высота спинки статора;
 - h_p - высота спинки ротора;
 - a_z - ширина зубца статора;
 - l_{Δ} - осевая длина воздушного зазор.
2. Далее осуществляется перебор переменных со следующей вложенностью:
 $l_{\Delta} \rightarrow h_m \rightarrow h_c, a_z, h_p$.
3. Осуществляется проверка, выходят ли индукции за допустимые пределы. Если выходят, зависящая переменная от индукции увеличивается.
4. Если отклонение величины потока в нейтральном сечении магнита от магнитного потока более 1%, то меняется величина потока в нейтральном потоке.
5. Если величина плотности тока в проводнике превосходит заданное значение, то увеличивается высота магнита.

Разработанный алгоритм представлен на рисунках 11–12 в виде блок-схемы.

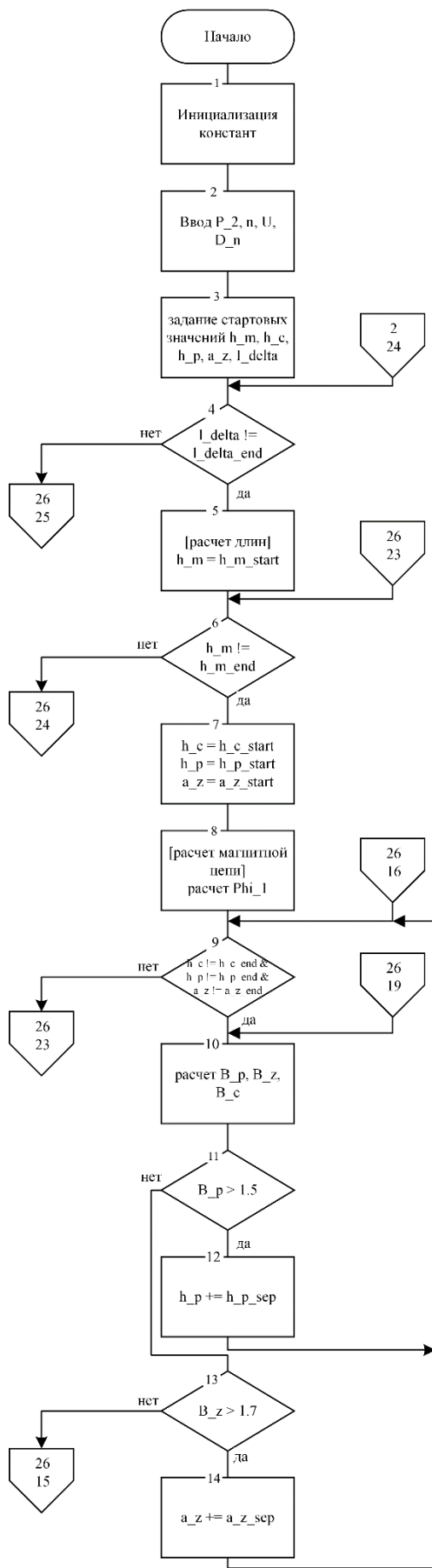


Рисунок 11 – Блок-схема алгоритм расчета

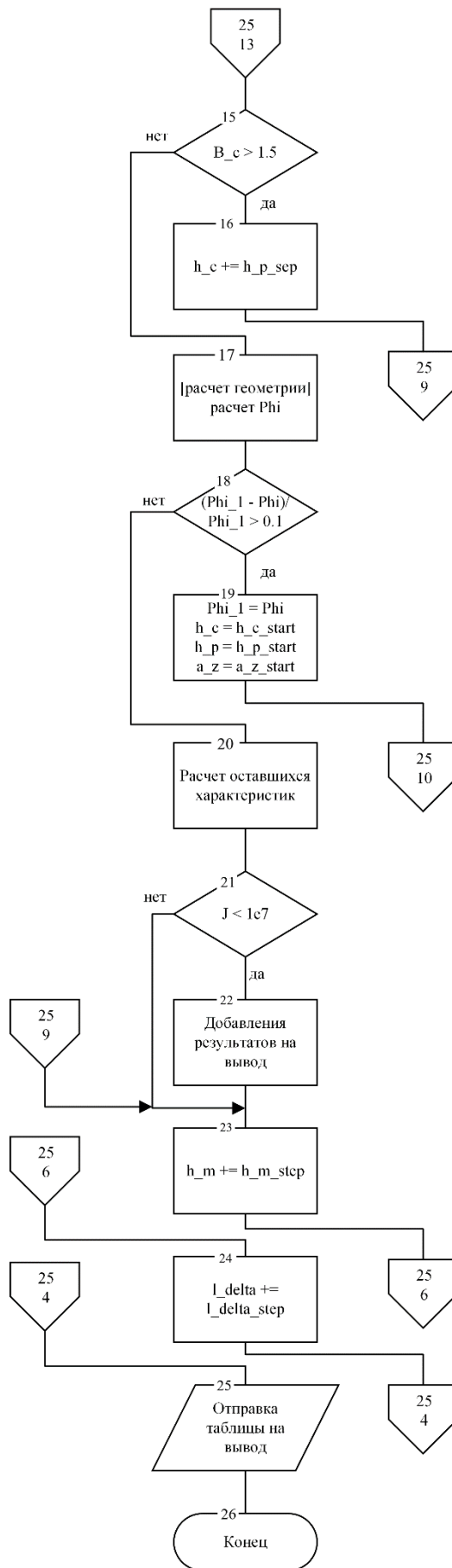


Рисунок 12 – Блок-схема алгоритма расчета

3.2. ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ

Для разработки программы был выбран язык программирования TypeScript с библиотекой React в среде Visual Studio Code. Использование React в связке с TypeScript объединяет мощь библиотеки для построения пользовательских интерфейсов с преимуществами строгой типизации и объектно-ориентированным программированием, предоставляемыми TypeScript. Вот ключевые преимущества такого сочетания:

1. Улучшенное автодополнение и проверка типов. Благодаря TypeScript редакторы могут предложить более точные предложения по автодополнению API React и ваших компонентов. Также имеется проверка типов во время компиляции: это позволяет избежать типичных ошибок, таких как неправильное использование props (свойств компонента) или состояний.
2. Безопасная работа с состоянием и пропсами. Указание точных типов для props и state ваших компонентов React помогает обеспечить их корректное использование в приложении.
3. Улучшенная поддержка IDE. Лучшие подсказки и рефакторинг: Типизированный код помогает IDE лучше понимать ваши намерения, что приводит к улучшенным функциям подсказок и рефакторинга кода.
4. Легкость масштабирования. Усиленный контроль за сложностью кода: Типизация помогает управлять сложностью, поскольку приложение растет, делая код более удобным для чтения и поддержки.
5. Лучшая поддержка в сообществе. Объединение React и TypeScript пользуется большой популярностью, благодаря чему существует обширное сообщество, готовое помочь с проблемами и вопросами.
6. Использование новейших возможностей JavaScript и TypeScript. TypeScript позволяет использовать новейшие возможности ECMAScript, а также уникальные особенности TypeScript, такие как декораторы и пространства имен.

В совокупности данные преимущества делают разработку на React в связке с TypeScript не только более безопасной и удобной, но и обеспечивают легкость в ведении и масштабировании больших проектов, помогая разработчикам сосредоточиться на создании качественного пользовательского опыта.

3.3. ФАЙЛОВАЯ СТРУКТУРА ПРОЕКТА

Для разработки приложения на была получена файловая структура, имеющая следующие основные каталоги файлов:

- info – вся документация для расчета, на основе которых был построен алгоритм расчета;
- index.html – верстка страницы;
- src – папка с основными файлами, среди которых:
 - data – информация о различных материалах в формате json;
 - components – код основной таблицы с характеристиками
 - app.tsx – ввод данных и вывод таблицы;
 - calculation.ts – код расчета;
 - main.css – дизайн интерфейса программы;
 - main.tsx – точка входа программы;
 - utils.ts – вспомогательные функции.

3.4. РАЗРАБОТКА ТРЕХМЕРНОЙ МОДЕЛИ

Была разработана трехмерная модель в SolidWorks. Для возможности модификации модели были добавлены файлы конфигурации, которые представляют собой Excel файл с возможностью добавления в него требуемых размеров. Этот файл «связывается» с моделью и, при открытии сборки, все размеры подгружаются и изменяют модель в соответствии с файлом. Сам файл генерирует программа в соответствии с рассчитанными параметрами. После, данные файлы добавляются в папку с моделью с заменой, чтобы заменить предыдущие файлы и

открывается модель сборки. Примерная сборка с ротором и статором представлена на рисунке 13.

В листинге 1 представлен код генерации файла конфигурации.

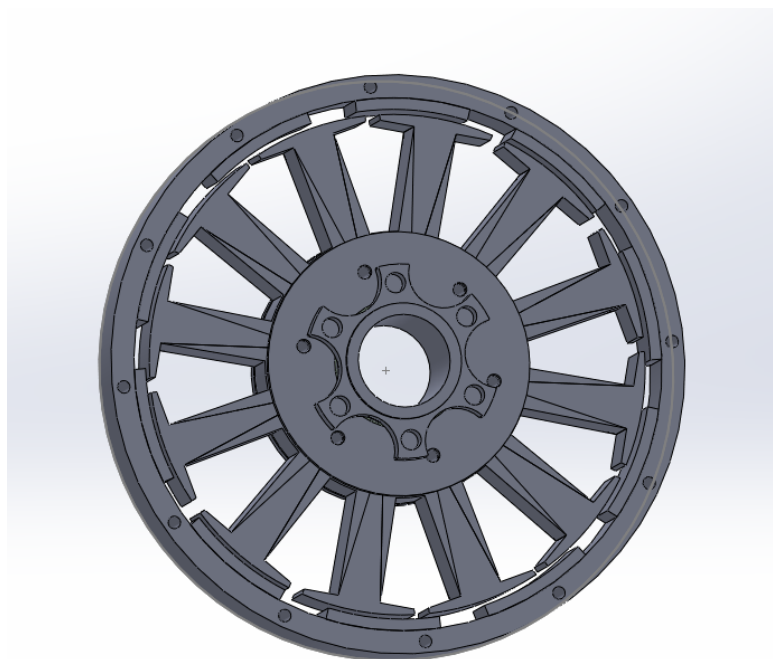


Рисунок 13 – Снимок экрана сборки в SolidWorks

В листинге 1 представлен код генерации файла конфигурации для магнита.

Листинг 1 – код создания файла конфигурации для магнита

```
const handleExportDataSolid = () => {
    const data = output.filter(o =>
selectedColumns.includes(o.id));

    const magnit = () => {
        var dataSolid = [
            ["Таблица параметров для: Магнит"],
            [, "ДиаметрМагнита@Эскиз1", "ВысотаМагнита@Эскиз1"],
        ]
        for (let i = 0; i < selectedColumns.length; i++) {
            if (i == 0){
                dataSolid.push(["По умолчанию",
                    "115",
                    (data[i].h_m * 1000).toString()])
            }else{
                dataSolid.push([(i+1).toString(),
                    "115",
                    (data[i].h_m * 1000).toString()])
            }
        }
    }
    const sheet = XLSX.utils.aoa_to_sheet(dataSolid);
```

Продолжение листинга 1

```
const workbook = XLSX.utils.book_new();
XLSX.utils.book_append_sheet(workbook, sheet, 'Лист1');
console.log(workbook.SheetNames);
Продолжение листинга 1
if(!workbook.Workbook) workbook.Workbook = {};
    if(!workbook.Workbook.Names) workbook.Workbook.Names =
[];
workbook.Workbook?.Names?.push({
    Name: "Family",
    Ref: "Лист1!$A$2"
});
console.log(workbook.Workbook?.Names);
XLSX.writeFile(workbook, "Магнит.xlsx");
}

if (selectedColumns.length <= 0) {
    return;
}

magnit()
}
```

Вид файла конфигурации представлен на рисунке 14.

Таблица параметров для: Магнит			
	ДиаметрМагнита@Эскиз1	ВысотаМагнита@Эскиз1	
По умолчанию	115	1,20	
2	120	1,50	
3	125	1,70	

Рисунок 14 – Вид файла конфигурации

Выводы по третьей главе

В данной главе был разработан алгоритм расчета характеристик двигателя, выбраны средства реализации программы, а также описана файловая структура проекта и способ создания трехмерной модели.

4. ТЕСТИРОВАНИЕ ПРОГРАММЫ

Тестирование программы проводилось в соответствии с тест планом в приложении А.

4.1. ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНОСТИ

Ввод данных

Проверка корректности ввода данных в поля таблицы. Было проведено тестирование ввода очень больших данных. Результат на рисунке 15.

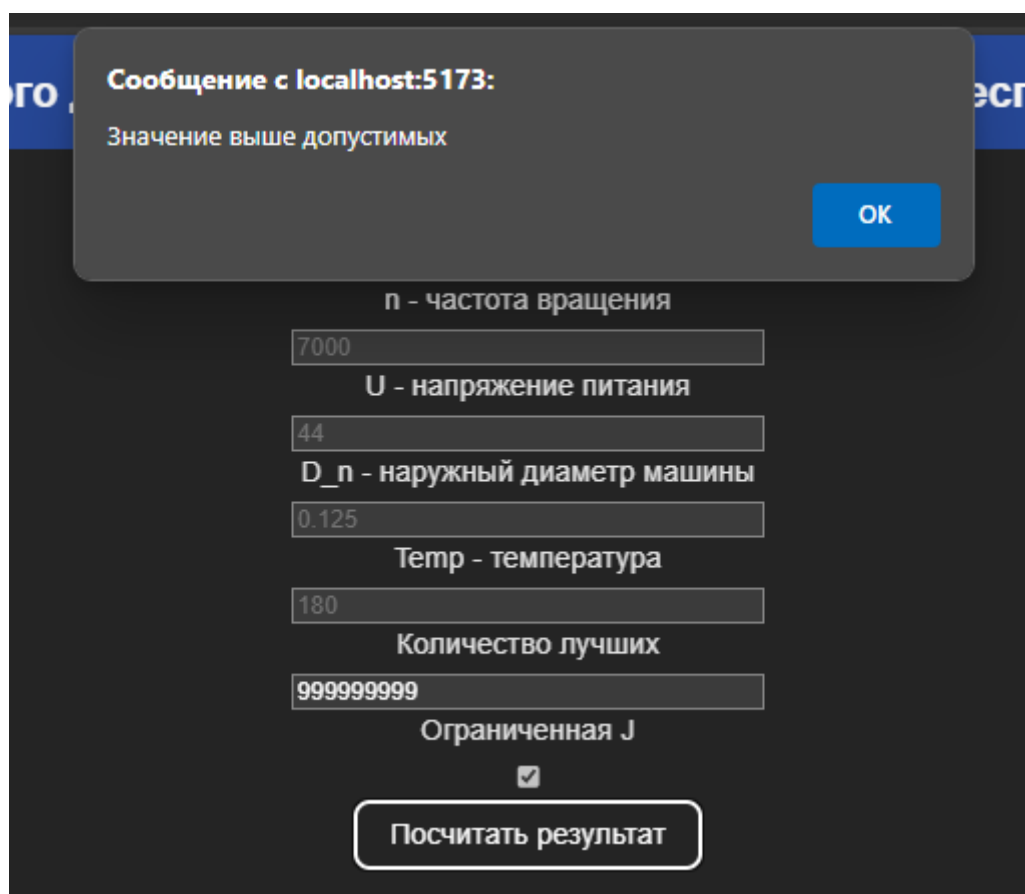


Рисунок 15 - Попытка ввода большого числа

Проверка обработки некорректных данных (например, отрицательные значения, символы, несоответствие формату). Также в форме присутствует возможность ввести только числа, что ограничивает возможность ввода букв. Результат тестирования представлен на рисунке 16.

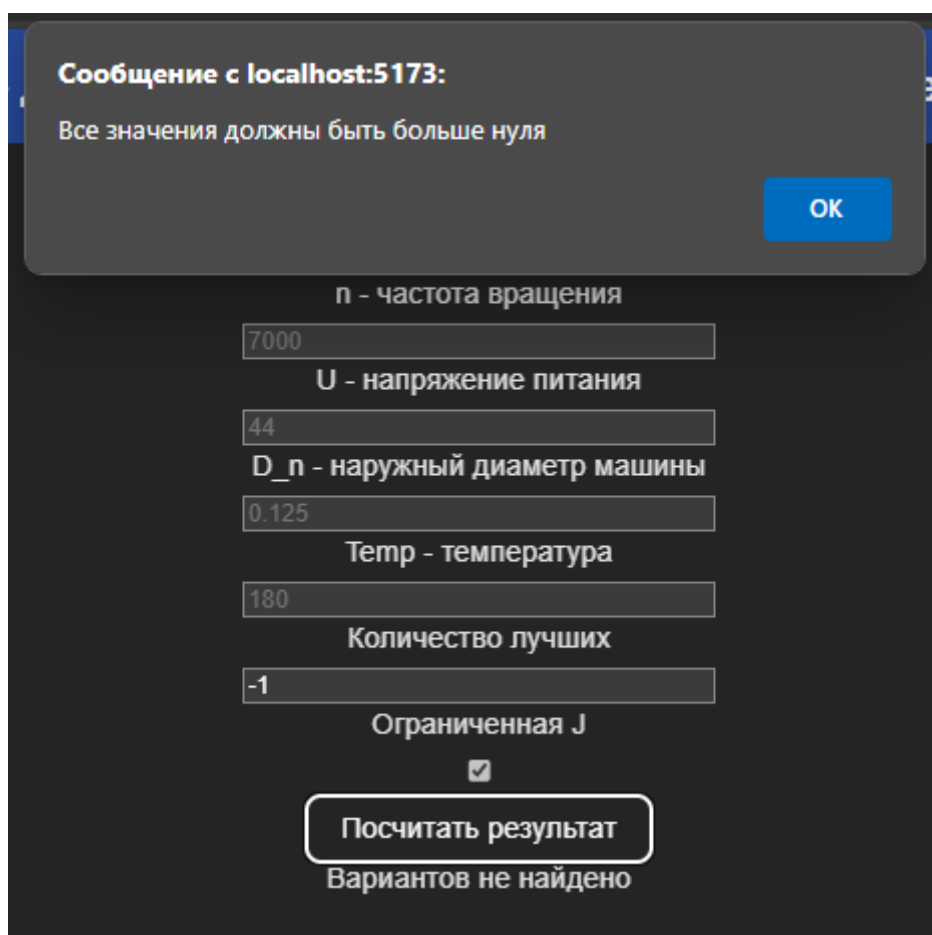


Рисунок 16 - Попытка ввода отрицательного числа

Корректность данных

Были проведены тесты сравнения заранее рассчитанных параметров и полученных в результате тестирования. Выявленные отклонения составляют не более 0,1%, что удовлетворяет требованиям заказчика.

Отображение таблицы

Проверка правильного отображения всех характеристик двигателя. Таблица отображается правильно. Результат на рисунке 17.

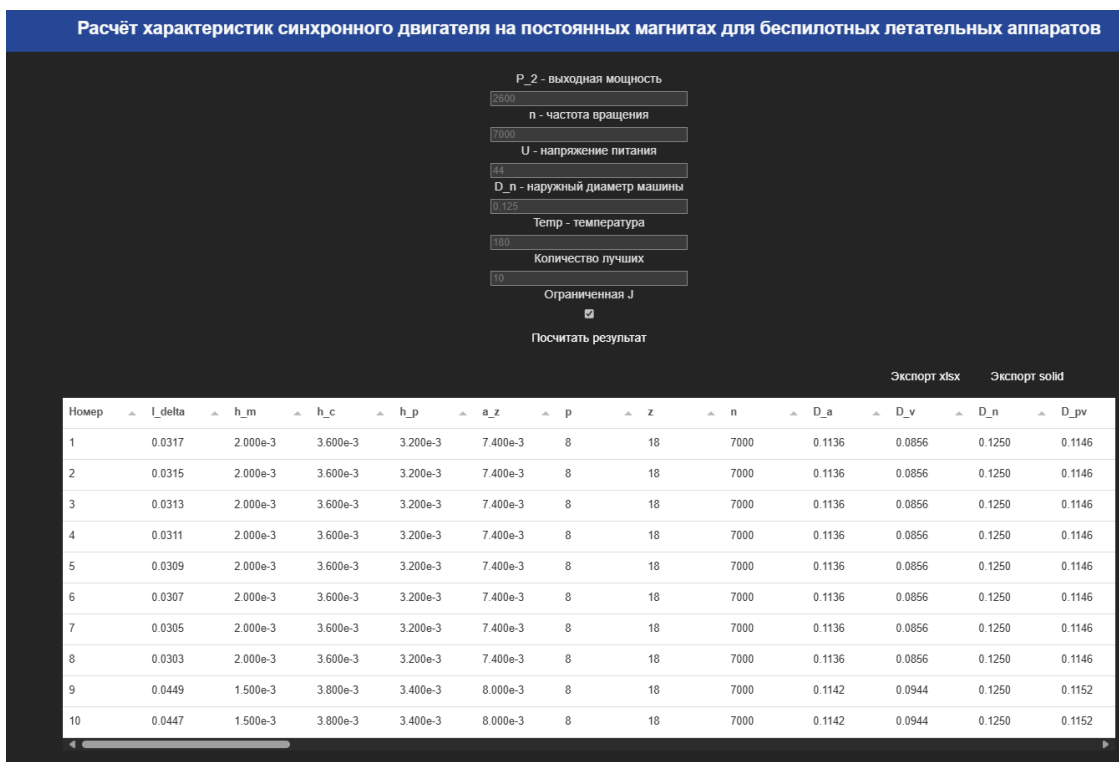


Рисунок 17 – Отображение таблицы

Проверка сортировки данных в таблице по различным столбцам. Результат на рисунке 18.

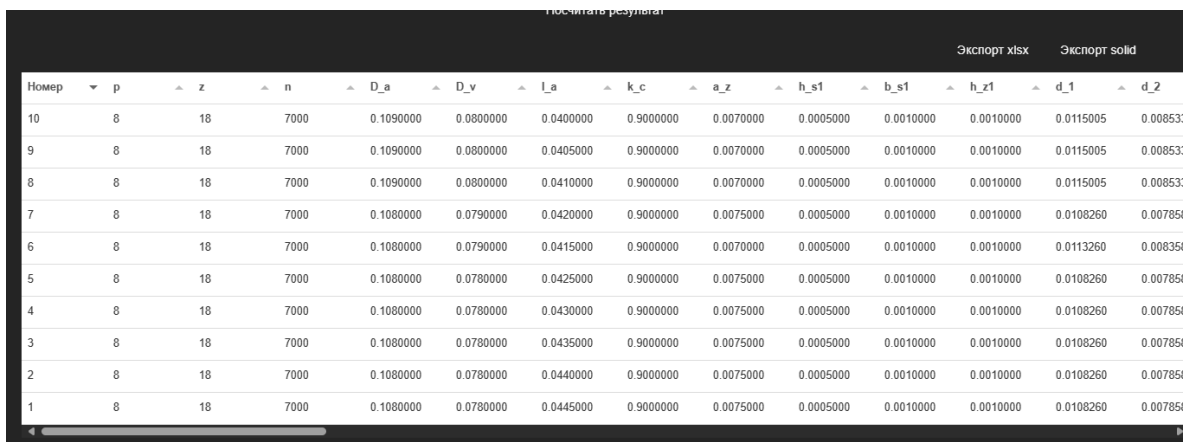


Рисунок 18 – Проверка сортировки данных

4.2. ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

Проверка времени загрузки таблицы при различных объемах данных. Загрузка с 1000 элементами занимает менее 1 секунды. Как видно из рисунка 19, время расчета и отрисовки таблицы занимает менее 1 секунды. Замеры проводились с помощью встроенных в браузер функций тестирования

производительности.

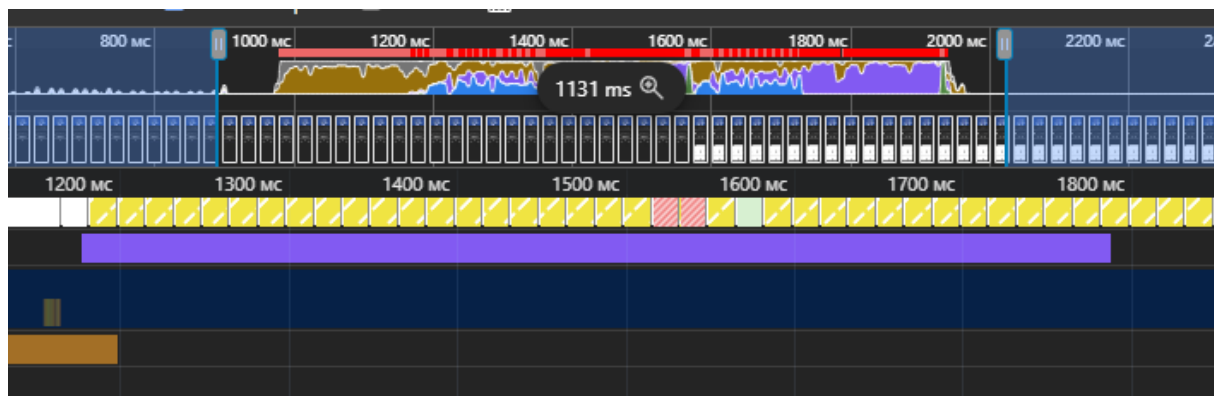


Рисунок 19 – Тестирование производительности

4.3. ТЕСТИРОВАНИЕ СОВМЕСТИМОСТИ

Проверка корректного отображения таблицы в различных веб-браузерах. Было проведено тестирования интерфейса и работоспособности в следующих браузерах: Edge, Opera GX, Firefox и Chrome. Все браузеры показали работоспособность и правильно отрисовали интерфейс. Результаты представлены на рисунках 20–23.

Расчёт характеристик синхронного двигателя на постоянных магнитах для беспилотных летательных аппаратов

Р_2 - выходная мощность
2600

n - частота вращения
7000

U - напряжение питания
44

D_n - наружный диаметр машины
0.125

Темр - температура
180

Количество лучших
10

Ограниченная J

[Экспорт xlsx](#) [Экспорт solid](#)

Номер	l_delta	h_m	h_c	h_p	a_z	p	z	n	D_a	D_v	D_n	D_pv
1	0.0317	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
2	0.0315	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
3	0.0313	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
4	0.0311	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
5	0.0309	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
6	0.0307	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
7	0.0305	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
8	0.0303	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
9	0.0449	1.500e-3	3.800e-3	3.400e-3	8.000e-3	8	18	7000	0.1142	0.0944	0.1250	0.1152
10	0.0447	1.500e-3	3.800e-3	3.400e-3	8.000e-3	8	18	7000	0.1142	0.0944	0.1250	0.1152

Рисунок 20 – Страница в браузере Edge

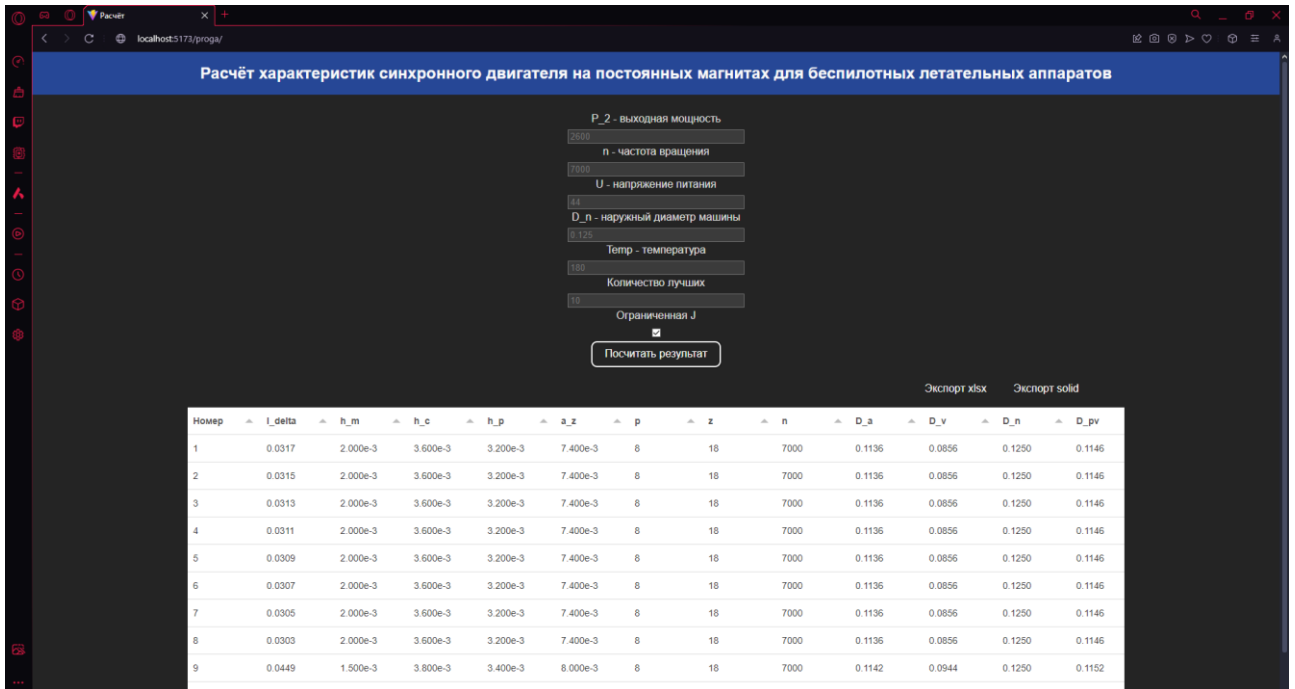


Рисунок 21 – Страница в браузере Opera GX

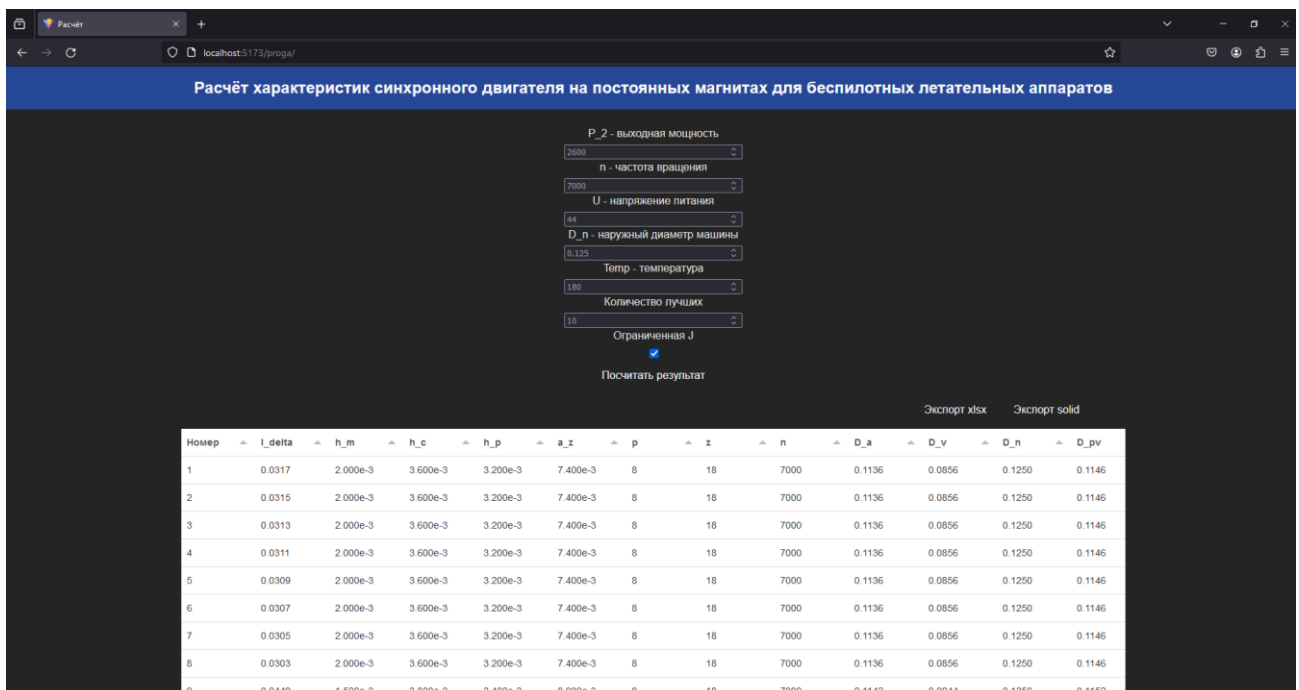


Рисунок 22 – Страница в браузере Firefox

Расчёт характеристик синхронного двигателя на постоянных магнитах для беспилотных летательных аппаратов

P₂ - выходная мощность
 2500
 n - частота вращения
 7000
 U - напряжение питания
 24
 D_п - наружный диаметр машины
 0.125
 Темр - температура
 100
 Количество лучших
 10
 Ограниченная J
 Посчитать результат

Экспорт xlsx Экспорт solid

Номер	I_delta	h_m	h_c	h_p	a_z	p	z	n	D_a	D_v	D_n	D_pv
1	0.0317	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
2	0.0315	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
3	0.0313	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
4	0.0311	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
5	0.0309	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
6	0.0307	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
7	0.0305	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
8	0.0303	2.000e-3	3.600e-3	3.200e-3	7.400e-3	8	18	7000	0.1136	0.0856	0.1250	0.1146
9	0.0449	1.500e-3	3.800e-3	3.400e-3	8.000e-3	8	18	7000	0.1142	0.0944	0.1250	0.1152

Рисунок 23 – Страница в браузере Google Chrome

Выводы по четвертой главе

В данном разделе было приведено функциональное тестирование, тестирование производительности и тестирования совместимости. Все этапы тестирования были завершены успешно, недочетов не выявлено.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была спроектирована и реализована программа для автоматизации расчета характеристик электродвигателя.

Для достижения поставленной цели были решены следующие задачи:

1. Выполнен анализ предметной области.
2. Выполнено проектирование программной среды.
3. Разработана программа для расчета характеристик двигателя.
4. Проведено тестирование программы.

Полученная программа полностью соответствует требованиям заказчика и существенно улучшает процесс расчета характеристик двигателей. Она позволяет значительно ускорить и упростить процесс вычислений, а также облегчает геометрическую проверку модели, что особо важно для производственных предприятий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Коржов, Е.Н. Математическое моделирование: учебное пособие. / Е. Н. Коржов. — Воронеж: ИПЦ ВГУ, 2012. — 74 с.
2. Королёв, А.Л. Компьютерное моделирование. Лабораторный практикум / А. Л. Королёв. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2013. — 296 с.
3. Пономарев, В.Б. Математическое моделирование технологических процессов : курс лекций / Пономарев В. Б., Лошкарев А.Б.— Екатеринбург : ГОУ ВПО УГТУУПИ, 2006.— 129 с.
4. Ахминин, В.Н. Введение в математическое моделирование: учебное пособие / В.Н. Ахминин, М.Б. Гитман, И. Э. Келлер [и др.]. — Москва : Логос, 2004. — 439 с. — Текст : непосредственный.
5. Горностаева, Т. Н. Математическое и компьютерное моделирование : учебное пособие / Горностаева Т.Н., Горностаев О.М. – М.: Мир науки, 2019. – URL: <https://izd-mn.com/PDF/50MNNPU19.pdf> (дата обращения 19.12.2023)
6. Виртуальная лаборатория математического моделирования. — URL: <http://mathmod.narod.ru/metods.htm#mm03> (дата обращения: 06.12.2023)
7. Звонарев, С.В. Основы математического моделирования: учебное пособие / С.В. Звонарев. — Екатеринбург : Изд-во Урал. ун-та, 2019. — 112 с.
8. Леонова, Н. Л. Компьютерное моделирование: курс лекций \ Н.Л. Леонова. — СПбГТУРП. - СПб., 2015. - 88 с
9. Петухов, О.А. Моделирование системное, имитационное, аналитическое: учеб. пособие / О.А. Петухов, А.В. Мороз, Е.О. Путхова. – 2- е изд., испр. и доп. – СПб.: Изд-во СЗТУ, 2008. – 288 с.
10. Маликов, Р.Ф. Основы математического моделирования : учебное пособие для вузов / Р. Ф. Маликов. — 2-е изд. — Москва : Юрайт, 2022. — 403 с.
11. Кравченко, Г.М. Дигитальная архитектура / Г.М. Кравченко, К.П. Подолько, Т.А. Литовченко // Инженерный вестник Дона. – 2017. – №4 – URL:http://www.ivdon.ru/uploads/article/pdf/IVD_161_kravchenko_N.pdf_1d7871f375.pdf (дата обращения: 19.12.2023)

12. Бабич, В.Н. О фрактальных моделях в архитектуре / В.Н. Бабич, А.Г. Кремлев // Архитектон: известия вузов. – 2010. – №. 2. – URL: https://www.gramota.net/articles/issn_1993-5552_2011_2_21.pdf (дата обращения: 21.12.2023)
13. Дерябкин, В. П. Моделирование систем на ЭВМ : учеб. пособие. / В.П. Дерябкин, М.А. Кораблин. – Куйбышев: Куйбышевский авиационный институт, 1977, - 82 с.
14. Очков, В.Ф. Типовые энергетические характеристики и расчеты турбоустановок в интернете / В.Ф. Очков, М.А. Буринов // Энергетик – 2004. – № 11. — URL: http://twi.mpei.ru/ochkov/MAS_Turbine/ (дата обращения: 19.12.2023).
15. Каменев, С.В. Основы метода конечных элементов в инженерных приложениях: учебное пособие / С.В. Каменев. – Оренбург: ОГУ, 2019. – 110 с.
16. Норенков, И. П. Основы автоматизированного проектирования: учеб. для вузов. — 4-е изд., перераб. и доп.— М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. — 430 с.
17. Чугунов, М. В. Программный модуль для решения задач оптимального проектирования в среде SolidWorks на базе API / М. В. Чугунов, Ю. А. Небайкина // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2011. – № 9. – С. 21
18. Ахминин, В. Н. Введение в математическое моделирование: учебное пособие / В. Н. Ахминин, М. Б. Гитман, И. Э. Келлер [и др.]. — Москва : Логос, 2004. — 439 с. — Текст : непосредственный.
19. Горностаева, Т.Н. Математическое и компьютерное моделирование : учебное пособие / Горностаева Т.Н., Горностаев О.М. – М.: Мир науки, 2019. – URL: <https://izd-mn.com/PDF/50MNNPU19.pdf> (дата обращения 19.12.2023)
20. Виртуальная лаборатория математического моделирования. — URL: <http://mathmod.narod.ru/metods.htm#mm03> (дата обращения: 06.12.2023)

- 21.Звонарев, С.В. Основы математического моделирования: учебное пособие / С. В. Звонарев. — Екатеринбург : Изд-во Урал. ун-та, 2019. — 112 с.
- 22.Леонова, Н. Л. Компьютерное моделирование: курс лекций \ Н. Л. Леонова. — СПбГТУРП. - СПб., 2015. - 88 с
- 23.Норенков, И. П. Основы автоматизированного проектирования: учеб. для вузов. — 4-е изд., перераб. и доп.— М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. — 430 с.
- 24.SolidWorks World 2015: как всегда, самая актуальная информация о мире SolidWorks // САПР и графика. – 2015. – № 3(221). – С. 48-52.
- 25.Тодорев, А. Н. Применение макросов при проектировании несущих конструкций в SolidWorks / А. Н. Тодорев, М. Н. Дятлов, Д. Ю. Брагин [и др.] // Молодой ученый. – 2016. – № 6(110). – С. 210-213.
- 26.SOLIDWORKS API // SolidWorks Help. – URL: https://help.solidworks.com/2022/russian/solidworks/sldworks/c_solidworks_api.htm (дата обращения 07.01.2024)
- 27.Разработка API-приложения для программного пакета SolidWorks 2012. Часть 1 // Хабр. - 2013. - URL: <https://habr.com/ru/articles/201700/> (дата обращения : 27.01.2024)
- 28.Запись и выполнение макроса // SOLIDWORKS Help. – URL: https://help.solidworks.com/2020/RUSSIAN/SolidWorks/sldworks/c_recording_playing_macros.htm?id=90facecd944f4adea8bb1a2b5dce34a5. (дата обращения: 27.01.2024).
- 29.Программа "Eldvig" // Официальный сайт программы Eldvig , - URL: <http://eldvig.bos.ru/eldvig.html> (дата обращения: 12.01.2024)

ПРИЛОЖЕНИЕ А

ПЛАН ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ РАСЧЁТА ХАРАКТЕРИСТИК ЭЛЕКТРОДВИГАТЕЛЯ

Цель тестирования

Оценка корректности и эффективности работы веб-приложения, предназначенного для расчета характеристик электродвигателей. Тестирование должно проверить функциональность, производительность, безопасность и удобство использования приложения.

Объекты тестирования

1. Входные данные: проверка приемки и обработки данных пользователем (например, тип двигателя, мощность, напряжение).
2. Интерфейс пользователя: оценка интуитивности и удобства интерфейса.
3. Производительность: анализ скорости обработки данных и отклика приложения.

Основные сценарии тестирования

1. Тестирование функциональности:
 - валидация корректного расчета характеристик при стандартных входных значениях;
 - проверка правильности расчета путем сравнения полученных результатов с эталонным решением.
2. Тестирование пользовательского интерфейса:
 - проверка отображения всех элементов интерфейса на различных устройствах;
 - проверка корректности сообщений об ошибках и подсказок.
3. Тестирование производительности:
 - оценка времени выполнения расчетов при максимальной загрузке.

Критерии тестирования

При прохождении всех тестовых сценариев тестирование считается пройденным.

ПРИЛОЖЕНИЕ Б

ЧАСТЬ КОДА РАСЧЕТА

```
export function makeCalculation(
  vars: Record<string, number>,
  topN,
  StrictMode,
  JStrictMode,
): Array<Record<string, number>> {
  h_mIterator.restart();
  h_cIterator.restart();
  h_pIterator.restart();
  a_zIterator.restart();
  console.clear();

  const kpdHolder: Array<Record<string, number>> = [];
  // Основные размеры
  var { P_2, n, U, D_n, Temp } = vars;
  var Strict = StrictMode;
  var Jstict = JStrictMode;
  if (P_2 < 0 || n < 0 || U < 0 || D_n < 0 || topN < 0) {
    alert("Все значения должны быть больше нуля");
    return kpdHolder;
  }
  if (P_2 > 1e5 || n > 1e5 || U > 1e5 || D_n > 1e5 || topN > 1e6) {
    alert("Значение выше допустимых");
    return kpdHolder;
  }
  # .. // Инициализация констант
  const lambda = P_2 / (D_n**3 * k * A * B_delta * n);
  const l = Math.ceil(lambda * D_n / 0.000206) * 0.000206; // 0.206
  const l_deltaIterator = createIterator(l, mm(45), mm(0.206));
  l_deltaIterator.restart();

  while (!l_deltaIterator.ended()) {
    var l_delta = l_deltaIterator.current;
    const l_a = l_delta;
    const l_m = l_delta;

    h_mIterator.restart();
    magneticIteration: while (!h_mIterator.ended()) {
      h_cIterator.restart();
      h_pIterator.restart();
      a_zIterator.restart();
      // Высота магнита
      var h_m = h_mIterator.current;

      inductionIteration: while (!h_cIterator.ended() && !h_pIterator.ended() && !a_zIterator.ended()){
        let h_c = h_cIterator.current;
        let h_p = h_pIterator.current;
        let a_z = a_zIterator.current;

        // Магнитная цепь
        var D_pv = D_n - 2 * h_m - 2 * h_p;
        var b_m = (alpha_delta * D_pv * Math.PI) / (2 * p);
        let Phi_1 = (H_c * mu_0 * l_m * b_m * h_m) / (delta * k_mu + h_m);
        var d_2 = 0.003;

        PhiIteraion: while (!h_cIterator.ended() && !h_pIterator.ended() && !a_zIterator.ended()) {
          let h_c = h_cIterator.current;
          let h_p = h_pIterator.current;
          let a_z = a_zIterator.current;

          var D_pv = D_n - 2 * h_m - 2 * h_p;
          var b_m = (alpha_delta * D_pv * Math.PI) / (2 * p);

          // Индукция в спинке ротора
          //showValue({ Phi_1 }, 2.9 * 10 ** (-4));
          var B_p = Phi_1 / (2 * h_p * l_m);
          if (B_p > 1.6) {
            h_pIterator.next();
            h_p = h_pIterator.current;
          }
        }
      }
    }
  }
}
```

```

        continue PhiIteraion;
    }
    else if (B_p < 1.2 && StrictMode) {
        h_mIterator.next();
        continue magneticIteration;
    }
    // Индукция в зубце
    var B_z = (Phi_1 * 2 * p) / (z * a_z * l_m * k_c * 0.8)
    if (B_z > 1.7) {
        a_zIterator.next();
        a_z = a_zIterator.current;
        continue PhiIteraion;
    }
    else if (B_z < 1.2 && StrictMode) {
        h_mIterator.next();
        continue magneticIteration;
    }
    // Индукция в спинке статора
    var B_c = Phi_1 / (2 * h_c * l_m * k_c);

    if (B_c > 1.6) {
        h_cIterator.next();
        h_c = h_cIterator.current;
        continue PhiIteraion;
    }
    else if (B_c < 1.2 && StrictMode) {
        h_mIterator.next();
        continue magneticIteration;
    }
    #.. // Часть кода расчета
    var Phi = (H_c * h_m) / R;
    // (Φ1 - Φ) / Φ1 * 100
    // допустимое отклонение 1%
    var deviation = Math.abs((Phi_1 - Phi) / Phi_1) * 100;
    if (deviation > 1) {
        Phi_1 = Phi;
        h_cIterator.restart();
        h_pIterator.restart();
        a_zIterator.restart();
        continue PhiIteraion;
    }
    #.. // Часть кода расчета
    var P_pot = P_el + P_a + P_z + P_mex + P_pov;
    var J = I_s / (q*a)
    if (J > 8.1e6 && Jstict){ //
        h_mIterator.next();
        continue magneticIteration;
    }
    if (J < 7.9e6 && Jstict){ //
        d_2 += mm(0.1);
        continue PhiIteraion;
    }
    // КПД
    var nyu = 1 - P_pot / (P_2 + P_pot);

    const result = {
        #.. // Вывод нужных переменных
    };
    kpdHolder.push(result);
    h_mIterator.next();
    break PhiIteraion;
}
break inductionIteration;
}
h_mIterator.next();
}
l_deltaIterator.next();
}

const sortedKpdHolder = kpdHolder.sort((a, b) => b.K_i12 - a.K_i12);
const topNKpd = sortedKpdHolder.slice(0, topN);

return topNKpd;
}

```