

**«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)**

Направление	09.03.04 — Программная инженерия
Профиль	Без профиля
Факультет	КТИ
Кафедра	МО ЭВМ

К защите допустить

И.о зав. кафедрой

А.А. Лисс

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ТЕМА: РАЗРАБОТКА ИНСТРУМЕНТА МОДЕЛИРОВАНИЯ И ВИЗУАЛИЗАЦИИ ФОРМЫ ПРОСТРАНСТВЕННОЙ КРИВОЙ, ВОССТАНОВЛЕННОЙ ПО ДАННЫМ, ИЗМЕНЯЮЩИМСЯ В РЕАЛЬНОМ ВРЕМЕНИ.

Студент		<hr/>	Д.С. Попов
		<i>подпись</i>	
Руководитель	д.т.н., профессор	<hr/>	А.Р. Лисс
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	
Консультант	к.т.н.	<hr/>	Е.И. Сергеева
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	
Консультант	к.э.н., доцент	<hr/>	Ю.А. Елисеева
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	
Консультант	к.т.н., доцент	<hr/>	М.М. Заславский
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	

Санкт-Петербург

2023

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Утверждаю

И.о зав. кафедрой МО ЭВМ

_____ А.А. Лисс

« ____ » _____ 2023 г.

Студент Д.С. Попов

Группа 9304

Тема работы: Разработка инструмента моделирования и визуализации формы пространственной кривой, восстановленной по данным, изменяющимся в реальном времени.

Место выполнения ВКР: СПбГЭТУ «ЛЭТИ» кафедра МО ЭВМ

Исходные данные (технические требования): Необходимо реализовать инструмент моделирования и визуализации формы пространственной кривой

Содержание ВКР: Введение, Описание основных методов интерполяции, Разработка инструмента визуализации, Тестирование программного продукта, Заключение.

Перечень отчетных материалов: пояснительная записка, иллюстративный материал.

Дополнительные разделы: нормативно-правовое регулирование интеллектуальной деятельности.

Дата выдачи задания

« ____ » _____ 2023 г.

Дата представления ВКР к защите

« ____ » _____ 2023 г.

Студент

_____ Д.С. Попов

Руководитель д.т.н., профессор

_____ А.Р. Лисс

Консультант к.т.н.

_____ Е.И. Сергеева

КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю

И.о зав. кафедрой МО ЭВМ

_____ А.А. Лисс

« ____ » _____ 2023 г.

Студент Д.С. Попов

Группа 9304

Тема работы: Разработка инструмента моделирования и визуализации формы пространственной кривой, восстановленной по данным, изменяющимся в реальном времени

№ п/п	Наименование работ	Срок выполнения
1	Обзор литературы по теме работы	20.04 – 01.05
2	Обзор предметной области	02.05 – 05.05
3	Формулировка требований к решению и постановка задачи	06.05 – 08.05
4	Архитектурная программная реализация	09.05 – 15.05
5	Исследование разработанного инструмента	16.05 – 18.05
6	Нормативно-правовое регулирование интеллектуальной деятельности	19.05 – 22.05
7	Оформление пояснительной записки	23.05 – 26.05
8	Оформление иллюстративного материала	27.05 – 29.05
9	Предзащита	01.06

Студент

Д.С. Попов

Руководитель Д.Т.Н., профессор
(Уч. степень, уч. звание)

А.Р. Лисс

Консультант

К.Т.Н.
(Уч. степень, уч. звание)

Е.И. Сергеева

РЕФЕРАТ

Пояснительная записка 50 стр., 18 рис., 8 табл., 24 ист., 00 прил.

Ключевые слова АЛГОРИТМ ВОССТАНОВЛЕНИЯ ДАННЫХ, ИНТЕРПОЛЯЦИЯ, СИСТЕМА ОРИЕНТАЦИИ ГИБКОЙ ПРОТЯЖЕННОЙ БУКСИРУЕМОЙ АНТЕННЫ, ПРОСТРАНСТВЕННАЯ КРИВАЯ, ИНСТРУМЕНТ ВИЗУАЛИЗАЦИИ

Объектом исследования являются гибкие протяженные буксируемые антенны с датчиками системы ориентации.

Предметом исследования методы оценки формы антенны по данным датчиков системы ориентации.

Цель работы состоит в разработке программного продукта для моделирования и визуализации, восстановленной по данным датчиков системы ориентации формы гибкой антенны, а также позволяющего оценить точность и производительность методов восстановления формы для заданных параметров антенны.

В современном мире гидроакустические системы играют важную роль в многих областях, таких как геофизика, морское исследование и военное дело. В составе таких систем широко применяются гибкие протяженные буксируемые антенны (ГПБА). Традиционно, обработка гидроакустической информации ГПБА выполняется в предположении о прямолинейной форме ГПБА. С включением датчиков системы ориентации в современные ГПБА появилась возможность получать информацию о текущей форме антенны в реальном времени и учитывать её в дальнейшей обработке. Датчики системы ориентации позволяют получить свои координаты в пространстве, но в практических случаях этих датчиков значительно меньше, чем акустических. Задача восстановления формы антенны состоит в получении координат каждого акустического приёмника и её можно сформулировать как интерполяцию кривой в пространстве. В настоящее время существует множество методов

восстановления формы гибких пространственных кривых, включая методы, основанные на кривых Безье, кривых Б-сплайнов, а также методы, использующие многочлены и рациональные функции. Каждый метод имеет свои преимущества и недостатки, выбор оптимального метода зависит от конкретных условий и требований. Однако, существующие методы восстановления формы кривой, не всегда обеспечивают необходимую точность. Кроме того, многие методы требуют вычислительно сложных алгоритмов, что может затруднять их использование в реальном времени.

Результаты работы будут полезны для исследования влияния учёта формы гибкой антенны на точность и эффективность обработки гидроакустической информации от антенны, а также при выборе метода восстановления для внедрения в штатную реализацию гидроакустической системы.

ABSTRACT

Hydroacoustic systems, such as geophysics, marine exploration and military science, play a significant role to a large extent. As part of such systems, flexible extended towed antennas (FETA) are widely used. Traditionally, hydroacoustic informatics processing of FETA has a special place in the assumption of a rectilinear form of FETA. With the connection of sensors of the orientation system in modern FETA, it became possible to obtain information about the antenna tuning depending on time and take it into account in further processing. Orientation system sensors allow you to get your coordinates indoors, but in practical cases, the sensors are much smaller than acoustic ones. The task of restoring the shape of the antenna is to obtain the coordinates of each acoustic receiver, and it can be formulated as an interpolation of a curve in a plane. Currently, there are many methods for detecting flexible open curves, including methods based on Bezier curves, B-spline curves, as well as methods using polynomials and drop-down functions. Each method has its own advantages and possibilities, the choice of a unique method depends on the specific conditions and safety. However, compliance with methods for restoring the shape of the curve does not always require compliance. In addition, many methods require computationally complex algorithms, which can lead to their use in a time environment.

The results of the work will be useful for studying the consideration of the flexible shape of the antenna for the efficiency and effectiveness of processing hydroacoustic information from the antenna, as well as for choosing a restoration method for use in a regular frequency hydroacoustic system.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1 ОПИСАНИЕ ОСНОВНЫХ МЕТОДОВ ИНТЕРПОЛЯЦИИ.....	10
1.1 ПОНЯТИЕ ИНТЕРПОЛЯЦИИ.....	10
1.2 ПОЛИНОМИАЛЬНАЯ ИНТЕРПОЛЯЦИЯ.	11
1.2.1 Интерполяция Эрмита.	11
1.2.2 Интерполяция Уиттакера-Шеннона.....	12
1.3 ДРОБНО-РАЦИОНАЛЬНАЯ ИНТЕРПОЛЯЦИЯ.	13
1.4 СПЛАЙНОВАЯ ИНТЕРПОЛЯЦИЯ.	14
1.4.1 Модифицированная интерполяция Акима.....	15
1.4.2 Кусочно-кубический полином Эрмита.....	16
1.4.3 Квадратичные В-сплайны.....	17
1.4.4 Кубические В-сплайны.	18
1.5 ВЫБОР КРИТЕРИЕВ ДЛЯ СРАВНЕНИЯ.	19
1.6 СРАВНЕНИЕ МЕТОДОВ ПО КРИТЕРИЯМ.	19
1.7 ВЫВОД ПО РАЗДЕЛУ.	20
2 РАЗРАБОТКА ИНСТРУМЕНТА ВИЗУАЛИЗАЦИИ	21
2.1 ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОМУ ПРОДУКТУ.....	21
2.2 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ.	21
2.3 ОБЩАЯ АРХИТЕКТУРА.....	22
2.3.1 Контроллер.....	24
2.3.2 Представление.	24
2.3.3 Модель.	25
2.4 ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ.	25
2.5 СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ.	31
2.6 ВЫВОД ПО РАЗДЕЛУ.	32
3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА	33
3.1 ПОДХОДЫ ТЕСТИРОВАНИЯ.....	33

3.2 ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПК.	34
3.3 ТЕСТИРОВАНИЕ ПРОГРАММЫ.	34
3.4 ВЫВОД ПО РАЗДЕЛУ.	39
4 НОРМАТИВНО-ПРАВОВОЕ РЕГУЛИРОВАНИЕ	
ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ.....	40
4.1 ОБЩИЕ ПОЛОЖЕНИЯ.....	40
4.2 ОПИСАНИЕ ОБЪЕКТА РАЗРАБОТКИ.....	42
4.3 РЕГИСТРАЦИЯ ПРОГРАММЫ ЭВМ.	42
4.4 РЕАЛИЗАЦИЯ ПРАВ ГОСУДАРСТВА НА ОБЪЕКТЫ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ.	43
4.5 НАЛОГООБЛОЖЕНИЕ.....	44
4.5 РАСПРОСТРАНЕНИЕ В СЕТИ «ИНТЕРНЕТ».	45
4.6 ПРЕСТУПЛЕНИЯ В СФЕРЕ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ.	46
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49

ВВЕДЕНИЕ

В современном мире одним из наиболее важных направлений развития технологий является создание новых технических устройств, способных улучшать качество и эффективность процессов в различных сферах деятельности человека. Одним из таких устройств является гибкая протяженная гидроакустическая буксируемая антенна с датчиками системы ориентации, которая может использоваться в многих областях, таких как морское исследование, геологические исследования, метеорология и другие [1, 2].

Однако одной из проблем, с которой сталкиваются исследователи, работающие с данными, получаемыми от таких устройств, является сложность восстановления формы гибкой антенны по этим данным. Для решения этой проблемы в данной работе предлагается реализовать инструмент моделирования и визуализации формы пространственной кривой, восстановленной по данным, изменяющимся в реальном времени [3, 4].

Таким образом, в качестве **объекта исследования** в данной работе являются гибкие протяженные гидроакустические буксируемые антенны с датчиками системы ориентации, а в качестве **предмета исследования** выступают методы оценки формы антенны по данным датчиков системы ориентации. Основная **цель исследования** состоит в разработке программного продукта для моделирования и визуализации, восстановленной по данным датчиков системы ориентации формы гибкой антенны, а также позволяющего оценить точность и производительность методов восстановления формы для заданных параметров антенны. Для успешного достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ существующих методов восстановления
2. Определится с критериями отбора методов восстановления для использования в программном продукте
3. Сравнить отобранные методы
4. Реализовать инструмент визуализации
5. Произвести тестирование разработанного продукта

1 ОПИСАНИЕ ОСНОВНЫХ МЕТОДОВ ИНТЕРПОЛЯЦИИ.

1.1 Понятие интерполяции.

Интерполяция является одним из основных методов аппроксимации функций и широко используется в различных областях, где требуется вычисление значений функции между заданными точками. В общем случае интерполяция — это процесс нахождения функции, которая проходит через некоторый набор точек [5]. На рис. 1 показан пример интерполяции, где красные точки представляют собой известные значения функции, через которые проводится синяя линия, представляющая интерполяционную функцию. Таким образом, можно вычислить промежуточные значения функции в других точках.

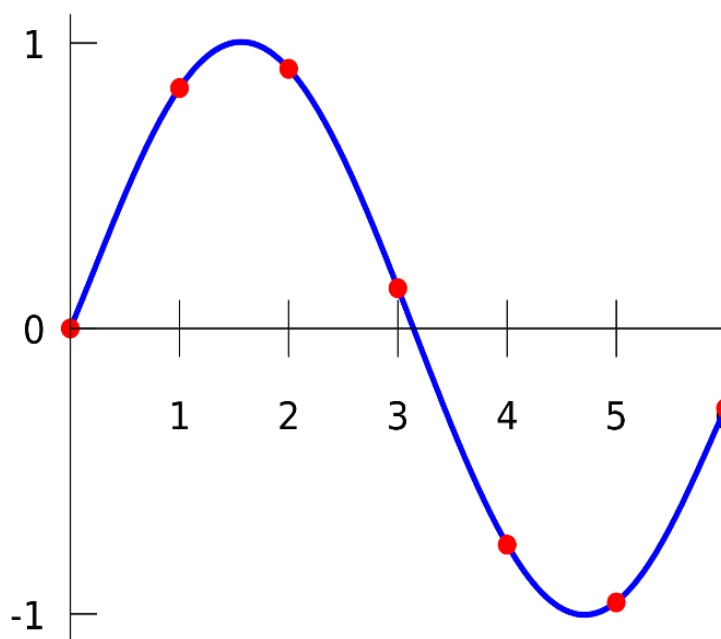


Рисунок 1 — Пример работы интерполяции

Существует несколько разновидностей интерполяции, включая полиномиальную, дробно-рациональную, сплайновую и кусочно-полиномиальную. В данном разделе будут рассмотрены основные методы интерполяции и принципы их работы.

1.2 Полиномиальная интерполяция.

Полиномиальная интерполяция — это метод, который позволяет приближенно вычислить значения функции $f(x)$ на заданном отрезке $[a, b]$ по ее значениям в узлах интерполяции x_0, x_1, \dots, x_n . При этом предполагается, что функция $f(x)$ достаточно гладкая и определена на всем отрезке $[a, b]$ [3].

Полиномиальная интерполяция основана на использовании многочлена Лагранжа [6] или многочлена Ньютона [7]. Многочлен Лагранжа определяется следующей формулой:

$$L_n(x) = \sum_{j=0}^n y_j l_j(x),$$

где $y_i = f(x_i)$ — значения функции в узлах интерполяции, а $l_i(x)$ — полиномы Лагранжа, которые определяются следующим образом:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Многочлен Ньютона выглядит следующим образом:

$$N_i(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n(x - x_0) \dots (x - x_{n-1}),$$

где a_0, a_1, \dots, a_n — разделенные разности, которые могут быть найдены с помощью таблицы разделенных разностей.

Оба этих метода позволяют построить многочлен, который проходит через все заданные узлы интерполяции и приближенно описывает функцию $f(x)$ на всем отрезке $[a, b]$.

1.2.1 Интерполяция Эрмита.

Интерполяция Эрмита — это метод интерполяции, который использует производные функции в заданных точках для создания интерполяционного полинома [8]. Этот метод может быть использован для интерполяции функций с разрывами или особенностями, и он обеспечивает более точные результаты, чем полиномиальная интерполяция.

Интерполяционный полином Эрмита $H_n(x)$ порядка n может быть записан в виде:

$$H_n(x) = \sum_{i=0}^n f(x_i) h_i(x)$$

где $f(x_i)$ — значение функции в точке x_i , $h_i(x)$ - интерполяционный полином, который задается формулой:

$$h_i(x) = [1 - 2(x - x_i)L'_i(x_i) + (x - x_i)L''_i(x_i)](L_i(x))^2$$

здесь $L_i(x)$ — полином Лагранжа степени n :

$$L_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

А $L'_i(x_i)$ и $L''_i(x_i)$ — первая и вторая производные полинома Лагранжа в точке x_i :

$$L'_i(x_i) = \sum_{j=0, j \neq i}^n \frac{1}{x_i - x_j},$$

$$L''_i(x_i) = \sum_{j=0, j \neq i}^n \frac{2}{(x_i - x_j)^2}$$

1.2.2 Интерполяция Уиттакера-Шеннона.

Интерполяция Уиттакера-Шеннона, также известная как семплирование Шеннона или теорема Котельникова-Шеннона, используется для восстановления непрерывной функции $f(x)$ по ее дискретному набору значений $f(n)$, взятым с некоторым шагом дискретизации T [9]. Формула для интерполяции Уиттакера-Шеннона выглядит следующим образом:

$$f(x) = \sum_{n=-\infty}^{\infty} f(n) \cdot \text{sinc}\left(\frac{x - nT}{T}\right),$$

где $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ — кардинальный синус.

Данная формула утверждает, что функция $f(x)$ может быть полностью восстановлена из ее значений $f(n)$ при условии, что шаг дискретизации T не превосходит половины наименьшей периодической составляющей функции $f(x)$.

1.3 Дробно-рациональная интерполяция.

Некоторые функции не могут быть приближены полиномами с достаточной точностью, либо полиномиальное приближение сходится очень медленно. В таких случаях дробно-рациональное приближение (иногда называемое рациональным) может быть более эффективным методом, которое соответствует отношению двух многочленов [10].

$$R(x) = \frac{a_0 + a_1x + \dots + a_px^p}{b_0 + b_1x + \dots + b_qx^q}, \quad p + q + 1 = n$$

Коэффициенты a_i, b_i можно найти из совокупности соотношений $R(x_i) = y_i, i = 1, \dots, n$, которые можно записать в виде:

$$\sum_{i=0}^p a_i x_i^i - f(x_i) \sum_{i=0}^q b_j x_i^j = 0, i = 1, \dots, n$$

Из этого процесса получается систему из n линейных алгебраических уравнений относительно $n + 1$ неизвестных. Если n нечетное и $p = q$ или n четное и $p - q = 1$, то функция $R(x)$ может быть записана явно. Для этого необходимо вычислить обратные разделенные разности в соответствии с определенными условиями

$$f^-(x_k; x_l) = \frac{x_k - x_l}{f(x_k) - f(x_l)}$$

и рекуррентным соотношением

$$f^-(x_k; \dots; x_l) = \frac{x_l - x_k}{f^-(x_{k+1}; \dots; x_l) - f^-(x_k; \dots; x_{l-1})}$$

после чего интерполирующая рациональная функция записывается в виде цепной дроби

$$f^-(x_k; \dots; x_n) = \frac{x - x_1}{f^-(x_1; x_2) + \frac{x - x_2}{f^-(x_1; x_2; x_3) + \dots + \frac{z - z_{N-1}}{f^-(z_k; \dots; z_n)}}$$

Для функции, которая имеет нерегулярное поведение, рациональное интерполирование может быть эффективным методом при условии, что узлы выбраны правильно.

1.4 Сплайновая интерполяция.

Сплайновая интерполяция — это метод интерполяции, при котором используется набор полиномов малой степени, называемых сплайнами, для аппроксимации кривой или поверхности, проходящей через заданные точки [11].

Для построения сплайновой интерполяции необходимо разбить исходный интервал на меньшие интервалы, называемые узлами. Затем для каждого узла строится полином, проходящий через узел и соседние узлы. Эти полиномы должны удовлетворять некоторым условиям, называемым граничными условиями, которые определяются в зависимости от конкретной задачи.

Обычно для сплайновой интерполяции используются кубические сплайны, т.е. полиномы третьей степени. При этом каждый кубический сплайн определяется следующим образом:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x_i \leq x \leq x_{i+1}$$

где x_i — узел интерполяции, a_i, b_i, c_i, d_i — коэффициенты полинома, которые необходимо найти.

Граничные условия могут быть разными. Например, для натурального сплайна используется условие равенства нулю второй производной на концах интервала:

$$S''(x_0) = S''(x_n) = 0$$

Для кубического сплайна на каждом интервале, кроме первого и последнего, также должны быть выполнены условия непрерывности первой и второй производных:

$$\begin{aligned} S_i(x_{i+1}) &= S_{i+1}(x_{i+1}), \\ S'_i(x_{i+1}) &= S'_{i+1}(x_{i+1}), \\ S''_i(x_{i+1}) &= S''_{i+1}(x_{i+1}) \end{aligned}$$

Таким образом, сплайновая интерполяция позволяет более гладко аппроксимировать кривую или поверхность, чем полиномиальная интерполяция, и имеет более гибкие граничные условия.

1.4.1 Модифицированная интерполяция Акима.

Модифицированная интерполяция Акима — это метод интерполяции данных, который использует локальные полиномы для аппроксимации данных в окрестности каждой точки исходного набора данных [12].

Для построения интерполяционной кривой на основе данных $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, где x_i — координаты точек на оси x , а y_i — соответствующие значения функции на оси y , сначала вычисляются конечные разности второго порядка:

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$$

где $\Delta y_i = y_i - y_{i-1}$.

Затем для каждой точки используются локальные полиномы, которые строятся по 5 точкам с помощью формулы:

$$p(x) = c_1 + c_2(x - x_i) + c_3(x - x_i)^2 + c_4(x - x_i)^3$$

где c_1, c_2, c_3 и c_4 - коэффициенты полинома, которые находятся из условий, что:

полином проходит через центральную точку исходного набора данных:

$$p(x_i) = y_i$$

первые производные полинома совпадают с конечными разностями:

$$p'(x_i) = \Delta y_i, p'(x_{i+1}) = \Delta y_{i+1}$$

полином является монотонным, то есть, $p'(x_i)$ и $p'(x_{i+1})$ имеют одинаковый знак.

Коэффициенты c_1, c_2, c_3 и c_4 могут быть найдены аналитически с помощью следующих формул:

$$c_1 = y_i$$

$$c_2 = \Delta y_i$$

$$c_3 = \frac{3\Delta^2 y_i - 2\Delta^2 y_{i-1} - \Delta^2 y_{i+1}}{x_{i+1} - x_i}$$

$$c_4 = \frac{2\Delta^2 y_i + \Delta^2 y_{i-1} - 3\Delta^2 y_{i+1}}{(x_{i+1} - x_i)^2}$$

Интерполяционная кривая может быть получена путем соединения всех полиномов.

1.4.2 Кусочно-кубический полином Эрмита.

Интерполяция кусочно-кубическим полиномом Эрмита — это метод интерполяции, который использует кубические полиномы для интерполяции функции и ее производной. Для каждого интервала $[x_i, x_{i+1}]$ метод определяет значения функции и ее производной в четырех точках: $f(x_i), f(x_{i+1}), f'(x_i), f'(x_{i+1})$. Затем на этом интервале строится кубический

полином, который проходит через эти точки и имеет те же значения производной на концах интервала, что и исходная функция [13].

Формулы для построения кусочно-кубического полинома Эрмита на интервале $[x_i, x_{i+1}]$ выглядят следующим образом:

$$H(x) = h_{00}(x)f(x_i) + h_{10}(x)f(x_{i+1}) + h_{01}(x)f'(x_i) + h_{11}(x)f'(x_{i+1}),$$

где $f(x_i), f(x_{i+1}), f'(x_i), f'(x_{i+1})$ — известные значения функции и ее производной в точках x_i и x_{i+1} , а $h_{00}(x), h_{10}(x), h_{01}(x)$ и $h_{11}(x)$ - кубические базисные функции Эрмита:

$$\begin{aligned} h_{00}(x) &= 2\left(\frac{x-x_i}{h}\right)^3 - 3\left(\frac{x-x_i}{h}\right)^2 + 1 \\ h_{10}(x) &= -2\left(\frac{x-x_{i+1}}{h}\right)^3 + 3\left(\frac{x-x_{i+1}}{h}\right)^2 \\ h_{01}(x) &= \left(\frac{x-x_i}{h}\right)^3 - 2\left(\frac{x-x_i}{h}\right)^2 + \frac{x-x_{i+1}}{h} \\ h_{11}(x) &= -2\left(\frac{x-x_{i+1}}{h}\right)^3 + \left(\frac{x-x_{i+1}}{h}\right)^2 \end{aligned}$$

где $h = x_{i+1} - x_i$ — длина интервала.

1.4.3 Квадратичные В-сплайны.

Интерполяция квадратичными В-сплайнами представляет собой метод интерполяции, основанный на построении кусочно-квадратичного сплайна с использованием функций Б-сплайнов второго порядка [14].

Формула квадратичного В-сплайна имеет вид:

$$S(x) = \begin{cases} \frac{(x-t_i)^2}{2(t_{i+1}-t_i)(t_{i+2}-t_i)}, & t_i \leq x \leq t_{i+1} \\ \frac{(x-t_{i+1}+1)(x-t_{i-1})}{(t_{i+1}-t_i)(t_{i+1}-t_{i-1})}, & t_{i+1} \leq x \leq t_{i+2} \\ \frac{(t_{i+2}-x)^2}{2(t_{i+2}-t_{i+1})(t_{i+2}-t_i)}, & t_{i+2} \leq x \leq t_{i+3} \end{cases}$$

где t_i — узлы интерполяции, $i = 0, 1, 2, \dots, n+1$.

Для интерполяции значения функции в узлах интерполяции используются условия:

$$S(t_i) = f(t_i), \quad S'(t_i) = f'(t_i)$$

где $f(t_i)$ и $f'(t_i)$ — значения функции и ее производной в узлах интерполяции.

Интерполяционная кривая представляет собой кусочно-квадратичный сплайн, который проходит через все узлы интерполяции и имеет непрерывные первую производную.

1.4.4 Кубические В-сплайны.

Интерполяция кубическими В-сплайнами — это метод интерполяции, который использует кубические функции-сплайны для аппроксимации кусочно-гладкой функции на заданном интервале. Кубические В-сплайны имеют преимущества перед другими методами интерполяции, такими как полиномиальная интерполяция, потому что они более устойчивы к выбросам и шумам [15].

Кубический В-сплайн на каждом отрезке $[x_i, x_{i+1}]$ представляется кубическим полиномом $S_i(x)$, который является решением системы линейных уравнений, удовлетворяющих следующим условиям:

$S_i(x_i) = f(x_i)$ и $S_i(x_{i+1}) = f(x_{i+1})$ — сплайн проходит через соседние узлы.

$S_{i-1}(x_i) = S_i(x_i)$ и $S'_{i-1}(x_i) = S'_i(x_i)$ — первая и вторая производные сопрягаются на точке пересечения.

$S''_{i-1}(x_n) = S''_n(x_n) = 0$ — краевые условия.

Кубический сплайн $S_i(x)$ на отрезке $[x_i, x_{i+1}]$ может быть записан в следующей форме:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

где a_i, b_i, c_i и d_i - коэффициенты, которые могут быть найдены из уравнений выше.

1.5 Выбор критериев для сравнения.

После того, как были рассмотрены основные методы интерполяции, необходимо проанализировать их достоинства и недостатки. Для этого нужно определиться с критериями, которые будут оказывать наибольшее влияние на программный продукт. Среди множества критериев выделяются следующие:

Сложность алгоритма – характеристика, которая указывает на то, сколько времени, либо какой объем памяти потребуется для выполнения.

Наличие готовой реализации позволит ускорить разработку продукта, а также снизить количество ошибок и некачественного кода.

Гибкость настройки отражает способность метода настраиваться под конкретную задачу, что благоприятно сказывается на точности вычисления, но увеличивает количество аспектов, на которые стоит обратить внимание при разработке.

1.6 Сравнение методов по критериям.

После выбора критериев было проведено сравнение, результаты представлены в таблице 1:

Таблица 1 — Сравнение методов интерполяции по критериям

Метод/ Критерий	Сложность алгоритма	Реализа- ция в биб- лиотеках	Гибкая настройка параметров
Интерполяция Уиттакера-Шеннона	$O(N)$	+	-
Кубическая интерполяция Эрмита	$O(\log(N))$	+	+
Кардинально-квадратичная В- сплайновая интерполяция	$O(N)$	+	+
Кардинально кубическая интерпо- ляция В-сплайна	$O(N)$	+	+
Барицентрическая рациональная интерполяция	$O(N)$	+	+
Модифицированная интерполяция Акима	$O(\log(N))$	+	+

1.7 Вывод по разделу.

Рассмотрены основные методы интерполяции, которые используются в данном приложении. Каждый метод имеет свои преимущества и ограничения, и выбор конкретного метода зависит от требований к точности и эффективности вычислений. Некоторые методы, такие как интерполяция Эрмита, могут обеспечить точность интерполяции при условии наличия дополнительной информации о функции. Другие методы, такие как сплайны, позволяют достичь гладкости и непрерывности интерполирующей функции. Все они доступны в приложении и могут быть использованы для интерполяции пространственных кривых.

2 РАЗРАБОТКА ИНСТРУМЕНТА ВИЗУАЛИЗАЦИИ

2.1 Требования к разрабатываемому продукту.

Опираясь на проведённый анализ существующих алгоритмов восстановления данных методами интерполяции, можно сформировать требования, которых должна придерживаться разрабатываемая программная система:

1. Продукт должен включать один или несколько алгоритмов восстановления.
2. В программной системе должны использоваться методы, реализованные в готовых математических библиотеках.
3. Программная система должна иметь возможность менять условия или параметры используемых алгоритмов интерполяции в зависимости от ситуации и предоставлять удобный пользовательский интерфейс.

Таким образом разрабатываемая система, производящая моделирование и визуализацию пространственной кривой по восстановленным данным, удовлетворит всем требованиям. Это позволит пользователям в реальном времени отслеживать ориентацию гибкой буксируемой антенны в пространстве во время буксировки. Использование быстрых и реализованных алгоритмов снизит вероятность ошибок при работе программы.

2.2 Используемые технологии.

Для решения поставленной задачи по разработке архитектуры приложения, удовлетворяющего описанным выше требованиям, были выбраны следующие технологии и цели их использования:

- C++ [16] - лучший вариант по производительности кода в высоконагруженных приложениях. На этом языке будет написана вся логика программы.
- boost [17] - математическая библиотека на языке C++. Из неё будут задействованы готовые алгоритмы интерполяции.

- Qt [18] - фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++. Он позволит быстро создать графический интерфейс продукта и смоделировать пространственную кривую.

Программная реализация подразумевает монолитную архитектуру. На вход приложению в реальном времени поступают потоковые данные с датчиков гибкой антенны. Алгоритм позволит сформировать модель кривой, при этом гарантируя её прохождение через имеющиеся точки. Затем на основе восстановленных данных моделируется пространственная ориентация антенны и результат выводится на экран. Обработка всех графических элементов производится в среде Qt.

2.3 Общая архитектура.

При проектировании и разработке приложений часто используют различные архитектурные паттерны, такие как MVC (Model-View-Controller) [19]. Этот паттерн помогает разделить приложение на три компонента: модель, представление и контроллер. Каждый компонент выполняет свою задачу, что облегчает сопровождение и развитие приложения. На рис. 2 представлена обобщенная архитектура решения.

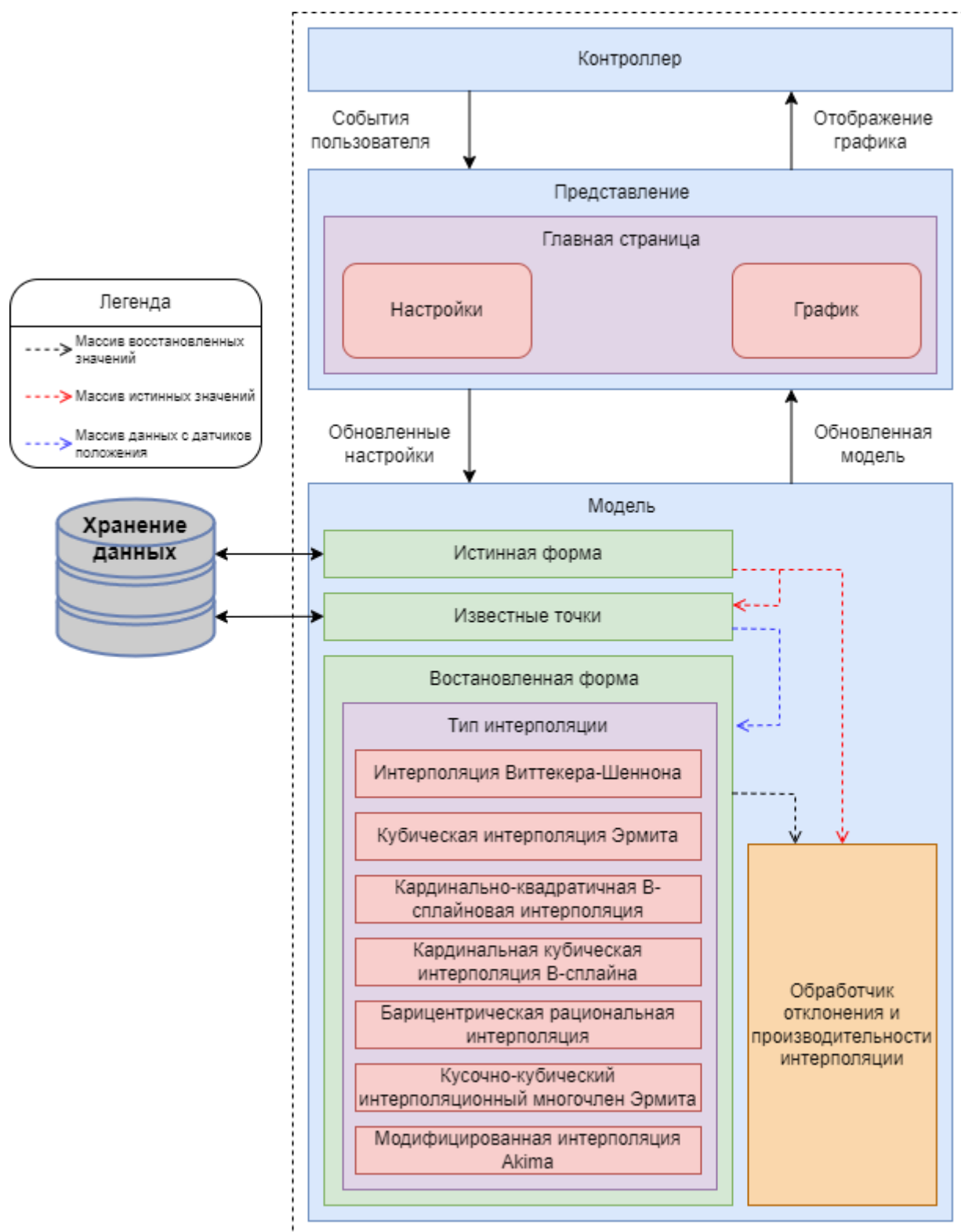


Рисунок 2 — Обобщенная архитектура решения

В этом разделе будет рассмотрена общая архитектура приложения, использующая паттерн MVC, и каждый из компонентов более подробно.

2.3.1 Контроллер.

Уровень контроллера в приложении отвечает за обработку пользовательского ввода и управление взаимодействием между компонентами приложения. Он является посредником между пользователем и моделью, а также управляет представлением, которое отображает данные, полученные от модели.

В данном случае, контроллер обрабатывает различные пользовательские действия, такие как добавление, удаление или изменение точек кривой, выбор определенного вида кривой для визуализации, изменение масштаба и поворота камеры в 3D пространстве и т.д. Контроллер также должен обеспечить правильное взаимодействие с моделью, так что любые изменения, сделанные пользователем, отображены в представлении.

2.3.2 Представление.

Уровень представления отвечает за отображение данных пользователю и обработку пользовательского ввода. В данном случае, главная страница приложения включает в себя два основных элемента:

Панель настройки, которая задает базовую модель, известные точки и метод восстановления, а также управляет визуальной составляющей отображающегося графика.

Поле визуализации, которое отображает график, построенный на основе введенных пользователем параметров. В этом поле пользователь может видеть исходную модель, известные точки, а также восстановленную кривую и взаимодействовать с ней (вращать, приближать/отдалять).

На уровне представления также реализована валидация пользовательского ввода. Более подробный разбор пользовательского интерфейса будет в соответствующем разделе.

2.3.3 Модель.

Уровень модели отвечает за хранение данных и бизнес-логику приложения. Он является независимым от пользовательского интерфейса и контроллера.

Модель представляет собой совокупность информации об истинной форме модели, наборе известных точек, а также восстановленной форме. Кроме того, модель включает блок обработки отклонений и производительности, который производит замеры времени и отвечает за сравнение восстановленной формы с исходной моделью.

Восстановленная форма включает в себя доступ к математическим библиотекам для выполнения процесса интерполяции и построения аппроксимирующих функций на основе имеющихся данных.

Для удобства пользователей приложения, предусмотрена возможность сохранения и загрузки информации. Все установленные данные будут сохранены на локальной машине, и загружены с новым запуском.

2.4 Интерфейс пользователя.

Интерфейс пользователя — это один из ключевых компонентов любого приложения. Он позволяет пользователям взаимодействовать с приложением, управлять его функциями и просматривать результаты работы. Хорошо спроектированный интерфейс должен быть интуитивно понятным, удобным и привлекательным для пользователя. На рис. 3 представлено главное окно приложения.

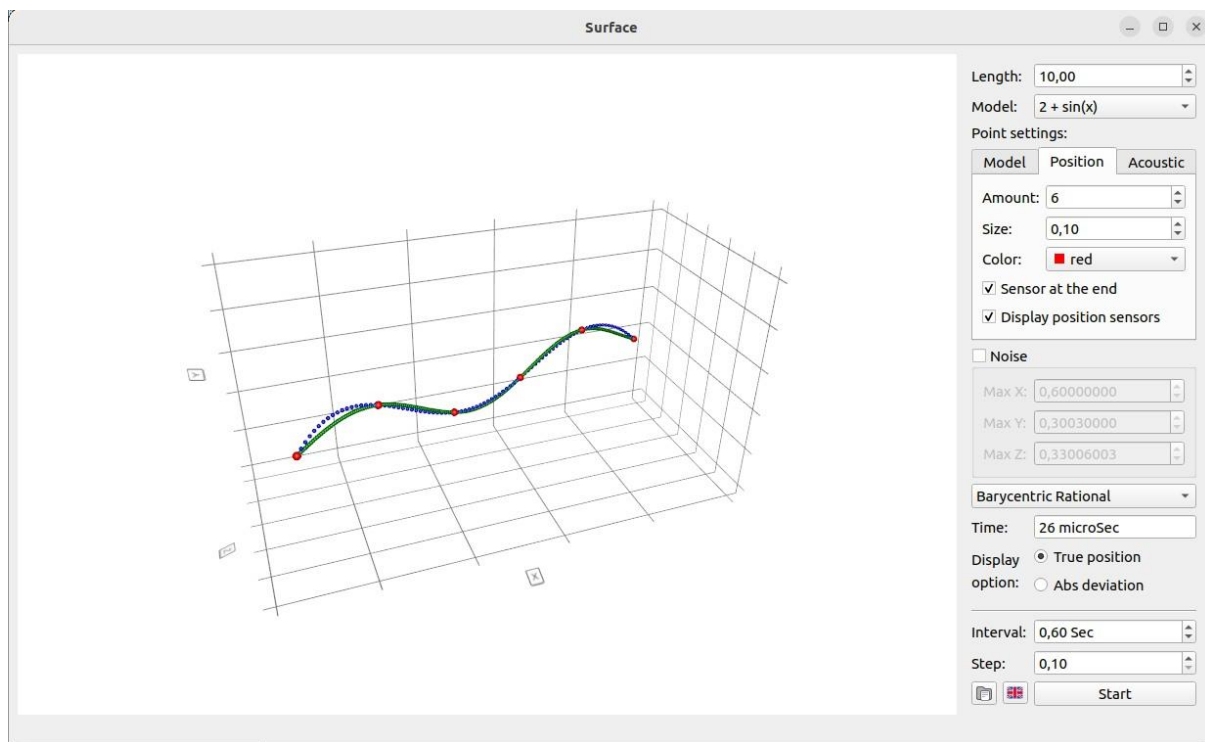


Рисунок 3 — Главное окно приложения

Основную часть всего главного окна занимает поле визуализации смоделированного графика. В данном окне пользователь имеет возможность выбрать наиболее подходящий для себя угол обзора, соответствующее приближение и в общих чертах оценить отклонение выбранного метода восстановления от базовой модели.

Меню настроек предоставляет богатый функционал по настройке исходных данных. Пользователь может влиять на следующие аспекты:

- Длину истинной модели
- Функцию истинной модели
- Количество, цвет и размер всех отображаемых точек
- Наличие отклонений и их максимальные значения по всем осям
- Тип метода восстановления
- Отображаемую информацию при выборе конкретной точки
- Интервал с которым будет обновляться эмуляция
- Шаг на который по графику будет продвигаться эмуляция
- Осуществлять запуск или остановку эмуляции

Настройка визуализаций всех категории точек производится в соответствующих меню, представленных на рис. 4, 5, 6 для истинной модели, известных точек и восстановленной формы соответственно.

The screenshot shows a configuration window with three tabs: 'Model', 'Position', and 'Acoustic'. The 'Model' tab is selected. It contains the following settings: 'Amount' is set to 200, 'Size' is set to 0,05, and 'Color' is set to green (indicated by a green square icon). At the bottom, there is a checked checkbox labeled 'Display model'.

Рисунок 4 — Настройка визуализации для истинной формы

The screenshot shows a configuration window with three tabs: 'Model', 'Position', and 'Acoustic'. The 'Position' tab is selected. It contains the following settings: 'Amount' is set to 88, 'Size' is set to 0,05, and 'Color' is set to blue (indicated by a blue square icon). At the bottom, there is a checked checkbox labeled 'Display acoustic sensors'.

Рисунок 5 — Настройка визуализации для восстановленных данных

The screenshot shows a configuration window with three tabs: 'Model', 'Position', and 'Acoustic'. The 'Position' tab is selected. It contains the following settings: 'Amount' is set to 6, 'Size' is set to 0,10, and 'Color' is set to red (indicated by a red square icon). At the bottom, there are two checked checkboxes: 'Sensor at the end' and 'Display position sensors'.

Рисунок 6 — Настройка для известных точек

Между настройкой точек для истинной формы и восстановленных данных различий нет, они идентичны, на всю длину выбирается количество отображаемых точек, размер, цвет и флаг видимости. Отличие же настроек известных точек состоит в наличии специального флага, который изменяет шаг таким образом, чтобы последняя известная точка находилась строго на конце модели.

Для симуляции отклонений используется меню, представленное на рис. 7.

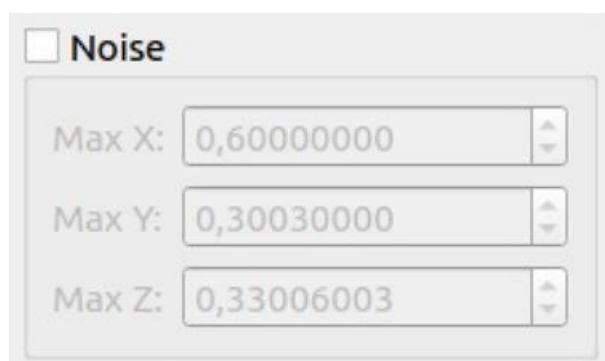


Рисунок 7 — Меню настройки отклонений

Наличие отклонений, а также их максимальное значение, определяет пользователь. Присутствующая валидация не позволяет устанавливать отклонения превышающее определенного значения, которое нарушит явный порядок точек.

Эмуляция движения модели осуществляется по нажатию на кнопку, изображенную на рис. 8.

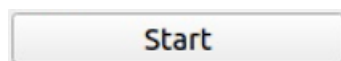


Рисунок 8 — Кнопка начала эмуляции

Таймер с заданным интервалом начинает двигаться по модели, шаг и интервал задается с помощью меню, изображенном на рис. 9.



Рисунок 9 — Меню интервала и шага эмуляции

Если включено отклонение, то на каждый шаг известные точки будут принимать новое отклонение, не превышающее заданного.

Для оценки метода восстановления пользователю предоставлен следующий функционал:

- Окно с временными затратами выбранного метода
- Панель с отклонениями для конкретной точки

На рис. 10 представлено окно, которое оценивает временные затраты, время указывается в микросекундах.

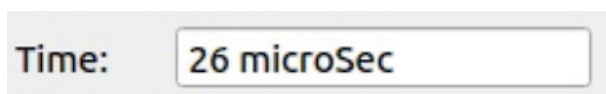


Рисунок 10 — Окно временных затрат выбранного метода восстановления

При выборе конкретной точки на визуализированном графике отображаются её координаты, пример представлен на рис. 11.

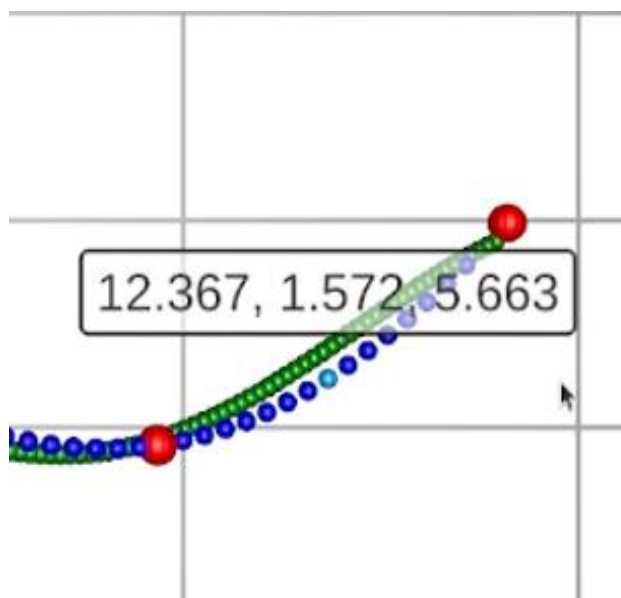


Рисунок 11 — Визуализация координат на графике

Если изменить тип отображения в соответствующем меню, представленном на рис. 12, то отображаемая информация будет указывать на модуль отклонения, пример отображаемого отклонения представлен на рис. 13.

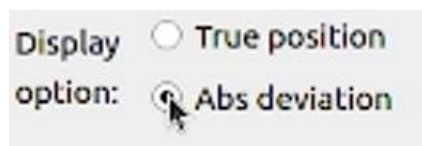


Рисунок 12 — Меню выбора типа отображающейся информации

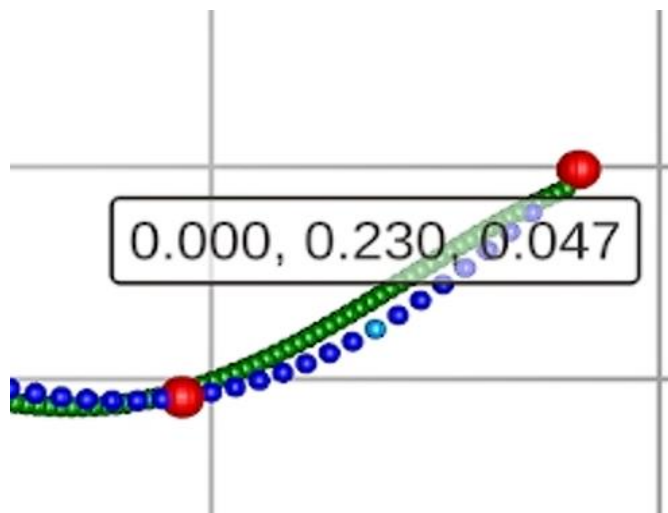


Рисунок 13 — Отклонение восстановленных данных от истинной формы

Для облегчения использования программы предоставлена возможность загрузки исходной модели и известных точек из файла формата .txt, кнопка выбора файла изображена на рис. 14.



Рисунок 14 — Кнопка выбора файла

Программа поддерживает Русский и Английский языки, смена происходит без необходимости перезагрузки программы, кнопка смены языка представлена на рис. 15, а переведенное главное окно на рис. 16.



Рисунок 15 — Кнопка смены языка

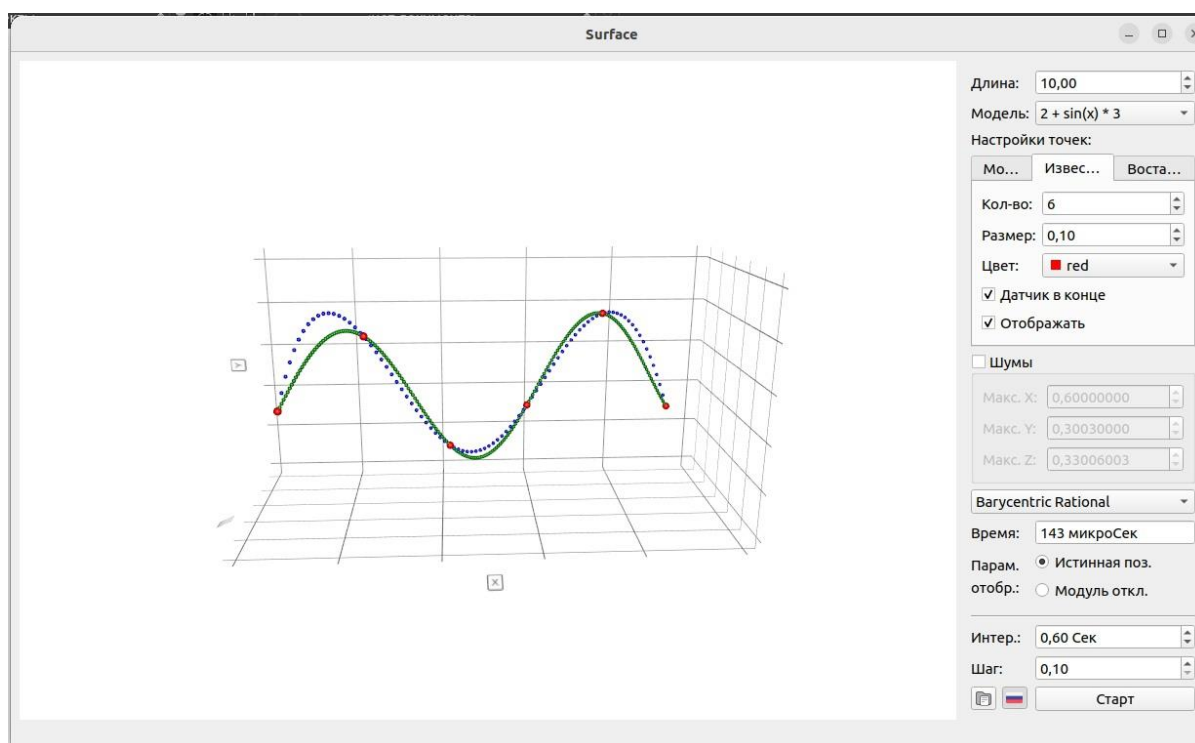


Рисунок 16 — Переведенное главное окно

2.5 Сценарии использования.

Использование данного решения заключается в создании графического отображения и анализе полученных результатов, в таблице 2 представлены сценарии использования.

Таблица 2 — Сценарии использования

Краткое описание	Выбор модели, метода обработки, количества датчиков позиции и настройка отображения.
Действующее лицо	Пользователь
Предусловие	Программа запущена, фокус на главном окне
Основной сценарии	
Пользователь	Система
Выбирает модель, длину	Обновляет поле с визуализированным графиком
Задаёт количество известных точек и параметры отображения	Обновляет поле с визуализированным графиком
Выбирает тип интерполяции и параметры отображения	Обновляет поле с визуализированным графиком

Продолжение таблицы 2

Задаёт время обновления и шаг	
Выставляет галочку в чекбоксе “Помехи”	Поля X, Y, Z в разделе “Помехи” становятся активны
Задает максимальное отклонение датчиков позиции по осям X, Y, Z	
Нажимает кнопку “Start”	Поле с графиком начинает обновляться согласно заданному интервалу и шагу. Датчики позиции на каждый шаг принимают отклонение, не превышающее заданного
Альтернативный сценарии №1 – Загрузка из файла	
Пользователь	Система
Нажимает на кнопку “Читать из файла”	Предоставляет системное окно выбора файла
Выбирает нужный файл	Считывает файл, устанавливает модель, метод обработки, количество датчиков позиции, помехи, шаг, интервал и настройки отображения
Нажимает кнопку “Start”	Поле с графиком начинает обновляться согласно заданному интервалу и шагу. Датчики позиции на каждый шаг принимают отклонение, не превышающее заданного

2.6 Вывод по разделу.

Для разработки программного модуля был выбран фреймворк Qt, который был использован для создания интерфейсного приложения. Исходный код модуля доступен в открытом репозитории на платформе GitHub [20].

3 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

3.1 Подходы тестирования

Тестирование GUI приложения включает в себя проверку правильности отображения графических элементов интерфейса, работоспособности кнопок и других элементов управления, а также общей удобства и эргономики пользовательского интерфейса.

Для производства тестирования GUI десктопного приложения можно использовать следующие подходы:

1. Ручное тестирование. Оценка интерфейса производится вручную. Тестировщик проверяет правильность отображения элементов интерфейса, корректность работы кнопок и других элементов управления, соответствие элементов интерфейса заданным спецификациям, удобство использования приложения.
2. Автоматизированное тестирование. Для этого используются специальные инструменты для тестирования GUI, такие как Selenium, Appium, TestComplete, Sikuli и другие. Эти инструменты позволяют записывать и воспроизводить действия пользователя на уровне интерфейса, а также проверять правильность отображения элементов интерфейса.
3. Тестирование с помощью "черного ящика". Тестировщик не знает внутренней структуры приложения, а проверяет только его внешние характеристики. Этот подход позволяет оценить приложение так, как это сделает реальный пользователь.
4. Тестирование с помощью "белого ящика". Тестировщик имеет доступ к исходному коду приложения и может проводить более глубокий анализ его работы и интерфейса.

Так как имеется доступ к исходному коду, то будет выполнено ручное тестирование с помощью "белого ящика". Для этого рассматривается логика каждого элемента управления и оценивается отзывчивость с удобством разработанного интерфейса.

3.2 Основные характеристики ПК.

Тактовая частота центрального процессора (CPU): 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz;

- Количество ядер центрального процессора: 6 ядер;
- Объем оперативной памяти компьютера (ОЗУ): 32 ГБ
- Объем памяти видеокарты: 2 ГБ;
- Размер свободного пространства на жестком диске или SSD: 256 ГБ;
- Версия Windows: Windows 10 Pro 64-bit

На рис. 17 представлены основные характеристики компьютера.

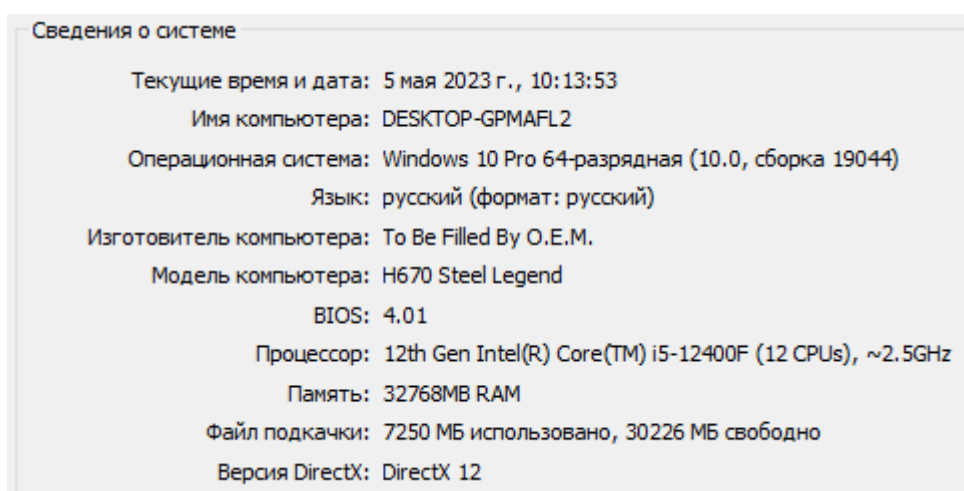


Рисунок 17 — Основные характеристики компьютера

3.3 Тестирование программы.

Первым делом необходимо проверить масштабируемость интерфейса, узнать, как программа изменяет пропорции элементов, расположенных на главном окне при раскрытии на весь экран. На рис. 18 представлено главное окно приложения, раскрытого на весь экран.

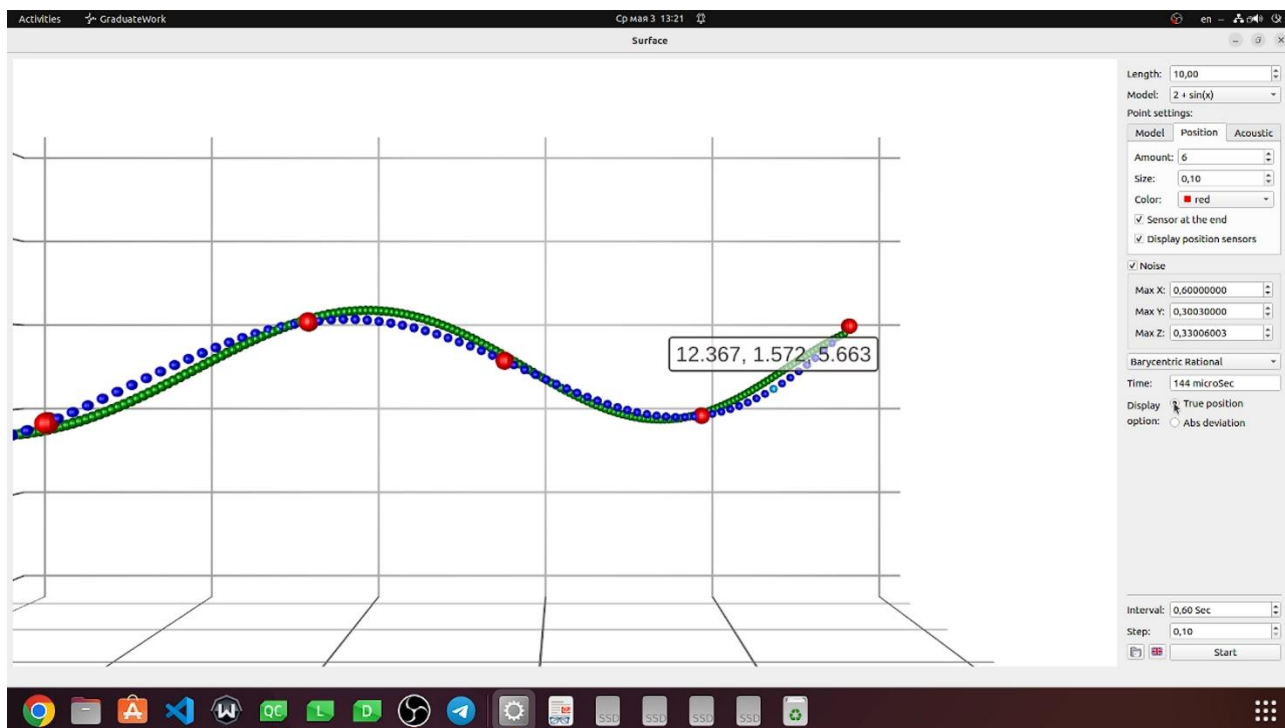


Рисунок 18 — Главное окно приложения

По рисунку 18 видно, что панель настроек сохраняет статический размер по горизонтали, а по вертикали элементы сгруппировались по соответствующим группам. Окно с визуализированным графиком занимает все доступное пространство.

Для всех имеющихся интерактивных объектов необходимо убедиться в правильности предоставленной валидации и логики их работы. Неверно поданные данные или некорректная логика обработки может привести к серьезным ошибкам, вплоть до непроизвольного завершения программы.

Поле ввода чисел необходимо пользователю для задания практически всех характеристик, необходимых для работы приложения. В таблице 3 указаны все элементы, которые принимают численные значения, их верхние и нижние границы, а также результат взаимодействия с ними.

Таблица 3 — Границы валидации и результат взаимодействия с полями, принимающие численные значения

Название поля	Нижняя граница	Верхняя граница	Результат взаимодействия
Длина	1.00	100.00	Изменение размера базовой модели
Количество точек базовой модели	3	10000	Изменение количества видимых точек на базовой модели
Размер точек базовой модели	0.01	1.00	Изменение размера видимых точек базовой модели
Количество точек восстановленных данных	3	10000	Изменение количества видимых точек восстановленных данных
Размер точек восстановленных данных	0.01	1.00	Изменение размера видимых точек восстановленных данных
Количество известных точек	6	10000	Изменение количества известных точек
Размер известных точек	0.01	1.00	Изменение размера отображаемых известных точек
Максимальное отклонение по оси X	0.00000000	Размер шага между известными точками / 2	Изменение максимально допустимого смещения известной точки по оси X
Максимальное отклонение по оси Y	0.00000000	8.00000000	Изменение максимально допустимого смещения известной точки по оси Y
Максимальное отклонение по оси Z	0.00000000	8.00000000	Изменение максимально допустимого смещения известной точки по оси Z
Интервал	0.10	30.00	Изменение интервала, с которым происходит движение по модели
Шаг	0.10	Длина базовой модели	Изменение размера шага по базовой модели

Флажки или CheckBoxs, тоже имеют свои численные значения, но они ограничены лишь двумя состояниями:

- Активен (true или 1)
- Неактивен (false или 0)

В таблице 4 описаны состояния всех имеющихся флажков приложения.

Таблица 4 — Описание состояния флажков

Название флажка	Активное состояние	Неактивное состояние
Показать базовую модель	Все точки базовой модели видны	Все точки базовой модели скрыты
Показать восстановленные данные	Все точки восстановленных данных видны	Все точки восстановленных данных скрыты
Показать известные точки	Отображены все известные точки	Скрыты все известные точки
Разместить известную точку на конце базовой модели	Пересчитывается шаг между известными точками таким образом, что последняя точка располагается на конце базовой модели	Шаг между известными точками высчитывается по формуле: длина базовой модели / кол-во известных точек

Переключатели или RadioButtons в отличии от флажков имеют одну интересную особенность: в один момент времени активным может быть только один переключатель. В таблице 5 указано влияние на программу активного переключателя.

Таблица 5 — Описание активного состояния переключателей

Название переключателя	Активное состояние
Истинная позиция	При выборе точки восстановленных данных на визуализированном графике, отображаемая информация будет указывать на позицию в пространстве
Модуль отклонения	При выборе точки восстановленных данных на визуализированном графике, отображаемая информация будет указывать модуль отклонения от базовой модели

Поле с выпадающим списком или ComboBox, представляет из себя список возможных значений, пользователь может ознакомиться со всеми представленными вариантами и выбрать нужное ему. В таблице 6 представлены все возможные значения выпадающих списков, а также их влияние на программу. Таблица 6 — Значения выпадающих списков и реакция приложения на выбор элемента

Название выпадающего списка	Значения списка	Реакция программы при выборе
Цвет точек отображения базовой модели/известных точек/восстановленных данных	Красный	Цвет отображаемых точек сменяется на выбранный из списка
	Зеленый	
	Синий	
	Золотой	
	Розовый	
	Оранжевый	
	Лаймовый	
Тип интерполяции	Интерполяция Виттекера-Шеннона	Перерасчет восстановленных данных по выбранному типу интерполяции
	Кубическая интерполяция Эрмита	
	Кардинально-квадратичная В-сплайновая интерполяция	
	Кардинально кубическая интерполяция В-сплайна	
	Барицентрическая рациональная интерполяция	
	Модифицированная интерполяция Акима	

Кнопки могут выполнять различные функции, такие как сохранение данных, запуск приложений, открытие файлов, переход на другие страницы и т.д.

Кнопки в зависимости от их назначения могут быть текстовыми или содержать иконки. В таблице 7 указана реакция приложения от нажатия конкретной кнопки.

Таблица 7 — Реакция программы на нажатия кнопки

Название кнопки	Реакция программы
Старт/Стоп	Запускает/останавливает эмуляцию
Открыть файл	Открывает меню выбора файла, где пользователь должен выбрать текстовый файл с позициями базовой модели и известных точек
Сменить язык	Меняет язык на Русский/Английский

3.4 Вывод по разделу.

В ходе выполнения тестирования были разобраны все элементы управления, масштабируемость, реакция системы на взаимодействия с ними, а также указаны допустимые значения валидации.

4 НОРМАТИВНО-ПРАВОВОЕ РЕГУЛИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

4.1 Общие положения.

Под интеллектуальной собственностью понимают особый вид гражданских прав (исключительное право) в отношении результатов интеллектуальной деятельности, таких как изобретения, промышленные образцы(дизайн), компьютерные программы, другие произведения науки, произведения литературы, искусства, которые принято называть объектами интеллектуальной собственности, а также различных средств индивидуализации производителя товаров и услуг, таких как товарные знаки, знаки обслуживания, фирменные наименования и др. Основным содержанием таких прав является монополия их владельца на использование этих объектов, включая право запретить или разрешить их использование другим, а также право переуступить другому лицу эти правомочия или отказаться от них вовсе.

Согласно определению интеллектуальной собственности, принятому в российском законодательстве, а также на основании определения Стокгольмской конференции от 14 июля 1967 г., программы для ЭВМ (компьютерные программы) и базы данных относятся к объектам интеллектуальной собственности. Программам для ЭВМ и базам данных предоставляется охрана нормами авторского права как литературным произведениям в соответствии с Бернской конвенцией, причем программы для ЭВМ охраняются как литературные произведения, а базы данных — как сборники.

В Российской Федерации вопросы предоставления правовой охраны программам для ЭВМ и базам данных регулируются Гражданским кодексом РФ, Часть 4 (ГК РФ Ч. 4).

Под программой для ЭВМ понимается *«. . . представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств в целях получения определенного результата»*. Кроме того, в понятие программы для ЭВМ входят *«. . .*

подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения».

С точки зрения программистов и пользователей программа для ЭВМ представляет собой детализацию алгоритма решения какой-либо задачи и выражена в форме определенной последовательности предписаний, обеспечивающих выполнение компьютером преобразования исходных данных в искомый результат. Можно выделить следующие объективные формы представления программы для ЭВМ:

- исходная программа (или исходный текст) — последовательность предписаний на алгоритмическом (понятном человеку) языке высокого уровня, предназначенных для автоматизированного перевода этих предписаний в последовательность команд в объектном коде;
- рабочая программа (или объектный код) — последовательность машинных команд, т. е. команд, представленных на языке, понятном ЭВМ;
- программа, временно введенная в память ЭВМ, — совокупность физических состояний элементов памяти запоминающего устройства ЭВМ(ОЗУ), сохраняющихся до прекращения подачи электропитания к ЭВМ;
- программа, постоянно хранимая в памяти ЭВМ, — представленная на языке машины команда (или серия команд), выполненная в виде физических особенностей участка интегральной схемы, сохраняющихся независимо от подачи электропитания.

Исходная и рабочая программы, как правило, представляются в виде записи на том или ином языке, выполненной на бумаге или машиночитаемом носителе данных: магнитном или оптическом диске, магнитной ленте и т. п.

Предоставляемая законодательством правовая охрана распространяется «. . . на все виды программ для ЭВМ (в том числе на операционные системы и программные комплексы), которые могут быть выражены на любом языке и

в любой форме, включая исходный текст и объектный код . . .». Преобразование исходного текста программы для ЭВМ в объектный (машинный) код не меняет сущности данной программы как произведения. Значит, если охраняется исходный текст программы, то охране подлежит и соответствующий ей объектный код. Обратное тоже справедливо. Правовая охрана программ для ЭВМ распространяется только в отношении формы их выражения и *« . . . не распространяется на идеи, концепции, принципы, методы, процессы, системы, способы, решения технических, организационных или иных задач, открытия, факты, языки программирования»* [21].

4.2 Описание объекта разработки.

В ходе выполнения данного дипломного проекта создана программа для ЭВМ *«Инструмент моделирования и визуализации формы пространственной кривой, восстановленной по данным, изменяющимся в реальном времени»*, которая является объектом интеллектуальной собственности и которой может быть предоставлена правовая охрана в рамках авторского права. Автором является Попов Дмитрий Сергеевич. Правообладателем является СПбГЭТУ «ЛЭТИ», т. к. программа создана при выполнении дипломной работы.

4.3 Регистрация программы ЭВМ.

В течение периода действия эксклюзивного права на программу для компьютера, её владелец может зарегистрировать её в федеральном органе по защите интеллектуальной собственности [22]. Однако следует учитывать, что программы, содержащие государственную тайну, не могут быть зарегистрированы. В случае нарушения этого правила, ответственность несет заявитель. Для регистрации программы необходимо подать заявление со следующими данными:

- Заявление о государственной регистрации с указанием правообладателя и автора, если он не отказался от упоминания.

- Депонируемые материалы, идентифицирующую программу для ЭВМ.

Если условия не соблюдаются, федеральная служба по интеллектуальной собственности направляет заявителю уведомление, после чего заявитель может предоставить исправленные документы в течение трех месяцев [22].

Для регистрации программы также требуется уплата пошлины. Федеральная служба проверяет этот факт и в случае, если оплата не была произведена или была произведена не в полном объеме, заявителю будет отправлено уведомление с требованием оплаты в течение двух месяцев со дня отправления уведомления.

После подачи заявки, федеральная служба проверяет наличие и правильность необходимых документов и материалов, и затем принимает решение о включении программы в Реестр программ для компьютеров. Причины для отказа могут быть следующие:

- Неуплата пошлины
- Представление документов вне установленного срока
- Нарушение условий государственной регистрации

Правообладатель также может передать исключительные права другому лицу. Для этого он должен подать заявление в орган исполнительной власти по защите интеллектуальной собственности, после чего изменения будут внесены в Реестр. Информация, занесенная в Реестр, считается достоверной, если не доказано обратное. Ответственность за правильность предоставленных данных несет заявитель.

4.4 Реализация прав государства на объекты интеллектуальной собственности.

При заключении государственных контрактов и договоров, связанных с реализацией прав государства на результаты научно-технической деятельности, организации, на которые распространяются эти права, обязаны предоставлять следующую информацию в уполномоченные федеральные органы:

- Сведения о заявках на охрану объектов интеллектуальной собственности.
- Информацию о получении охраняемых документов.
- Детали сделок, касающихся прав на результаты научно-технической деятельности.
- Информацию об использовании результатов научно-технической деятельности.

4.5 Налогообложение.

Статья 149 НК РФ устанавливает, что продажа исключительных прав на программы для ЭВМ, включенных в единый реестр российских программ, а также прав на использование таких программ, в том числе удаленный доступ к ним через Интернет, не облагается налогом. Однако, права, которые связаны с распространением или получением рекламной информации через Интернет, подлежат налогообложению [23].

Также при определении налоговой базы не учитываются доходы в виде исключительных прав на программы для ЭВМ, созданных при реализации государственного контракта [23].

Одной из особенностей налогообложения программ для компьютеров является то, что они относятся к нематериальным активам [23]. В этом случае, первоначальная стоимость амортизируемых нематериальных активов определяется как сумма расходов на их приобретение и подготовку к использованию. Если нематериальный актив создается организацией, то его стоимость определяется как сумма фактических расходов на его создание. Остаточная стоимость нематериальных активов определяется как разница между их первоначальной стоимостью и суммой амортизации, начисленной за период эксплуатации. Существуют также особенности, связанные с тем, что расходы, связанные с приобретением права на использование программ на компьютере по договорам с правообладателем, относятся к прочим расходам [23].

4.5 Распространение в сети «интернет».

Если информация распространяется через сеть "Интернет", то организатор несет несколько обязательств в соответствии с законом [24]:

1. Он должен уведомить уполномоченный федеральный орган о запуске программ, которые предназначены или используются для передачи, доставки, обработки или приема сообщений пользователей сети "Интернет".
2. Он должен хранить информацию о фактах передачи, доставки, обработки или приема сообщений пользователей сети "Интернет" и информацию о пользователях, и предоставлять её уполномоченным государственным органам.
3. Он должен предоставлять необходимую информацию для декодирования электронных сообщений, которые передаются, доставляются, принимаются или обрабатываются.
4. Он должен обеспечить выполнение указанных требований для программного и технического обеспечения, используемого им в работе.

Несоблюдение указанных требований приводит к наложению административного штрафа, размер которого зависит от типа нарушения и нарушителя. В случае повторного нарушения размер штрафа увеличивается. Размеры штрафа представлены в таблице 8.

Таблица 8 — Размеры штрафа для правонарушения в сети «интернет»

Тип нарушения	Нарушитель	Пункт 1	Пункт 2	Пункт 3	Пункт 4
Первичное	Гражданин	1-3 тыс. руб.	3-5 тыс. руб.	3-5 тыс. руб.	3-5 тыс. руб.
	Должностное	10-30 тыс. руб.	30-50 тыс. руб.	30-50 тыс. руб.	30-50 тыс. руб.
	Юридическое	100-300 тыс. руб.	0.8-1 млн. руб.	0.8-1 млн. руб.	300-500 тыс. руб.

Продолжение таблицы 8

Повторное	Гражданин	5-10 тыс. руб.	15-30 тыс. руб.	15-30 тыс. руб.	15-30 тыс. руб.
	Должностное	50-100 тыс. руб.	100-500 тыс. руб.	100-500 тыс. руб.	100-500 тыс. руб.
	Юридическое	0.5-1 млн. руб.	2-6 млн. руб.	2-6 млн. руб.	2-6 млн. руб.

4.6 Преступления в сфере компьютерной информации.

Глава 28 УК РФ определяет преступления, связанные с компьютерной информацией, и соответствующие наказания за них:

1. Создание, распространение или использование компьютерных программ или информации, которые могут несанкционированно уничтожить, скопировать, заблокировать или изменить компьютерную информацию или нейтрализовать средства защиты. Наказание за нарушение может быть следующим:
 - Ограничение свободы на срок до четырех лет
 - Принудительные работы на срок до четырех лет
 - Лишение свободы на тот же срок со штрафом до 200 тыс. рублей или дохода за период до 18 месяцев.
2. Нарушение правил использования средств хранения, обработки, передачи охраняемой компьютерной информации, блокирование, уничтожение, копирование или изменение информации, которое причинило значительный ущерб. Наказание за нарушение может быть следующим:
 - штраф в размере до 500 тыс. рублей или дохода за период до 18 месяцев исправительные работы на срок от шести месяцев до одного года
 - ограничение свободы на срок до двух лет принудительные работы на срок до двух лет
 - лишение свободы на срок до двух лет

Если правонарушение привело к тяжким последствиям:

- Принудительные работы на срок до пяти
- Лишение свободы на срок до пяти лет

ЗАКЛЮЧЕНИЕ

Во время выполнения работы были рассмотрены следующие алгоритмы восстановления:

- Интерполяция Уиттакера-Шеннона
- Кубическая интерполяция Эрмита
- Кардинально-квадратичная В-сплайновая интерполяция
- Кардинально-кубическая В-сплайновая интерполяция
- Барицентрическая рациональная интерполяция
- Модифицированная интерполяция Акима

Определены требования для успешного написания программного продукта:

- Продукт должен включать один или несколько алгоритмов восстановления.
- В программной системе должны использоваться методы, реализованные в готовых математических библиотеках.
- Программная система должна иметь возможность менять условия или параметры используемых алгоритмов интерполяции в зависимости от ситуации и предоставлять удобный пользовательский интерфейс.

Разработан инструмент моделирования и визуализации формы пространственной кривой, восстановленной по данным, изменяющимся в реальном времени, который полностью соответствует заданным требованиям, а также включает в себя все рассмотренные алгоритмы восстановления.

Произведено ручное тестирования пользовательского интерфейса по методу “белого ящика”, в котором были подробно рассмотрены все элементы управления, а также реакция системы на взаимодействие с ними.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. А.В. Желтаков, С.А. Семенова. Система ориентации гидроакустической гибкой протяженной буксируемой антенны (ГПБА). Средства и методы подводных исследований. ОАО “Концерн Океанприбор”.
2. A Nonlinear Data-Driven Towed Array Shape Estimation Method Using Passive Underwater Acoustic Data. Key Laboratory of Underwater, Acoustic Signal Processing of Ministry of Education, Southeast University, Nanjing 210096, China
3. Measurement of towed array position, shape, and attitude. IEEE, October 1989
4. В.Ф. Жирков, Л.Т. Сушкова, А.И. Королев, К.Н. Большаков, А.А. Обеднин, Г.В. Прокофьев. Полиномиальная интерполяция в цифровой обработке сигналов при высоких требованиях к точности. Журнал радиоэлектроники, ISSN 1684-1719, №4, 2017.
5. Интерполяция [Электронный ресурс] URL: https://en.wikipedia.org/wiki/1911_Encyclop%C3%A6dia_Britannica/Interpolation (дата обращения 07.05.2023)
6. Полином Лагранжа [Электронный ресурс] URL: https://encyclopediaofmath.org/index.php?title=Lagrange_interpolation_formula (дата обращения 07.05.2023)
7. Полином Ньютона [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Newton_polynomial (дата обращения 07.05.2023)
8. Интерполяционный полином Эрмита [Электронный ресурс] URL: <https://mathworld.wolfram.com/HermiteInterpolatingPolynomial.html> (дата обращения 07.05.2023)
9. Интерполяционная формула Уиттакера-Шеннона [Электронный ресурс] URL: <https://dic.academic.ru/dic.nsf/ruwiki/641458> (дата обращения 07.05.2023)

10. В. Г. Чередниченко, “Рациональная интерполяция, аналитическое решение”, Сиб. матем. журн., 43:1 (2002), 188–193; Siberian Math. J., 43:1 (2002), 151–155
11. Интерполяция сплайнами: теоретические основы [Электронный ресурс] URL: <http://statistica.ru/branches-maths/interpolyatsiya-splaynami-teor-osnovy/> (дата обращения 07.05.2023)
12. Модифицированная интерполяция Акима [Электронный ресурс] URL: <https://blogs.mathworks.com/cleve/2019/04/29/makima-piecewise-cubic-interpolation/#3b78d49a-0e90-4c1d-9f16-49a23a175132> (дата обращения 07.05.2023)
13. Журнал атмосферных и океанических технологии [Электронный ресурс] URL: <https://journals.ametsoc.org/view/journals/atot/37/4/jtech-d-19-0211.1.xml> (дата обращения 07.05.2023)
14. Типы В-сплайнов [Электронный ресурс] URL: <https://bsplines.org/flavors-and-types-of-b-splines/> (дата обращения 07.05.2023)
15. Кубические В-сплайны [Электронный ресурс] URL: http://www.cs.tau.ac.il/~turkel/notes/numeng/spline_note.pdf (дата обращения 07.05.2023)
16. Практическое программирование на C++ [Электронные ресурсы] URL: <https://archive.org/details/acceleratedcprac2000koen/page/n7/mode/2up> (дата обращения 07.05.2023)
17. Boost C++ libraries [Электронный ресурс] URL: https://www.boost.org/doc/libs/1_81_0/libs/math/doc/html/index.html (дата обращения 05.05.2023)
18. Qt [Электронный ресурс] URL: <https://www.qt.io/> (дата обращения 05.05.2023)
19. Шаблон проектирования MVC [Электронный ресурс] URL: <https://www.geeksforgeeks.org/mvc-design-pattern/> (дата обращения 07.05.2023)

20. Исходный код разработанного приложения [Электронный ресурс] URL: https://github.com/DimonPopov/PopovDS_9304_GraduateWork (дата обращения 07.05.2023)
21. Гражданский кодекс Российской Федерации. (Часть четвертая) от 18.12.2006, № 230-ФЗ
22. О государственной регистрации распоряжения исключительным правом на изобретение, полезную модель, промышленный образец, товарный знак, знак обслуживания, зарегистрированные топологию интегральной микросхемы, программу для ЭВМ, базу данных по договору и перехода исключительного права на них без договора: постановление Правительства Рос. Федерации от 24.12.2015 № 1416
23. Налоговый кодекс Российской Федерации (Часть вторая) от 05.08.2000, № 117-ФЗ
24. Кодекс Российской Федерации об административных правонарушениях от 30.12.2001, № 195-ФЗ