

Санкт-Петербургский государственный университет

*Лень Юлия Александровна*

Выпускная квалификационная работа

Разработка прототипа системы  
моделирования навигации роботов в  
условиях неопределенности

Уровень образования: магистратура

Направление *09.04.04 «Программная инженерия»*

Основная образовательная программа *ВМ.5666.2019 «Программная инженерия»*

Научный руководитель:  
д.ф.-м.н., проф., О.Н. Граничин

Рецензент:  
инженер по машинному обучению ООО «Звук» Д.Р. Ужва

Санкт-Петербург  
2024

Saint Petersburg State University

*Julia Len*

Master's Thesis

# Development of Simulation System for Robot Navigation under Uncertainty

Education level: master

Speciality *09.04.04 «Software Engineering»*

Programme *BM.5666.2019 «Software Engineering»*

Profile: *Software Engineering*

Scientific supervisor:  
Sc.D, prof. O.N. Granichin

Reviewer:  
machine learning engineer at 'Zvuk' LLC D.N. Uzhva

Saint Petersburg  
2024

# Оглавление

<b>1. Введение</b>	<b>4</b>
<b>2. Постановка задачи</b>	<b>5</b>
<b>3. Обзор</b>	<b>6</b>
3.1. Навигационные системы . . . . .	6
3.2. Задача коррекции навигационной системы . . . . .	8
<b>4. Реализация</b>	<b>12</b>
4.1. Инструменты для реализации . . . . .	12
4.2. Компоненты системы . . . . .	13
4.3. Графический интерфейс прототипа системы . . . . .	15
4.4. Программная реализация методов поиска фрагмента . .	16
<b>5. Эксперименты</b>	<b>18</b>
5.1. Описание эксперимента . . . . .	18
5.2. Работоспособность алгоритмов при разных видах шумов и искажении фрагмента . . . . .	18
5.3. Скорость работы алгоритмов . . . . .	19
5.4. Общие выводы по экспериментам . . . . .	21
<b>6. Заключение</b>	<b>22</b>
<b>Список литературы</b>	<b>23</b>

# 1. Введение

Роботизация в разных сферах жизни человека значительно выросла в последние годы [3]. Активное развитие в науке и технике активно поспособствовало этому явлению. Сегодня роботы выполняют самый широкий круг задач: от сложных хирургических операций до доставки посылок в труднопроходимые местности. Однако вместе с развитием технологий в области робототехники выявился ряд проблем, которые необходимо решать. Одной из таких является навигация роботов в условиях неопределенности [1].

Неопределенность может возникнуть из-за различных факторов, таких как изменение окружающей среды, неточность измерений в датчиках, наличие препятствий. В таких условиях традиционные методы навигации могут оказаться неэффективными, поэтому необходимо разрабатывать новые методы, которые смогут адаптироваться к изменяющимся условиям.

В рамках навигации актуальным вопросом является выбор алгоритма, который будет использоваться для решения задачи. В зависимости от типа навигационной системы и доступных для анализа данных происходит первоначальный выбор группы алгоритмов, позволяющих достичь поставленной цели (например, доехать из точки А в точку Б).

Однако прежде чем использовать алгоритмы на практике и реализовывать их для конкретных платформ и типов роботов, необходимо проверить, как разработанный или существующий алгоритм будет вести себя при разных условиях. В связи с этим встает вопрос о разработке системы, где будет реализована возможность промоделировать поведение робота с разными алгоритмами и выбрать тот, который будет лучше всего подходить в рамках поставленной перед роботом и человеком задачей.

## 2. Постановка задачи

Целью работы является разработка прототипа системы моделирования навигации роботов в условиях неопределенности. Для достижения поставленной цели необходимо выполнить следующие задачи:

- выполнить обзор методов в области моделирования навигации роботов;
- разработать прототип системы моделирования для алгоритмов навигации в условиях неопределенности;
- провести эксперименты с разными модификациями данных.

## 3. Обзор

### 3.1. Навигационные системы

Навигация при разработке роботов является одной из наиболее важных задач. Широкий спектр, решаемых роботами, привел к широкой классификации внутри навигационных систем (НС), которые можно разделить на несколько видов категорий, в зависимости от разных критериев разбиения. По типу работы с координатами [6] НС разделяют на:

- глобальные НС;
- локальные НС;
- персональные НС.

В глобальных НС позиция робота рассчитывается в абсолютных координатах. Например, значения ширины и долготы при использовании GPS являются ярким примером такой системы. В локальных же наоборот, система выбирает некую условную точку начала координат и запоминает позиционирование относительно этой точки. В качестве примера можно вспомнить робот-пылесос, у которого заранее известна ограниченная область, где он будет передвигаться, а точкой отсчета робот считает свою станцию. Персональная НС чаще всего используется для роботов-манипуляторов, так как положение всех объектов считается относительно частей робота.

Также в навигации системы подразделяют по принципу получения информации на активные и пассивные [7]. Активные НС рассчитывают свое положение самостоятельно, в то время как пассивные получают эту информацию извне. Исходя из этого описания можно сделать вывод, что в большинстве случаев глобальные НС получают информацию пассивно, персональные — активно, а локальные могут быть как активными, так и пассивными.

НС также разделяют по используемым типам датчиков, которые используют для получения информации об окружающем пространстве.

Например, можно выделить следующие наиболее популярные виды:

- Спутниковые навигационные системы [2]. Одними из самых известных представителей данной категории являются GPS и ГЛОНАСС. Эти системы используют сигналы от спутников для определения и корректировки местоположения объекта.
- Инерциальные навигационные системы (ИНС) [5]. Для этих систем характерно использование акселерометров, гироскопов и других датчиков для измерения ускорения и угловой скорости объекта. На основе этих параметров НС производит ориентацию и корректировку своего положения в пространстве.
- Визуальные навигационные системы [16]. Определение местоположения в рамках этого подхода производится с помощью изображений с камер. На полученных изображениях выделяются некие ключевые объекты и их положение относительно друг друга и относительно объекта НС (например, робота), благодаря чему происходит ориентация в пространстве.
- Геомагнитные навигационные системы [11]. Данные НС используют данные карт магнитного поля Земли для определения своего местоположения.
- Радиочастотные навигационные системы. С помощью сигналов от радиомаяков НС определяет местоположения объекта.

В рамках работы будут рассмотрены корреляционно-экстремальные навигационные системы, так как именно этот тип систем использует корреляцию между наблюдаемым сигналом и эталонным сигналом для определения местоположения объекта. Эта разновидность НС широко используется в навигации и управлении движением [4, 9] особенно в условиях высокой неопределенности или отсутствия точных данных о местоположении.

## 3.2. Задача коррекции навигационной системы

Среди задач в области навигации стоит отметить задачу корректировки НС по данным датчика и карты [8], так как в последнее время ее актуальность растет. Интерес к ней вызван проблемой поиска альтернативы спутниковым системам и работой с зашумленными данными.

В рамках корректировки важным шагом является поиск текущего положения объекта, если известны только последние несколько точек маршрута и карта местности. В представленной выпускной квалификационной работе будут рассмотрены варианты решения этой подзадачи корректировки НС с учетом разных вариантов искажения информации.

### 3.2.1. Поиск положения на карте

Пусть известен маршрут передвижения и его карта высот. Карту высот для маршрута строит, например, Яндекс Карта [18], если выбрать вариант путешествия "Пешком" в мобильном приложении (Рис. 1).

При нажатии на маршрут появляется детализация (Рис. 2), показывающая количество шагов, крутые спуски и подъемы, приблизительную длительность маршрута, время прибытия в пункт назначения, а также гистограмму с высотами вдоль всего маршрута.

Пусть даны гистограмма или карта высот для маршрута и фрагмент, который был пройден и записан в ограниченный буфер с помощью некоего измерителя. Задача состоит в том, чтобы найти положение записанного в буфер фрагмента на маршруте. Поиск положения может осложняться тем, что измеритель может получать данные с погрешностью или искажаться в силу внешних воздействий на измеритель. Например, в процессе измерения высоты, датчик обо что-то ударился. В результате полученное в этот момент измерение будет содержать в себе погрешность самого измерителя и искажение измерения в результате удара.

В рамках работы будет рассмотрено три метода для поиска фрагментов маршрута: метод перебора, метод корреляции и метод на основе быстрого преобразования Фурье.

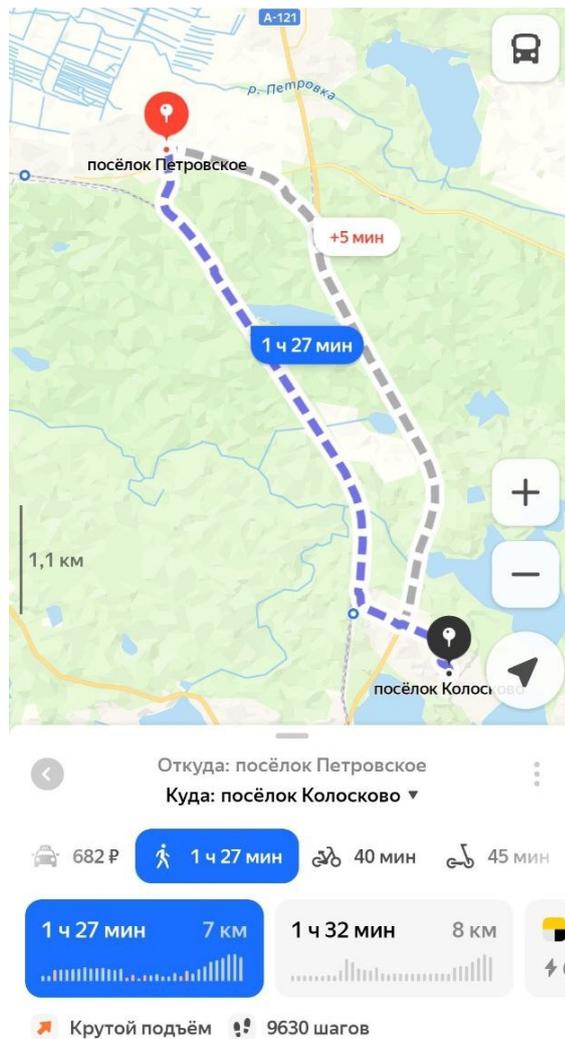


Рис. 1: Скриншот мобильного приложения Яндекс Карты с отображением маршрута на карте (фиолетовая линия) и карты высот маршрута.

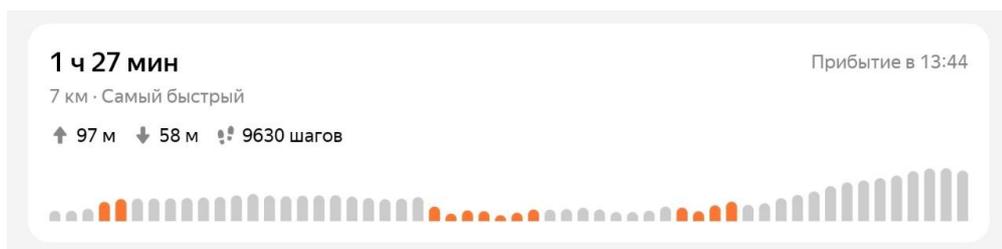


Рис. 2: Скриншот мобильного приложения Яндекс Карты: отображение карты высот маршрута.

### 3.2.2. Метод перебора

Метод перебора является самым простым методом поиска данных в массиве. В рамках текущей задачи искомый фрагмент будет последовательно сравниваться пооконно с оригинальной картой. Главным пре-

имуществом этого метода является его простота и скорость вычислений, однако данный подход перестает работать в случае зашумленных данных.

### 3.2.3. Метод корреляции

Метод корреляции также основан на сравнении фрагмента с каждым доступным участком всего маршрута, только теперь для каждого участка дороги рассчитывается значение корреляции фрагментов.

Пусть  $M$  — это маршрут (карта) длины  $m$  карты высот,  $P$  — пройденный маршрут длины  $p$ , измеренный датчиком,  $M^j \in M$  — это фрагмент маршрута равный по длине пройденному маршруту. Теперь для всех фрагментов  $j = 1, 2..M-P$  рассчитаем значение корреляции по следующей формуле:

$$r_j = \frac{\sum_{i=1}^p (M_i^j - \bar{M}^j)(P_i - \bar{P})}{\sqrt{\sum_{i=1}^p (M_i^j - \bar{M}^j)^2 \sum_{i=1}^p (P_i - \bar{P})^2}}, \quad (1)$$

где  $\bar{M}^j$ ,  $\bar{P}$  — средние значения фрагмента карты и пройденного маршрута соответственно. Так как значение коэффициента корреляции находится в диапазоне от  $-1$  до  $1$ , то для того, чтобы найти фрагмент, где значение коэффициента максимально приближено к единице.

Преимущество такого подхода заключается в том, что он позволяет найти фрагменты, максимально похожие на измеренный путь, даже если измерения были произведены с помехами. Однако вычислительная скорость такого подхода больше, чем у перебора, так как теперь для каждого фрагмента необходимо рассчитать коэффициент корреляции, а потом найти максимум среди них.

### 3.2.4. Метод на основе преобразования Фурье

Преобразование Фурье активно применяется в областях, связанных с анализом временных рядов и поиском похожих фрагментов. В частности можно использовать преобразование Фурье для расчета взаимной корреляции двух функций.

Пусть этими функциями будут карта и искомый фрагмент, тогда алгоритм поиска пройденного пути на карте с использованием быстрого преобразования Фурье (БПФ, FFT), будет выглядеть следующим образом:

1.  $M$  — карта,  $P$  — пройденный маршрут (фрагмент).
2. Вычитаем среднее значение из каждого элемента фрагмента:  $P_{cp} = P - \bar{P}$ .
3. Расширяем фрагмент  $P_{cp}$  до размера карты и заполняем нулями новые значения.
4. Переворачиваем расширенный фрагмент таким образом, что первый элемент становится последним, а последний первым. Обозначим полученный массив  $P_{inv}$ .
5. Применяем БПФ к карте и расширенному фрагменту:  $M_{fft} = FFT(M)$ ,  $P_{fft} = FFT(P_{inv})$ .
6. Перемножаем поэлементно преобразованные карту и фрагмент:  $PM = M_{fft} \cdot P_{fft}$ .
7. Применяем обратное преобразование Фурье к произведению:  $PM_{rfft} = RFFT(PM)$ .
8. Ищем индекс, где последовательность имеет максимальное значение. Это и будет искомым положением фрагмента.

Данный подход позволяет выполнить расчет корреляции более быстрым способом.

## 4. Реализация

### 4.1. Инструменты для реализации

В главе будут описаны инструменты, с помощью которых был реализован прототип системы, и основные компоненты, из которых она состоит.

Для программной реализации прототипа системы был выбран язык программирования Python версии 3.10 [14]. С помощью интерактивного блокнота Jupyter [13] проводилось предварительное тестирование методов. Jupyter дает возможность выполнять код по частям, что упрощает процесс тестирования и локальную отладку функций. Также данный инструмент позволяет отображать графики, диаграммы и другие графические представления результатов на той же странице, что и код.

Графический интерфейс разработан при помощи Qt Designer [15]. Эта среда является кросс-платформенным инструментом для разработки графических интерфейсов приложений. Разработанный интерфейс сохраняется в файл, который подключается к создаваемой системе с помощью специальных методов библиотеки Qt для языка программирования Python. Файл с описанием интерфейса записывается в xml-формате, что позволяет при необходимости внести изменения в интерфейс в любом текстовом редакторе.

В качестве источника данных для карты высот используются данные, предоставленные ресурсом Open Topo Data [12]. Доступ к картам осуществляется с помощью общедоступного API, который дает доступ к несколькими открытыми наборами данных высот. Более известным источником карт высот является API Google Maps Elevation [17]. Open Topo Data обладает совместимостью с сервисом от Google, но в отличие от последнего является бесплатным, что становится одним из основных преимуществ источника. Однако возможность бесплатно пользоваться картами высот с помощью Open Topo Data накладывает определенные технические ограничения при его использовании:

- В одном запросе можно получить значения высот по координатам

GPS максимум для 100 точек.

- Максимальное количество запросов — 1 запрос в секунду.
- Максимум запросов к API в день — не более 1000 запросов.

## 4.2. Компоненты системы

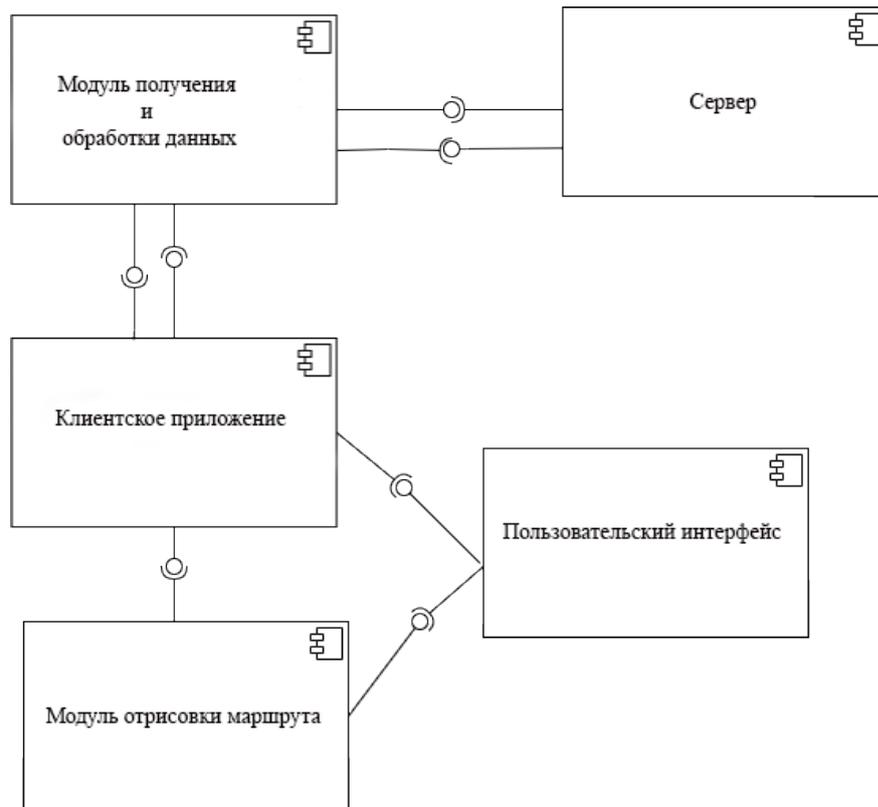


Рис. 3: Диаграмма компонентов системы.

Прототип системы (далее — прототип или система) состоит из следующих компонентов:

- **Пользовательский интерфейс**, с помощью которого пользователь может взаимодействовать с прототипом посредством кнопок, полей ввода и других графических и текстовых элементов.
- **Клиентское приложение** отвечает за взаимодействие системы с пользователем и управление данными. Оно обрабатывает ввод пользователя и запускает процесс получения данных с сервера.

- **Модуль получения и обработки данных** отвечает за общение с сервером, содержащим карты высот с помощью API, и обработку полученных от клиентского приложения и сервера данных. В этом модуле находятся реализованные методы для поиска фрагмента.
- **Сервер** обрабатывает запросы от модуля получения данных и возвращает необходимые для отображения и расчетов данные карты высот.
- **Модуль отрисовки карт** выводит карты высот, полученные от клиентского приложения и отображает пользователю эту информацию в пользовательском интерфейсе.

Основной сценарий взаимодействия пользователя и системы и элементов внутри системы выглядит следующим образом.

1. Пользователь загружает маршрут в формате списка GPS-координат в формате текстового файла. Текстовый файл состоит из строк, содержащих пару координат, разделенных запятой, каждая пара координат записывается с новой строки.
2. На основании полученных координат система делает запрос к карте высот через API и строит график, который отображает изменение высот вдоль выбранного пользователем маршрута.
3. Пользователь выбирает или загружает фрагмент для поиска на графике изменения высот. Выбрать можно заранее известные точки GPS-координат или указать номера шагов, между которыми находится искомый фрагмент.
4. Пользователь настраивает параметры для эксперимента: выбирает тип накладываемых шумов из предложенных и алгоритм для поиска фрагмента. В рамках системы для поиска фрагмента были реализованы метод перебора, метод на основе быстрого преобразования Фурье и метод корреляции.

5. Пользователь запускает моделирование с помощью кнопки "Начать".
6. Система запускает выполнение алгоритма.
7. Система выдает окно с результатом, где отображается найден ли фрагмент, совпал ли он с изначальным искомым местоположением, а также время исполнения программы.

Благодаря описанному выше сценарию, пользователь имеет возможность за 4 взаимодействия провести эксперимент и проверить свои гипотезы относительно работоспособности методов в разных условиях.

### 4.3. Графический интерфейс прототипа системы

Прототип пользовательского интерфейса системы представлен на рисунке 4. В верхней части экрана по центру расположено окно для отображения карты высот местности, которую загружает пользователь, через пункт "Файл—>Загрузить карту".

Ниже расположена панель выбора координат искомого фрагмента. Ввести координаты можно тремя способами:

- Через пункт меню сверху: "Файл—>Загрузить фрагмент". Файл с фрагментами содержит список высот, где каждая высота записывается с новой строки.
- Через указание координат начала и конца в GPS-формате.
- Через указание номеров шагов на исходной карте.

Слева располагается элемент, отображающий искомый фрагмент. График высот фрагмента отображается после его загрузки через меню "Файл —> Загрузить фрагмент" или после проведения первого моделирования с выбранными данными.

В правой части пользователю предлагается с помощью радиокнопки выбрать метод, которым будет производится поиск фрагмента на карте, и из выпадающего списка выбрать тип зашумления или искажения

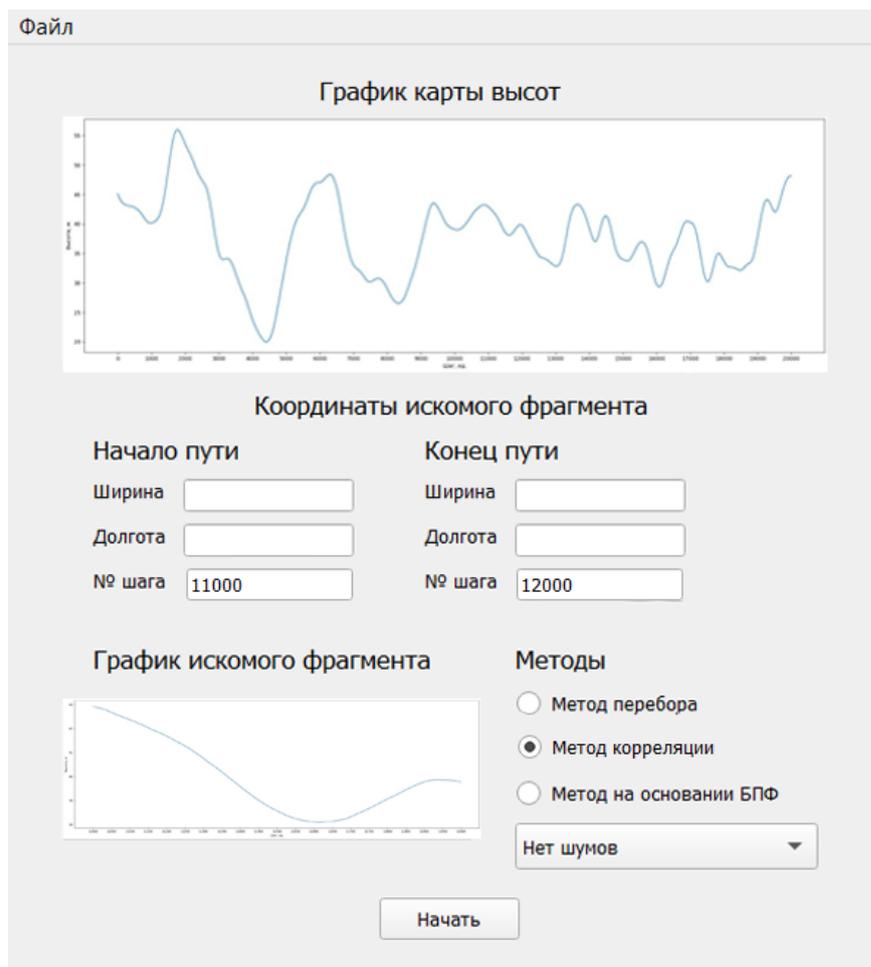


Рис. 4: Прототип пользовательского интерфейса

фрагмента. Для разных типов зашумления внизу появляются поля с необходимыми для заполнения параметрами.

Кнопка "Начать" в самом низу формы запускает программу поиска фрагмента на карте. После моделирования пользователю показывается окно, которое содержит найденное местоположение, время поиска фрагмента. Если искомое положение было заранее известно, то выдается сообщение о совпадении или несовпадении найденного и заданного фрагментов.

#### 4.4. Программная реализация методов поиска фрагмента

В рамках прототипа были реализованы метод перебора, метод корреляции и метод основанный на быстром преобразовании Фурье с по-

мощью языка Python с использованием математической библиотеки NumPy.

Метод перебора имеет самую простую реализацию. На вход программа получает 2 массива: карту и искомый фрагмент. Циклом происходит проверка на совпадение фрагмента с частью массива карты по длине равной фрагменту. Если находится поэлементное совпадение, то номер итерации в цикле записывается в массив для совпадений. В результате алгоритм возвращает массив с индексами искомого фрагмента в карте.

Метод корреляции имеет следующую реализацию. Программа получает 2 массива - карту и фрагмент. В цикле высчитывается значение корреляции между фрагментом и частью карты, равной по длине фрагменту. В полученном массиве корреляций находим элемент с наибольшим значением, то есть наиболее близким к 1. В результате алгоритм возвращает индекс наиболее вероятного начала фрагмента на карте.

Метод, основанный на быстром преобразовании Фурье, реализован следующим образом. На вход программа получает 2 массива: карту и искомый фрагмент. Из каждого элемента массива искомого фрагмента вычитается среднее значение, затем массив расширяется до размера карты и переворачивается таким образом, что первый элемент становится последним, а последний - первым. Применяем быстрое преобразование Фурье к обоим массивам и перемножаем полученные результаты поэлементно. После применяем обратное преобразование и находим индекс с наибольшим значением. Индекс и будет искомым положением начала фрагмента на карте.

## 5. Эксперименты

### 5.1. Описание эксперимента

В рамках эксперимента была взята дорога длиной около 42 000 точек в Ленинградской области. Точки были сформированы в формате GPS-координат, для которых потом строилась карта высот. Данные для карты высот были взяты данные из набора данных ASTER (The Advanced Spaceborne Thermal Emission and Reflection Radiometer) [10], которая была разработана Министерством экономики, торговли и промышленности (METI) Японии и НАСА. Карта имеет разрешение в 1 угловую секунду, что соответствует разрешению в 30 метров на экваторе.

Было проведено два типа экспериментов: качество работы алгоритмов при наличии разных шумов и искажений на фрагменте и скорость работы алгоритмов при разном объеме имеющихся данных.

### 5.2. Работоспособность алгоритмов при разных видах шумов и искажении фрагмента

В рамках эксперимента проверялось качество работы алгоритма при следующих параметрах:

- на фрагмент не накладывался шум;
- на фрагмент накладывался гауссовский шум;
- искомый фрагмент масштабировался (значение умножалось на константу).

В ходе экспериментов подтвердилось, что при отсутствии шумов все алгоритмы смогли однозначно найти искомый фрагмент на карте. При добавлении гауссовских шумов и умножении значений фрагмента на константу метод перебора перестает работать. В противовес к этому, метод корреляции и метод на основе преобразования Фурье правильно определили положение фрагмента на карте.

### 5.3. Скорость работы алгоритмов

В рамках текущего эксперимента проводилось сравнение скорости работы алгоритмов без наложения шумов, так как искажение фрагмента не влияет на производительность методов. Также метод перебора также принимал участие в сравнении, как самый простой из алгоритмов. Для сравнения скорости работы были проведены эксперименты со следующими параметрами:

- Сравнилось влияние положения искомого фрагмента карты: в начале, середине или в конце карты.
- Для сравнения брали длины карты ( $m$ ) равны 100, 1000, 10 000 точек искомым фрагмент был длины ( $p$ ) 10, 100, 1000 точек.
- Для каждого метода поиск проводился 1000 раз и рассчитывался 95% доверительный интервал.

В таблицах 1 – 6 представлены результаты замеров скорости работы алгоритмов с разными параметрами положения искомого фрагмента и размерами фрагмента и карты.

	Фрагмент в начале, мс	Фрагмент в середине, мс	Фрагмент в конце, мс
Метод перебора	$0.03 \pm 0.01$	$0.03 \pm 0.01$	$0.03 \pm 0.01$
Метод корреляции	$2.00 \pm 0.05$	$1.98 \pm 0.04$	$1.97 \pm 0.04$
Метод Фурье	$0.08 \pm 0.01$	$0.08 \pm 0.01$	$0.07 \pm 0.01$

Таблица 1: Эксперимент:  $m = 100, p = 10$ .

	Фрагмент в начале, мс	Фрагмент в середине, мс	Фрагмент в конце, мс
Метод перебора	$0.36 \pm 0.03$	$0.34 \pm 0.03$	$0.36 \pm 0.03$
Метод корреляции	$7.33 \pm 0.06$	$7.35 \pm 0.06$	$7.70 \pm 0.07$
Метод Фурье	$0.33 \pm 0.03$	$0.32 \pm 0.03$	$0.33 \pm 0.03$

Таблица 2: Эксперимент:  $m = 1000, p = 10$ .

	Фрагмент в начале, мс	Фрагмент в середине, мс	Фрагмент в конце, мс
Метод перебора	$0.74 \pm 0.05$	$0.74 \pm 0.04$	$0.76 \pm 0.04$
Метод корреляции	$47.71 \pm 0.28$	$48.75 \pm 0.49$	$47.49 \pm 0.36$
Метод Фурье	$0.33 \pm 0.04$	$0.34 \pm 0.04$	$0.33 \pm 0.04$

Таблица 3: Эксперимент:  $m = 1000, p = 100$ .

	Фрагмент в начале, мс	Фрагмент в середине, мс	Фрагмент в конце, мс
Метод перебора	$1.93 \pm 0.03$	$1.89 \pm 0.03$	$1.92 \pm 0.03$
Метод корреляции	$109.21 \pm 0.26$	$106.38 \pm 0.28$	$109.76 \pm 0.84$
Метод Фурье	$1.90 \pm 0.03$	$1.93 \pm 0.03$	$2.33 \pm 0.05$

Таблица 4: Эксперимент:  $m = 10000, p = 10$ .

	Фрагмент в начале, мс	Фрагмент в середине, мс	Фрагмент в конце, мс
Метод перебора	$5.32 \pm 0.07$	$4.26 \pm 0.05$	$3.86 \pm 0.04$
Метод корреляции	$736.47 \pm 3.98$	$651.52 \pm 3.42$	$650.81 \pm 3.18$
Метод Фурье	$4.25 \pm 0.04$	$3.86 \pm 0.04$	$3.76 \pm 0.04$

Таблица 5: Эксперимент:  $m = 10000, p = 100$ .

	Фрагмент в начале, мс	Фрагмент в середине, мс	Фрагмент в конце, мс
Метод перебора	$19.20 \pm 0.15$	$16.18 \pm 0.07$	$16.30 \pm 0.07$
Метод корреляции	$5308.21 \pm 57.52$	$4976.18 \pm 96.86$	$6294.52 \pm 65.71$
Метод Фурье	$15.94 \pm 0.07$	$15.97 \pm 0.06$	$23.26 \pm 0.08$

Таблица 6: Эксперимент:  $m = 10000, p = 1000$ .

Из таблиц представленных выше можно сделать вывод, что положение искомого фрагмента в последовательности практически не влияет на скорость вычислений для всех алгоритмов.

Метод перебора является самым быстрым методом при размере карте в 100 точек, однако при увеличении размеров карты и искомого фрагмента начинает проигрывать методу на основе быстрого преобразования Фурье.

Метод корреляции вычисляется дольше всех остальных алгоритмов,

даже на небольшом размере карты скорость вычислений отличается в десятки раз.

Метод на основе преобразования Фурье дает вычислительное преимущество при увеличении размера карты и размера фрагмента для поиска. При размере фрагмента  $p = 10$  и карты до  $m = 10000$  метод по скорости примерно совпадает с методом перебора, однако уже на последнем измерении метод на основании Фурье начинает немного выигрывать в скорости у метода перебора.

#### **5.4. Общие выводы по экспериментам**

В результате двух экспериментов было проведено сравнение трех методов поиска фрагмента. Метод корреляции оказался самым медленным из алгоритмов, но при этом смог найти фрагмент на карте при искажении и зашумлении фрагмента. Метод перебора является самым быстрым алгоритмом на картах небольшого размера, однако при наложении шумов на фрагмент, метод больше не находит необходимый фрагмент. Метод на основе преобразования Фурье превосходит два других алгоритма по скорости работы при увеличении размеров фрагмента и карты, а также позволяет найти фрагмент на карте даже после накладывания шумов и искажений.

## 6. Заключение

В ходе выпускной квалификационной работы были выполнены следующие задачи.

- Выполнен обзор методов в области моделирования навигации роботов. Рассмотрены классификации навигационных систем, определена какой является навигационная система, созданная в рамках работы. Были выбраны методы, которые необходимы для поиска фрагмента на карте.
- Разработан прототип системы моделирования навигации роботов, в который входит программная реализация алгоритмов, рассмотренных в обзоре, на языке программирования Python, графический интерфейс для визуализации и работы пользователем, который был разработан с помощью Qt Designer.
- Проведены эксперименты с разными модификациями данных. Сравнение методов показало, что метод корреляции находит фрагмент даже при искажении и зашумлении, но работает медленнее остальных. Метод перебора самый быстрый на небольших картах, но не находит фрагменты при наложении шумов. Метод на основе преобразования Фурье работает быстрее на больших картах, позволяет найти фрагменты после искажений и добавления шумов.

## Список литературы

- [1] Адаптивное управление с прогнозирующими моделями при переменной структуре пространства состояний с приложением к системам сетевого управления движением и автоматизации медицинского оборудования / О. Н. Граничин, Н. О. Амелина, А. В. Проскурников и др. // Адаптивное управление с прогнозирующими моделями при переменной структуре пространства состояний. — 2018. — С. 5–28.
- [2] Артюхин В.С., Логункова А.А. Глобальные спутниковые навигационные системы Глонасс и GPS // Славянский форум. — 2017. — № 3. — С. 278–284.
- [3] Байнов А.М., Зарипова Р.С. Робототехника и компьютерное моделирование: задачи и перспективы применения // International Journal of Advanced Studies in Computer Engineering. — 2018. — № 2. — С. 4–7.
- [4] Гурский Б.Г. Современные корреляционно-экстремальные системы для высокоточной навигации летательных аппаратов // Гироскопия и навигация. — 2006. — № 2. — С. 114–115.
- [5] Инерциальные навигационные системы морских объектов / Д.П. Лукьянов, А.В. Мочалов, А.А. Одинцов, И.Б. Вайсгант // Судостроение. — 1989.
- [6] Кремповский П.Р., Луцков Ю.И. Навигационные системы автоматизированных робототехнических комплексов // IEEE Transactions on Automatic Control. — 2016. — Т. 61, № 5. — С. 58–61.
- [7] Поддубный В. В. Рестриктивная фильтрация в навигационных системах // Вестник Томского государственного университета. — 2002. — № 275. — С. 202–215.
- [8] Степанов О.А. Методы обработки навигационной измерительной информации. — СПб: Университет ИТМО, 2017. — С. 196.

- [9] Степанов О. А., Носов А. С. Алгоритм коррекции навигационной системы по данным карты и измерителя, не требующий предварительного оценивания значений поля вдоль пройденной траектории // Гироскопия и навигация. — 2020. — Т. 54, № 2 (109). — С. 70–90.
- [10] ASTER - OPEN TOPO DATA. — URL: <https://asterweb.jpl.nasa.gov/gdem.asp>.
- [11] Key technologies of geomagnetic aided inertial navigation system / Xiaojiao Ma, Hongwei Liu, Di Xiao, Hankui Li // 2009 IEEE Intelligent Vehicles Symposium / IEEE. — 2009. — P. 464–469.
- [12] Open Topo Data. — URL: <https://www.opentopodata.org/>.
- [13] Project Jupyter|Home. — URL: <https://jupyter.org/>.
- [14] Python. — URL: <https://www.python.org/>.
- [15] Qt Design Studio - UI Design Tool for Application. — URL: <https://www.qt.io/product/ui-design-tools#eval-form-modal>.
- [16] A visual navigation system for autonomous land vehicles / Allen Waxman, Jacqueline LeMoigne, Larry Davis et al. // IEEE Journal on Robotics and Automation. — 1987. — Vol. 3, no. 2. — P. 124–141.
- [17] Начать | Elevation API | Google For Developers. — URL: <https://developers.google.com/maps/documentation/elevation/start?hl=ru>.
- [18] Яндекс Карты. — URL: <https://yandex.ru/maps>.