
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И
МАТЕМАТИКИ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

КАФЕДРА ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ
ТЕХНОЛОГИИ

Пояснительная записка
к дипломной работе на тему:

**РАЗРАБОТКА СИСТЕМЫ ПОЗИЦИОНИРОВАНИЯ ТРАНСПОРТА ПО
СИГНАЛАМ СОТОВЫХ СЕТЕЙ**

Студент: Ковалев М.М.

Руководитель проекта: Столяров Д.О.

Допущен к защите: «__» _____ 2012 г.

Консультанты проекта:

Охрана труда: Михайлов Е.Б.

Заведующий кафедрой: Азаров В.Н.

Москва, 2012г.

“Утверждаю”
Зав. кафедрой
д.т.н., проф. В. Н. Азаров

“ ” 2012 г.

Срок сдачи проекта
“01” июня 2012 г.

ЗАДАНИЕ
на дипломное проектирование

студенту группы С-104 дневного отделения
Ковалеву Максиму Максимовичу
Ф.И.О. полностью

Тема утверждена приказом по институту
от “ ” 2012 г. № ____

1. Тема проекта

Разработка системы позиционирования транспорта по сигналам сотовых сетей/

2. Функционал разрабатываемой системы

1. Приём от пользователя маршрутов движения позиционируемых устройств.
2. Сбор данных об уровнях принимаемых сигналов при известном местонахождении устройства.
3. Поиск предполагаемого местонахождения устройства на основании данных об уровнях принятых сигналов.

3. Содержание работы

3.1. Обзорно-аналитическая часть

1. Обзор существующих систем позиционирования устройств в условиях города.
2. Анализ недостатков и возможных путей избавления от них.
3. Анализ математических методов, потенциально применимых в данной задаче.
4. Выбор средств разработки.

3.2. Технологическая часть

1. Обзор существенных для проекта характеристик сотовых сетей.
2. Оценка предполагаемой нагрузки на систему.
3. Выбор и оценка СУБД для системы.

3.3. Разработка

1. Проектирование архитектуры системы.
2. Синтез математического метода для работы системы.
3. Реализация разработанного метода.
4. Разработка сетевой части.
5. Разработка базы данных.
6. Перспективы разработки.

3.4. Экспериментальная часть

1. Оценка соответствия системы техническому заданию.
2. Сбор тестовых данных в реальных условиях города.
3. Проверка качества позиционирования.

3.5. Охрана труда

1. Исследование возможных опасных и вредных факторов при эксплуатации ЭВМ и их влияния на пользователей.

2. Методы и средства защиты пользователей от воздействия на них опасных и вредных факторов.

4. Перечень материалов, представляемых на защите

1. Результаты тестирования системы.

5. Подписи сторон

Консультанты:

Специальная часть "___" ____ 2012г. _____ / _____ /

Технологическая часть "___" ____ 2012г. _____ / _____ /

Охрана труда "___" ____ 2012г. _____ /Михайлов Е.Б./

Задание выдал "___" ____ 2012 г. Задание получил "___" ____ 2012 г.

_____ /Столяров Д.О./ _____ /Ковалев М.М./

Задание оформляется в двух экземплярах.

Один экземпляр после утверждения темы выдается студенту для включения в пояснительную записку после титульного листа.

Аннотация

В данной работе создан и проверен новый метод позиционирования мобильных устройств на основе информации о принимаемых ими уровнях сигналов от базовых станций сетей GSM, рассчитанный на стационарную установку позиционируемых устройств на наземном общественном транспорте. Использование априорной информации о неизменном маршруте движения каждого транспортного средства позволило свести задачу от двухмерной (поиск широты и долготы) к одномерной (поиск расстояния от фиксированной точки на маршруте) и полностью отказаться от применения метода триангуляции в пользу статистического подхода.

Разработанный алгоритм был реализован в прототипе системы позиционирования, после чего проверен на тестовом отрезке пути в реальных городских условиях. Достигнута средняя погрешность определения положения в 49 метров, что лучше, чем у современных систем, основанных на триангуляции, но ещё не позволяет сравниться с системами спутниковой навигации, а потому для достижения возможности внедрения наравне со спутниковыми системами, требуется дополнительная научная и техническая работа.

Исходные коды компонентов разработанной системы, текстовые журналы экспериментов, а также исходные коды данной пояснительной записи в форматах L^AT_EX и Graphviz вместе с использованными растровыми изображениями доступны в репозитории Subversion по адресу <http://svn.auditory.ru/repos/tatmon>.

В соответствие с GNU Free Documentation License [1], данная работа является свободным документом:

Copyright © 2012 Maxim Kovalev. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Оглавление

Техническое задание	1
Аннотация	3
1 Введение	7
2 Обзорно-аналитическая часть	10
2.1 Существующие системы позиционирования	10
2.1.1 Спутниковая навигация	10
2.1.2 Позиционирование в сотовых сетях	12
2.1.2.1 Распространение сигнала на открытой местности	12
2.1.2.2 Погрешность определения расстояния	14
2.1.2.3 Координаты базовых станций	17
2.1.2.4 Факторы, снижающие точность позиционирования	18
2.1.3 Заключение	19
2.2 Позиционирование общественного транспорта	19
2.2.1 Снижение точности из-за двухмерной избыточности	21
2.2.2 Снижение точности из-за экстраполяции	22
2.2.3 Гипотеза о возможности повышения точности	23
2.3 Анализ возможности применения существующих алгоритмов	23
2.3.1 Постановка задачи	23
2.3.2 Расстояние Махalanобиса	24
2.3.3 Байесовский классификатор	25
3 Технологическая часть	27
3.1 Общая архитектура системы	27
3.1.1 Процесс обработки данных	27
3.1.2 Состав системы	28
3.2 Средства разработки	29

3.2.1	Система контроля версий	29
3.2.2	Мобильное приложение	29
3.2.3	Сервер и СУБД	29
3.2.4	Клиентская часть	30
3.2.5	Пояснительная записка	30
4	Охрана труда	32
5	Разработка	35
5.1	Алгоритм	35
5.1.1	Исходные данные	35
5.1.2	Интерполяция	35
5.1.3	Вычисление псевдоплотности вероятности	37
5.1.4	Вычисление ответа	41
5.2	Мобильное приложение	41
5.2.1	Протокол передачи данных	42
5.2.2	Интерфейсы	42
5.2.2.1	Графический интерфейс	44
5.2.2.2	Данные об уровнях сигнала	45
5.2.2.3	Данные спутниковой навигации	45
5.2.2.4	Взаимодействие с сетью	45
5.3	Серверная часть	45
5.3.1	Режим сбора данных	46
5.3.1.1	Проверка входящих данных	46
5.3.1.2	Преобразование координат	47
5.3.1.3	Взаимодействие с базой данных	48
5.3.1.4	Взаимодействие с клиентами	48
5.3.1.5	Главный модуль	49
5.3.2	Режим позиционирования	50
5.3.2.1	Интерполяция и псевдоплотность	50
5.3.2.2	Результирующая псевдоплотность и позициони- рование	54
5.3.2.3	Главный модуль	55
5.4	База данных	56
5.4.1	Добавление информации в БД	59
5.5	Клиентская часть	60

5.5.1	Режим сбора данных	60
5.5.2	Режим позиционирования	60
6	Экспериментальная проверка	63
6.1	Условия эксперимента	63
6.2	Сбор данных	63
6.3	Позиционирование	64
6.3.1	Входные данные	64
6.3.2	Выборка собранных данных	66
6.3.3	Интерполяция	66
6.3.4	Псевдоплотности вероятности	66
6.3.5	Результирующая псевдоплотность	66
6.4	Погрешность	70
6.5	Анализ экспериментальных данных	71
6.5.1	Степени интерполяционных многочленов	71
6.5.2	Возможные дальнейшие эксперименты	73
Заключение		74
Литература		77
Список иллюстраций		81
Список таблиц		84
Список публикаций		86

1 Введение

Актуальность. На рубеже нулевых и десятых годов XXI века, параллельно идущие процессы удешевления как средств, так и услуг мобильного доступа в сеть Интернет, а также удешевления и миниатюризации средств спутниковой навигации привели к появлению принципиально нового вида онлайн-услуг — служб, основанных на местоположении (location-based service, LBS) [2]. Одним из таких сервисов является удалённое наблюдение за местонахождением транспортных средств, с помощью установленных на них автономных «маяков», различными способами определяющих собственное местонахождение и передающих данные о нём через мобильный интернет. Владельцы личных автомобилей преимущественно используют подобные средства в следящих противоугонных системах [3], но эти средства применимы и в городском транспортном хозяйстве. Помимо непосредственно городских властей, в чьи интересы входит наблюдение за транспортными потоками для задач городского планирования, пресечения злоупотребления служебным положением со стороны недобросовестных водителей и прочих глобальных целей, подобные данные могут быть интересны и простым пассажирам. В связи с трудностью прогнозирования транспортной обстановки, реальный график движения наземного общественного транспорта в часы пик может существенным образом отличаться от расписания. Наблюдение же реальной динамики движения позволило бы создать систему, делающую динамически изменяющиеся прогнозы фактического времени прибытия транспорта и предполагаемого времени в пути, о результатах которого уведомляющую пассажиров через такие каналы как электронные табло на остановках и в самих транспортных средствах, через беспроводную связь подключённые к серверу системы, web-сервис, доступный через стационарные или планшетные компьютеры, приложение для смартфонов (см. рис. 1.1), SMS или USSD сервис для любого типа мобильных телефонов и другие. Подобные системы уже начинают внедрять по всему миру [4] [5] [6] [7], однако по-прежнему сравнительно высокая стоимость приёмников GPS и ГЛОНАСС тормозит процесс внед-

рения. С другой стороны, существуют сервисы позиционирования мобильных телефонов, использующие для своей работы только информацию о сигналах от базовых станций сотовой сети, но их проблемой является недостаточная точность позиционирования. При этом, на общественном транспорте используются те же системы, что и на личном, хотя между ними есть огромная разница: общественный транспорт движется по заранее известным и неизменным маршрутам, что фактически позволяет свести задачу его позиционирования от двухмерного случая к одномерному. Но эта технологическая ниша оказалась фактически незанятой.

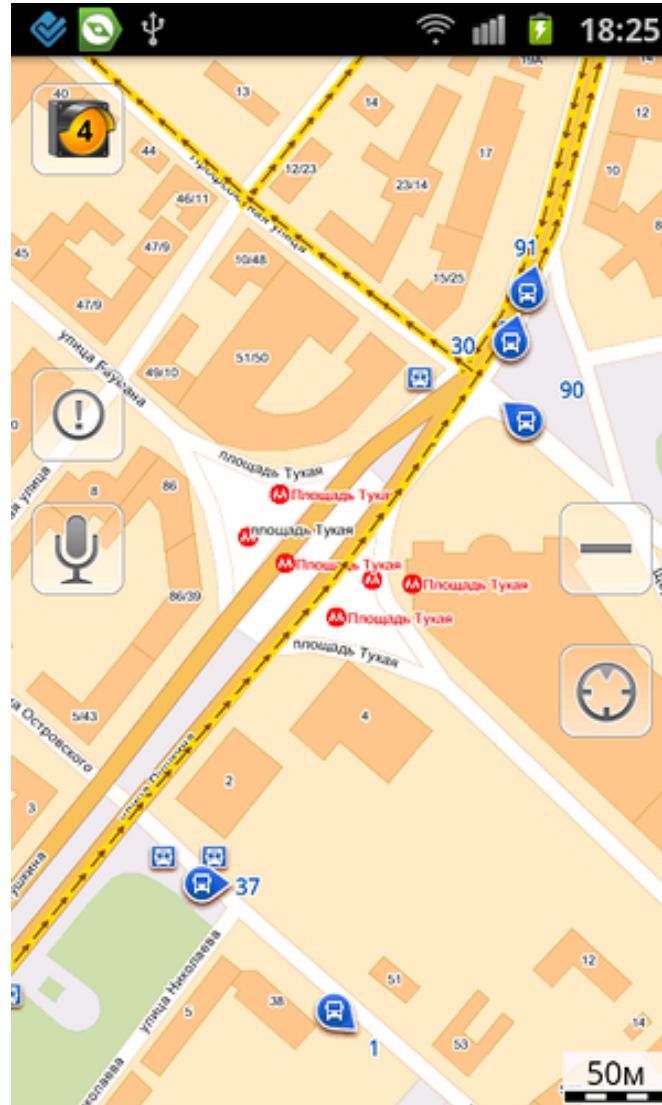


Рис. 1.1: Приложение от Яндекс, в реальном времени показывающее автобусы Казани. Изображение из блога Яндекса [7]

Целью данной работы является проверка предположения о возможности существенного повышения точности позиционирования транспортных средств, по сравнению с существующими универсальными системами, при условии заранее известного маршрута движения, а также удешевления подобной системы,

по сравнению со случаем использования спутниковой навигации. Для достижения этой цели требуется создать и протестировать прототип подобной системы позиционирования, который, в случае успеха, в дальнейшем может быть доработан до промышленного образца, внедрён и коммерциализирован, однако эти цели уже выходят за рамки данной работы.

Задачи, которые требуется решить на пути к данной цели, таковы:

- Формулировка одномерной задачи позиционирования на местности;
- Анализ принципов работы существующих систем позиционирования и поиск в них избыточности относительно одномерной задачи позиционирования;
- Создание математического метода, устраниющего указанную избыточность в пользу большей точности;
- Проектирование архитектуры системы с учётом разработанного метода и реализация её прототипа;
- Тестирование созданного прототипа.

Публикации. Помимо данной пояснительной записки, результаты данной работы также были дважды представлены на разных стадиях на конференции студентов, аспирантов и молодых специалистов МИЭМ (см. «Список публикаций» на стр. 86). Данная работа также выиграла конкурс грантов «Участник Молодёжного Научно-Иновационного Конкурса» в марте 2011 года [8].

2 Обзорно-аналитическая часть

2.1 Существующие системы позиционирования

На сегодняшний день, наибольшее распространение получили [2] два вида систем позиционирования: основанные на спутниковой навигации и основанные на принимаемых уровнях сигналов от базовых станций сотовых сетей.

2.1.1 Спутниковая навигация

Спутниковые системы позиционирования, такие как GPS и ГЛОНАСС, в качестве опорных точек используют орбитальную группировку спутников, специально запущенных для целей навигации. Орбитальные элементы для всех спутников известны, а потому вычислить местонахождение каждого из них в любой момент времени возможно с большой точностью. После того, как они вычислены, можно осуществлять непосредственно позиционирование.

GPS использует метод трилатерации, в котором измеряется расстояние от приёмника до трёх видимых спутников. Три сферы, пересекаясь, дают две возможные точки, причём расположенные на разной высоте (см. рис. 2.1), а потому правильной считается ближайшая к поверхности Земли. Для измерения расстояния оценивается время прохождения сигнала от спутника до приёмника. Средняя высота орбит спутников GPS над поверхностью Земли составляет порядка $2 \cdot 10^7$ метров [9], что, с учётом скорости света, равной $3 \cdot 10^8$ м/с, даёт характерное время прохождения сигнала около 60 миллисекунд. Чтобы при таком порядке задержки обеспечить достаточную точность позиционирования, все спутники оборудованы синхронизированными атомными часами, дающими точность около 14 наносекунд [9], причём для её обеспечения приходится учитывать эффекты специальной теории относительности, применительно к движению спутника. Было бы крайне дорого обеспечивать каждый приёмник такими же часами, а потому точное время он вычисляет каждый раз, когда принимает сигнал от спутника, решая уравнения метода трилатерации и относительно

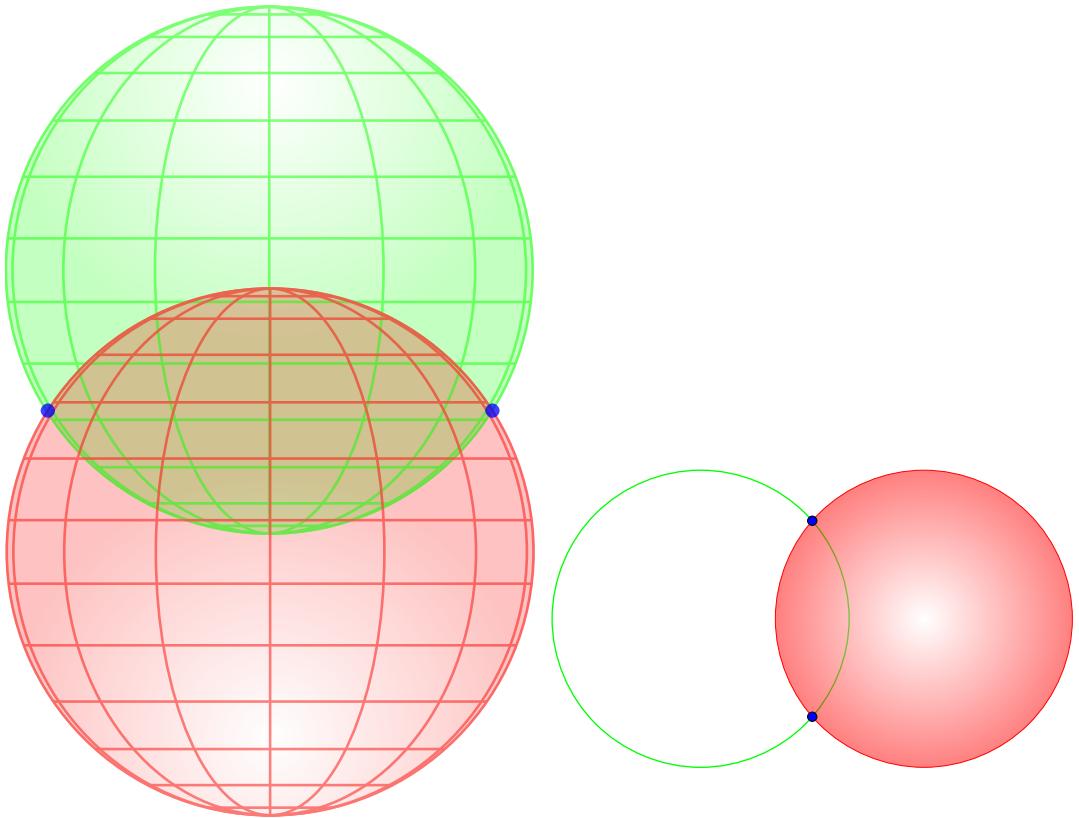


Рис. 2.1: Две пересекающиеся сферы образуют окружность. Третья сфера, пересекая эту окружность, образует две точки. Изображения из фонда Wikimedia Commons.

положения, и относительно времени.

В случае, когда в области видимости находится более, чем 4 спутника, система уравнений, описывающих положение приёмника, становится избыточной, количество уравнений превышает количества неизвестных. В этом случае, с одной стороны, можно выбрать набор спутников, использование которых даёт наибольшую точность позиционирования, оценивая величину геометрического ухудшения точности (geometric dilution of precision — GDOP), с другой — решать избыточную систему уравнений одним из приближённых численных методов, таких как метод наименьших квадратов, чтобы получить лучшую оценку положения автоматически [9].

Характерное же расстояние от мобильного устройства до базовой станции сотовой сети, в зависимости от стандарта, составляет 10^2 – 10^4 метров. Для обеспечения соответствующей точности трилатерации в таких сетях, потребовалось бы пропорционально снизить погрешность точного времени, что представляется трудноосуществимым, поскольку для синхронизации часов базовых станций используется как раз сигнал GPS [10]. В связи с этим, в отсутствие возможности использования сигнала GPS (отсутствие приёмника или видимости спутников),

применяются иные методы.

2.1.2 Позиционирование в сотовых сетях

Как было показано в пункте 2.1.1, метод оценки расстояния между источником и приёмником, с помощью измерения задержки прохождения сигнала для сотовых сетей неприменим. Однако, остаётся возможность измерить амплитуду принимаемого сигнала, убывающую обратно пропорционально квадрату расстояния. Стандартный GSM-модуль может предоставить информацию об амплитуде принимаемого сигнала не только от базовой станции, через которую идёт передача данных, но и обо всех окружающих, сигнал которых можно принять. Информация эта представляется в виде логарифмической величины received signal strength indicator (RSSI), преобразованную из стандартной единицы измерения — децибел мощности (dBm) — в дискретную единицу измерения asu, представляющую из себя целое число в диапазоне от 0 до 31, где 0 соответствует самому малому уровню сигнала, доступному в шкале, а 31 — самому большому [11].

2.1.2.1 Распространение сигнала на открытой местности

Максимальный радиус действия одной базовой станции в сети GSM составляет 35 километров на открытой местности, а в условиях городской застройки плотность их расстановки может составлять вплоть до одной станции на 400 метров [12]. Максимальная излучаемая мощность базовой станции стандарта DCS 1800 составляет 40 Вт [13]. Длина волны частотой 1800 МГц составляет $3 \cdot 10^8(\text{м}/\text{с})/1800 \cdot 10^6(\text{Гц}) = 16(\text{см})$. Примем эту величину за характерный размер излучателя и вычислим плотность потока электромагнитного излучения, считая, что 40 Вт излучения проходят через сферу диаметром 16 см:

$$\frac{40(\text{Вт})}{\pi \cdot 0,16^2(\text{м})} = 497(\text{Вт}/\text{м}^2) \quad (2.1)$$

Отсюда, зависимость мощности излучения от расстояния до базовой станции составит:

$$P = \frac{40^2}{497 \cdot \pi \cdot R^2} \quad (2.2)$$

Построим график этой зависимости.

На рис. 2.2 видно, что мощность, измеренная в ваттах, близка к нулю для всех 35 возможных километров. В связи с этим, целесообразно введение лога-

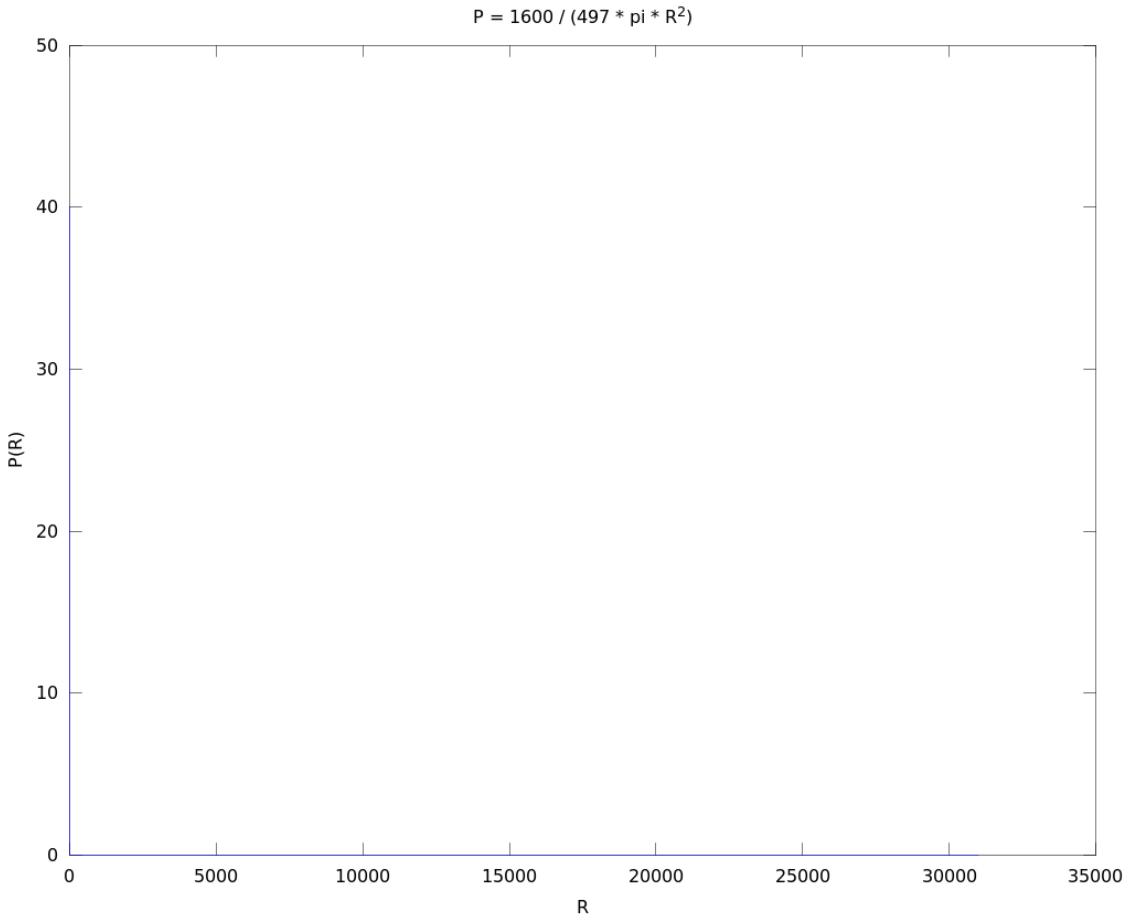


Рис. 2.2: Зависимость мощности излучения в ваттах от расстояния в метрах до базовой станции GSM DCS 1800 мощностью 40 Вт. Гипербола столь быстро сходится к асимптотам, что на рисунке сливаются с осями координат.

рифмической величины децибел мощности (dBm), связанной с ваттом следующим соотношением [14]:

$$x = 10 \cdot \log_{10}(1000 \cdot P) \quad (2.3)$$

Подставим формулу 2.2 в 2.3 и построим график получившейся зависимости.

На рис. 2.3 видно, что использование логарифмического масштаба позволяет получить кривую, намного более равномерно заполняющую координатную плоскость, чем в линейном масштабе как на рис. 2.2.

Наконец, возьмём связь между единицами dBm и asu [11]:

$$\text{asu} = \frac{\text{dBm} + 113}{2} \quad (2.4)$$

Подставим формулы 2.2 и 2.3 в 2.4 и получим:

$$\text{asu} = \frac{10 \cdot \log_{10}(1000 \cdot \frac{40^2}{497 \cdot \pi \cdot R^2}) + 113}{2} \quad (2.5)$$

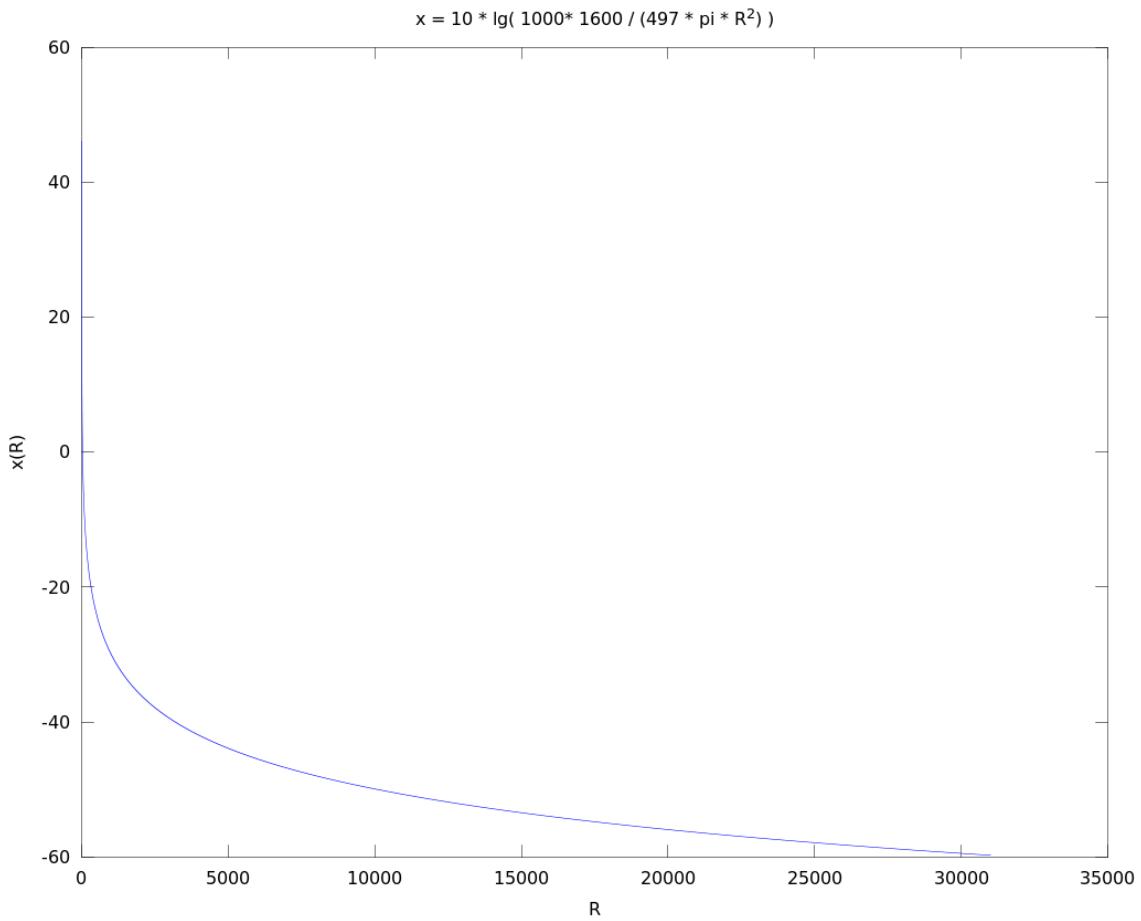


Рис. 2.3: Зависимость мощности излучения в децибелах мощности от расстояния в метрах до базовой станции GSM DCS 1800 мощностью 40 Вт.

2.1.2.2 Погрешность определения расстояния

Рассмотрим формулу 2.5 и найдём, на каких расстояниях от базовой станции будет меняться asu . В таблице 2.1 для наглядности мы намеренно пренебрегли ограничением на максимальное значение asu , равное 31, и это допустимо, поскольку при использовании базовой станции с меньшей мощностью передатчика, значения asu также сдвинутся в меньшую сторону, и допустимый диапазон asu будет достигаться не на границе радиуса действия базовой станции, а в его середине или начале.

Поскольку GSM-модуль передаёт для дальнейшей обработки только целую часть asu , расстояние между двумя соседними точками, где asu на самом деле имеет целое значение, можно считать погрешностью изменения расстояния на этом интервале. А так как измеряемой величиной является расстояние до базовой станции, отношение длины этого интервала к расстоянию до неё будет относительной погрешностью измерения. Из таблицы 2.1 видно, что относи-

Таблица 2.1: Расстояния до точек изменения asu от базовой станции GSM DCS 1800 мощностью 40 Вт. Теоретическая оценка.

Расстояние от станции, метров	Значение asu	Расстояние до следующей точки изменения asu	Отношение расстояния до следующей точки изменения asu к расстоянию от станции
143	50	37	0.259
180	49	47	0.261
227	48	58	0.255
285	47	74	0.260
259	46	93	0.259
359	46	93	0.259
452	45	117	0.259
569	44	148	0.260
717	43	185	0.258
902	42	234	0.259
1136	41	294	0.259
1430	40	370	0.259
1800	39	466	0.259
2266	38	587	0.259
2853	37	739	0.259
3592	36	930	0.259
4522	35	1170	0.259
5692	34	1475	0.259
7167	33	1856	0.259
9023	32	2336	0.259
11359	31	2941	0.259
14300	30	3700	0.259
18000	29	4660	0.259
22660	28	5870	0.259
28530	29		

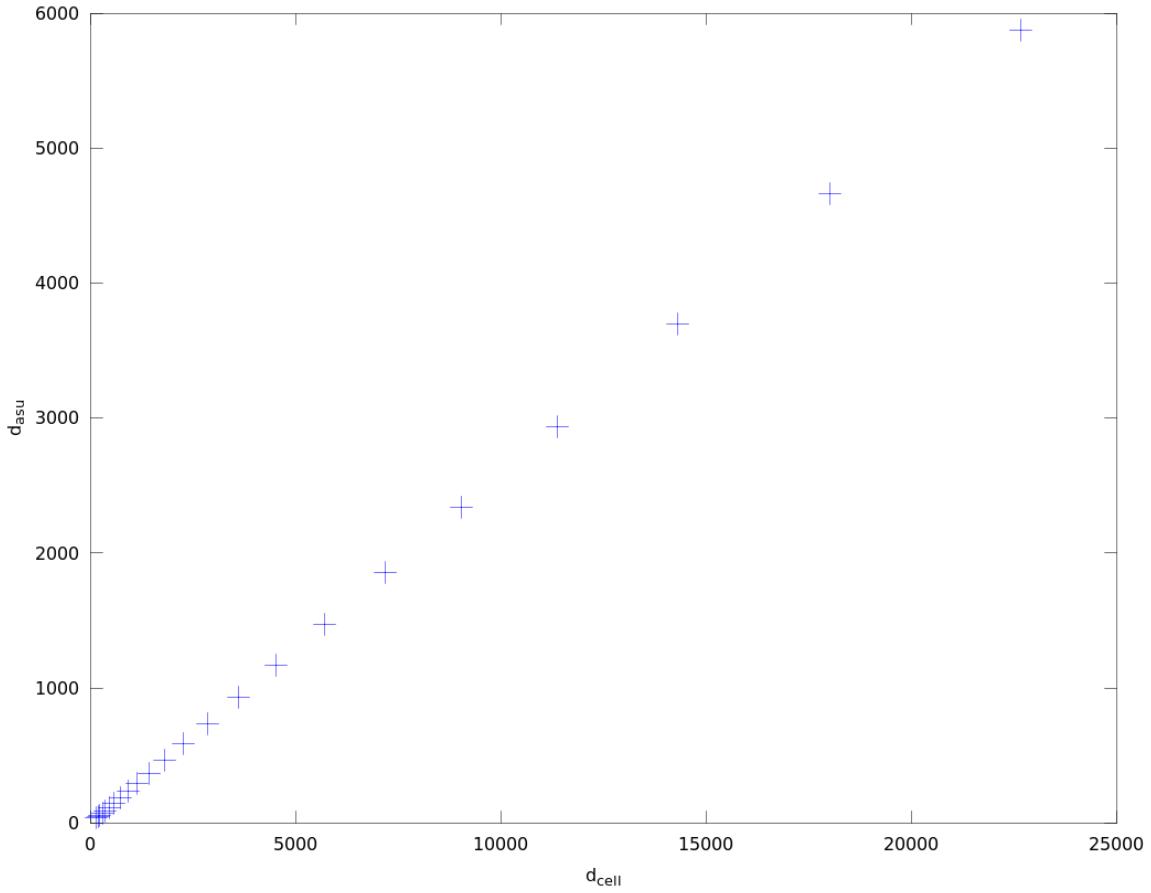


Рис. 2.4: Зависимость расстояния между двумя точками, где asu принимает целочисленное значение, от расстояния этих точек до базовой станции, по осям отложены метры

тельная погрешность остаётся постоянной на любом расстоянии. На рис. 2.4 наглядно показано, что абсолютная погрешность при этом, как и следует ожидать, зависит от расстояния линейно.

При этом, стоит заметить, что эти значения погрешности были вычислены в предположении, что единственным фактором, ограничивающим точность измерений, является «цена деления» GSM-модуля, измеряющего asu . Однако в реальности, следует ожидать вмешательства случайной погрешности. Если считать, что в некоторой местности уровень электромагнитных помех примерно одинаков для любой точки, то с ростом расстояния до базовой станции отношение сигнал/шум будет падать, что внесёт дополнительную погрешность.

Таким образом, видно, что измерение расстояния от базовой станции до мобильного устройства, основанное на амплитуде сигнала, позволяет достигнуть точности от десятков метров до единиц километров, в зависимости от расстояния до базовой станции и мощности её передатчика. Наибольшая точность

достигается вблизи станции, но при условии, что измеренное значение RSSI, выраженное в единицах asu, не выходит за пределы допустимого диапазона.

2.1.2.3 Координаты базовых станций

Для определения абсолютного местоположения мобильного устройства, а не только его положения относительно некоторых опорных точек, требуется знать координаты самих опорных точек. В случае использования спутниковой навигации, спутники немногочисленны — на 24.05.2010 их было 31 [9] на всю Землю — а их координаты в любой момент времени можно рассчитать с помощью законов орбитальной механики. В случае же базовых станций сотовых сетей, их количество для одной только России на конец 2011 года превысило 165 тысяч [15]. При этом, популярными провайдерами геолокационных сервисов зачастую являются не сотовые операторы, а сторонние компании, такие как Google, предоставляющий сервис Google Latitude, или Яндекс.Локатор от компании Яндекс [2]. Такие провайдеры могут работать в зонах покрытия множества различных операторов сотовой связи, а также, при наличии соответствующего приёмного оборудования на мобильном устройстве, использовать для повышения точности позиционирования и другие беспроводные сети, такие как WiFi или WiMAX [16] [17].

В связи с этим, у провайдера геолокационного сервиса обычно нет априорных данных о координатах базовых станций, которые используются для позиционирования. Работа любой подобной системы начинается с того, что измеряются RSSI видимых базовых станций во множестве точек на местности, координаты которых получаются с помощью спутниковой навигации. Когда достаточное количество подобных замеров осуществлено, решается обратная задача — вместо позиционирования устройства по известным координатам базовых станций, приближённо вычисляются координаты базовых станций, при известных координатах точек замеров RSSI. [18]. Также распространена ситуация, в которой данные о координатах базовых станций уточняются непосредственно в процессе работы системы. В этом случае, провайдеры изыскивают способы вынудить самих пользователей сервиса посыпать данные, помогающие уточнить базу данных. Так, пользовательское соглашение сервиса Яндекс.Локатор требует обязательной посылки данных нетмониторинга в случае, если от пользователя к сервису поступает более тысячи запросов в день [19].

Поскольку предполагаемые координаты базовых станций представляют со-

бой большой объём данный, к тому же, постоянно изменяющийся, они хранятся не на позиционируемом устройстве, а на сервере провайдера геолокационного сервиса. Таким образом, само устройство вычислить свои координаты не может, оно лишь отправляет запрос с результатами измерений на сервер, откуда получает ответ о предполагаемом местонахождении [20]. Поэтому для работы таких систем обязательно наличие подключения к интернету.

Отсюда следует, что:

- 1 Подобные системы являются самообучающимися: чем больше их используют, тем точнее они работают;
- 2 Подобные системы могут работать только в тех местах, где кто-то ранее измерял RSSI и посыпал эти данные провайдеру вместе с определёнными по GPS координатами точки измерения.

2.1.2.4 Факторы, снижающие точность позиционирования

Реальные условия работы системы могут существенным образом отличаться от рассмотренного случая. В силу ряда факторов, системы позиционирования в мобильных сетях не достигают такой точности, как спутниковые. В отличие от спутниковых систем, сигналы точного времени которых могут быть либо приняты, либо нет, а на время прохождения сигнала влияет только расстояние, уровень сигнала от станций сотовых сетей подвержен множеству аналоговых искажений. Сюда можно включить:

- 1 Поглощение излучения городской застройкой, деревьями, атмосферными осадками;
- 2 Частичное отражение излучения от городской застройки, в результате которого сигнал от базовой станции достигает мобильного устройства не по прямой, а по ломанной, причём иногда, в результате частичного отражения, одновременно несколькими путями;
- 3 Волновые эффекты распространения сигнала между домами и воздушными силовыми и сигнальными кабелями, которые могут работать как волноводы;
- 4 Анизотропия диаграммы направленности базовых станций, в связи с которой на одном и том же расстоянии от неё, но в разных точках, RSSI может быть разным;

- 5 Анизотропия диаграммы направленности мобильных устройств, в связи с которой одно и то же устройство в одном и том же месте может показывать различные RSSI для всех окружающих базовых станций, в зависимости от своей ориентации в пространстве;
- 6 Помехи на частотах работы базовых станций.

Эти и многие другие факторы могут приводить сразу к двум негативным последствиям:

- 1 На этапе сбора данных, искажённые RSSI приведут к ошибочному определению координат базовых станций;
- 2 На этапе позиционирования мобильных устройств, точность будут снижать и ошибки в определении текущих уровней сигналов, и ранее накопленные ошибки позиционирования базовых станций.

2.1.3 Заключение

В таблице 2.2 сведены характеристики, рассмотренные в данном разделе. Видно, что спутниковая навигация выигрывает у позиционирования в сотовых сетях практически по всем параметрам. К её недостаткам можно отнести необходимость нахождения под открытым небом для работы спутниковой навигации, а в случае использования в автономных устройствах — наличие приёмника GPS помимо и так необходимого модуля GSM, что повышает стоимость и энергопотребление прибора.

2.2 Позиционирование общественного транспорта

Позиционирование транспортных средств с помощью установленных на них автономных «маяков» является частным случаем задачи позиционирования мобильных устройств. В случае работы с легковыми и грузовыми автомобилями, требования не будут существенно отличаться от задачи позиционирования мобильного телефона или навигатора. Однако, если речь идёт об общественном транспорте, в задаче появляется новая сущность — маршрут, от движения по которому транспортное средство не может отклоняться.

Трамвай не может отклониться от маршрута вообще, троллейбус — не более, чем на длину штангового токоприёмника, автобус — на ширину проезжей части.

Таблица 2.2: Сравнение свойств спутникового позиционирования и позиционирования в сотовых сетях

Характеристика	GPS	Сотовые сети
Опорные точки	Десятки спутников на околоземных орbitах	Сотни тысяч стационарных базовых станций
Метод определения расстояния	Измерение задержки прохождения сигнала	Измерение затухания амплитуды сигнала
Координаты опорных точек	Известны с большой точностью	Необходимо приблизительно оценивать с помощью дополнительных измерений
Погрешность позиционирования	Единицы–десятки метров, зависит от видимости спутников	Десятки–сотни метров, зависит от множества факторов
Покрытие	Всемирное	Только там, где проводились предварительные замеры сигнала
Автономность устройства	Для работы достаточно приёмника	Нужен обмен данными с сервером провайдера
Условие прохождения сигнала	Нахождение под открытым небом с таким углом обзора, чтобы были видны хотя бы три спутника	Везде, где есть сотовая связь

Иными словами, задачу позиционирования общественного транспорта можно считать практически одномерной. Для каждого транспортного средства маршрут движения известен заранее, определять его не требуется. Таким образом, возможность позиционирования устройства в произвольном месте двухмерной карты является избыточной по отношению к данной задаче.

2.2.1 Снижение точности из-за двухмерной избыточности

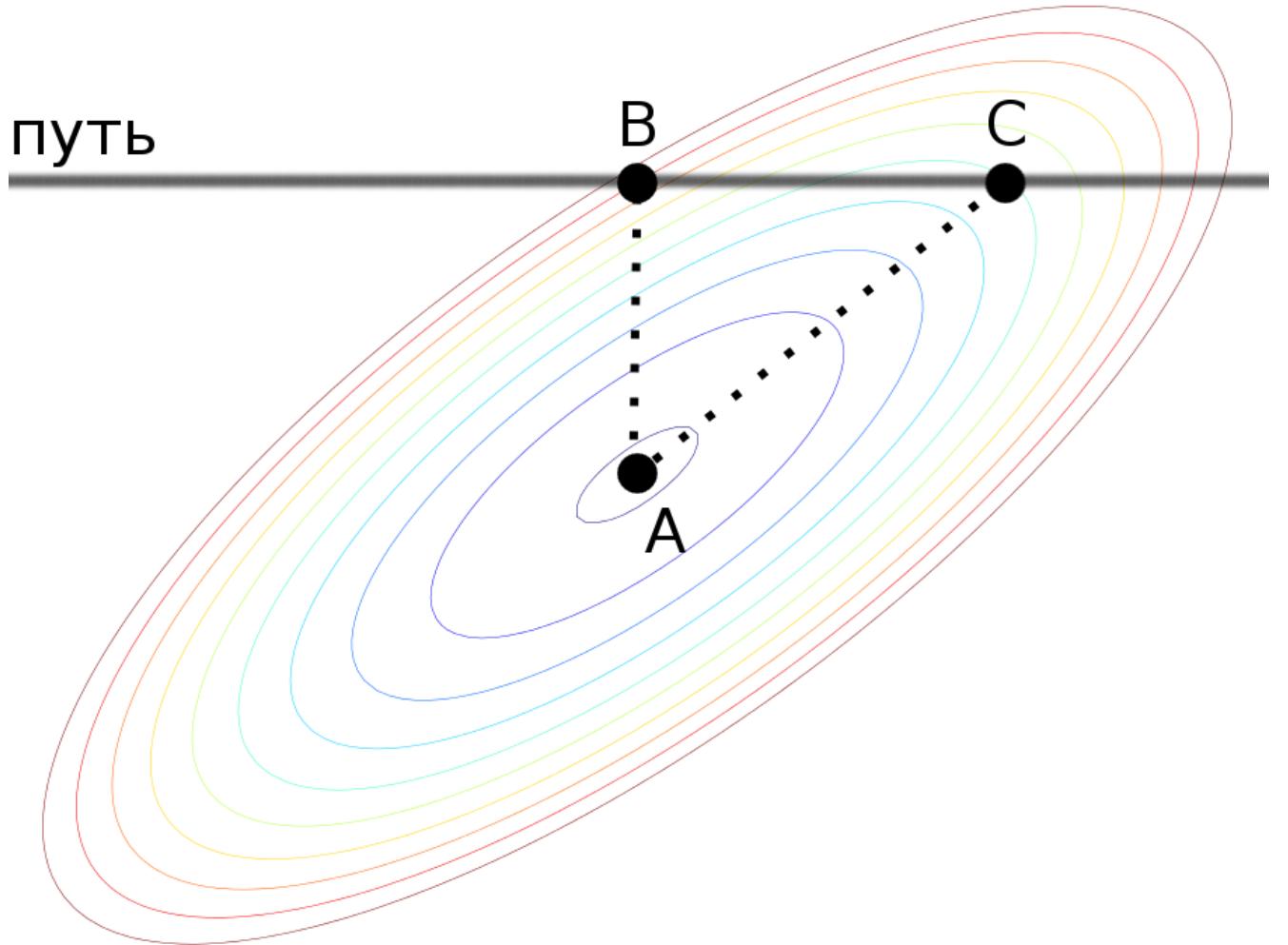


Рис. 2.5: Возможная ситуация при позиционировании транспорта. Изолинии показывают целевую функцию метода наименьших квадратов. Точка А — глобальный минимум, она будет считаться результатом позиционирования. В — проекция минимума на путь, по которому движется транспортное средство. С — точка на пути движения, в которой достигается наименьшее значение целевой функции.

Рассмотрим случай движения транспортного средства по прямой и позиционирования на ней. Как было отмечено в пункте 2.1.1, в случае, когда имеются

избыточные данные о расстояниях до опорных точек, применяются различные методы приближённого решения, такие как метод наименьших квадратов. Конечным этапом применения такого метода будет определение для каждой точки на карте целевой функции, выражающей степень неуверенности в том, что данная точка является искомой, и поиск глобального минимума этой функции [21]. На рис. 2.5 изображено одно из возможных взаимных расположений пути следования транспортного средства и целевой функции. Глобальный минимум, который геолокационный сервер выдаст в качестве ответа на запрос местоположения, из-за ошибок измерения, находится в точке А, не лежащей на маршруте, хотя на самом деле, истинное положение находится где-то на маршруте. Если нам дана только точка А и маршрут движения, то наилучшим приближением будет проекция точки А на маршрут — точка В. Однако, если учесть маршрут движения ещё на этапе, когда доступными являются все значения целевой функции, а не только информация о расположении минимума, то можно вычислить точку С, рассмотрев все точки, лежащие на маршруте, и найдя среди них такую, для которой значение целевой функции минимально, и это будет лучшим приближением.

2.2.2 Снижение точности из-за экстраполяции

Как было отмечено в подпункте 2.1.2.4, ошибки в ходе позиционирования возникают сразу на двух этапах: и на этапе составления карты базовых станций, и на этапе непосредственно позиционирования. Сам метод триангуляции обязывает применять именно такую последовательность действий, однако необходимо обосновать применение триангуляции как таковой. В случае позиционирования ручных и автомобильных устройств, необходимость применения этого метода обусловлена тем, что невозможно измерить уровни принимаемых сигналов во всех точках, где необходимо покрытие геолокационного сервиса. Так, если считать, что для триангуляции достаточно трёх замеров в точках, не лежащих на одной прямой, то для вычисления координат трёх базовых станций, достаточно трёх измерений, если учесть, что их зоны покрытия перекрываются. Три таких замера дадут возможность после этого позиционировать устройства во всей области, где перекрываются зоны покрытия этих станций.

В случае же общественного транспорта ситуация совершенно иная. Для примера рассмотрим московскую трамвайную сеть. На 46 маршрутов находится 1000 трамваев [22]. Если каждый трамвай хотя бы раз в день полностью прохо-

дит свой маршрут, это уже даст $1000/46 = 22$ вагона на одной линии. Приборы, оборудованные системой спутниковой навигации, и собирающие данные об RSSI, будучи установленными всего на один из каждого из 22 вагонов, уже дадут полное обновление карты покрытия для каждой точки каждого маршрута.

Необходимости экстраполировать собранные данные на окружающие точки во всём городе в такой ситуации нет. Достаточно осуществить интерполяцию между имеющимися точками в одномерном случае.

2.2.3 Гипотеза о возможности повышения точности

Анализ факторов, изложенных в пунктах 2.2.1 и 2.2.2, позволяет предположить, что использование свойств, специфичных для наземного общественного транспорта, позволит отказаться от стадий процесса позиционирования, снижающих его точность. В данной работе экспериментально проверяется возможность этого.

2.3 Анализ возможности применения существующих алгоритмов

Рассмотрим, с помощью каких алгоритмов и математических методов можно решать задачу позиционирования, если не использовать триангуляцию.

2.3.1 Постановка задачи

- 1 Даная числовая прямая, обозначающая маршрут движения. Задача отображения двухмерных координат на эту прямую решается отдельно;
- 2 Некоторое количество переменных, выражающих принимаемый уровень сигнала от каждой станции, являются функциями от положения на прямой, определены на всей её длине (отсутствие сигнала можно трактовать как нулевой уровень), но подвержены также случайным флуктуациям;
- 3 Для ряда точек, случайно расположенных на прямой, даны предварительно измеренные значения переменных. Возможна и ситуация неоднократных замеров в одной точке, давших разные результаты, они все попадают в массив измерений;

4 Для кортежа из нескольких измеренных значений найти точку на прямой, где они с наибольшей вероятностью были осуществлены.

Таким образом, требуется найти алгоритм, который будет способен сравнивать точки на прямой по степени их похожести друг на друга, основываясь на значениях набор дискретных, но упорядоченных переменных (то есть, учитывать то, что, например, 13 больше похоже на 14, чем на 42), устойчивый к выбросам (то есть, если какая-то одна станция неожиданно даст сигнал, резко отличающийся от своего нормального значения, это не должно стать помехой) и учитывающий ситуацию в соседних точках (если в некоторой точке никогда не было зафиксировано некое конкретное значение RSSI, но метром больше и метром ближе — было, то для данной точки оно также должно быть представительным). Также желательно учесть распределение результатов измерений для каждой точки, хотя подойдёт и использование среднего значения.

Математического метода, удовлетворяющего всем перечисленным требованиям, найдено не было, однако рассмотрим ряд существующих, наиболее приближающихся к заданным требованиям, на основе которых в дальнейшем будет синтезироваться новый.

2.3.2 Расстояние Махalanобиса

Расстояние Махalanобиса — это обобщение Евклидовой метрики на случай, когда известно, что сравниваемые векторы состоят из случайных переменных [23]. Для одного из сравниваемых векторов определяется матрица ковариации его компонентов, таким образом чётко выражена разница между целым набором статистических данных об измерениях, сделанных в конкретной точке, и вектором, однократно измеренным во время позиционирования. В дальнейшем, эта матрица используется для того, чтобы определить величину дисперсии значений в направлении между сравниваемыми векторами. В случае, когда дисперсия мала (то есть, известно, что получение значений, сильно отличающихся от среднего, маловероятно), расстояние получается больше Евклидова, когда велика — меньше. Определяется расстояние по формуле 2.6.

$$D_M(x) = \sqrt{(x - \mu)^T \cdot S^{-1} \cdot (x - \mu)} \quad (2.6)$$

В формуле 2.6:

- $D_M(x)$ — расстояние Махalanобиса от вектора x до вектора μ ;

- x — однократно измеренный вектор;
- μ — вектор с известным распределением;
- S — матрица ковариации вектора μ .

В поставленной задаче расстояние Махalanобиса можно вычислить для всех точек на маршруте, после чего найти среди них наименьшее — оно и будет точкой наиболее вероятного местонахождения устройства. Оценивая его применимость, отметим, что такой метод идеально справляется с учётом статистических данных об измеренных уровнях, учитывает упорядоченность множества возможных значений уровней, средне справляется с выбросами: если для какой-то конкретной станции известно, что её сигнал в данной точке нестабилен, она будет учитываться с меньшим весом в конечном результате. Однако если выброс даст «надёжная станция», он в значительной мере изменит оценку.

Главным недостатком является тот факт, что такой метод не позволяет учесть непрерывность самого маршрута. Расстояние Махalanобиса можно определить только для одной точки, не учитывая соседние, либо для интервала, в рамках которого точки будут неразличимыми. Если же в некоторой точке уровни не были измерены никогда, то она никогда не будет результатом позиционирования, даже если есть данные о двух соседних.

2.3.3 Байесовский классификатор

Задача позиционирования по своей сути схожа с задачей классификации: если рассматривать каждую точку на маршруте как класс, то он будет определён имеющимися статическими данными о замерах RSSI в данном месте, а само позиционирование будет отнесением вектора измеренных значений к одному из классов. По этой причине в обзор также включён Байесовский классификатор.

Байесовский классификатор обладает следующими свойствами [24]:

- Строится на основании ранее полученных экземпляров класса, даже если сами экземпляры сильно искажены флюктуациями; сложение большого их количества обеспечивает выявление скрытых закономерностей, характерных для класса;
- Крайне устойчив к выбросам; мерой веса каждой из переменных является то, в какой мере она уменьшает энтропию распределения вероятности отнесения примера к различным классам, поэтому чем в большей мере

нормальные значения переменных свидетельствуют в пользу конкретного класса, тем меньший вес имеют выбросы;

- Позволяет учесть распределение переменных в экземплярах одного класса.

Критическим же недостатком этого классификатора является то, что он расчитан на дискретные классы и дискретные значения переменных, не позволяя учесть не только соседние точки, но и ближайшие значения RSSI, упорядоченность множества значений переменных тут не учитывается.

С другой стороны, именно как попытка в некотором смысле обобщения Байесовского классификатора на непрерывный случай и был изобретён метод, предлагаемый в данной работе, и Байесовское понимание вероятности как степени уверенности в суждении, а не частоты появления события, легло в его основу.

3 Технологическая часть

3.1 Общая архитектура системы

3.1.1 Процесс обработки данных

Независимо от того, какой именно математический метод будет осуществлять позиционирование устройства на основе имеющихся данных, можно с уверенностью сказать, что в системе будут поочерёдно или параллельно происходить два процесса: обучение и использование.

Процесс обучения состоит из следующих этапов:

- 1 Измерение RSSI видимых базовых станций устройством, оборудованным модулем спутниковой навигации;
- 2 Передача данных о базовых станциях и положении устройства на сервер;
- 3 Приём и первоначальная обработка данных сервером. В этот этап могут входить такие процессы как проверка корректности, преобразование координат из пары широта/долгота в пару маршрут/расстояние и так далее;
- 4 Сохранение собранной информации в базе данных;
- 5 После набора достаточного количества данных — вычисление необходимых данных для алгоритма позиционирования (обучение).

Процесс позиционирования проходит следующим образом:

- 1 Измерение RSSI видимых базовых станций устройством, не оборудованым модулем спутниковой навигации;
- 2 Приём данных сервером;
- 3 Вычисление предполагаемого расположения устройства с помощью обученного алгоритма;

- 4 Если требуется — обратное преобразование координат в двухмерные (например, для отображения на карте), хотя этот шаг может и отсутствовать (так, для электронного табло на остановке более подходит именно расстояние по маршруту);
- 5 Передача данных о положении устройства клиентам.

3.1.2 Состав системы

На основании данного описания логики работы системы, предложена архитектура, изображённая на рис. 3.1.

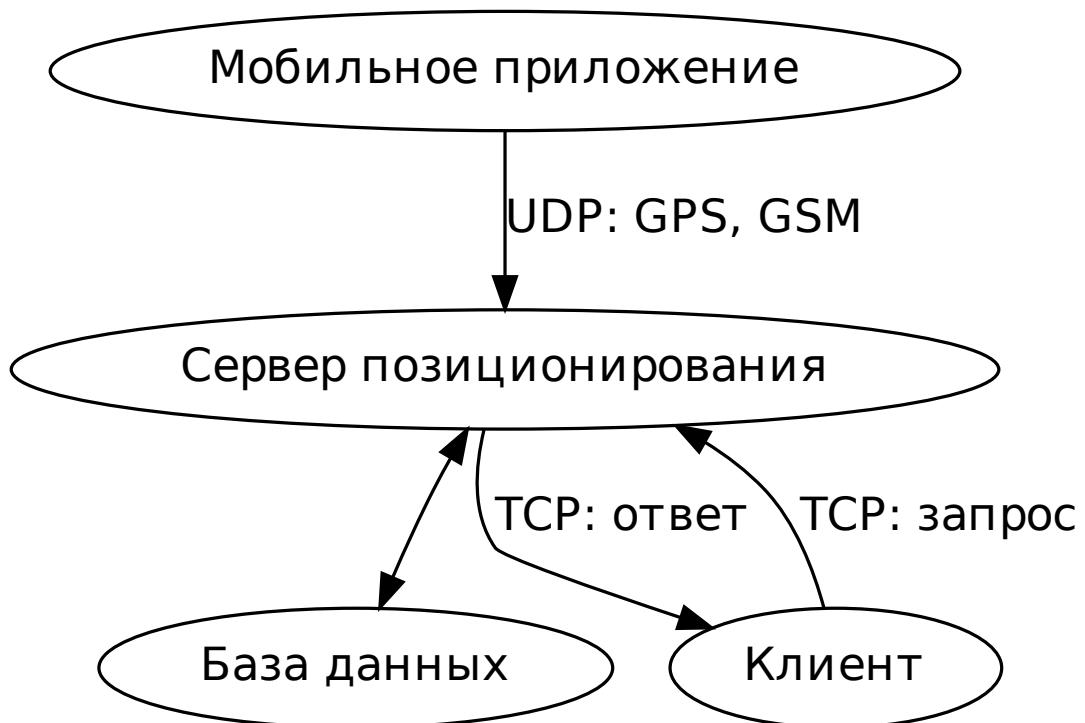


Рис. 3.1: Общая архитектура разрабатываемой системы

Приложение, работающее на мобильном устройстве, в зависимости от типа последнего, передаёт на сервер данные о своём положении и видимых базовых станциях, либо только о базовых станциях. Для передачи используется протокол UDP. Выбор обусловлен тем, что доставка подобных данных не является для системы критичной, а поток данных идёт только в одном направлении.

Связь с клиентами осуществляется по протоколу TCP, причём по запросу клиента. Вызвано это тем, что сервер заранее не обладает информацией о том, куда ему передавать данные, а сами клиенты могут не иметь возможности принять входящее соединение.

3.2 Средства разработки

3.2.1 Система контроля версий

С целью достижения надёжного хранения исходных кодов, возможности получать и изменять код на разных машинах, в том числе и на сервере, возможности отката к предыдущим версиям и предоставления кода всем желающим, была использована система контроля Subversion, официальный кафедральный репозиторий которой находится по адресу <http://svn.auditory.ru/>.

3.2.2 Мобильное приложение

В качестве как позиционируемого устройства, так и прибора для контрольных замеров RSSI в данной работе использован телефон под управлением операционной системы Android. Данная операционная система имеет удобные средства для работы как с данными об RSSI [11], так с GPS [25]. Пользовательские приложения данной ОС исполняются на виртуальной машине dalvik [26], основным языком программирования для которой является Java, SDK для которой представляют разработчики ОС [27].

Разработка под Android возможна как и с помощью ручного управления файлами, так и через IDE. Одной из официально поддерживаемых IDE является Eclipse, который и был выбран для разработки.

3.2.3 Сервер и СУБД

Платформой для серверной части был выбран дистрибутив Gentoo Linux, VPS под управлением которого возможно было использовать для тестирования проекта. Имеющаяся там СУБД mysql также подошла для проекта.

Языком программирования был выбран Python. Его стандартная библиотека покрыла все нужды, кроме быстрых вычислений и соединения с базой данных. Для вычислений использовалась библиотека NumPy [28], а для соединения с базой данных — MySQLdb [29]. Будучи написанной на интерпретируемом языке

ке и использующей только библиотечные средства взаимодействия с внешними компонентами, программа может работать везде, где имеется интерпретатор языка Python версии не ниже 2.6 с установленными библиотеками NumPy и MySQLdb.

Вместо IDE в разработке данной части применялся консольный текстовый редактор vim [30]. Он предоставляет огромные возможности для редактирования исходных кодов, а тот факт, что его интерфейс является полностью консольным, позволил использовать его для редактирования кода серверной части непосредственно на VPS через подключение по удалённой консоли SSH.

3.2.4 Клиентская часть

Для проверки работы системы была написана графическая клиентская часть, отображающая положение позиционируемого устройства на карте тестового участка. В её разработке также использован язык программирования Python вместе с библиотекой Pygame [31] для отображения графики. Эта библиотека также реализована под множество платформ, и клиентскую часть можно запустить под любой из них.

3.2.5 Пояснительная записка

Для оформления пояснительной к данному дипломному проекту, являющейся его важной частью [32], главным образом применялись следующие средства:

- L^AT_EX [33] — система вёрстки, специально разработанная для оформления научных работ, и рекомендованная многими специалистами для оформления также и дипломных проектов [34];
- JabRef [35] — работающая в связке с L^AT_EX система для создания и редактирования библиографии;
- Graphviz [36] — система для автоматического отображения графов и диаграмм по их текстовому описанию;
- vim — текстовый редактор для создания данных текстов;
- GNU Octave [37] — математический пакет, использованный для построения графиков и гистограмм;

- GNU Make [38] — система для автоматизированного управления компиляцией проекта в формат PDF.

Исходные коды содержания и графиков пояснительной записки также находятся под управлением Subversion.

4 Охрана труда

Осознавая первоочередную важность сохранения жизни и здоровья пользователей любой системы и первоочередной приоритет требований безопасности перед всеми остальными, рассмотрим, прежде, чем начинать разработку, какой потенциальный вред пользователю может нанести подобная система.

Все виды опасностей на производстве можно разделить на следующие виды [39]:

- 1 Механические, такие как удары, падения, поражения высоким давлением, а также иные физические, такие как свет, ионизирующее излучение, звук, электрический ток и тому подобные;
- 2 Биологические, такие как вирусы, бактерии, грибки, паразиты и так далее;
- 3 Химические, такие как кислоты, основания, тяжёлые металлы и другие;
- 4 Заболевания опорно-двигательного аппарата;
- 5 Психологические, такие как моббинг [40], эмоциональное выгорание [41], негативное воздействие среды, принуждающей к незддоровой активности (такой как неумеренное употребление спиртных напитков, курение и так далее) как необходимому условию карьерного роста и интеграции в коллектив.

Важно отметить, что со временем меняются не только непосредственно опасные факторы, как результат научно-технического прогресса, но и представления о них, как официальные, так и неофициальные. Рассмотрим, например, такой вредный фактор как моббинг. Согласно данным аналитической системы Google Trends, слово *mobbing* является популярным поисковым запросом уже с 2004 года, и, судя по тенденции, было таковым раньше, но только в 2008 начался рост его упоминаний в новостях (см. рис. 4.1). Найти же нормативные

документы, упоминающие моббинг как официально принимаемый во внимание вредный фактор не удалось вообще.

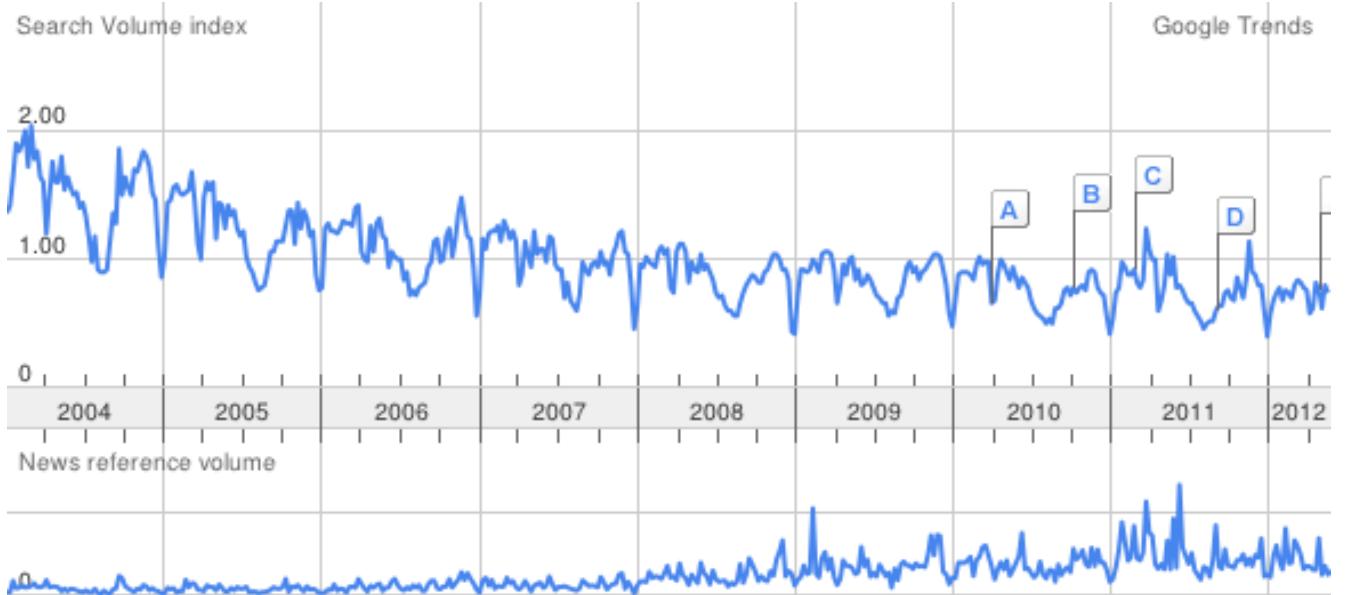


Рис. 4.1: Частота поиска и упоминания в новостях слова *mobbing* по данным аналитической системы Google Trends.

Рассматривая потенциальный вред от разрабатываемой системы, отметим, что все её компоненты являются исключительно программными средствами, а потому физический вред нанести в той же мере, в которой те аппаратные средства, на которых система выполняется. Однако ни в коем случае нельзя оставлять за пределами рассмотрения потенциальный психологический вред. Так, известно наличие в сети Интернет сайтов, таких как 4chan, длительное пребывание на которых, по мнению многих аналитиков [42], может приводить к деформации личности.

К счастью, можно сразу сказать, что система позиционирования транспорта не предполагает взаимодействие между её пользователями, а сама по себе, хоть и использует алгоритмы машинного обучения, не является в достаточной мере интеллектуальной, чтобы наносить психический вред самостоятельно.

Таким образом, имеет смысл рассматривать вред, который может нанести система через ЭВМ. Здесь также можно отметить значительное устаревание имеющихся нормативных документов. Так, основной нормативный документ, описывающий санитарные требования к рабочему месту оператора ЭВМ — САНПИН 2.2.2/2.4.1340-03 [43] уделяет большое внимание вопросу рентгеновского излучения ЭЛТ-мониторов, несмотря на то, что те из них, которые могли сохранить работоспособность до 2012 года, гарантированно относятся к поколению, в котором вся необходимая защита встроена в корпус трубки. С дру-

гой стороны, такому явлению, как чтению с экрана, уделено недопустимо мало внимания. Такая характеристика как контрастность вообще не рассматривается, считаются только знаки, как будто они все равнозначны. При том, что многие люди действительно не осведомлены о том, что контрастность текста измерять нужно, и делать это в чёрно-белом варианте, в связи с чем недопустим, например, светло-зелёный текст на розовом фоне, хотя эти цвета очень легко различить.

Не уделяется там и внимание такому фактору, давно изучаемому специалистами по интерфейсам, но не специалистами по охране труда, как шрифты. А ведь замечено, что неправильно подобранный шрифт может существенно уменьшить время, которое человек может провести за чтением текста, не устав.

Ещё одним фактором является мерцание и быстрая смена картинки. На это тоже нет никаких нормативов, в связи с чем, зная, что серверная часть разрабатываемой системы будет выводить на экран много текста с большой скоростью, авторам работы остаётся лишь порекомендовать пользователям, во избежание раздражения и покраснения глаз, не злоупотреблять использованием консольного интерфейса сервера, отдавая предпочтение графическому клиенту. Нормировать этот параметр не представляется возможным за отсутствием необходимых норм.

Таким образом, несомненно, можно выделить много потенциально негативных факторов в работе с любым программным продуктом, но те требования, которые были предъявлены 10 лет назад и продолжают предъявляться до сих пор, не соответствуют реальным потребностям повышения юзабилити, не могут быть выполнены и не выполняются в современных условиях, при том, что новые опасные факторы не нормируются или нормируются с большим опозданием.

5 Разработка

5.1 Алгоритм

В данном подразделе рассматривается алгоритм позиционирования, начиная с того момента, когда данные об уровнях сигнала уже привязаны к точкам на маршруте, а после позиционирования не требуется осуществлять обратное преобразование. Вопрос прямого и обратного преобразования рассмотрен в подпункте 5.3.1.2.

5.1.1 Исходные данные

Исходными данными алгоритма являются, с одной стороны, хранящиеся в базе данных тройки $\langle CID, RSSI, Dist \rangle$ (CID — уникальный идентификатор базовой станции, $RSSI$ — уровень принятого сигнала от неё в точке с координатой $Dist$), которые были измерены ранее, и значение $Dist$ для которых посчитано на основе данных спутниковой навигации, а с другой — пары $\langle CID, RSSI \rangle$ измеренные устройством, координаты которого требуется найти.

Эскиз возможного вида таких данных показан на рис. 5.1. Хранящиеся в базе данных замеры соединены прямыми линиями только для того, чтобы не путать замеры сигналов от разных станций на рисунке, реально хранятся только точки. Из базы данных осуществляется выборка тех замеров, CID которых соответствует CID фактически принятых сигналов.

5.1.2 Интерполяция

Для того, чтобы перейти от дискретной оси расстояния к непрерывной, используется интерполяция. Данные о каждой из рассматриваемых базовых станций интерполируются многочленами с помощью линейной регрессии. Из переменной $Dist$ создаётся набор $\langle 1, Dist, Dist^2, Dist^3, \dots, Dist^n \rangle$. Этот набор вместе со значениями $RSSI$ составляет систему уравнений, решаемую методом нор-

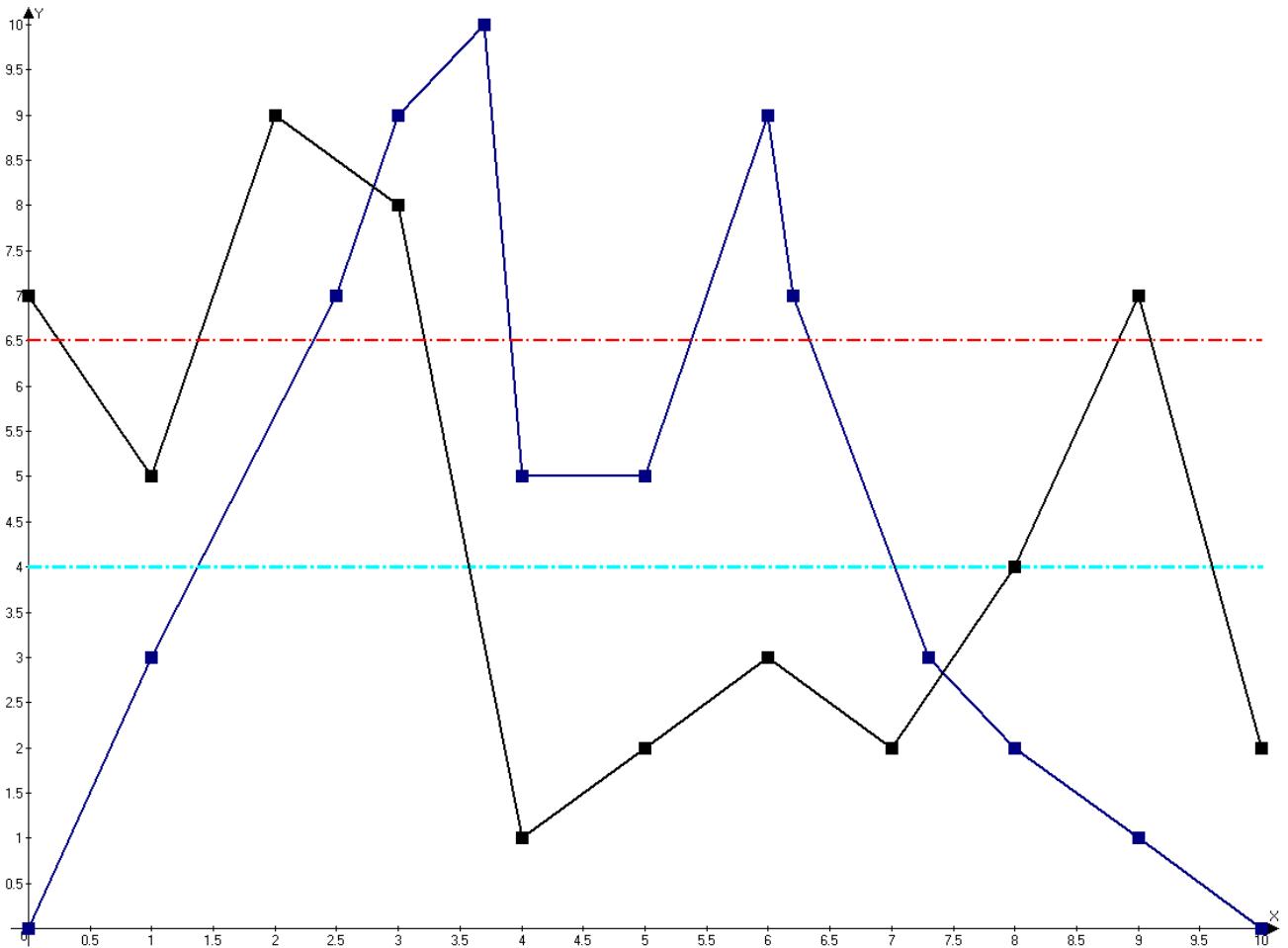


Рис. 5.1: Эскиз возможных входных данных для алгоритма позиционирования. Вертикальная ось — уровень сигнала, горизонтальная — координата. Штрих-пунктирными линиями обозначены уровни фактически принятых сигналов от двух базовых станций (поскольку их координаты неизвестны, изображены на всём протяжении маршрута). Точки, соединённые сплошными отрезками — хранящиеся в базе данных замеры RSSI этих станций с известными координатами.

мальных уравнений. Результатом являются коэффициенты многочлена, оптимизированные по методу наименьших квадратов.

Эскиз этого процесса изображён на рис. 5.2.

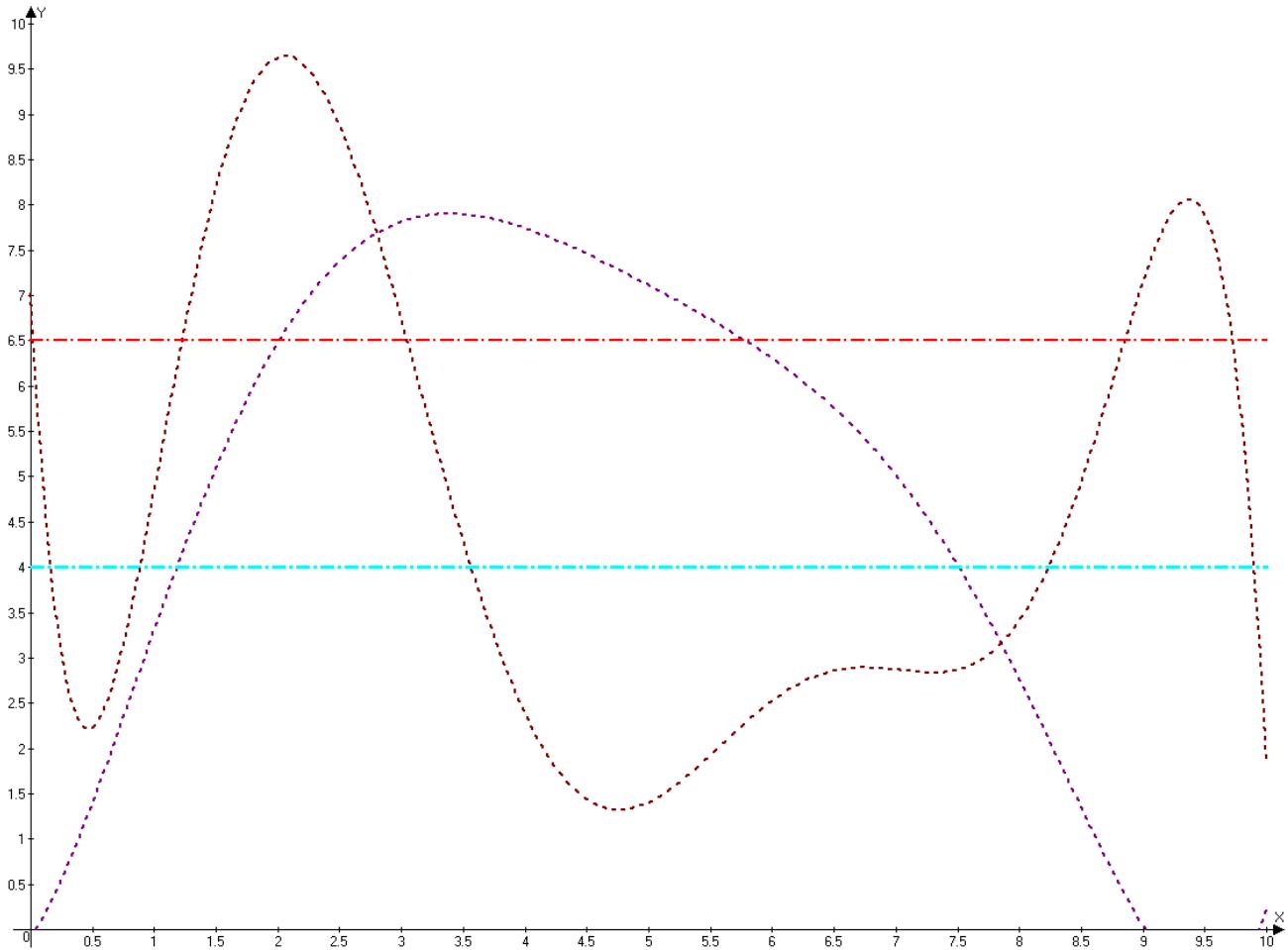


Рис. 5.2: Эскиз возможной интерполяции в ходе алгоритма позиционирования. Вертикальная ось — уровень сигнала, горизонтальная — координата. Штрих-пунктирными линиями обозначены уровни фактически принятых сигналов от двух базовых станций (поскольку их координаты неизвестны, изображены на всём протяжении маршрута), штриховыми — функции, интерполирующие имеющиеся в базе данных замеры.

5.1.3 Вычисление псевдоплотности вероятности

Одним из ключевых достижений данной работы является введение и применение для позиционирования функции псевдоплотности вероятности.

Определим функцию псевдоплотности вероятности как функцию двух переменных $P(f(x), y)$, где y — фактически измеренное значение RSSI в некоторой неизвестной точке, а $f(x)$ — значение интерполяционного многочлена в известной точке x , удовлетворяющую следующим условиям:

$$1 \quad \forall_{f(x),y} P(f(x), y) \in (0, 1]$$

$$2 \quad \forall_{f(x),y} f(x) = y \Leftrightarrow P(f(x), y) = 1$$

$$3 \lim_{|f(x)-y| \rightarrow \infty} P(f(x), y) = 0$$

Функция, удовлетворяющая данным критериям, будет обладать следующим свойством: в любой точке маршрута, чем более фактический замер похож на значением многочлена в данной точке, тем ближе эта функция к 1, а чем дальше — к 0. Этим свойством данная функция похожа на Байесовское понимание вероятности, а потому названа псевдоплотностью вероятности. Более того, также аналогично Байесовской вероятности, умножение двух таких функций, вычисленных для разных базовых станций, позволяет одновременно сравнивать похожесть по двум параметрам. За счёт того, что операцией, связывающей две псевдоплотности вероятности, является не сложение, как в функции расстояния Махalanобиса (пункт 2.3.2 на странице 24), а умножение, как в Байесовском классификаторе (пункт 2.3.3 на странице 25), все выбросы, если только не случится так, что в результате флюктуаций сигнал сразу от нескольких станций станет напоминать другую точку, взаимно уничтожаются.

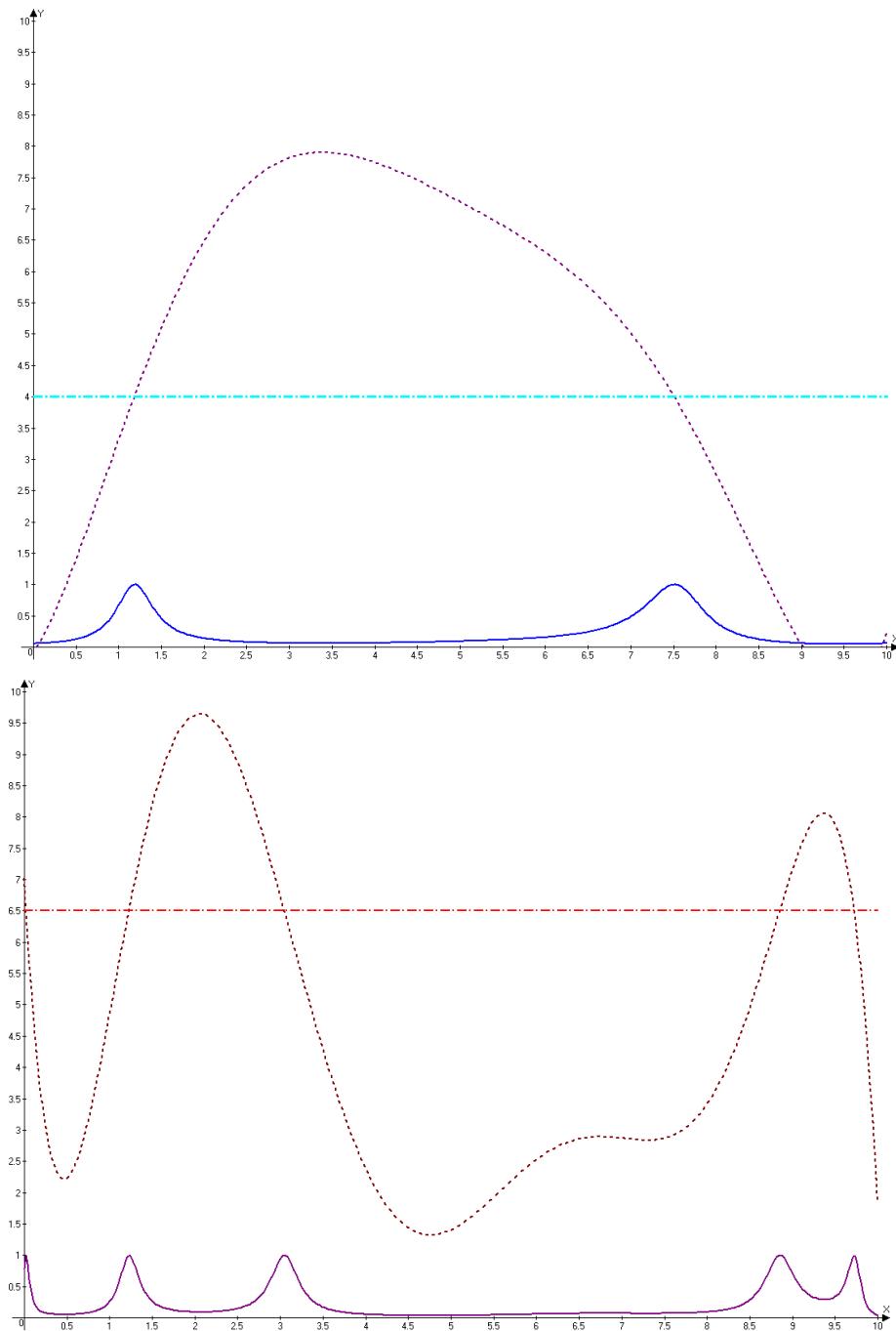


Рис. 5.3: Эскиз процесса расчёта псевдоплотности вероятности для двух базовых станций. Горизонтальная ось — расстояние, вертикальная — RSSI и значение псевдоплотности (не в масштабе). Штрихпунктирная линия — фактически измеренные значения RSSI, штриховая — интерполяционный многочлен, сплошная — значение псевдоплотности.

Отличает такую функцию от настоящей плотности вероятности следующее:

- 1 Она может точно достигать значения 1, а не только приближаться к нему;
- 2 Не требуется, а потому не гарантируется то, что интеграл этой функции по всей области определения равен 1;

3 Даже если нормировать интеграл этой функции на 1, не гарантируется, что интеграл по некоторому участку действительно в точности равен частотной вероятности нахождения там устройства, хотя в Байесовском понимании она отражает степень уверенности в том, что устройство именно там.

Вычисление истинной плотности вероятности не требуется, потому что для работы алгоритма её не нужно интегрировать, а только умножать и искать аргумент максимального значения. А в отношении этих свойств, данная функция сохраняет свойства, присущие плотности вероятности.

Отказ от возможности интегрирования, с другой стороны, позволяет ограничиться очень простым видом функции псевдоплотности вероятности, таким как 5.1.

$$P(f(x), y) = \frac{1}{1 + (f(x) - y)^2} \quad (5.1)$$

Эскиз процесса вычисления псевдоплотности вероятности показан на рис. 5.3.

5.1.4 Вычисление ответа

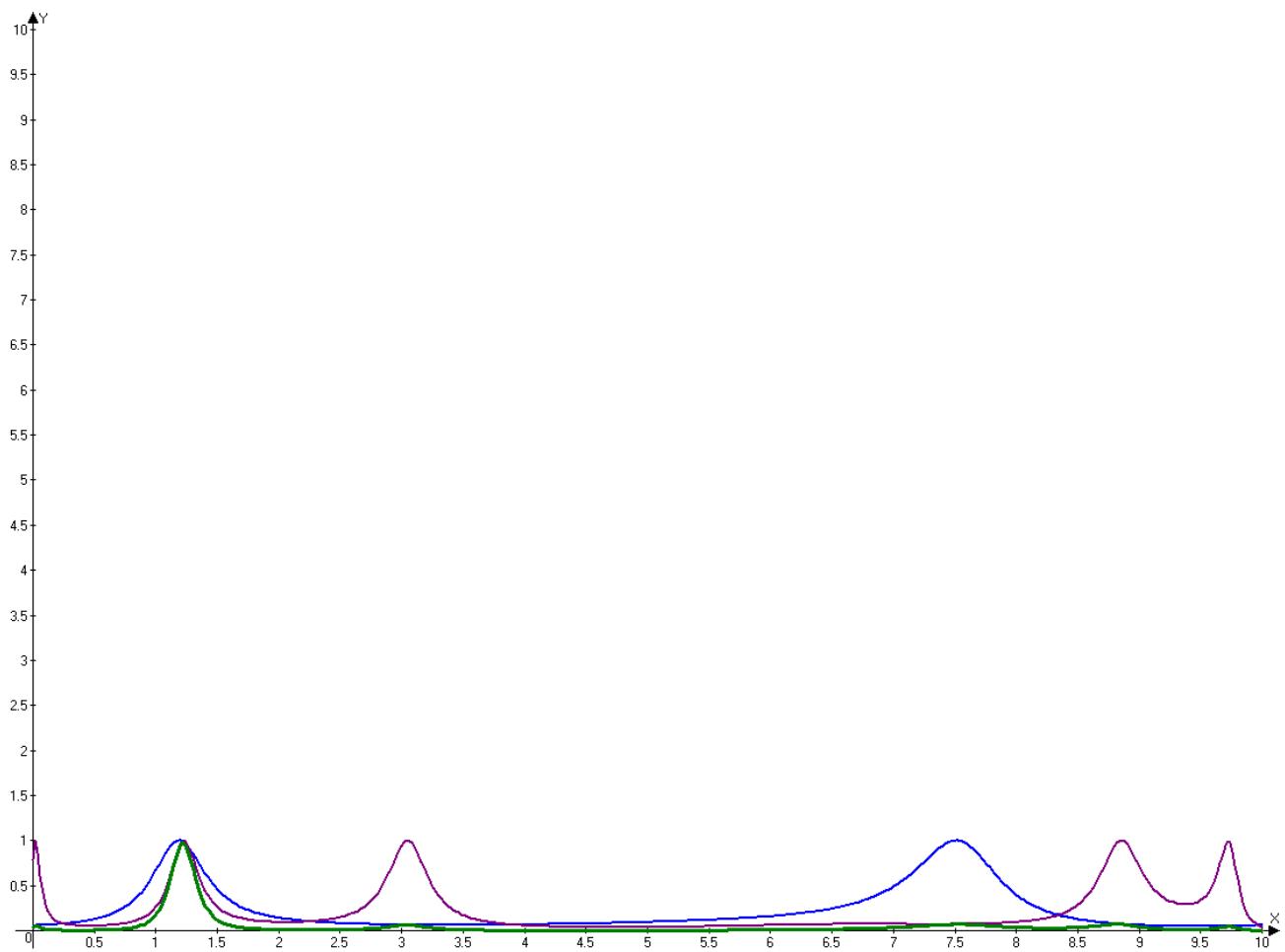


Рис. 5.4: Эскиз произведения функций псевдоплотности для нескольких базовых станций. Горизонтальная ось — точки на маршруте, вертикальная — значения псевдоплотности. Результирующая псевдоплотность имеет всего один хорошо выраженный глобальный максимум между координатами 1 и 1,5.

После расчёта значений псевдоплотности вероятности для каждой базовой станции, процесс позиционирования сводится к их перемножению и поиску аргумента глобального максимума. Эскиз этого процесса показан на рис. 5.4.

5.2 Мобильное приложение

Для ОС Android, выполняющейся на устройстве, оборудованном модулями GPS и GSM, а также предоставляющем доступ в интернет, требуется написать приложение, получающее данные об RSSI видимых базовых станций и своим местоположении, после чего передающее их на сервер.

5.2.1 Протокол передачи данных

Собранные данные передаются по протоколу UDP на сервер, будучи записанными в формате JSON в сообщения следующего вида:

```
{  
    "GSM": {  
        "cellcount": 2,  
        "cells": [  
            {  
                "CID": 11531,  
                "Psc": -1,  
                "RSSI": 26,  
                "type": "EDGE"  
            },  
            {  
                "CID": 32779,  
                "Psc": -1,  
                "RSSI": 22,  
                "type": "EDGE"  
            }  
        ]  
    },  
    "GPS": {  
        "lng": 37.64814019203186,  
        "ltd": 55.75437605381012,  
        "acc": 24.0  
    }  
}
```

Значения содержимого полей описаны в таблице 5.1.

5.2.2 Интерфейсы

Особенностью программирования под Android SDK является тот факт, что в программе нет аналога функции или метода Main из многих популярных языков программирования. Вся программа состоит из обработчиков различных событий, происходящих в графическом интерфейсе или внутри ОС Android. За-

Таблица 5.1: Значения полей пакета в формате JSON, передаваемого от мобильного устройства на сервер.

Поле	Тип	Значение
GSM	Объект	Набор данных о видимых базовых станций
cellcount	Число	Количество станций, передаваемых в данном пакете
cells	Массив	Набор объектов, каждый из которых соответствует одной базовой станции
CID	Число	Уникальный идентификатор базовой станции, 2 байта
Psc	Число	Зарезервировано для сетей 3G
RSSI	Число	Уровень принятого сигнала в asu
type	Строка	Тип соединения. Возможные значения: «GPRS», «EDGE», «UMTS», «UNKNOWN»
GPS	Объект	Набор данных спутниковой навигации
lng	Число	Долгота в градусах
ltd	Число	Широта в градусах
acc	Число	Погрешность позиционирования в метрах

пустить код, не зависящий от обработчиков, можно только создав отдельный поток во время обработки одного из событий в главном.

В связи с этим, приложение можно полностью описать через его взаимодействие с иными компонентами.

5.2.2.1 Графический интерфейс



Рис. 5.5: Графический интерфейс написанного мобильного приложения, запущенного на телефоне HTC Hero под управлением Android 2.2.

Графический интерфейс приложения изображён на рис. 5.5. По нажатию кнопки «Start»/«Stop» начинается или приостанавливается работа фонового потока, обеспечивающего сбор и передачу данных. Этому потоку передаётся адрес и порт, по которому нужно посыпать данные, значение флага GPS, определяющего, надо ли передавать данные спутникового позиционирования, или же только уровни сигнала. При неустановленном флаге UDP, фоновый поток не передаёт данные в сеть, а только возвращает их потоку графического интерфейса, который при наступлении такого события как приход сообщения о

фонового, обновляет отображаемый список.

Поле Device ID и флаг Secure в описываемой версии не используются.

5.2.2.2 Данные об уровнях сигнала

Информацию об уровнях сигналов видимых базовых станций предоставляет-
ся объект класса TelephonyManager, который инстанцируется операционной
системой в главном потоке. Информацию он предоставляет через синхронный
вызов функции, возвращающей ответ. В связи с этим, обработка этих данных
ведётся в фоновом потоке.

5.2.2.3 Данные спутниковой навигации

Данные же GPS поступают асинхронно. При включённом модуле GPS операци-
онная система сама отслеживает факт измерения координат, и создаёт событие
обновления, которое можно обработать. Эта обработка осуществляется в глав-
ном потоке, после чего её результаты передаются фоновому.

5.2.2.4 Взаимодействие с сетью

Передача данных через UDP осуществляется синхронным вызовом функции, а
потому вместе с обработкой данных об уровнях сигнала вынесена в фоновый
поток.

5.3 Серверная часть

Сервер, в целом, осуществляет две задачи: приём и обработка поступающих
данных и передача результатов работы клиенту. Разница между этими про-
цессами заключается в том, что приходящие по UDP данные обрабатываются
синхронно, а передача клиенту по TCP — асинхронно. В связи с этим, сервер
работает в два потока, передача данных между которыми осуществляется через
библиотечный класс Queue. Схема этого процесса изображена на рис. 5.6.

Сервер может работать в двух режимах — сбор данных и тестирование по-
зиционирования. Часть классов и модулей является общей для обоих режимов,
часть — специфичной для позиционирования. Начнём, поэтому, с описания ре-
жима сбора данных, в рамках описания которого опишем и общие классы.

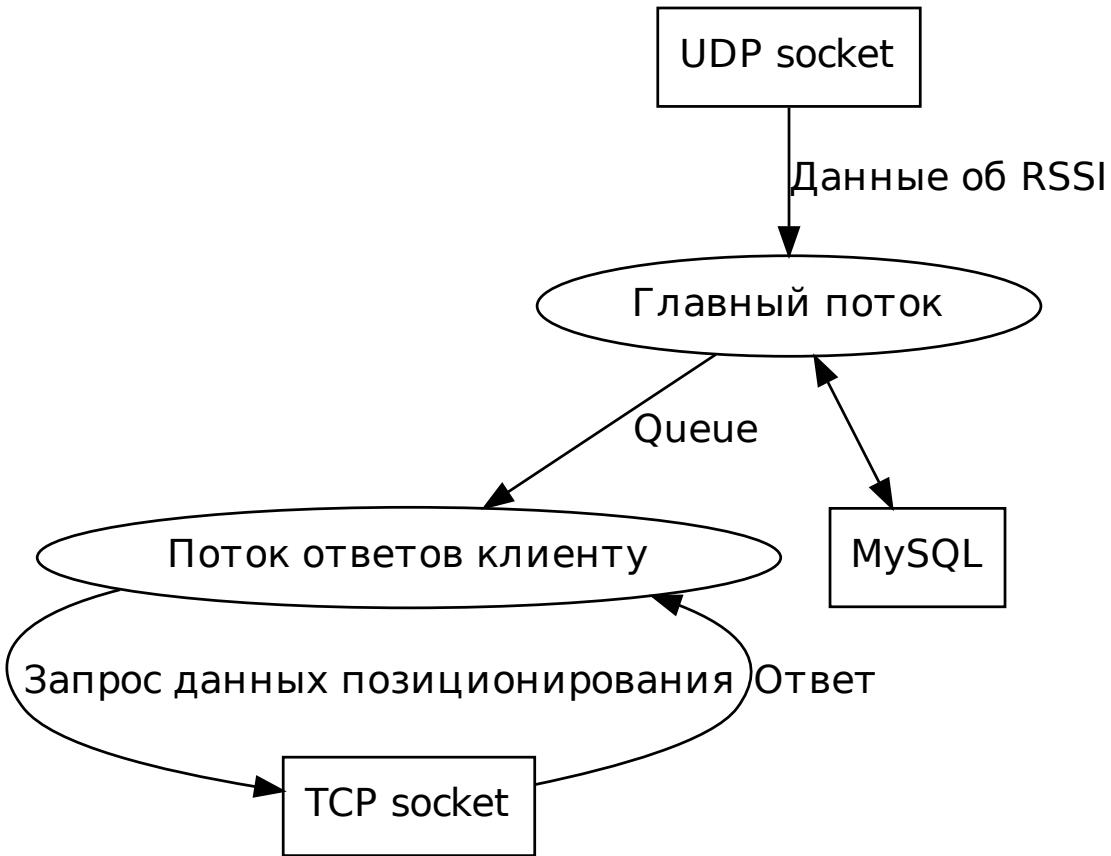


Рис. 5.6: Взаимодействие между потоками сервера (эллипсы) и внешними интерфейсами (прямоугольники).

5.3.1 Режим сбора данных

Общая схема направлений потоков данных в режиме сбора данных изображена на рис. 5.7. Не отображено преобразование данных в формате JSON во внутренне представление Python, но поскольку и прямое, и обратное преобразование осуществляются каждой одной функцией, их в описании потоков опускаем. Рассмотрим теперь подробнее каждый из классов и модулей, участвующих в обработке данных.

5.3.1.1 Проверка входящих данных

Модуль **validator** содержит единственную функцию с таким же названием, осуществляющую проверку входящих данных на наличие и корректные значения всех полей, описанных в 5.1. В случае, если находится несоответствие, функция возвращает специальное значение `None`, получение которого обрабатывается, в

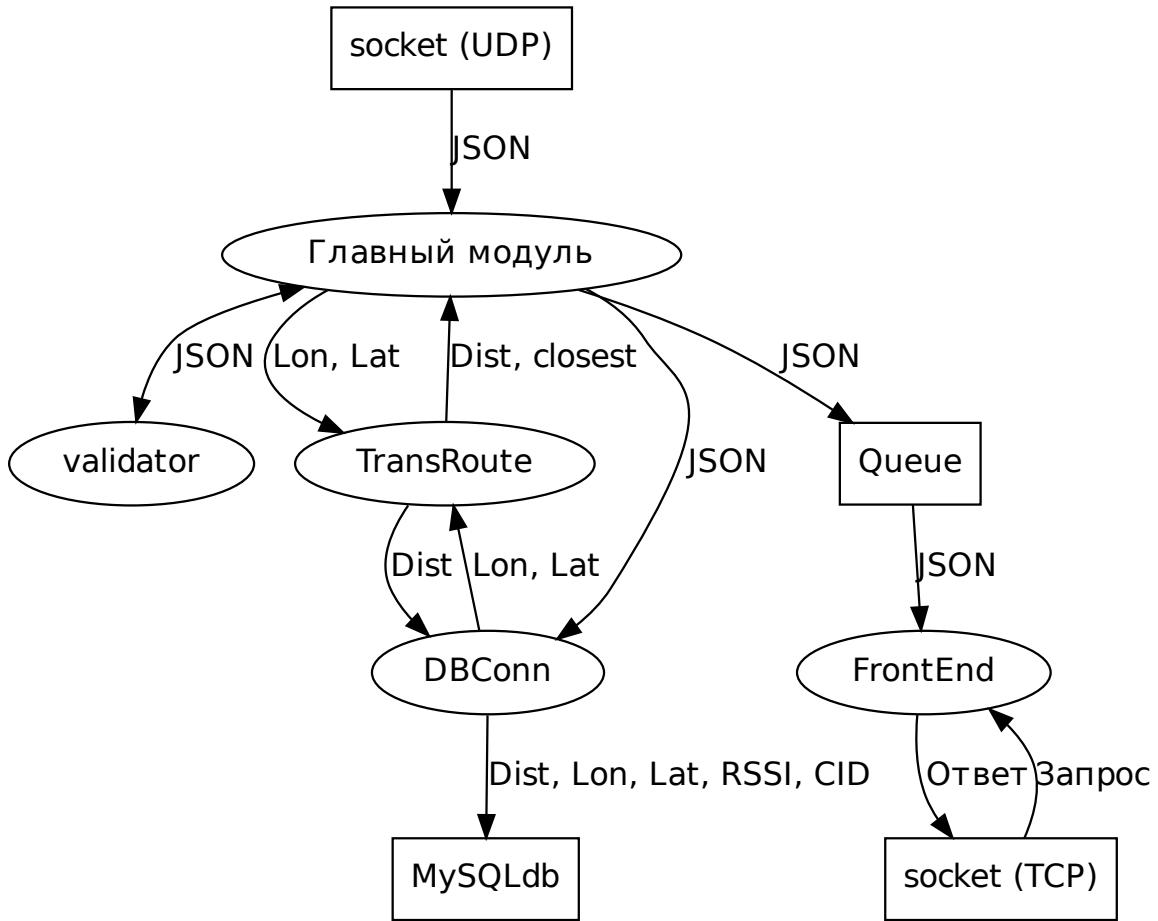


Рис. 5.7: Потоки данных между классами сервера (эллипсы) и внешними интерфейсами (прямоугольники) в режиме сбора данных.

противном случае — переданный объект.

5.3.1.2 Преобразование координат

Класс **TransRoute** осуществляет три операции: преобразование пары широта/долгота в расстояние от начала маршрута, обратно — расстояние в широту и долготу, а также находит точку на маршруте, ближайшую к переданной.

В прототипе системы была реализована обработка единственного маршрута, представляющего собой прямую линию с известными координатами начала и конца. Нетривиальной в таком случае является только одна задача: поиск точки на маршруте, ближайшей к данной. Необходимость её решения вызвана тем, что GPS сам обладает некоторой погрешностью, и может возвращать координаты не строго на маршруте, а немного сбоку от него, поэтому сначала полученную точку надо найти её проекцию на маршрут.

Координаты начала и конца маршрута передаются в конструктор класса, сам класс имеет следующие методы:

- **closest** — принимает описание широты и долготы точки в формате JSON, возвращает кортеж из двух координат её проекции на маршрут;
- **point_to_dist** — принимает описание широты и долготы точки в формате JSON, возвращает число — расстояние от начала маршрута до проекции переданной точки на него. Расстояние имеет знак, если выйти за границу маршрута со стороны начала, будет также корректным расстоянием по модулю, но отрицательным;
- **dist_to_longlat** — принимает расстояние от начала маршрута до искомой точки, возвращает её географические координаты.

5.3.1.3 Взаимодействие с базой данных

Все запросы к базе данных осуществляются через класс **DBConn**. Подробное описание базы данных находится в подразделе 5.4, а здесь дано общее описание осуществляемых запросов:

- **save_netdata** — принимает объект JSON с данными о GPS и RSSI, записывает их в базу данных как в виде географических координат, так и в виде точек на маршруте; для преобразования использует класс TransRoute;
- **avg_rssi** — принимает CellID, возвращает среднее значение RSSI по всем точкам;
- **all_cells** — возвращает список из всех CellID, данные о которых имеются в базе;
- **samples_by_sell** — принимает CellID, возвращает все результаты замеров для неё.

5.3.1.4 Взаимодействие с клиентами

Класс **FrontEnd** наследует библиотечный класс **threading.Thread** и перегружает единственный метод **run**, выполняющийся в отдельном потоке. Метода **run**, будучи запущенным, открывает сокет для входящих соединений от клиентов, после чего в бесконечном цикле ждёт их. После установки соединения,

во вложенном бесконечном цикле пытается считать сообщение из очереди, идущей от главного потока, и если это удалось, посыпает его в сокет. При обрыве соединения вложенный бесконечный цикл завершается с исключением, которое обрабатывается внешним, после чего последний снова ожидает соединений.

Передача данных начинается сразу по мере установки соединения клиентом, без ожидания запросов с его стороны. Передаются данные в формате JSON следующего вида:

```
{  
    "GPSerrm": 5.8398814870603832,  
    "Route": {  
        "lng": 37.647922706087265,  
        "dstm": 260.0450561256485,  
        "dst": 0.0023370777009036913,  
        "ltd": 55.754401640921643  
    },  
    "GPSerr": 5.2484200248492898e-05,  
    "GPS": {  
        "lng": 55.75437605381012,  
        "ltd": 37.647968530654907  
    }  
}
```

Значения полей описаны в таблице 5.2.

5.3.1.5 Главный модуль

Алгоритм работы главного модуля в режиме сбора данных состоит из инициализации всех классов, открытия UDP-сокета для входящих датаграмм от мобильных устройств и перехода в следующий бесконечный цикл:

- 1 Принятие сообщения;
- 2 Проверка сообщения модулем validate;
- 3 Запись сообщения в базу данных модулем DBConn;
- 4 Вычисление проекции принятой точки на маршрут, её расстояния от начала маршрута, расстояния между исходной точкой и её проекцией;

Таблица 5.2: Значения полей пакета в формате JSON, передаваемого от сервера в режиме сбора данных клиентам.

Поле	Тип	Значение
GPSerrm	Число	Расстояние между точкой и её проекцией на маршрут в метрах
Route	Объект	Информация о точке на маршруте
lng	Число	Долгота в градусах
ltd	Число	Широта в градусах
dst	Число	Расстояние от начала маршрута в градусах
dstm	Число	Расстояние от начала маршрута в метрах
GPSerr	Число	Расстояние между точкой и её проекцией на маршрут в градусах

5 Запись посчитанных значений в формат JSON и отправка в очередь потока FrontEnd.

5.3.2 Режим позиционирования

Режим позиционирования отличается от режима сбора данных тем, что реакцией на приход сообщения от мобильного устройства является не помещение этой информации в базу данных, а позиционирование на основе собранных данных, проверка качества работы на основе пришедших данных спутникового позиционирования и отправка этих данных клиенту. Соответственно, в архитектуру добавляются классы, обеспечивающие процесс позиционирования. Общий вид потоков данных в режиме позиционирования изображён на рис. 5.8.

5.3.2.1 Интерполяция и псевдоплотность

Класс **Interpol** предоставляет набор функций для построения интерполирующего многочлена, вычисления его значений для конкретных точек, а также вычисления псевдоплотности вероятности для данной точки на маршруте и данного уровня сигнала. Один экземпляр данного класса описывает перечисленные сущности для одной базовой станции.

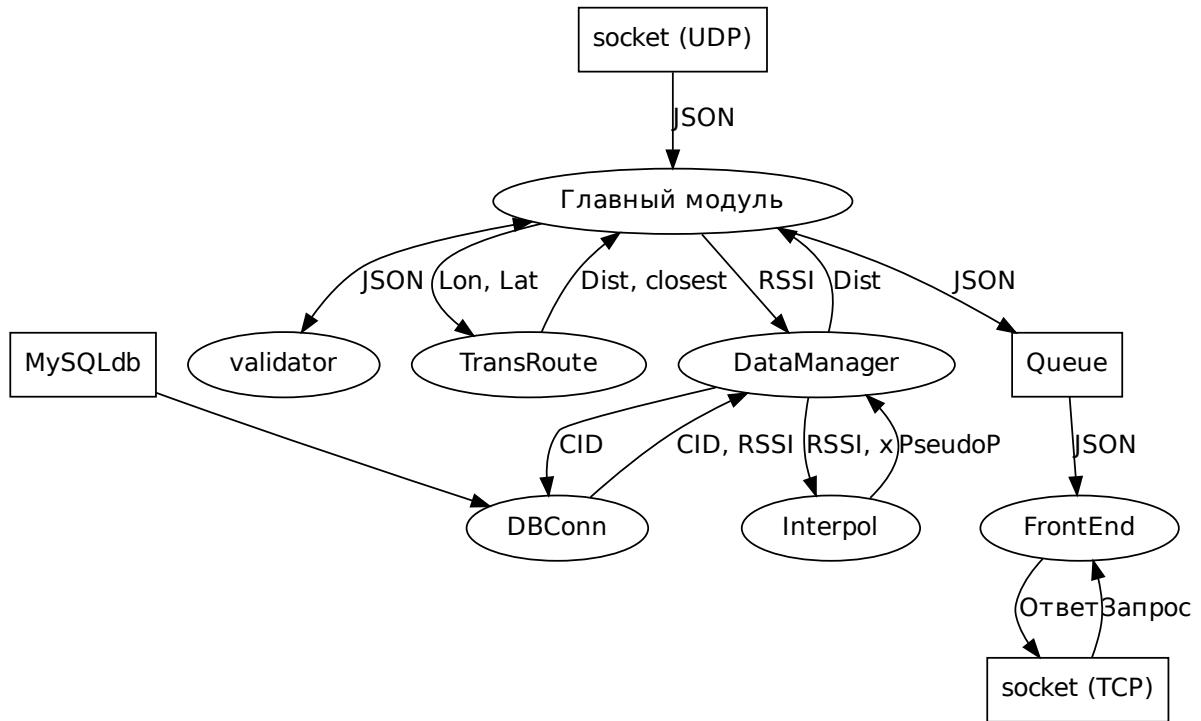


Рис. 5.8: Потоки данных между классами сервера (эллипсы) и внешними интерфейсами (прямоугольники) в режиме позиционирования.

Код этого класса, как реализующего основу разработанного математического метода, приводится целиком:

```
#-*- coding: utf8 -*-
import numpy
import numpy.linalg

import sys
class Interpol(object):
    def __init__(self, order=10):
        self.order = order
        self.theta = [0]*(self.order+1)

    def create_vars(self, data):
        l = len(data)
        print l
        self.X = numpy.empty([l, self.order+1])
        self.Y = numpy.empty([l, 1])
```

```

try:
    for j in xrange(1):
        for i in xrange(0, self.order+1):
            self.X[j][i] =\
                data[j][0]**i
            self.Y[j][0] = data[j][1]
except Exception as e:
    print e
    print "i:", i, "j:", j
    print "order:", self.order
    print "data[j]:", data[j]
    print "self.X[j][i]", self.X[j][i]
    print "self.Y[j][0]:", self.Y[j][0]
    sys.exit(0)

def solve_theta(self):
    self.theta = numpy.transpose(self.X)
    self.theta = numpy.dot(self.theta, self.X)
    self.theta = numpy.linalg.pinv(self.theta)
    self.theta = numpy.dot(self.theta,\n
                           numpy.transpose(self.X))
    self.theta = numpy.dot(self.theta, self.Y)

def value(self, x):
    res = 0
    for i in xrange(0, self.order+1):
        res += self.theta[i] * x**i
    return res

def pseudop(self, x, y):
    return 1.0/(1.0+(self.value(x)-float(y))**2)

```

Данный класс использует библиотеку NumPy для быстрых вычислений, предоставляющую интерфейс к ряду математических библиотек, конкретный набор которых зависит от платформы, под GNU/Linux используется GNU Scientific

Library [44]. Вычисления, даже будучи по своей природе векторными, в случае применения GSL, осуществляются в один поток, однако для данной задачи этого оказалось достаточно.

Конструктор класса `__init__` в качестве необязательного аргумента принимает максимальную степень требуемого интерполяционного многочлена, после чего сохраняет её значение в поле `order` и создаёт массив `theta` для его коэффициентов.

Два метода должны быть вызваны в обязательном порядке:

- 1 `create_vars` — этот метод в качестве аргумента принимает список `data`, состоящий из пар $\langle Dist, RSSI \rangle$ и содержащий все ранее измеренные значения RSSI исследуемой базовой станции в известных точках `Dist`:

$$data = \begin{pmatrix} Dist_0 & RSSI_0 \\ Dist_1 & RSSI_1 \\ \vdots & \vdots \\ Dist_{len(data)-1} & RSSI_{len(data)-1} \end{pmatrix} \quad (5.2)$$

Результатом работы данного метода является запись в поля класса двух массивов: массива `self.X`, состоящего из векторов $\langle 1, Dist_i, Dist_i^2, \dots, Dist_i^{order} \rangle$, и массива `self.Y`, состоящего из значений RSSI, расположенных в том же порядке, что соответствующие им значения $Dist_i$ в массиве `self.X`:

$$self.X = \begin{pmatrix} 1 & Dist_0 & Dist_0^2 & \dots & Dist_0^{order} \\ 1 & Dist_1 & Dist_1^2 & \dots & Dist_1^{order} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & Dist_{len(data)-1} & Dist_{len(data)-1}^2 & \dots & Dist_{len(data)-1}^{order} \end{pmatrix} \quad (5.3)$$

$$self.Y = \begin{pmatrix} RSSI_0 \\ RSSI_1 \\ \vdots \\ RSSI_{len(data)-1} \end{pmatrix} \quad (5.4)$$

- 2 `solve_theta` — результатом вызова этого метода является нахождение коэффициентов интерполяционного многочлена по методу нормальных уравнений и запись их в массив `self.theta`:

$$self.theta = (self.X^\top \cdot self.X)^+ \cdot self.X^\top \cdot self.Y \quad (5.5)$$

Отметим, что в ходе решения используется не обычная инверсия, а нахождение псевдообратной матрицы, что позволяет сохранить работоспособность системы даже в случае, когда система неразрешима в рамках обычного метода нормальных уравнений.

После того, как были вызваны эти методы, становится доступным вызов метода **value**, возвращающего значение интерполяционного многочлена в переданной точке, и метода **pseudop**, для заданной точки и заданного уровня RSSI возвращающего значение псевдоплотности вероятности.

5.3.2.2 Результирующая псевдоплотность и позиционирование

Класс **DataManager** отвечает за хранение коллекции экземпляров класса **Interpol**, соответствующих различным базовым станциям, и использование их для позиционирования устройств. В прототипе системы приняты следующие соглашения:

- 1 Интерполяционные многочлены строятся при инициализации сервера в режиме позиционирования, сразу для всех имеющихся в базе данных CellID;
- 2 Одновременно сервер работает только в одном из режимов, а потому во время работы в режиме позиционирования, состояние базы данных изменяться не может, и перестраивать многочлены гарантированно не потребуется;
- 3 Весь процесс выборки данных из базы, соответственно, происходит только во время инициализации, после чего в течение всего времени работы сервера интерполяционные многочлены хранятся в оперативной памяти.

С учётом этих соглашений, рассмотрим работу класса **DataManager**. При вызове конструктора класса создаётся подключение к базе данных через класс **DBConn**, после чего в обязательном порядке должны быть вызваны следующие методы:

- 1 **fetch_cells** — этот метод передаёт классу **DBConn** запрос на получение списка всех CellID, имеющихся в базе данных, после чего записывает результат в своё поле `self.cells`;
- 2 **build_interpols** — для каждого из полученных CellID, этот метод передаёт классу **DBConn** запрос на получение всех результатов измерений,

связанных с данной базовой станцией, после чего создаёт объект класса Interpol, у которого вызывает метод `create_vars` со списком измерений в качестве аргумента, после чего вызывает метод `solve_theta`. Полученный объект, готовый к использованию для вычисления псевдоплотности, помещается в ассоциативный массив `self.cellfuncs` с CellID в качестве индекса.

После вызова этих методов, становятся доступными следующие:

- **total_pseudop** — для заданного списка $\langle CID, RSSI \rangle$ и заданной точки на маршруте, вычисляет значения псевдоплотности, после чего перемножает их между собой и возвращает результат.
- **position** — для заданного списка $\langle CID, RSSI \rangle$ перебирает все возможные значения x и возвращает то, для которого значение общей псевдоплотности было максимальным. Для работы прототипа системы, даже такого простого метода поиска максимума оказалось достаточно, хотя, в случае реализации для промышленного применения, его стоит заменить на более эффективный.

5.3.2.3 Главный модуль

Алгоритм работы главного модуля в режиме позиционирования схож с алгоритмом в режиме сбора данных (см. подпункт 5.3.1.5), а отличается тем, что в начале работы, среди прочих, инициализируется класс DataManager, а бесконечный цикл состоит из следующих этапов:

- 1 Приём данных от мобильного устройства;
- 2 Валидация принятых данных;
- 3 Позиционирование по данным GSM с помощью класса DataManager;
- 4 Перевод координат полученной точки из одного числа — расстояния от начала маршрута — в пару широта/долгота;
- 5 Сравнение полученных данных с данными GPS, также принятыми от мобильного устройства;
- 6 Отправка всех вычисленных данных в формате JSON в очередь класса FrontEnd.

Клиенту от сервера передаётся сообщение в формате JSON следующего вида:

```
{  
    "RouteGPS": {  
        "dstd": 0.0033751292307418156,  
        "dstm": 375.73660822935375,  
        "fromtrue": 13.135102486978056,  
        "lng": 55.755307975824707,  
        "ltd": 37.648428777250331  
    },  
    "RouteGSM": {  
        "dstm": 336.50000000000193,  
        "lng": 37.648256950398967,  
        "ltd": 55.755000247022828  
    },  
    "TrueGPS": {  
        "acc": 8.0,  
        "lng": 55.755250453948975,  
        "ltd": 37.648531794548035  
    }  
}
```

Описание значения полей даны в таблице 5.3.

5.4 База данных

Необходимость хранить данные об отношении между точками на маршруте и результатами измерений уровней сигнала в них, по определению, говорит о необходимости использования реляционной модели.

В разработанном прототипе системы используется простая база данных, состоящая из двух таблиц, хранящих результаты измерений RSSI в различных точках маршрута для различных базовых станций. Никакие другие данные длительного хранения не требуют.

Имеющиеся таблицы объявлены следующим образом:

```
CREATE TABLE IF NOT EXISTS raw_rssi_data (  
    Longitude DOUBLE NOT NULL,
```

Таблица 5.3: Значения полей пакета в формате JSON, передаваемого от сервера в режиме позиционирования клиентам.

Поле	Тип	Значение
RouteGPS	Объект	Набор данных о проекции истинного положения устройства на маршрут
dsts	Число	Расстояние от начала маршрута до проекции положения в градусах
dstm	Число	Расстояние от начала маршрута до проекции положения в метрах
fromtruem	Число	Расстояние от истинного положения до его проекции на маршрут в метрах
lng	Число	Долгота в градусах
ltd	Число	Широта в градусах
RouteGSM	Объект	Набор данных о результатах позиционирования по сигналу GSM
TrueGPS	Объект	Набор данных об истинном положении мобильного устройства по данным спутниковой навигации
acc	Число	Погрешность GPS в метрах

Таблица 5.4: Значения столбцов в таблицах базы данных сервера

Столбец	Значение
Longitude	Долгота в градусах
Latitude	Широта в градусах
CellID	Уникальный идентификатор базовой станции, целое число
ObservedRSSI	Измеренное значение уровня сигнала от данной базовой станции в данной точке вasu
Observations	Количество раз, которое уровень сигнала от данной базовой станции принимал в данной точке данное значение
Distance	Расстояние от начала маршрута в градусах
Route	Номер маршрута, зарезервировано для дальнейшего развития

```

Latitude DOUBLE NOT NULL,
CellID INT NOT NULL,
ObservedRSSI INT NOT NULL,
Observations INT NOT NULL,
PRIMARY KEY (Longitude , Latitude , CellID , ObservedRSSI) ,
INDEX (CellID )
) ENGINE=InnoDB ;

```

```

CREATE TABLE IF NOT EXISTS route_rssi_data (
    Route INT NOT NULL,
    Distance DOUBLE NOT NULL,
    CellID INT NOT NULL,
    ObservedRSSI INT NOT NULL,
    Observations INT NOT NULL,
    PRIMARY KEY (Route , Distance , CellID , ObservedRSSI) ,
    INDEX (CellID )
) ENGINE=InnoDB ;

```

Значения их столбцов описаны в таблице 5.4. Видно, что имеющие таблицы

дублируют друг друга. Такая архитектура была выбрана в связи с тем, что для работы сервера требуется только данные из таблицы `route_rssi_data`, ассоциация же результатов измерений с их истинным положением, а не его проекцией на маршрут, требуется только для целей сбора и анализа статистики. Таким образом, таблицу `raw_rssi_data` следует считать хранилищем отладочной информации, а реально используемой базой данных — только `route_rssi_data`.

5.4.1 Добавление информации в БД

Хранение для каждой точки и каждой базовой станции не среднего результата всех замеров, а количества наблюдений для каждого из наблюдавшегося уровня по отдельности, по сути, представляет собой хранение распределения. Это позволяет однозначно восстановить все когда-либо сделанные замеры (что используется для увеличения множества, на котором строятся интерполяционные многочлены), при этом не перегружая базу уникальными идентификаторами для каждого наблюдения.

С другой стороны, это усложняет добавление информации о новом наблюдении в базу. Оно разделяется на два случая:

- 1 Если для измеренной тройки $\langle Distance, CellID, ObservedRSSI \rangle$ в таблице уже имеется строка с точно такими же значениями, требуется увеличить на 1 значение ячейки `Observations` этой строки;
- 2 Если же таких значений нет, подобную строку надо добавить в таблицу, установив значение ячейки `Observations` в 1.

Поскольку MySQL не поддерживает триггеры типа «INSTEAD OF INSERT» [45], осуществляет эту операцию сервер по следующему алгоритму:

- 1 Начало транзакции;
- 2 Запрос вида: «`SELECT Longitude, Latitude, CellID, ObservedRSSI FROM raw_rssi_data WHERE Longitude = %4.15f AND Latitude = %4.15f AND CellID = %d AND ObservedRSSI = %d`»;
- 3
 - Если результат запроса пустой — запрос вида: «`INSERT INTO raw_rssi_data(Longitude, Latitude, CellID, ObservedRSSI, Observations) VALUES (%4.15f, %4.15f, %d, %d, %d)`»;

- Иначе — запрос вида: «UPDATE raw_rssi_data SET Observations = Observations + 1 WHERE Longitude = %4.15f AND Latitude = %4.15f AND CellID = %d AND ObservedRSSI = %d»;

4 Конец транзакции.

5.5 Клиентская часть

Несмотря на то, что основная информация о работе сервера собиралась с помощью текстовых журналов, для контроля его работы в реальном времени был написан графический клиент на языке Python с использованием библиотеки Pygame.

Данное приложение отображает на экране вид сверху на местность, где осуществлялось тестирование приложения (снимок взят из приложения Google Earth [46] на условиях Fair use [47]), поверх которого рисует жёлтым цветом линию маршрута, обозначив его начало кругом того же цвета (см. рис. 5.9).

Приложение, будучи запущенным, открывает соединение с сервером по протоколу TCP и начинает ждать передачи данных. При приходе сообщения в формате JSON, клиент определяет режим работы сервера по содержимому принятого сообщения (см. подпункты 5.3.1.4 и 5.3.2.3), после чего сам переключается в соответствующий режим (менять режим возможно с каждым новым пакетом, если это потребуется).

5.5.1 Режим сбора данных

В режиме сбора данных (рис. 5.9а) клиент отображает на карте зелёной точкой истинное положение мобильного устройства по данным GPS, а голубой — её проекцию на маршрут. На экране отображается параметр Dist, означающий расстояние в метрах от начала маршрута до проекции положения мобильного устройства на него, и Err — расстояние от истинного положения до его проекции на маршрут в метрах.

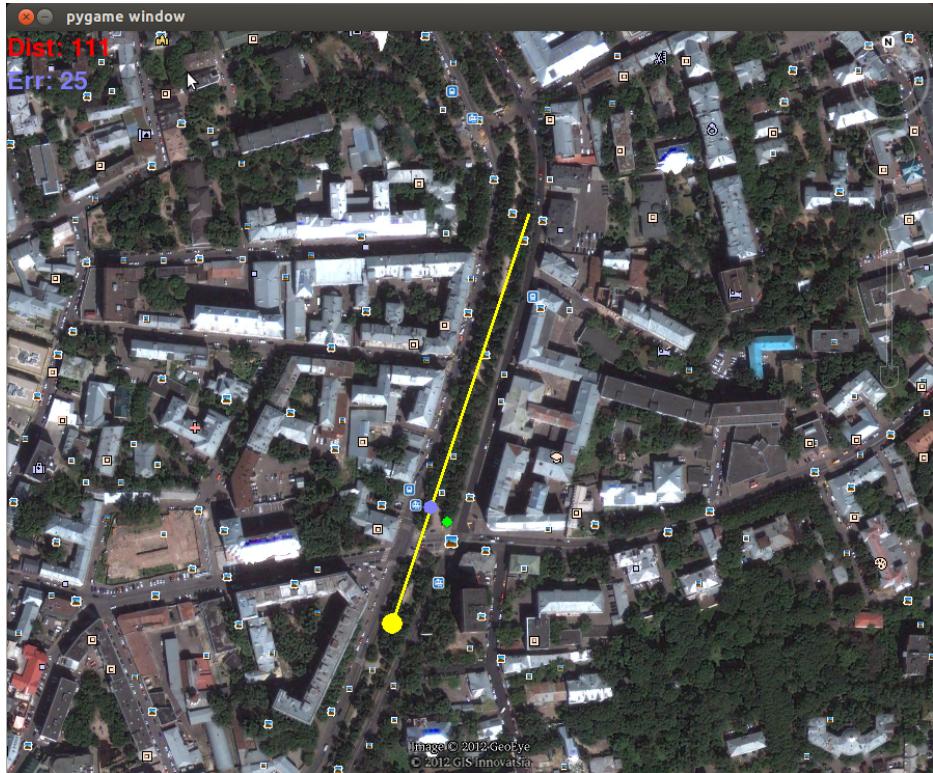
5.5.2 Режим позиционирования

В режиме позиционирования (рис. 5.9б) клиент отображает на карте:

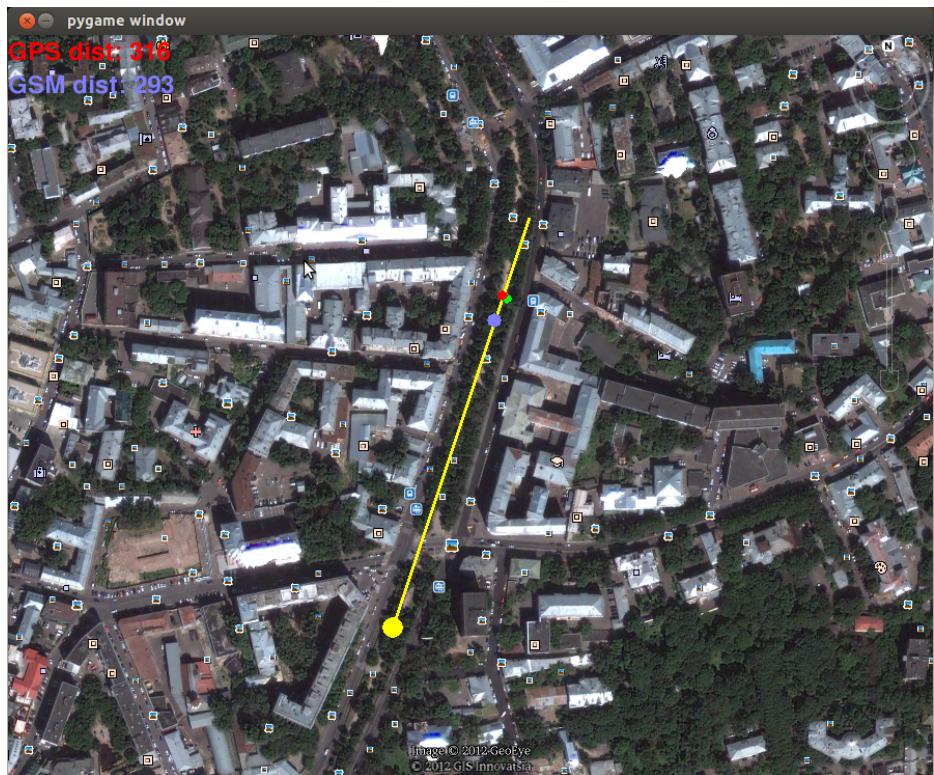
- **Зелёная точка** — истинное положение устройства по данным GPS;
- **Красная точка** — проекция истинного положения на маршрут;

- **Голубая точка** — результат позиционирования по данным GSM.

Параметры GPS dist и GSM dist означают расстояние от начала маршрута до положения устройства по данным GPS и GSM соответственно.



(a) Сбор данных



(b) Позиционирование

Рис. 5.9: Снимки работы графического клиента в двух возможных режимах. Изображение местности взято из приложения Google Earth [46] на условиях Fair use [47].

6 Экспериментальная проверка

6.1 Условия эксперимента

- Проверка работы созданного прототипа осуществлялась в мае 2012 года на прямом тестовом участке длиной около 400 метров, проходящем по Покровскому и Яузскому бульварам в Москве (см. рис. 5.9);
- В качестве мобильного устройства использовался HTC Hero под управлением ОС Android 2.2 (см. рис. 5.5);
- Сервер был запущен на VPS vkphoto.auditory.ru под управлением дистрибутива Gentoo ОС GNU/Linux версии 2.6.28-vs2.3.0.36.4-gentoo-ws0 с интерпретатором Python версии 2.6.4;
- Передача данных от мобильного устройства серверу по протоколу UPD осуществлялась через порт 31415;
- Входящие соединения TCP для передачи данных клиентам сервер принимал через порт 31416.

6.2 Сбор данных

Данные собирались в течение часа, по возможности равномерным образом по всей длине маршрута. В ходе эксперимента выяснился факт сравнительно редкого обновления данных GPS под Android, поэтому во время перемещения по маршруту передача данных выключалась до тех пор, пока показания широты и долготы не поменяются, затем включалась заново. Дополнительно осуществлялся контроль правильности обработки данных о местоположении путём визуального сравнения показаний клиентского приложения (см. рис. 5.9а), запущенного на портативном компьютере, подключённым к сети Интернет через общий канал с мобильным устройством, с фактической обстановкой вокруг.

Таблица 6.1: Координаты крайних точек тестового прямого маршрута.

Точка	Широта, градусов	Долгота, градусов
Начало	55.75236111111114	37.64678333333333
Конец	55.755430555555556	37.64849722222222

Таблица 6.2: Статистика по собранным данным во время экспериментальной проверки системы.

Всего замеров уровней сигнала	22675
Замеров с различающимися значениями $< CID, Dist, RSSI >$	10020
Всего различных базовых станций	22
Время сбора данных	1 час

Координаты тестового маршрута отражены в таблице 6.1. Общая статистика собранных данных отражена в таблице 6.2.

6.3 Позиционирование

Рассмотрим трассировку процесса позиционирования, осуществлённого после сбора данных с произвольно выбранным сообщением от мобильного устройства.

6.3.1 Входные данные

Было выбрано следующее сообщение:

```
{
    "GSM": {
        "cellcount": 6,
        "cells": [
            {
                "CID": 41526,
                "Psc": -1,
                "RSSI": 21,
                "type": "EDGE"
            }
        ]
    }
}
```

```

        } ,
        {
          "CID":32778,
          "Psc":-1,
          "RSSI":28,
          "type": "EDGE"
        } ,
        {
          "CID":32777,
          "Psc":-1,
          "RSSI":20,
          "type": "EDGE"
        } ,
        {
          "CID":12056,
          "Psc":-1,
          "RSSI":18,
          "type": "EDGE"
        } ,
        {
          "CID":14296,
          "Psc":-1,
          "RSSI":20,
          "type": "EDGE"
        } ,
        {
          "CID":32779,
          "Psc":-1,
          "RSSI":15,
          "type": "EDGE"
        }
      ]
    },
    "GPS": {
      "lng":37.647523283958435,
      "ltd":55.7536518573761,

```

```
    "acc": 16.0  
}  
}
```

По данным класса TransRoute, расстояние от начала маршрута до проекции этой точки составляет 165,5 метров.

6.3.2 Выборка собранных данных

На этом этапе для всех имеющихся в массиве cells базовых станций осуществляется выборка ранее измеренных значений из базы данных. Графики этих значений, для каждой станции по отдельности, изображены на рис. 6.1.

6.3.3 Интерполяция

На данном этапе для выбранных множеств результатов предварительных замеров сигналов базовых станций строятся интерполяционные многочлены. На рис. 6.2 приведены такие многочлены, степень каждого из которых определялась для каждой выборки индивидуально и равна одной десятой её длины. В пункте 6.5.1 вопрос выбора степени для интерполяционных многочленов рассмотрен более подробно.

6.3.4 Псевдоплотности вероятности

На данной стадии осуществляется сравнение статистических данных из базы, представленных в виде их интерполяционных многочленов, и фактически принятых RSSI. На рис. 6.3 можно видеть, как приближение значения многочлена к фактически принятому ведёт к повышению значения псевдоплотности для данной точки.

6.3.5 Результирующая псевдоплотность

Последний оставшийся шаг — это перемножение независимо посчитанных псевдоплотностей для разных базовых станций между собой и поиск аргумента максимума полученной функции. На рис. 6.4 изображена эта функция. Аргумента её максимума — 143 метра, реальное положение устройства по данным GPS — 165,5 метров. Итоговая ошибка составила 22,5 метров.

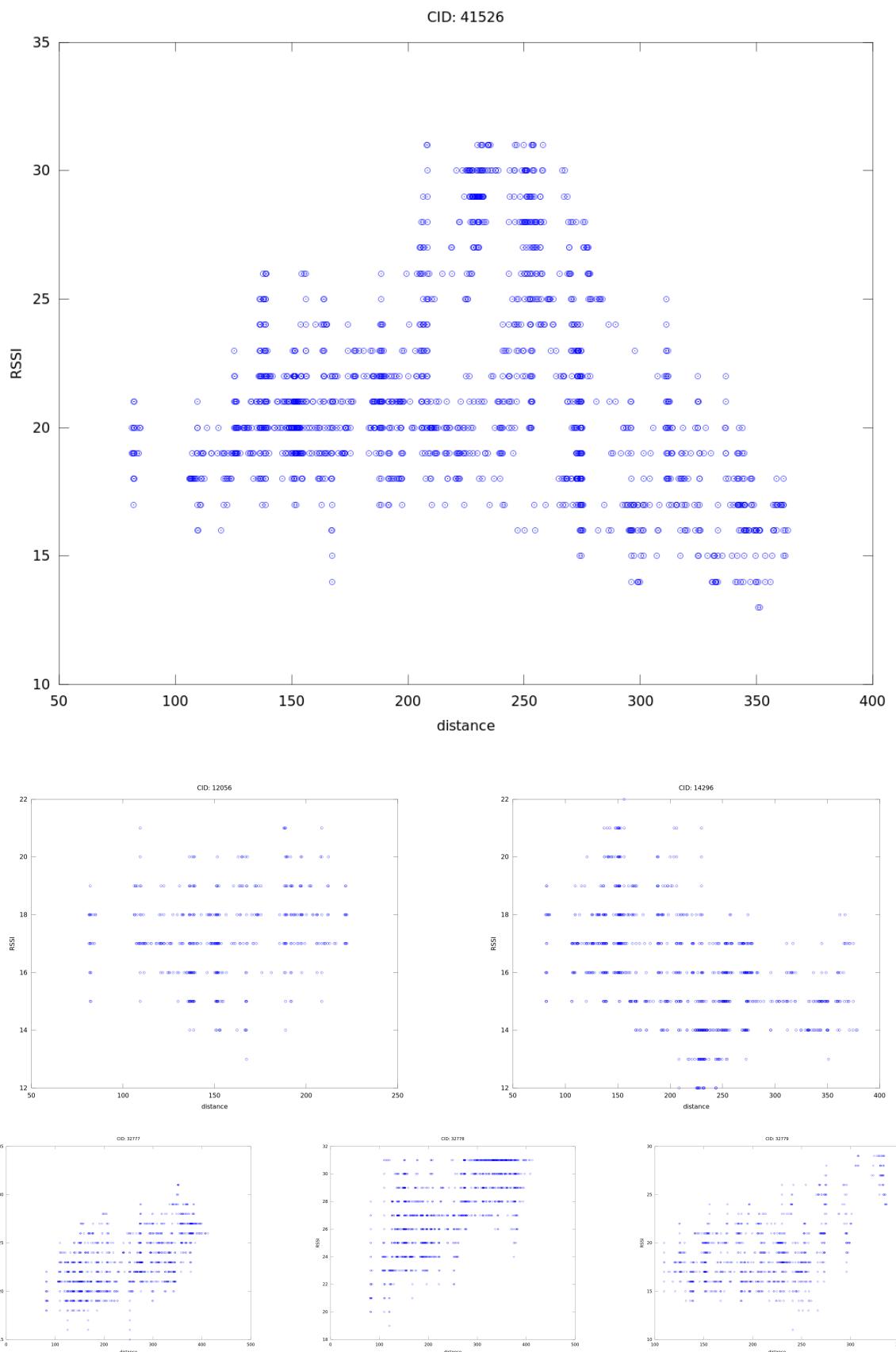


Рис. 6.1: Выборка данных для шести базовых станций, сигнал которых был принят позиционируемым устройством. Вертикальная ось — asu, горизонтальная — метры от начала маршрута.

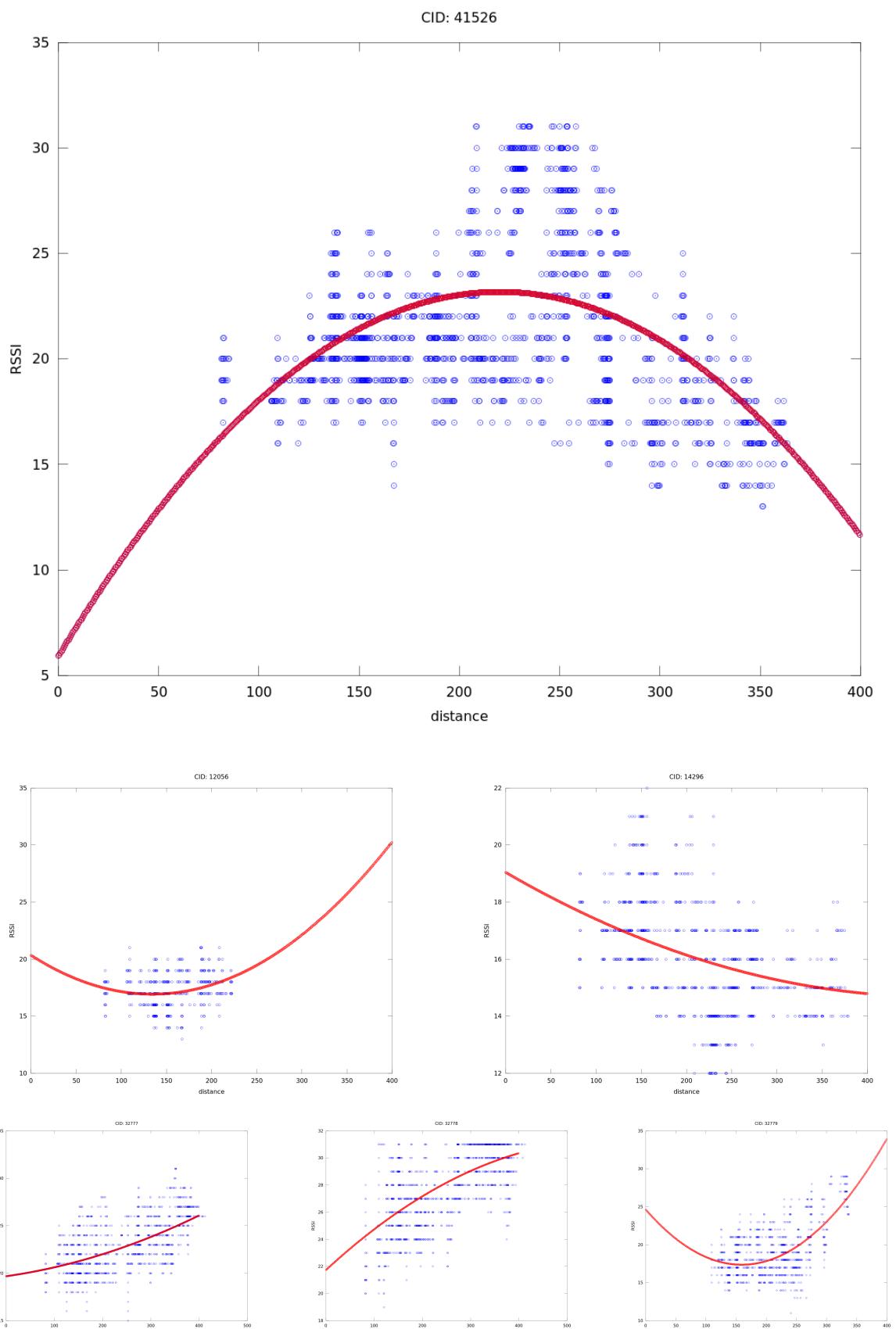


Рис. 6.2: Выборка данных о рассматриваемых станциях (точки) и интерполяционные многочлены для них (сплошные линии), полученные для степени, равной размерам выборок, делённым на 10. Вертикальная ось — asu, горизонтальная — метры от начала маршрута.

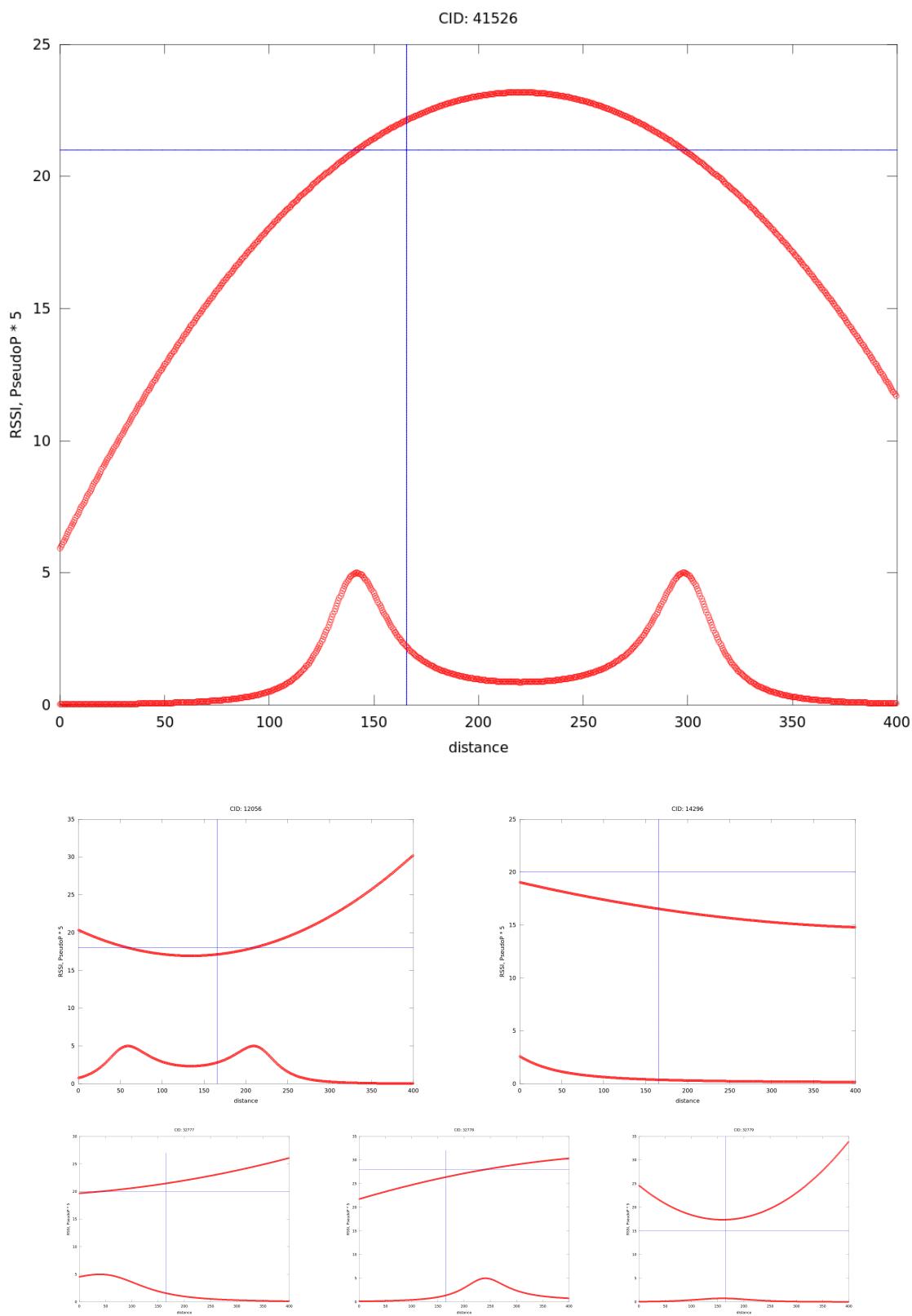


Рис. 6.3: Интерполяционные многочлены (толстая кривая сверху) и псевдоплотности вероятности (толстая кривая снизу) для принятых сигналов. Тонкая вертикальная прямая — координата истинного положения, тонкая горизонтальная — реально принятый RSSI для данной базовой станции. Вертикальная ось — asu и единицы псевдоплотности, умноженные на 5, горизонтальная — метры от начала маршрута.

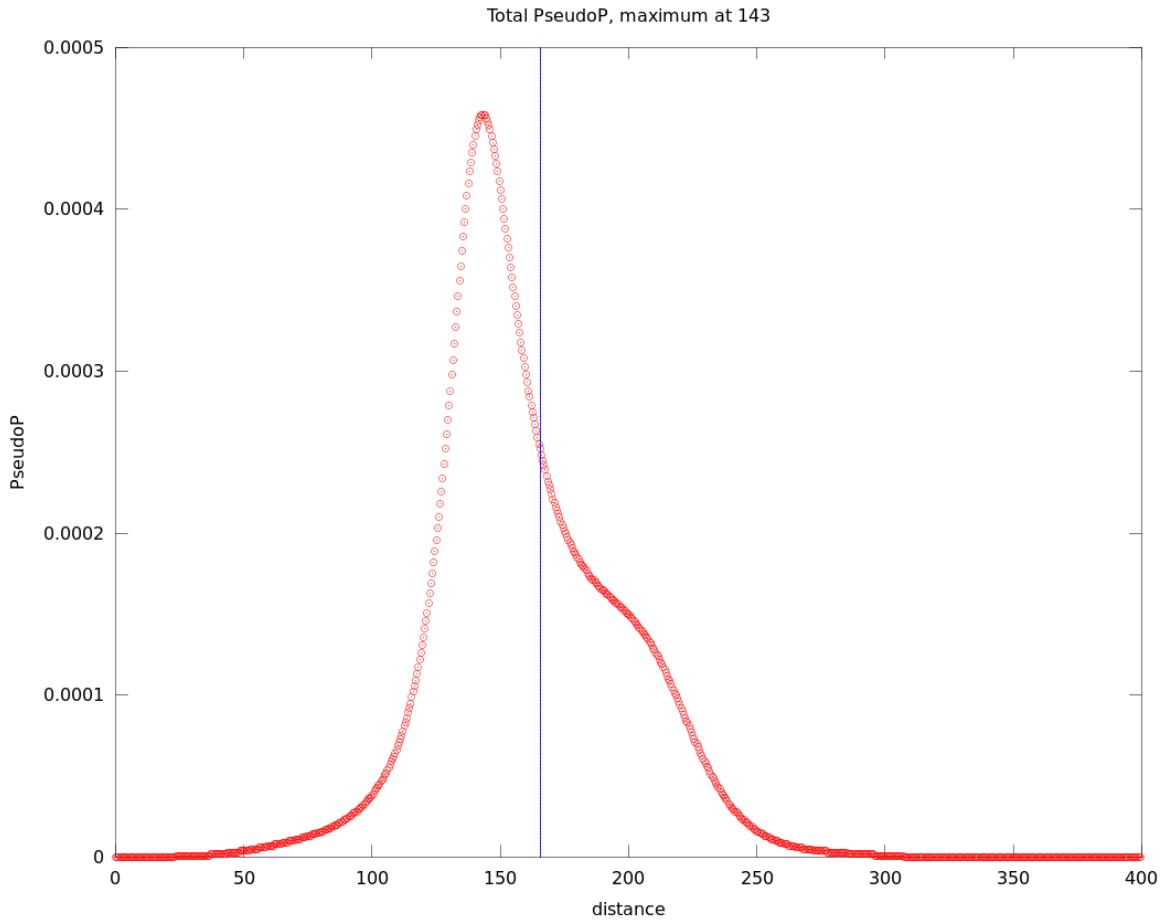


Рис. 6.4: Значения псевдоплотности в дискретно расположенных точках, обрабатывавшихся при поиске максимума. Вертикальная линия — координата истинного положения устройства. Вертикальная ось — единицы псевдоплотности, горизонтальная — метры. Аргумент максимального положения — 143 метра.

6.4 Погрешность

Перейдя от рассмотрения однократного процесса поиска местонахождения устройства к статистике этого процесса, рассмотрим, какое значение ошибки является типичным для реализованного алгоритма в условиях поставленного эксперимента.

Было поставлено 176 экспериментов, по возможности распределённых равномерно по всей длине маршрута. Для каждого из них отмечалось истинное значение расстояния от начала маршрута и результат его вычисления на основе данных GSM. После этого было вычислено расстояние между истинным и рассчитанным положением. Гистограмма полученной выборки изображена на рис. 6.5. Математическое ожидание ошибки составило 49 метров.

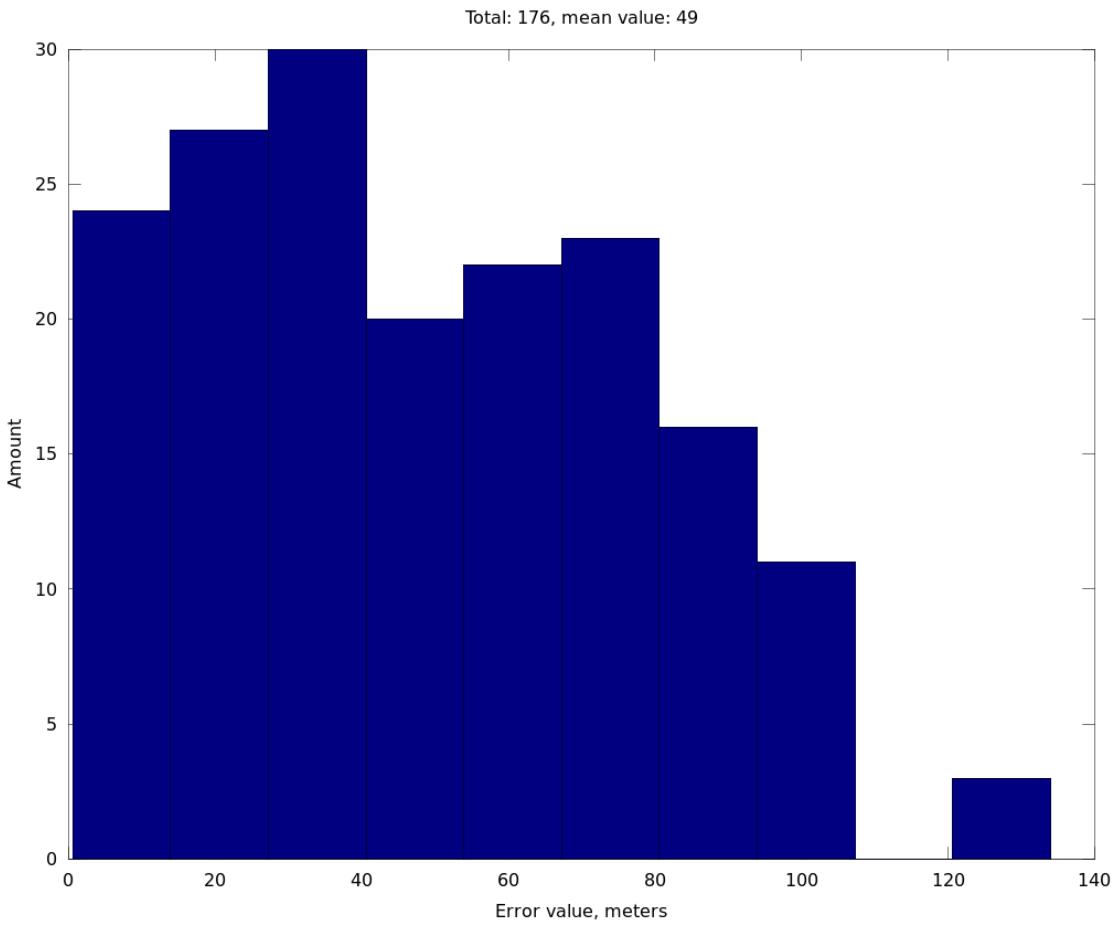


Рис. 6.5: Гистограмма величин ошибок позиционирования. Горизонтальная ось — величина ошибки в метрах, вертикальная — количество появлений такой ошибки в экспериментах. Всего 176 примеров, математическое ожидание ошибки — 49 метров.

6.5 Анализ экспериментальных данных

Точность позиционирования, достигнутая в ходе экспериментов, превышает современные системы, основанные на триангуляционном методе, но ещё не достигает точности спутниковых систем, что было бы идеальным результатом. Рассмотрим, какие факторы могут стоять на пути к повышению точности.

6.5.1 Степени интерполяционных многочленов

Самый заметный из факторов видно на рис. 6.3 — очень простой вид полученных интерполяционных многочленов. Все они либо имеют лишь один видимый экстремум, либо позволяют предположить его наличие за пределами графиков (поскольку конечный многочлен с положительными степенями асимптотически

сходиться к какому-либо конечному числу на может). Изначальная гипотеза предполагала наличие более сложного вида этой зависимости, который бы привёл к более выраженным экстремумам функции псевдоплотности вероятности и позволил достигнуть большей точности.

Проанализируем, с чем это может быть связано. В ходе процесса построения многочленов, описанного в пункте 6.3.3, в качестве максимальной степени для каждого многочлена выбиралось значение, разное одной десятой размера выборки. Для разных базовых станций размеры выборок различаются, начинаясь от десятков замеров и заканчиваясь тысячами. Таким образом, и степени многочленов получаются существенным образом разные.

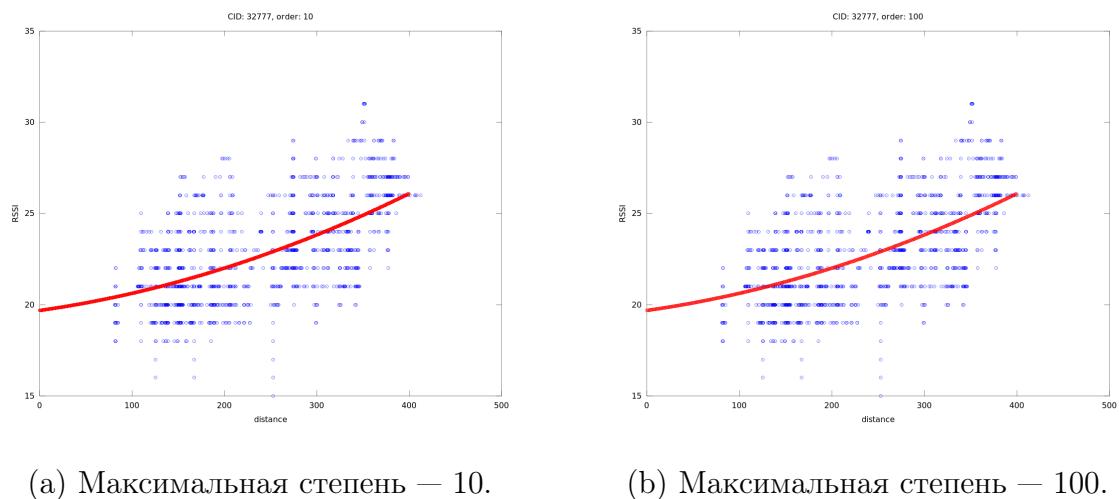


Рис. 6.6: Вид интерполяционных многочленов для выборки базовой станции 32777 при разной степени. Вертикальная ось — asu, горизонтальная — метры от начала маршрута.

Возьмём одну конкретную базовую станцию, пусть это будет 32777, для которой в базе имеется 2953 замеров, и построим для неё интерполяционные многочлены с разными степенями. На рис. 6.6 показаны интерполяционные многочлены десятой и сотой степени — разницы между ними нет. Таким образом получается, что полученные в пункте 6.3.3 многочлены действительно являются лучшими полиномиальными приближениями имеющихся массивов данных (причём стоит отметить, что на отрезке 400 метров многочлен сотой степени может с большой точностью приблизить практически любую зависимость, имеющую физический смысл).

6.5.2 Возможные дальнейшие эксперименты

Итак, зависимость уровня от сигнала от расстояния до станции, выявленная в ходе экспериментов, имеет вид, отличающийся от теоретического предсказания, но имеет простой вид, а не обладает множеством экстремумов, порождённых отражениями сигнала, поглощением городской застройкой и прочими систематическими факторами. В связи с этим, важным вопросом, который должен быть выяснен в ходе дальнейших экспериментов, является такой: действительно ли погрешности носят преимущественно случайный характер, и в ходе экспериментов была выявлена единственная систематически наблюдаемая закономерность?

Одним из таких экспериментов видится следующий: сравнение статистических данных об уровнях сигнала в ряде точек с учётом и без учёта соседних. При многократных изменениях на первый план должны выйти именно постоянные факторы, а не случайные. Таким образом, если во многих точках, распределённых по маршруту, средние результаты многократных измерений будут сходиться к интерполяционному многочлену, даже несмотря на то, что измерения усредняются только по одной точке, в многочлен учитывает все, это будет сильным аргументом в пользу того, что выявленные зависимости являются единственными. И, следовательно, полученная точность близка к теоретическому пределу точности позиционирования, которую можно получить в сотовых сетях.

Если же такой сходимости наблюдаться не будет, это будет означать, что статистические методы как таковые могут работать точнее, но необходимо работать над совершенствованием методов, чтобы выявить дополнительные скрытые закономерности.

Заключение

В данной работе был предложен и опробован принципиально новый метод позиционирования устройств на основе сигналов базовых станций сетей GSM, рассчитанный на использование в системах наблюдения за наземным общественным транспортом. Качество работы метода превысило то, которое дают существующие, основанные на триангуляции в сетях GSM, но ещё не достигло точности, сравнимой со спутниковыми системами, а потому требуются и были предложены дополнительные экспериментальные и теоретические изыскания, направленные на дальнейшее совершенствование метода.

В обзорно-аналитической части были рассмотрены существующие методы позиционирования устройств: спутниковая навигация и триангуляция в сетях GSM. Были подробно исследованы особенности метода триангуляции и его практического применения, был сделан вывод о роли городской застройки как, предположительно, главного фактора, ограничивающего точность подобного метода. Было отмечено, что особенности общественного транспорта, как движущегося всегда по ограниченному и заранее известному набору маршрутов, позволяют отказаться от метода триангуляции, с присущими ему недостатками, в пользу статистических методов. Сформулирована гипотеза о том, что такой переход позволит значительно повысить точность позиционирования, возможно, вплоть до значений, обеспечиваемых спутниковой навигацией.

После этого был осуществлён анализ существующих методов поиска наибольшего правдоподобия аргумента при сравнении векторов случайных переменных, таких как расстояние Махalanобиса и Байесовский классификатор, в результате которого сделан вывод о том, что ни один из них в полной мере не удовлетворяет требованиям, предъявляемым к алгоритму позиционирования транспорта, хотя каждый из них обладает своими достоинствами, в результате чего сделан вывод о необходимости создания нового математического метода.

Создание требуемого алгоритма является одним из ключевых достижений данной работы. Была предложена функция псевдоплотности вероятности, поз-

воляющая создать алгоритм, напоминающий Байесовский классификатор и обладающий всеми его достоинствами, такими как большая устойчивость к выбросам и шумам в значениях анализируемых случайных величин, но работающий с бесконечными, причём континуальными и упорядоченными множествами как возможных значений случайных величин, так и классов. Из возможности работы над континуальными множествами следует то, что алгоритм, во-первых, для каждой возможной точки учитывает не только данные о ней самой, но и данные о соседних точках, во-вторых, пользуется не дискретным сравнением совпадения данного ответа с хранящимся в базе, а использует нечёткую шкалу похожести, а в-третьих, позволяет успешно обрабатывать ситуацию, когда даётся ответ, никогда ранее не дававшийся, но похожий на имеющиеся.

Этот алгоритм был положен в основу прототипа программно-аппаратного комплекса для позиционирования, состоящего из четырёх звеньев:

- 1 Мобильное устройство, собирающее данные об уровнях сигналов видимых базовых станций GSM в разных точках маршрута и передающее их на сервер; в качестве устройства использовался смартфон со специальной программой, написанной на языке Java и работающей под управлением ОС Android;
- 2 Сервер, принимающий и сохраняющий данные об уровнях сигнала, а затем использующий их в работе созданного алгоритма позиционирования; написан на языке Python с использованием библиотеки NumPy для векторных вычислений;
- 3 База данных, хранящая информацию об уровнях сигнала; использована СУБД MySQL;
- 4 Клиентское приложение, получающее информацию о процессе сбора данных или позиционирования с сервера в реальном времени и отображающее её на карте местности; написано на языке Python с использованием библиотеки pygame для отображения графики.

После завершения разработки системы была предложена методика проведения экспериментальной проверки гипотезы о возможности повышения точности позиционирования с использованием данной системы. Проверка осуществлялась на прямом тестовом участке на Покровском и Яузском бульварах в Москве.

В ходе предварительного сбора данных было осуществлено более 20 тысяч замеров уровней сигналов базовых станций в разных точках, при этом в разное время в зоне видимости были видны 22 различные базовые станции. Когда данные были собраны, было поставлено 176 экспериментов по позиционированию устройства, пользуясь собранными данными и разработанным алгоритмом, контроль при этом осуществлялся по данным GPS.

В результате, математическое ожидание ошибки позиционирования составило 49 метров, что лучше современных систем, основанных на триангуляционном методе, но ещё не достигает точности спутниковой навигации. На основании интерпретации экспериментальных данных, в том числе и трассировки процесса позиционирования на реальных данных, были выдвинуты гипотезы о факто-рах, которые могут мешать дальнейшему повышению точности, и предложены эксперименты для выяснения, возможно ли дальнейшее повышение точности с помощью статистических методов, или же был достигнут технологический предел точности позиционирования на основе данных об уровнях сигналов базовых станций.

Вывод из данной работы можно сформулировать следующим образом: созданный математический метод показал большую точность позиционирования, чем триангуляция, и гипотеза о возможности повышения точности, используя статистический подход, оправдалась. При этом, достигнуть той точности, которую даёт спутниковая навигация, пока не удалось, но были предложены пути для дальнейшего совершенствования метода.

Литература

1. Free Software Foundation Inc <<http://www.gnu.org/copyleft/fdl.html>>. Gnu free documentation license, November 2008.
2. Location-based service Википедия.
http://ru.wikipedia.org/wiki/location-based_service.
3. Vehicle tracking system Wikipedia.
http://en.wikipedia.org/wiki/vehicle_tracking_system.
4. Transport Canada: Intelligent Transportation Systems in 98 B-Line Rapid Bus Service: Advanced Technology at Work.
<http://www.tc.gc.ca/eng/programs/environment-utsp-intelligenttransportationsystems-945.htm>.
5. ХаброХабр: Spb Transport Online. <http://habrahabr.ru/post/143229/>.
6. ХаброХабр: Триумф геопозиционирования (ГЛОНАСС+GPS) общественного транспорта.
<http://habrahabr.ru/post/77259/>.
7. Маршруты маршрутки и автобусы в новых Картах для Android.
http://clubs.ya.ru/company/replies.xml?item_no=41137.
8. <http://miem.edu.ru/news/Призёры-научно-технической-конференции-студентов-аспирантов-и-молодых-специалистов-МИЭМ-2011.html>.
9. Wikipedia: GPS. http://en.wikipedia.org/wiki/global_positioning_system.
10. Portable Calibration of GSM Base Station Clocks.
http://www.spectracomcorp.com/desktopmodules/bring2mind/dmx/download.aspx?entryid=434&command=core_download&method=attachment&portalid=0&tabid=59.

11. NeighboringCellInfo | Android Developers.
<http://developer.android.com/reference/android/telephony/neighboringcellinfo.html>.
12. Cell site Wikipedia. http://en.wikipedia.org/wiki/cell_site.
13. Digital cellular telecommunications system (Phase 2+) Radio transmission and reception (GSM 05.05).
http://www.etsi.org/deliver/etsi_gts/05/0505/05.00.00_60/gsmts_0505v050000p.pdf. c.c. 10–11.
14. Wikipedia: dBm. <http://en.wikipedia.org/wiki/dbm>.
15. Мегафон лидирует в большой тройке по количеству базовых станций и числу. <http://decod.ru/news/a-74.html>.
16. Яндекс.Локатор Руководство разработчика: Данные для определения местоположения.
http://api.yandex.ru/locator/doc/dg/api/geolocation-api_data.xml.
17. Location source and accuracy Google Maps for Mobile Help.
<http://support.google.com/gmm/bin/answer.py?hl=en&answer=81873>.
18. Espen Talberg. Positioning with cell id. (Institue of Informatics Ostfold Universiry College, Halden, Norway)
<http://mobapp.hiof.no/wp-content/uploads/talberg06pwc.pdf> c.c. 16–19.
19. Пользовательское соглашение Яндекс.Локатор.
<http://api.yandex.ru/locator/eula.xml>.
20. Яндекс.Локатор Руководство разработчика: Определение местоположения.
<http://api.yandex.ru/locator/doc/dg/concepts/geolocation.xml>.
21. Wikipedia: Least Squares. http://en.wikipedia.org/wiki/least_squares.
22. Википедия: Московский трамвай.
http://ru.wikipedia.org/wiki/Московский_трамвай.
23. Wikipedia: Mahalanobis distance.
http://en.wikipedia.org/wiki/mahalanobis_distance.
24. Янгель Б.К. Байесовский подход и Акинатор – Спецкурс "Байесовские методы машинного обучения".

<http://www.machinelearning.ru/wiki/images/7/78/BayesML-2010-Yangel-Akinator.pdf>.

25. Location Manager Android Developers.

<http://developer.android.com/reference/android/location/locationmanager.html>.

26. Wikipedia: Dalvik.

[http://en.wikipedia.org/wiki/dalvik_\(software\)](http://en.wikipedia.org/wiki/dalvik_(software)).

27. Android Developers. <http://developer.android.com/index.html>.

28. Scientific computing tools for python. <http://numpy.scipy.org/>.

29. MySQLdb User's Guide. <http://mysql-python.sourceforge.net/mysqlDb.html>.

30. vim. <http://www.vim.org/>.

31. Pygame. <http://www.pygame.org/news.html>.

32. Максим Ковалев. Рекурсия. <http://wiki.auditory.ru/Рекурсия>.

33. LaTeX – A document preparation system. <http://www.latex-project.org/>.

34. Как оформить диплом в LaTeX?

<http://mydebianblog.blogspot.com/2008/11/latex.html>.

35. JabRef reference manager. <http://jabref.sourceforge.net/>.

36. Graphviz Graph Visualization Software. <http://www.graphviz.org/>.

37. GNU Octave. <http://www.gnu.org/software/octave/>.

38. GNU Make. <http://www.gnu.org/software/make/>.

39. Wikipedia: Occupational safety and health.

http://en.wikipedia.org/wiki/occupational_safety_and_health.

40. Wikipedia: Mobbing. <http://en.wikipedia.org/wiki/mobbing>.

41. Wikipedia: Burnout. [http://en.wikipedia.org/wiki/burnout_\(psychology\)](http://en.wikipedia.org/wiki/burnout_(psychology)).

42. How Society Rejected Men, Created 4chan, and Anonymous.

<http://www.inmalafide.com/blog/2011/03/09/how-society-rejected-men-and-created-4chan-and-anonymous/>.

43. ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ПЕРСОНАЛЬНЫМ ЭВМ И ОРГАНИЗАЦИИ РАБОТЫ (САНПИН 2.2.2/2.4.1340-03).
44. GNU Scientific Library.
<http://www.gnu.org/software/gsl/>.
45. MySQL 5.0 Reference Manual Trigger Syntax.
<http://dev.mysql.com/doc/refman/5.0/en/trigger-syntax.html>.
46. Google Earth.
<http://www.google.com/earth/index.html>.
47. Wikipedia: Fair use.
http://en.wikipedia.org/wiki/fair_use.

Список иллюстраций

1.1	Приложение от Яндекс, в реальном времени показывающее автобусы Казани. Изображение из блога Яндекса [7]	8
2.1	Две пересекающиеся сферы образуют окружность. Третья сфера, пересекая эту окружность, образует две точки. Изображения из фонда Wikimedia Commons.	11
2.2	Зависимость мощности излучения в ваттах от расстояния в метрах до базовой станции GSM DCS 1800 мощностью 40 Вт. Гипербола столь быстро сходится к асимптотам, что на рисунке сливаются с осями координат.	13
2.3	Зависимость мощности излучения в децибелах мощности от расстояния в метрах до базовой станции GSM DCS 1800 мощностью 40 Вт.	14
2.4	Зависимость расстояния между двумя точками, где asu принимает целочисленное значение, от расстояния этих точек до базовой станции, по осям отложены метры	16
2.5	Возможная ситуация при позиционировании транспорта. Изолинии показывают целевую функцию метода наименьших квадратов. Точка А — глобальный минимум, она будет считаться результатом позиционирования. В — проекция минимума на путь, по которому движется транспортное средство. С — точка на пути движения, в которой достигается наименьшее значение целевой функции.	21
3.1	Общая архитектура разрабатываемой системы	28
4.1	Частота поиска и упоминания в новостях слова mobbing по данным аналитической системы Google Trends.	33

5.1	Эскиз возможных входных данных для алгоритма позиционирования. Вертикальная ось — уровень сигнала, горизонтальная — координата. Штрихпунктирными линиями обозначены уровни фактически принятых сигналов от двух базовых станций (поскольку их координаты неизвестны, изображены на всём протяжении маршрута). Точки, соединённые сплошными отрезками — хранящиеся в базе данных замеры RSSI этих станций с известными координатами.	36
5.2	Эскиз возможной интерполяции в ходе алгоритма позиционирования. Вертикальная ось — уровень сигнала, горизонтальная — координата. Штрихпунктирными линиями обозначены уровни фактически принятых сигналов от двух базовых станций (поскольку их координаты неизвестны, изображены на всём протяжении маршрута), штриховыми — функции, интерполирующие имеющиеся в базе данных замеры.	37
5.3	Эскиз процесса расчёта псевдоплотности вероятности для двух базовых станций. Горизонтальная ось — расстояние, вертикальная — RSSI и значение псевдоплотности (не в масштабе). Штрихпунктирная линия — фактически измеренные значения RSSI, штриховая — интерполяционный многочлен, сплошная — значение псевдоплотности.	39
5.4	Эскиз произведения функций псевдоплотности для нескольких базовых станций. Горизонтальная ось — точки на маршруте, вертикальная — значения псевдоплотности. Результирующая псевдоплотность имеет всего один хорошо выраженный глобальный максимум между координатами 1 и 1,5.	41
5.5	Графический интерфейс написанного мобильного приложения, запущенного на телефоне HTC Hero под управлением Android 2.2. .	44
5.6	Взаимодействие между потоками сервера (эллипсы) и внешними интерфейсами (прямоугольники).	46
5.7	Потоки данных между классами сервера (эллипсы) и внешними интерфейсами (прямоугольники) в режиме сбора данных. . . .	47
5.8	Потоки данных между классами сервера (эллипсы) и внешними интерфейсами (прямоугольники) в режиме позиционирования. . .	51

5.9	Снимки работы графического клиента в двух возможных режимах. Изображение местности взято из приложения Google Earth [46] на условиях Fair use [47].	62
6.1	Выборка данных для шести базовых станций, сигнал которых был принят позиционируемым устройством. Вертикальная ось — asu, горизонтальная — метры от начала маршрута.	67
6.2	Выборка данных о рассматриваемых станциях (точки) и интерполяционные многочлены для них (сплошные линии), полученные для степени, равной размерам выборок, делённым на 10. Вертикальная ось — asu, горизонтальная — метры от начала маршрута.	68
6.3	Интерполяционные многочлены (толстая кривая сверху) и псевдодоплотности вероятности (толстая кривая снизу) для принятых сигналов. Тонкая вертикальная прямая — координата истинного положения, тонкая горизонтальная — реально принятый RSSI для данной базовой станции. Вертикальная ось — asu и единицы псевдодоплотности, умноженные на 5, горизонтальная — метры от начала маршрута.	69
6.4	Значения псевдодоплотности в дискретно расположенных точках, обрабатывавшихся при поиске максимума. Вертикальная линия — координата истинного положения устройства. Вертикальная ось — единицы псевдодоплотности, горизонтальная — метры. Аргумент максимального положения — 143 метра.	70
6.5	Гистограмма величин ошибок позиционирования. Горизонтальная ось — величина ошибки в метрах, вертикальная — количество появлений такой ошибки в экспериментах. Всего 176 примеров, математическое ожидание ошибки — 49 метров.	71
6.6	Вид интерполяционных многочленов для выборки базовой станции 32777 при разной степени. Вертикальная ось — asu, горизонтальная — метры от начала маршрута.	72

Список таблиц

2.1	Расстояния до точек изменения asu от базовой станции GSM DCS 1800 мощностью 40 Вт. Теоретическая оценка.	15
2.2	Сравнение свойств спутникового позиционирования и позициони- рования в сотовых сетях	20
5.1	Значения полей пакета в формате JSON, передаваемого от мо- бильного устройства на сервер.	43
5.2	Значения полей пакета в формате JSON, передаваемого от серве- ра в режиме сбора данных клиентам.	50
5.3	Значения полей пакета в формате JSON, передаваемого от серве- ра в режиме позиционирования клиентам.	57
5.4	Значения столбцов в таблицах базы данных сервера	58
6.1	Координаты крайних точек тестового прямого маршрута.	64
6.2	Статистика по собранным данным во время экспериментальной проверки системы.	64

Список публикаций

- 1 М.М.Ковалев. Позиционирование наземного транспорта с помощью сигнала GSM. Научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ. Тезисы докладов. – М.: МИЭМ, 2011. — 420. ISBN 978-5-94506-257-3. с.с. 29–30.
- 2 М.М.Ковалев. Прототип программно-аппаратного комплекса для сбора данных об уровнях сигнала GSM на местности. Научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ, посвящённая 50-летию МИЭМ. Тезисы докладов. – М.: МИЭМ, 2012. — 445 с. ISBN 978-5-94506-314-3. с.с. 185–187.