

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(государственный университет)»

ФАКУЛЬТЕТ РАДИОТЕХНИКИ И КИБЕРНЕТИКИ

КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

## **Способы классификации по типу движения и валидации геолокационных данных**

Выпускная квалификационная работа  
(бакалаврская работа)

Направление подготовки: 03.03.01 Прикладные математика и физика

Выполнил:

студент 211 группы \_\_\_\_\_ Саргин Даниил Владимирович

Научный руководитель: \_\_\_\_\_ Соболев Артемий Анатольевич

Москва 2016

## Аннотация

Системы навигационного позиционирования широко распространены: на данный момент практически каждый мобильный телефон имеет возможность приема сигнала GPS (Global Positioning System). Помимо этого, мобильные устройства снабжены другими навигационными сенсорами: гироскоп, акселерометр, педометр, которые являются частью ИНС (Инерциальная Навигационная Система). В то время как прием сигнала GPS расходует значительное количество энергии и недоступен в зданиях, ИНС остается включенной практически всегда. Использование данных ИНС может дать дополнительную информацию для валидации сигнала GPS. Особый случай использования ИНС: определение типа движения по её данным. Знание о типе движения может помочь в выставлении частоты приема сигнала GPS, снижая потребляемую мощность.

В данной работе рассматриваются различные классификации типа движения по навигационным данным. Также рассматривается возможность валидации сигнала GPS при различных обстоятельствах и использование данных ИНС для достижения оптимальной частоты обновлений данных GPS на устройстве.

# Оглавление

<b>1</b>	<b>Введение</b>	<b>3</b>
1.1	Предпосылки . . . . .	3
1.2	Аналогичные работы . . . . .	4
<b>2</b>	<b>Описание модели</b>	<b>6</b>
2.1	Получение и обработка данных . . . . .	6
2.2	Используемое оборудование . . . . .	8
<b>3</b>	<b>Теория</b>	<b>9</b>
3.1	Трансформация координат . . . . .	9
3.2	Методы изучения особенностей сигнала . . . . .	11
<b>4</b>	<b>Анализ движения человека по данным ИНС</b>	<b>13</b>
4.1	Сбор данных ИНС . . . . .	15
4.2	Поиск особенностей данных и их классификация . . . . .	16
4.3	Схема алгоритма классификации . . . . .	18
4.4	Результаты испытаний . . . . .	19
<b>5</b>	<b>Оптимизации в приеме сигнала GPS на основе типа движения</b>	<b>21</b>
5.1	Выставление частоты обновления сигнала GPS . . . . .	21
5.2	Фильтрация точек . . . . .	22
<b>6</b>	<b>Заключение</b>	<b>26</b>
6.1	Выводы . . . . .	26

6.2	Планы на дальнейшую работу . . . . .	27
-----	--------------------------------------	----

# Глава 1

## Введение

### 1.1 Предпосылки

Развивающийся рынок смартфонов и других типов носимых устройств делает возможность использования локации пользователя для различных целей все более и более привлекательным для приложений. Основным способом позиционирования остается широко известная ГПС (Глобальная Система Позиционирования, далее GPS). Однако, GPS до сих пор остается довольно затратным в плане энергоэффективности, к тому же максимальная точность может достигаться только при позиционировании на открытом пространстве. Для грубого позиционирования обычно используется триангуляция по ближайшим базовым станциям, но её точность заметно уступает GPS.

В настоящее время ведутся активные разработки для позиционирования в зонах, где сигнал GPS недоступен, методом навигационного счисления по данным ИНС (Инерциальная Навигационная Система) [1], так как акселерометр, гироскоп и педометр доступны в каждом смартфоне, и они работают всегда (для примера, кэшированные данные ИНС можно запросить в системе iOS в любой момент и сразу же начать считывание новых данных) [2]. Однако в стандартных библиотеках iOS и Android пока нет реализа-

ции методов навигационного счисления. Для позиционирования в зданиях Apple внедряет технологию iBeacon, а Google – Eddystone. Это Bluetooth маяки, которые сообщают устройству, что оно находится в регионе данного маяка. Но данные технологии требуют размещения специальных передатчиков, а также должны поддерживаться на мобильных устройствах [2, 3]. Поэтому в данной работе предлагается рассмотрение алгоритмов комбинирования данных GPS и ИНС для повышения точности позиционирования пользователя и повышения энергоэффективности использования GPS. Для достижения цели требуется выполнить следующие подзадачи:

- 1 Детектирование типа движения пользователя по данным ИНС.
- 2 Выставление частоты приема сигнала GPS на основе типа движения.
- 3 Вычисление средней длины шага на основе данных GPS и ИНС.
- 4 Использование полученных данных для навигационного счисления в случаях, когда GPS недоступно.

В работе рассматриваются первые два пункта, остальные вынесены для последующей работы.

## 1.2 Аналогичные работы

Существует множество работ, связанных с определением типа движения по данным ИНС и их применения для навигационного счисления, причем они рассматривают разные случаи расположения датчиков относительно пользователя: на поясе в [4], других частях тела (ноги, руки) в [5]. Общие методы детектирования движения основаны на рассмотрении циклического поведения вертикального ускорения, к которому применяется принцип скользящего окна с детектированием локальных экстремумов и прохождение сигнала через ноль [6–8]. Есть работы, в которых рассматриваются боковые случаи движения, и при обработке данных учитывается ускорение на всех трех осях движения [9]. В данной работе рассматриваются базовые

типы движения: ходьба, бег и отсутствие движения с закрепленным на поясе датчиком.

Определение ориентации в пространстве остается проблемной частью для навигационного счисления, так как магнетометры часто дают ошибочные показания и чувствительны к магнитным полям. Для решения данной проблемы предложены различные методы, в которых учитываются показания гироскопа вместе с GPS для улучшения определения направления движения [4, 6–8]. Однако в данной работе это не входит в рассмотрение.

В работе будут рассмотрены общие методы детектирования и классификации движения. В некоторых работах для классификации применялись алгоритмы скрытых Марковских цепей [10, 11], но в данной работе решено ограничиться на логическом классификаторе движения. Результаты этих работ показывают, что выигрыш от использования скрытых Марковских цепей заметен лишь в промежутки времени, когда происходит смена типа движения, но на практике смена типа движения происходит не настолько часто. Всё, что требуется в данном случае — обеспечить частоту обновления результатов классификации на достаточно высоком уровне, чтобы эффекты при смене движения не имели значительного влияния. К тому же, целью является создать простое, но эффективное решение, которое можно без особого труда интегрировать для использования в мобильных устройствах с разными операционными системами.

# Глава 2

## Описание модели

### 2.1 Получение и обработка данных

Общая структура модели представлена на рис. 2.1.

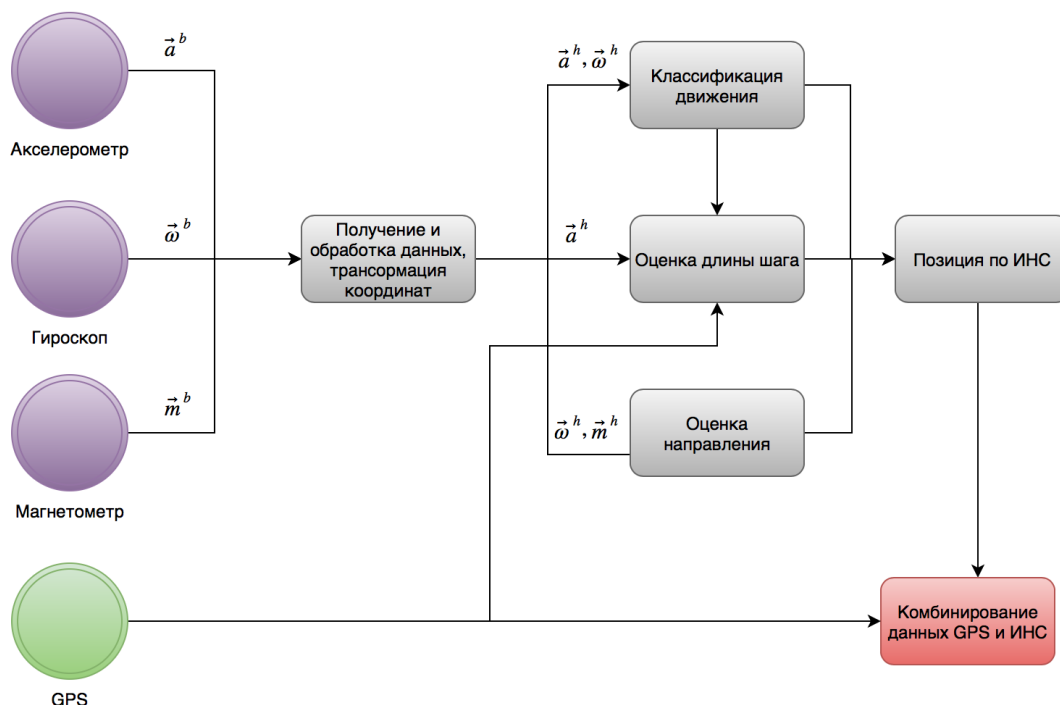


Рис. 2.1: Полная схема модели обработки данных.



Данные приходят от акселерометра, гироскопа, магнетометра и GPS в реальном времени, причем первые три сенсора имеют большую частоту обновления. На современных мобильных устройствах данный факт не вызывает проблем с энергоэффективностью, так как ИНС обычно работает на вспомогательном сопроцессоре.

GPS остается самым требовательным к энергии звеном системы, и его работу всегда требуется сокращать к минимуму. В iOS предусмотрено множество механизмов для этого:

- 1 отложенные обновления, когда приложение находится в бэкграунд-режиме: GPS-чип кэширует новые локации, не взаимодействуя с приложением, разгружая основной процессор устройства. Система пробуждает приложение только когда выполняются заданные условия: пользователь преодолел заданную дистанцию либо прошло заданное количество времени;
- 2 поддержка отслеживания посещений вместо постоянного обновления локаций;
- 3 поддержка отслеживания прибытия пользователя в определенные зоны;
- 4 поддержка запуска приложения, когда пользователь переместился на большое расстояние. То есть если пользователь обеспокоен о потреблении энергии приложением и принудительно останавливает его работу, то приложение все равно имеет доступ к каким-либо обновлениям, но до тех пор, пока доступ к обновлениям локаций не будет заблокирован через настройки телефона.

При поступлении данных та часть из них, которая относится к ИНС, проходит предварительную обработку: происходит трансформация в горизонтальную систему координат, затем — классификация типа движения и оценка ориентации в пространстве. Комбинация данных о типе движения в купе с данными GPS дает возможность измерить среднюю длину шага,

которая затем используется при навигационном счислении, которое может дать меньшую ошибку, чем GPS в случаях, когда сигнал слабый. Стрелками обозначена передача данных

В данной работе рассмотрена упрощенная модель, которая представлена на рис. 2.2, так как проводится лишь базовая классификация движения и оптимизация приема сигнала GPS на основании типа движения. Помимо этого, проводится валидация для полученных точек GPS и их фильтрация, если точек много.

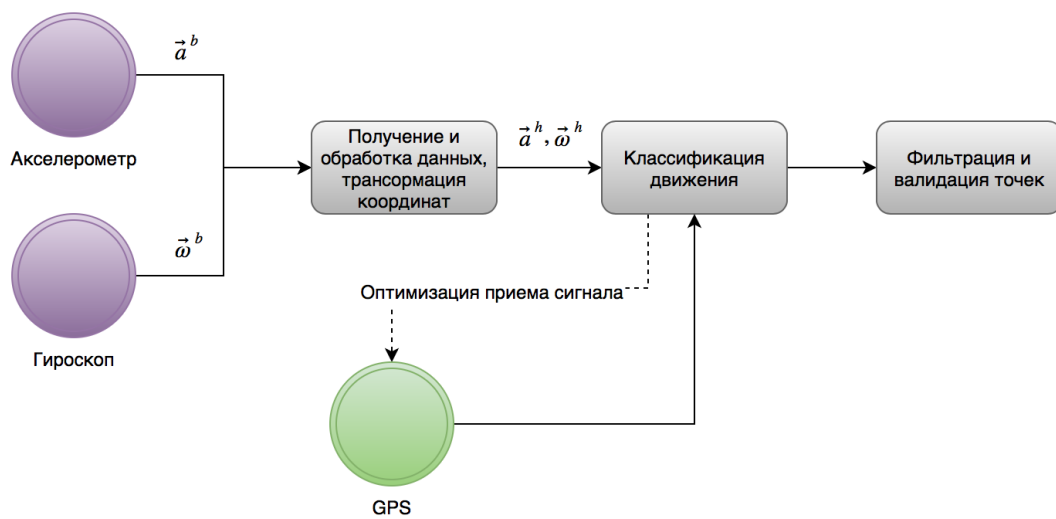


Рис. 2.2: Упрощенная схема модели обработки данных.

## 2.2 Используемое оборудование

В качестве устройства для снятия данных использовался Apple iPhone 5s с операционной системой iOS 9.3.1. Специально для этих целей написано приложения для считывания и сохранения показателей всех датчиков, которые доступны на устройстве: GPS, акселерометр, педометр, гироскоп и магнетометр.

# Глава 3

## Теория

### 3.1 Трансформация координат

При анализе ускорения возникает необходимость разделения ускорения на вертикальную и горизонтальную части. Рассматривается случай, когда устройство закреплено на поясе человека. Будем рассматривать систему, связанную с телом, считая, что корпус человека при ходьбе или беге, как правило, имеет наклон вперед. Трансформацию координат будем проводить также в систему, связанную с телом человека, но ось  $y$  которой направлена перпендикулярно поверхности Земли (см. рис. 3.1).

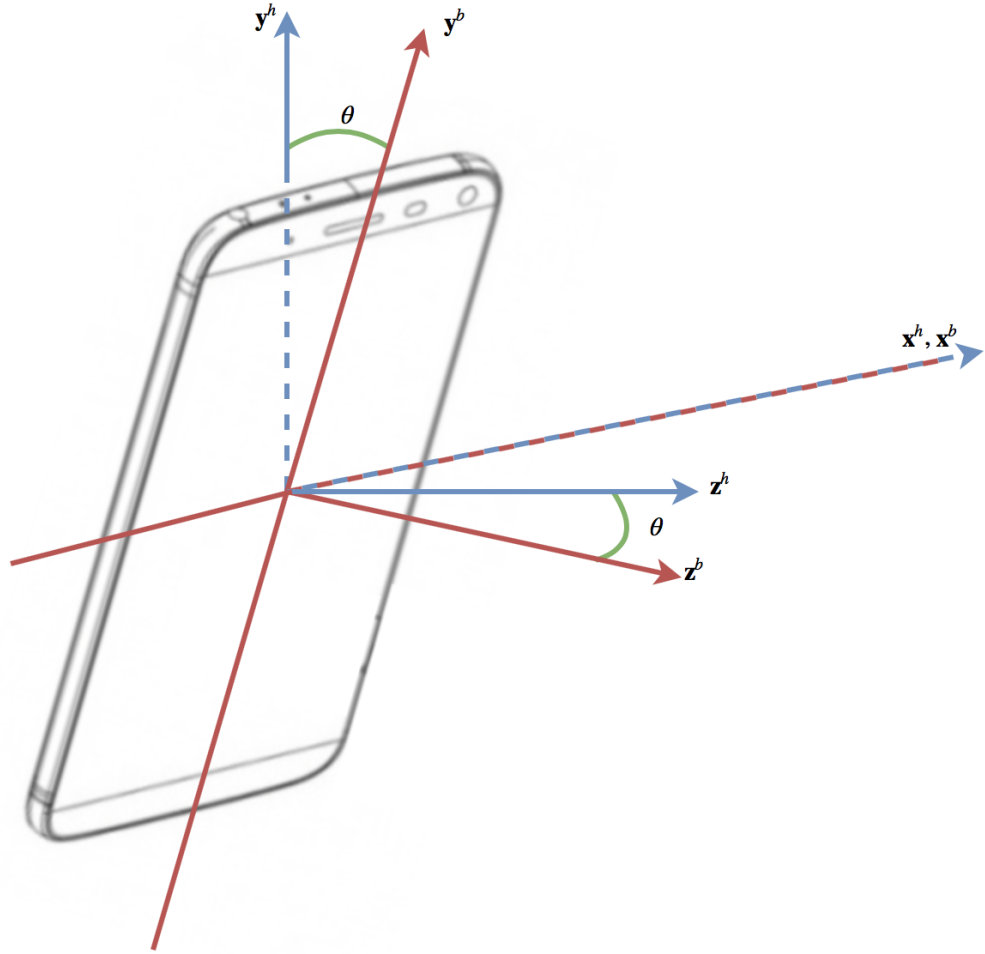


Рис. 3.1: Иллюстрация положения устройства в двух координатных системах: в одной ось  $y^b$  связана с устройством, в другой ось  $y^h$  направлена перпендикулярно Земле (далее будем называть её горизонтальной координатной системой). Устройство имеет наклон на угол  $\theta$  вокруг оси  $x^b$ . Поворот вокруг оси  $y^b$  дал бы угол  $\phi$ .

Уравнения для трансформации ускорения и угловой скорости к горизонтальной системе координат:

$$a^h = R_b^h(g^b + a^b) \quad (3.1)$$

$$\omega^h = R_b^h\omega^b \quad (3.2)$$

Где  $R_b^h$  – матрица поворота от системы координат, связанной с телом, к

горизонтальной системе координат, которая имеет вид:

$$R_b^h = \begin{pmatrix} -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \\ -\cos \theta & -\sin \theta \sin \phi & -\sin \theta \cos \phi \\ 0 & -\cos \phi & \sin \phi \end{pmatrix} \quad (3.3)$$

где  $\theta$  и  $\phi$  - углы поворота в системе координат, связанной с телом (углы Эйлера), вокруг осей  $\mathbf{x}$  и  $\mathbf{y}$  соответственно.

Далее в работе под ускорением тела в различных направлениях подразумеваются величины в горизонтальной системе координат, обозначенные как  $a_x, a_y, a_z$ .

## 3.2 Методы изучения особенностей сигнала

После трансформации сигнала ускорения в горизонтальную систему координат ставится задача изучения особенностей сигнала. Используется принцип скользящего окна, что в данном случае подразумевает наличие буфера точечных измерений ускорения. Окно характеризуется размером буфера  $N$  и перекрытием (величина, отражающая насколько данное окно перекрывается с предыдущим). Выбор подходящей длины окна производится с тем условием, что сигнал внутри окна можно рассматривать как стационарный и устоявшийся. Перекрытие влияет на частоту обновления результатов и выбирается исходя из того, насколько часто нужны результаты обработки.

В качестве меры разброса сигнала относительно его среднего значения берется вариация. Вариация сигнала  $\mathbf{x}$  записывается как

$$s^2 = \frac{1}{N-1} \sum_{n=1}^N (x_n - \bar{x})^2. \quad (3.4)$$

Квадратный корень вариации называется стандартным отклонением, обо-

значается как  $s$ , а  $x$  – среднее значение сигнала внутри окна, которое определяется как

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (3.5)$$

Для анализа сигнала по частоте можно применять Дискретное Преобразование Фурье (ДПФ):

$$X_k = N \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \quad (3.6)$$

Применение этой формулы в прямом виде дает сложность  $O(N^2)$ , но существуют алгоритмы Быстрого Преобразования Фурье (БПФ), имеющие сложность  $O(N \log N)$ , которые и применялись при вычислениях [12].

ДПФ применимо к периодическим сигналам. Использование скользящего окна может привести к трудностям, так как возможно возникновение нежелательных граничных эффектов. Чтобы избежать данной проблемы, используется окно Ханна(Хеннинга), функция которого определяется как

$$\omega(n) = 0.5 \left\{ 1 - \cos \frac{2\pi n}{N-1} \right\} \quad (3.7)$$

а формула ДПФ в данном случае принимает вид:

$$X_{k,m} = N \sum_{n=0}^{N-1} x_n \omega(n-m) e^{-i2\pi k \frac{n}{N}} \quad (3.8)$$

## Глава 4

# Анализ движения человека по данным ИНС

Чтобы сделать предположения для классификации движения, необходимо проанализировать данные сенсоров. Для примера рассмотрим рис. 4.1, на котором представлен график вертикального ускорения. На 17 секунде пользователь начинает бежать, что сопровождается увеличением стандартного отклонения (см. рис. 4.2).

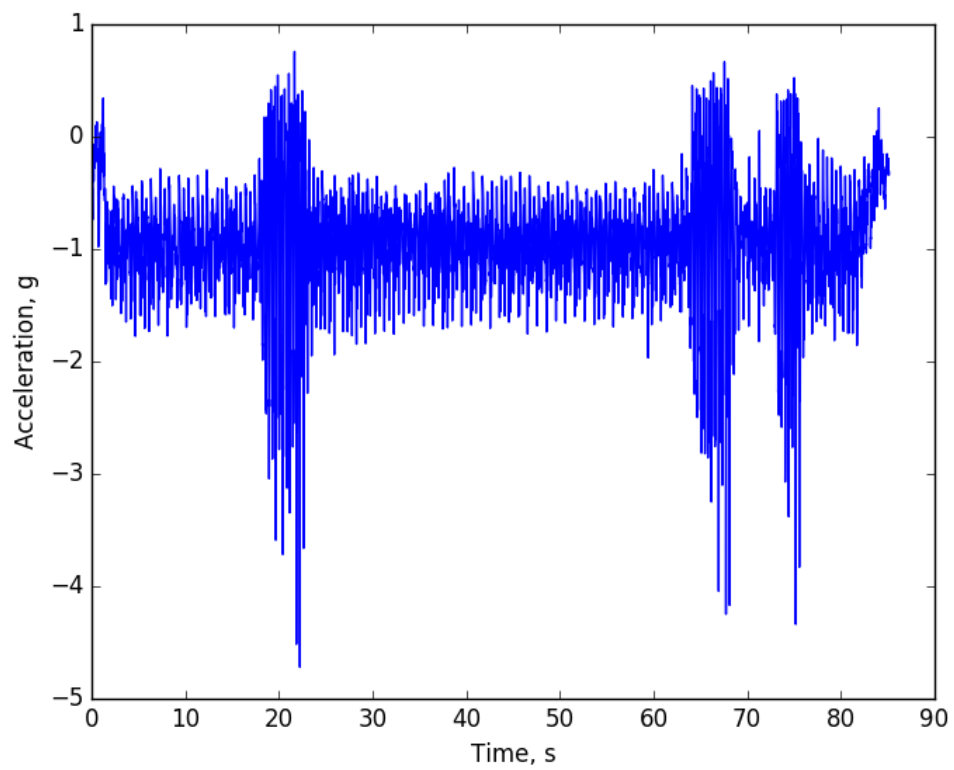


Рис. 4.1: График показаний сенсора ускорения при чередовании бега и ходьбы.



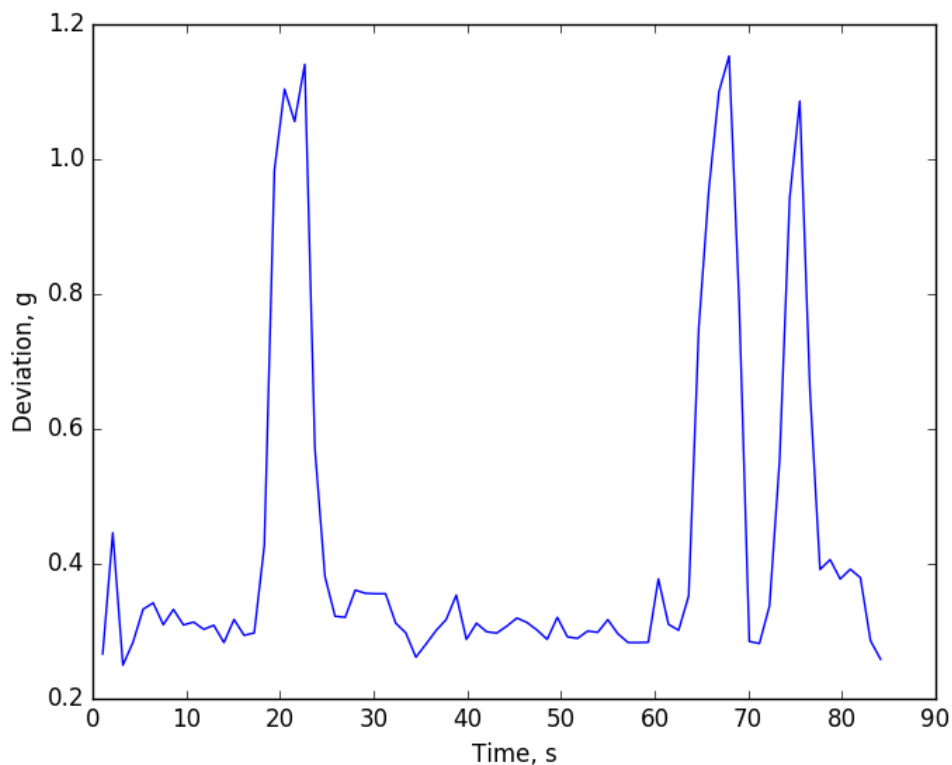


Рис. 4.2: Стандартное отклонение ускорения при чередовании бега и ходьбы.

## 4.1 Сбор данных ИНС

Проведено несколько сборов данных для анализа. Данные собирались как внутри зданий, учитывая только показатели сенсоров ИНС, так и снаружи, где добавлялись данные GPS. В обоих случаях устройство было расположено на поясе. Для данных акселерометра направление оси  $z$  совпадает с направлением движения, ось  $y$  направлена вверх и ось  $x$  направлена влево. Частота обновления данных акселерометра составляет около 200 Гц, данные GPS поступали без определенной частоты.

Так как необходимо учитывать, что движение у людей может иметь разный характер, был организован сбор данных, в котором приняло участие несколько человек. Всего участвовало 4 человека со следующими биомет-

рическими данными: все мужского пола, вес в пределах 60-90, рост 170-187 см. Для каждого тестируемого собраны данные на длинные прогулки по 5-7 минут, бег одного круга на стадионе длиной 400 м и комбинация движения ходьба/бег.

## 4.2 Поиск особенностей данных и их классификация

При движении наблюдается циклическое поведение в сигнале вертикального ускорения. Каждый локальный максимум и локальный минимум в графике сигнала сопоставим с шагом (см. рис. 4.3).

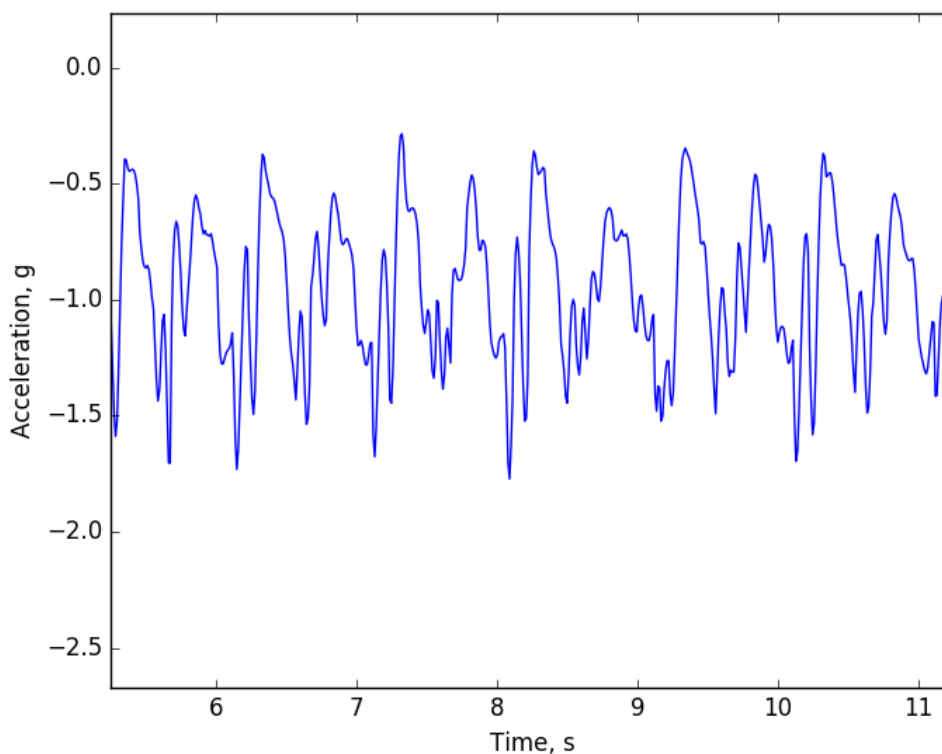


Рис. 4.3: Сигнал акселерометра при ходьбе.

При рассмотрении спектра сигнала вертикального ускорения при хождении доминирует частота 2 Гц (см. рис. 4.4). Сигнал от других осей менее

циклический, поэтому его спектр более размазан (см. рис. 4.5). При беге испытуемых максимальная частота, которая была замечена - 3 Гц. Самая высокая частота шагов, которая была замечена у легкоатлетов при беге на дистанцию 100 м – около 5 Гц, но данное движение не учитывается, так как оно носит локальный характер и довольно редкое (частота шагов Хусейна Болта – рекордсмена по бегу на 100 м, к примеру, составляет 4,8 Гц) [13]. Поэтому решено принять максимально возможную частоту 4 Гц.

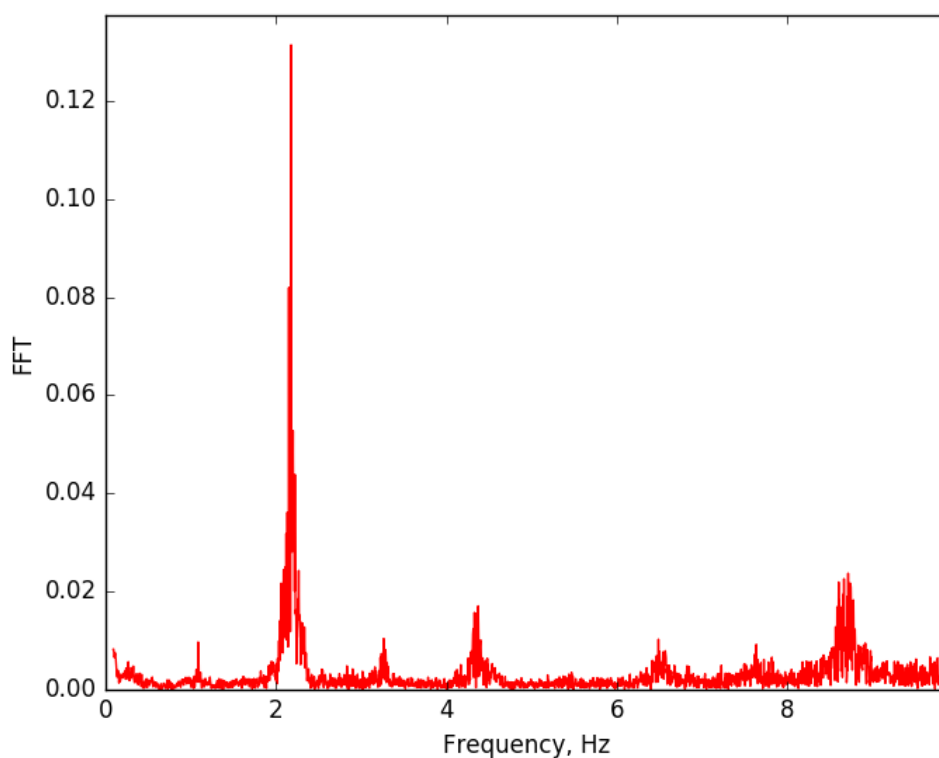


Рис. 4.4: Разложение сигнала вертикального ускорения по спектру в случае ходьбы.

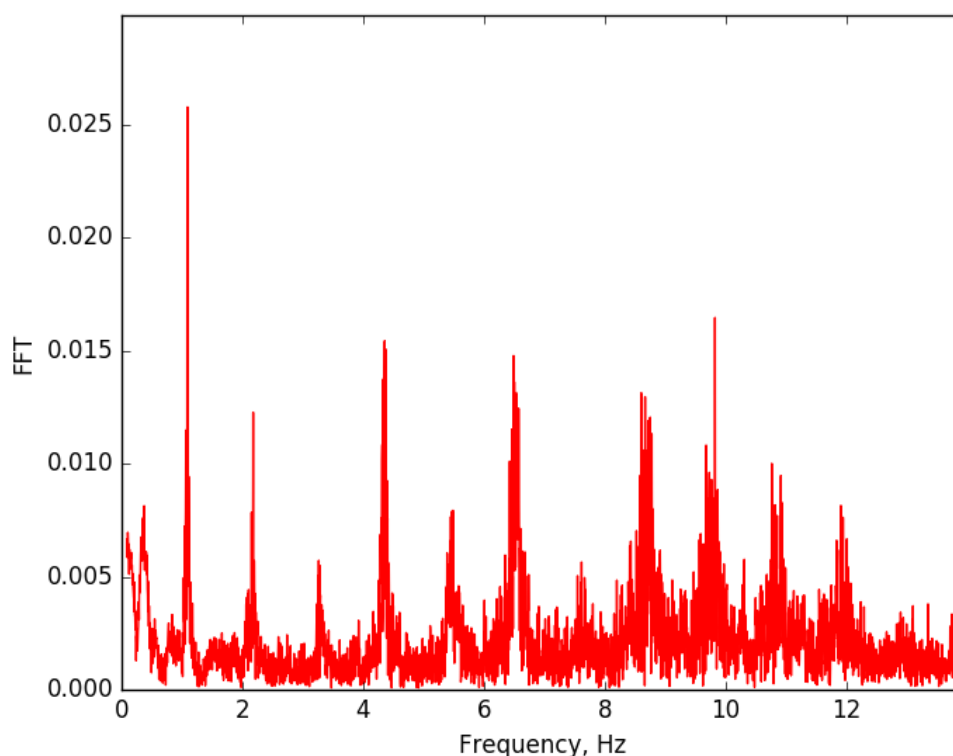


Рис. 4.5: Разложение сигнала горизонтального ускорения по спектру в случае ходьбы.

Стандартное отклонение амплитуды имеет большой разброс в зависимости от скорости ходьбы и бега, а также носит индивидуальный характер, поэтому выбор каких-либо фиксированных значений для их классификации неуместен. В качестве решения предлагается калибровать значения данного параметра, отталкиваясь от значения стандартного отклонения амплитуды в случае, когда пользователь стоит на месте. Размер скользящего окна при вычислениях  $N = 200$ . Он обеспечивает обновление результата классификации каждую секунду.

### 4.3 Схема алгоритма классификации

На рис. 4.6 представлена схема алгоритма классификации. Здесь  $Std(a_y)$  – стандартное отклонение вертикального ускорения (ось  $y$  направлена вверх).

$FFT(a_y)$  – БПФ для сигнала ускорения. Данные об ускорении поступают с акселерометра с частотой 200 Гц. Блоки схемы проверяют, удовлетворяет ли  $Std(a_y)$  какому-либо случаю движения, и если оно подходит для бега или ходьбы, то происходит дополнительная проверка, имеет ли сигнал выраженную частоту, допустимую для данного типа движения. Если спектр сигнала расплывчатый, то движение относится к неопознанному типу.

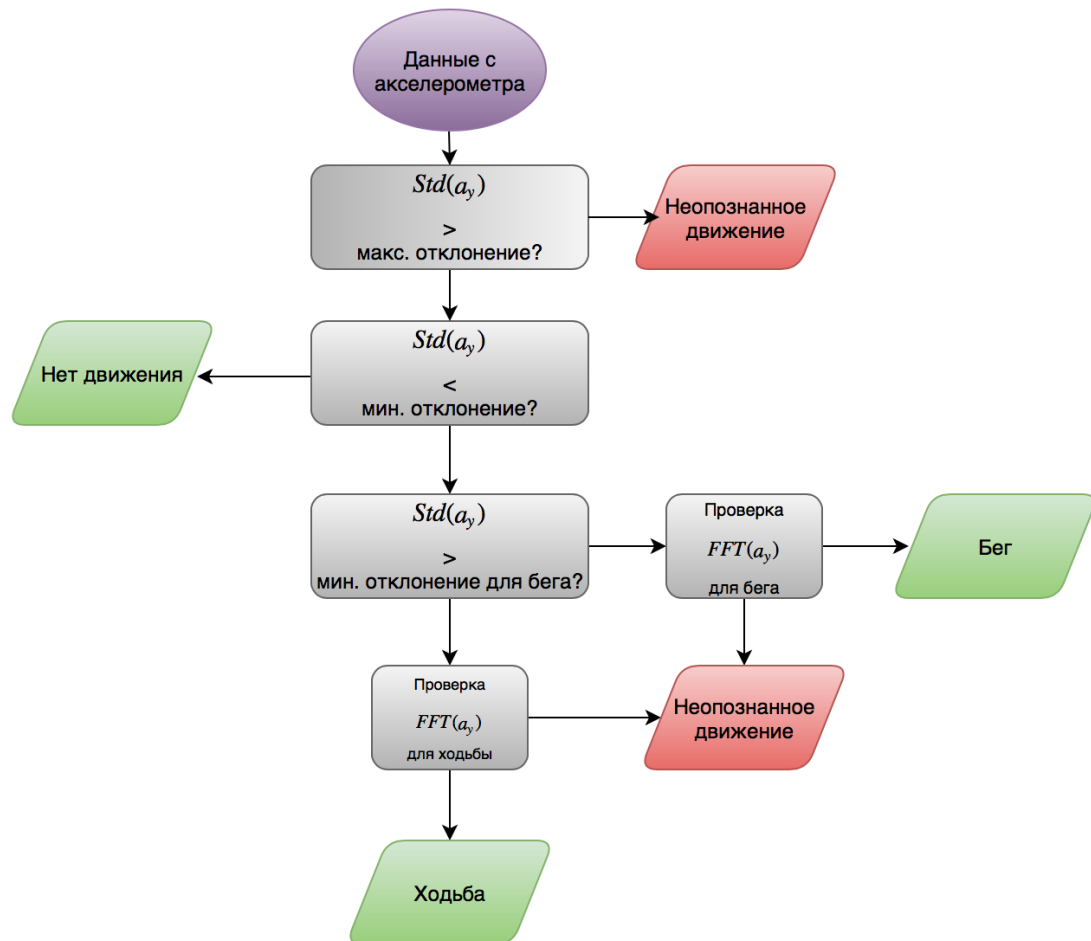


Рис. 4.6: Схема алгоритма классификации.

## 4.4 Результаты испытаний

В таблице 4.1 приведены результаты испытаний классификатора. Для испытаний использовались записи с частой сменой движения. По результатам ошибка не превысила 7%, но в тестировании приняло участие всего 4 чело-

	Время трека, с	Ошибочный результат, с	Ошибка
Тест 1	127	8	6.3%
Тест 2	160	7	4.3%
Тест 3	181	5	2.8%
Тест 4	197	9	4.5%
Тест 5	204	14	6.8%
Тест 6	140	3	2.1%
Тест 7	135	6	4.4%
Тест 8	144	9	6.3%

Таблица 4.1: Таблица результатов испытаний классификатора.

века, и устройство было постоянно закреплено на поясе, поэтому ошибка может быть занижена. Стоит отметить, что ошибочный результат возникал в основном в начале движения и на границе смены движения, и если учитывать, что пользователь в среднем меняет тип движения гораздо реже, чем каждую секунду, то результаты можно считать вполне приемлемыми.

## Глава 5

# Оптимизации в приеме сигнала GPS на основе типа движения

Так как прием сигнала GPS потребляет большое количество энергии, то предлагается рассмотреть, как можно использовать данные о типе движения для оптимизации приема сигнала GPS на мобильном устройстве. В операционной системе iOS можно напрямую выставить расстояние обновления данных GPS, а также точность, с которой он принимается, поэтому самый очевидный путь в данном случае — изменять параметры приема сигнала в зависимости от типа движения.

### 5.1 Выставление частоты обновления сигнала GPS

При известном типе движения предлагается выставить расстояние между обновлениями локации. При тестировании приемлемую частоту обновлений удалось получить при следующих параметрах:

- 1 отсутствие движения – отключение GPS и ожидание начала движения;
- 2 ходьба - 30 м;
- 3 бег - 80 м.

В реальности при выставлении данных фильтров на устройствах iOS обновления локаций могут приходиться чаще, особенно в случаях, когда максимальная точность GPS не достигнута, и параметры подбирались, учитывая этот факт. На устройствах iOS данная фильтрация задается выставлением поля `distanceFilter` у объекта класса `CLLocationManager` стандартной библиотеки iOS MapKit.

## 5.2 Фильтрация точек

В данной секции предлагается алгоритм фильтрации точек для уменьшения их числа на локационном треке с сохранением формы трека. Данная фильтрация уместна, если локационные данные передаются по сети, чтобы снизить объем передаваемой информации. Считается, что новые локационные точки приходят по одной при каждой итерации.

**Data:** location

**Result:** locationIsValid: Bool

**if** *previousLocation* == *nil* **then**

    | store location as *previousLocation*; return location;

**end**

**if** *filterMinDistancePass(location)* OR *filterCourseChange(location)* OR *filterStationaryCase(location)* **then**

    | store location as *previousLocation*; return location;

**end**

### Algorithm 1: Фильтр LocationFilter

Для снижения частоты точек расстояние для фильтрации выставляется в



зависимости от типа движения: для езды на автомобиле оно больше, чем для бега и ходьбы:

**Data:** location, previousLocation, minDistance = defaultMinDistance

**Result:** locationIsValid: Bool

**if** *type of motion for location is automotive* **then**

    | minDistance = automotiveMinDistance;

**end**

**if** *distance between location and previousLocation*  $\geq$  *minDistance* **then**

    | return True;

**else**

    | return False;

**end**

### **Algorithm 2:** Функция filterMinDistancePass

Чтобы снизить срезы на повороте, отслеживается изменение курсе между точками, в данном случае пропускаются любые точки, курс которых изменился больше, чем на 40 градусов:

**Data:** location, previousLocation

**Result:** locationIsValid: Bool

**if** *type of motion for location is stationary* **then**

    | return False;

**end**

**if** *course change for location*  $\geq$  *40 degrees* **then**

    | return True;

**else**

    | return False;

**end**

### **Algorithm 3:** Функция filterCourseChange

Для стационарного случая нужно учесть тот факт, что обновления GPS продолжают приходить, когда пользователь стоит неподвижно, даже если точность новых локаций меньше. Для этого все новые точки, которые не точнее предыдущей, либо попадают в радиус ошибки предыдущей, отсеи-

ваются (для наглядности см. рис. 5.1):

**Data:** location, previousLocation

**Result:** locationIsValid: Bool

**if** *type of motion for previousLocation is NOT stationary* **then**

| return False;

**end**

**if** *type of motion for locaion is NOT stationary* **then**

| return False;

**end**

**if** *location.coordinate == previousLocation.coordinate* **then**

| return False;

**end**

**if** *location.accuracy > previousLocation.accuracy* **then**

| return False;

**end**

**if** *location.accuracy < previousLocation.accuracy* **then**

| **if** *distance between location and previousLocation <*  
*location.accuracy + previousLocation.accuracy* **then**

| | return True;

| **else**

| | return False;

| **end**

**else**

| return False;

**end**

**Algorithm 4:** Функция filterStationaryCase

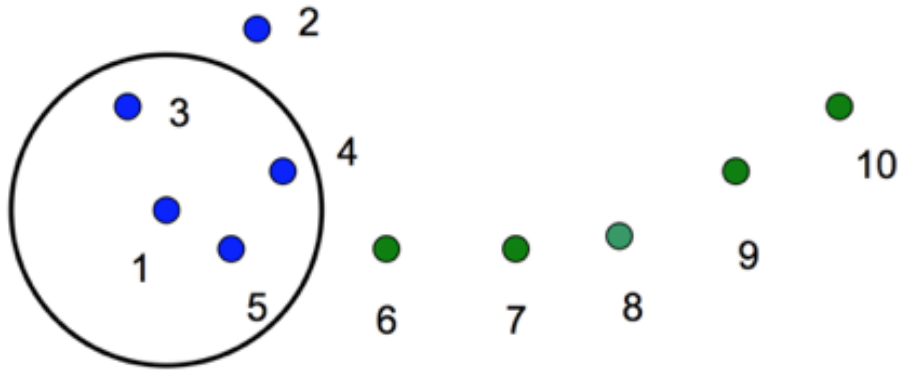


Рис. 5.1: Иллюстрация к алгоритму функции `filterStationaryCase`. Точки 2, 3, 4 и 5 входят в радиус ошибки точки 1 (радиус ошибки в данном случае складывается из радиусов ошибок двух точек), остальные точки последовательно пропускаются фильтром как отдельные.

В работе фильтра предлагается использовать классификатор типа движения, доступный в стандартной библиотеке Apple CoreMotion. Он различает случаи ходьбы, бега, езды на автомобиле и велосипеде. Однако у него замечены проблемы с определением стационарного типа движения, что учтено в фильтрации. Задержка при смене типа движения у данного классификатора составляет примерно 3 секунды.

# Глава 6

## Заключение

### 6.1 Выводы

В результате работы удалось создать простой, но эффективный классификатор по трем типам движения: стационарное, ходьба и бег с частотой обновления результатов в одну секунду. Работа классификатора проверялась только в случае положения устройства на поясе пользователя, поэтому он нуждается в доработках, чтобы применять его для широкого использования. Также написан, с проверкой на практических данных, фильтр обновлений GPS. Фильтр справляется с проблемой ненужных обновлений статической локации, но для случая движения может потребоваться индивидуальная установка параметров, в зависимости от желаемой частоты обновлений локаций. В процессе работы решено не заострять дальнейшее внимание на проработке классификатора движения, так как стало известно о существовании стандартных библиотек для этого. Однако следует определить, подходят ли такие классификаторы для применения их в навигационном счислении, что отведено на дальнейшую работу.

## 6.2 Планы на дальнейшую работу

В качестве продолжения планируется перейти к изучению возможности создания программного модуля навигационного счисления для мобильных устройств. Среди пунктов работы:

- 1 проверка возможности применения классификатора стандартной библиотеки iOS CoreMotion для определения средней длины шага;
- 2 работа над алгоритмом по определению средней длины шага пользователя по данным классификатора типа движения и GPS;
- 3 работы над программным модулем, определяющего местоположение пользователя на основе данных ИНС при условии доступности информации о средней длине шага пользователя;
- 4 дальнейшая валидация и совершенствование фильтра GPS локаций.

# Литература

- [1] *Kakiuchi, Noriaki*. Pedestrian dead reckoning for mobile phones through walking and running mode recognition / Noriaki Kakiuchi, Shunsuke Kamijo // *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. — 2013. — Pp. 261–267.
- [2] Apple Official iOS Documentation. <https://developer.apple.com/ios/documentation/>.
- [3] Google Official Beacons Documentation. <https://developers.google.com/beacons/>.
- [4] Strap-Down Pedestrian Dead-Reckoning System / P. Goyal, V. Ribeiro, H. Saran, A. Kumar // *International Conference on Indoor Positioning and Indoor Navigation (IPIN 2011)*. — 2011. — Pp. 1–7.
- [5] *Susi, Melania*. Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users / Melania Susi, Valérie Renaudin, Gérard Lachapelle // *Sensors*. — 2013.
- [6] An Effective Pedestrian Dead Reckoning Algorithm Using a Unified Heading Error Model / W. Chen, R. Chen, Y. Chen et al. // *Position Location and Navigation Symposium (PLANS)(IEEE/ION 2010)*. — 2010. — Pp. 340–347.
- [7] An Integrated GPS and Multi-Sensor Pedestrian Positioning System for 3D Urban Navigation / W. Chen, Z. Fu, R. Chen et al. // *Urban Remote Sensing Event*. — 2009. — Pp. 1–6.

- [8] *Ladetto, Q.* On Foot Navigation: Continuous Step Calibration Using Both Complementary Recursive Prediction and Adaptive Kalman Filtering / Q. Ladetto // *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*. — 2000. — Pp. 1735–1740.
- [9] Digital Magnetic Compass and Gyroscope for Dismounted Soldier Position and Navigation / Q. Ladetto, J. van Seeters, S. Sokolowski et al. // *Military Capabilities enables by Advances in Navigation Sensors, Sensors and Electronics Technology Panel, NATO-RTO meetings*. — 2002.
- [10] Towards HMM Based Human Motion Recognition Using MEMS Inertial Sensors / G. Shi, Y. Zou, Y. Jin et al. // *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*. — 2009. — Pp. 1762–1766.
- [11] A Hybrid HMM/SVM Classifier for Motion Recognition Using IMU Data / W. Wan, H. Liu, L. Wang et al. // *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*. — 2007. — Pp. 115–120.
- [12] Fast Fourier Transform, MATLAB documentation. <https://www.mathworks.com/help/>.
- [13] *Romanov.* Analysis of Usain Bolt’s running technique / Romanov. <https://posemethod.com/usain-bolts-running-technique/>.