|  |
| --- |
| **МИНОБРНАУКИ РОССИИ**  федеральное государственное бюджетное образовательное  учреждение высшего образования  **«Национальный исследовательский университет «МЭИ»** |

|  |  |
| --- | --- |
| **Институт** | ИРЭ |
| **Кафедра** | РТС |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(магистерская диссертация)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Направление** | | | | | 11.04.01 Радиотехника | | | | |
| (код и наименование) | | | | | | | | | |
| **Направленность (программа)** | | | | | | | | | **Радиотехнические системы** |
|  | | | | | | | | | |
|  | | | | | | | | | |
| **Форма обучения** | | | | | | | **очная** | | |
|  | | (очная/очно-заочная/заочная) | | | | | | | |
| **Тема:** | | **Разработка и исследование алгоритма кластеризации** | | | | | | | |
| **радиолокационных паттернов для повышения эффективности** | | | | | | | | | |
| **обработки информации в системах пассивной радиолокации** | | | | | | | | | |
|  | | | | | | | | | |
| **Студент** | | | **ЭР-12м-20 Н.В. Масалкова** | | | | | | |
| группа подпись фамилия и инициалы | | | | | | | | | |
| **Научный руководитель** | | | | | | **к.т.н. зав. каф. РТС Р.С. Куликов** | | | |
| уч. степень должность подпись фамилия и инициалы | | | | | | | | | |
| **Консультант** | | | | **ассистент каф. РТС Т.А. Бровко** | | | | | |
| уч. степень должность подпись фамилия и инициалы | | | | | | | | | |
| **Консультант** | | | |  | | | | | |
| уч. степень должность подпись фамилия и инициалы | | | | | | | | | |
|  | | | | | | | | | |
| **«Работа допущена к защите»** | | | | | | | | | |
| **Зав. кафедрой РТС** | | | | | | **к.т.н. доцент Р.С. Куликов** | | | |
| уч. степень звание подпись фамилия и инициалы | | | | | | | | | |
| **Дата** |  | | | | | | |  | |
|  | | | | | | | |  | |
| **Москва, 2022**   |  | | --- | | **МИНОБРНАУКИ РОССИИ**  федеральное государственное бюджетное образовательное  учреждение высшего образования  **«Национальный исследовательский университет «МЭИ»** |  |  |  | | --- | --- | | **Институт** | ИРЭ | | **Кафедра** | РТС |   **ЗАДАНИЕ**  **НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ (магистерскую диссертацию)**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Направление** | | | | 11.04.01 Радиотехника | | | | | (код и наименование) | | | | | | | | | **Направленность (профиль)** | | | | | | | **Радиотехнические системы** | |  | | | | | | | | |  | | | | | | | | | **Форма обучения** | | | | | | **очная** | | |  | (очная/очно-заочная/заочная) | | | | | | | | **Тема:** | **Разработка и исследование алгоритма кластеризации** | | | | | | | | **радиолокационных паттернов для повышения эффективности** | | | | | | | | | **обработки информации в системах пассивной радиолокации** | | | | | | | | |  | | | | | | | | | **Студент** | | **ЭР-12м-20 Н.В. Масалкова** | | | | | | | группа подпись фамилия и инициалы | | | | | | | | | **Научный руководитель** | | | | | **к.т.н. зав. каф. РТС Р.С. Куликов** | | | | уч. степень должность подпись фамилия и инициалы | | | | | | | | | **Консультант** | | | **ассистент каф. РТС Т.А. Бровко** | | | | | | уч. степень должность подпись фамилия и инициалы | | | | | | | | | **Консультант** | | |  | | | | | | уч. степень должность подпись фамилия и инициалы | | | | | | | | | **Зав. кафедрой** | | | | | **к.т.н. доцент Р.С. Куликов** | | | | уч. степень звание подпись фамилия и инициалы | | | | | | | | |  | | | | | | | | | **Место выполнения работы** | | | | | | | Кафедра РТС «НИУ «МЭИ» | |  | | | | | | | |   **1. Обоснование выбора темы выпускной квалификационной работы**  Распознавание целей в воздушном пространстве является одним из приоритетных направлений радиолокации. Характерным примером обработки радиолокационной информации является задача обнаружения летательных аппаратов и оценка их принадлежности к определенному классу или типу. Проблема аналогична задаче кластеризации и последующей классификации в области прикладной статистики: необходимо из множества одиночных отметок (радиоимпульсов) выбрать несколько центров группирования, которые соответствуют обнаруживаемым целям.  Таким образом, возможность повышения эффективности обработки информации в системах пассивной радиолокации за счет отождествления сигналов с целями путем кластеризации принятых радиоимпульсов является актуальной научно-практической задачей. Это подтверждается большим количеством публикаций на данную тему.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Научный руководитель** | | Р.С. Куликов | **дата** | 02.09.2021 | |  | | | | | | **Студент** | Н.В. Масалкова | | **дата** | 02.09.2021 |   **2. Консультации по разделу**   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  | | | | | |  | | | | | |  | | | | | |  | | | | | | **Подпись консультанта** |  | **дата** |  |   **3. Консультации по разделу**   |  |  |  |  | | --- | --- | --- | --- | |  | | | | |  | | | | |  | | | | |  | | | | |  | | | | | **Подпись консультанта** |  | **дата** |  |   **4. План выполнения выпускной квалификационной работы**   |  |  |  |  | | --- | --- | --- | --- | | **№**  **п\п** | **Содержание разделов** | **Срок**  **выпол-**  **нения** | **Трудоём-**  **кость,**  **%** | | I. | Теоретическая часть   1. Обзор и анализ этапов обработки радиолокационной информации. 2. Обзор и анализ алгоритмов кластеризации, выбор подходящего алгоритма кластеризации для решения поставленной задачи. 3. Разработка имитационной модели принимаемой последовательности радиоимпульсов. | январь 2021  февраль 2020  август 2021 | 5%  5%  5% | | II. | Экспериментальная часть   1. Апробация алгоритма кластеризации к модели принимаемой последовательности радиоимпульсов. 2. Апробация алгоритма кластеризации к реальной принимаемой последовательности радиоимпульсов. 3. Апробация алгоритма кластеризации к модели принимаемой последовательности радиоимпульсов, с целью выявления характерных паттернов. 4. Апробация алгоритма кластеризации к реальной принимаемой последовательности радиоимпульсов, с целью выявления характерных паттернов. | октябрь  2021  декабрь  2021  февраль  2022  апрель  2022 | 5%  10%  15%  15% | | III. | Публикации   1. Подача публикации на IEEE EDM 2022 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM) to the 100th anniversary of the legendary NETI rector Georgy Lyshchinsky. | март  2022 | 15% | | IV. | Оформление диссертации   1. Оформление расчетно-пояснительной записки выпускной квалификационной работы в соответствии с требованиями 2. Подготовка презентации для доклада на заседании ГЭК. | апрель  2021  май  2021 | 15%  10% |   **5. Рекомендуемая литература**   |  | | --- | | 1. А.Л. Татузов. Нейронные сети в задачах радиолокации. Кн. 28. – М.: | | Радиотехника, 2009. – 432 с.: ил. (Научная серия «Нейрокомпьютеры и их | | применение). | | 2. Князев Н.Л., Денисова Л.А., Методы распознавания динамических | | объектов в радиолокационном пространстве. УДК 621.396.96:004.021. | | Омский государственный технический университет, г. Омск, Россия. |   ***Примечания:***   1. Задание брошюруется вместе с выпускной работой после титульного листа (страницы задания имеют номера 2, 3, 4, 5). 2. Отзыв руководителя, рецензия(и), отчет о проверке на объем заимствований и согласие студента на размещение работы в открытом доступе вкладываются в конверт (файловую папку) под обложкой работы. | | | | | | | | | |

**АННОТАЦИЯ**

К магистерской диссертации на тему «Разработка и исследование алгоритма кластеризации паттернов принятых сигналов для повышения эффективности обработки информации в системах пассивной радиолокации»

Выпускная работа магистра посвящена синтезу и исследованию алгоритма кластеризации паттернов принятых сигналов для повышения эффективности обработки информации в системах пассивной радиолокации.

Магистерская диссертация состоит из введения, четырех основных глав, заключения, списка используемых источников информации и приложений с листингом программ.

В первой главе представлен обзор по теме алгоритмов обработки для современных систем позиционирования внутри помещений научно-технических источников, проиндексированных в базах: РИНЦ, ВАК, Scopus, WoS.

Во второй главе изложен синтез алгоритма комплексирования измерений локальной навигационной системы на безе сверхширокополосных сигналов и датчика угловых скоростей, а также аналитическая оценка точности синтезированного алгоритма фильтрации.

В третьей главе приведены результаты имитационного моделирования синтезированного алгоритма.

В четвертой главе представлено описание структуры макета локальной навигационной системы, на базе которого проводились эксперименты по исследованию синтезированного алгоритма. Здесь же изложены сравнительные оценки погрешностей определения координат, полученные с использованием разработанного алгоритма.

В заключении констатируется достижение цели работы и приводятся основные результаты ВКР.

Работа содержит 102 страниц, 31 рисунок, 7 таблиц и Приложения.

Оглавление

[Список сокращений 9](#_Toc75024753)

[Введение 10](#_Toc75024754)

[Глава 1. Обзор существующих решений 16](#_Toc75024755)

[1.1 Алгоритмы в системах локальной навигации на базе СШП сигналов 18](#_Toc75024756)

[1.2 Алгоритмы в инерциальных системах локальной навигации 22](#_Toc75024757)

[1.3 Комплексные алгоритмы в системах локальной навигации на базе СШП и ИНС 27](#_Toc75024758)

[Глава 2. Синтез алгоритма комплексирования 32](#_Toc75024759)

[2.1 Постановка задачи 32](#_Toc75024760)

[2.2 Синтез комплексного фильтра 34](#_Toc75024761)

[2.3 Оценка ошибок фильтрации 39](#_Toc75024762)

[Глава 3. Моделирование комплексного фильтра 46](#_Toc75024763)

[Глава 4. Экспериментальное исследование алгоритма комплексирования СШП ЛНС И ДУС 59](#_Toc75024764)

[4.1 Архитектура экспериментального стенда 59](#_Toc75024765)

[4.1.1. Сверхширокополосная локальная навигационная система 59](#_Toc75024766)

[4.1.2. Инерциальная навигационная система 63](#_Toc75024767)

[4.1.3. Макет комплексной локальной навигационной системы 65](#_Toc75024768)

[4.2. Результаты экспериментального исследования синтезированного алгоритма 68](#_Toc75024769)

[Заключение 78](#_Toc75024770)

[Список литературы 79](#_Toc75024771)

[Приложение А 82](#_Toc75024772)

[Приложение Б 83](#_Toc75024773)

[Приложение В 91](#_Toc75024774)

[Приложение Г 92](#_Toc75024775)

[Приложение Д 93](#_Toc75024776)

# Список сокращений

|  |  |
| --- | --- |
| AOA | Angle Of Arrival |
| API | Application Programming Interface |
| DMP | Digital Motion Processor |
| NLOS | Non-Line-Of-Sight |
| PDR | Pedestrian Dead Reckoning |
| RTDOA | Reverse Time Difference Of Arrival |
| TDOA | Time Difference Of Arrival |
| TOA | Time Of Flight |
| ZUPT | Zero-Velocity Update |
| АСП | Акустическая Система Позиционирования |
| ГНСС | Глобальные Навигационные Спутниковые Системы |
| ГФ | Геометрический Фактор |
| ДУС | Датчик Угловых Скоростей |
| ИИБ | Инерциально-Измерительный Блок |
| ИНС | Инерциальная Навигационная Система |
| ЛНС | Локальная Навигационная Система |
| МЭМС | Микроэлектромеханическая Система |
| РФК | Расширенный Фильтр Калмана |
| СШП ЛНС | Сверхширокополосная Локальная Навигационная Система |

# Введение

Современные радиолокаторы должны работать в условиях, для которых характерны высокая скорость изменения внешней обстановки и большое количество объектов, подлежащих обнаружению [1]. Обработка столь большого количества информации в требуемые сжатые сроки не может быть эффективно осуществлена человеком-оператором [8]. Поэтому перед радиолокационными станциями (РЛС), являющимися основным источником информации, ставится задача проведения законченного цикла её автоматической обработки. Актуальность решения данной задачи в следующем, что в отличие от применявшихся ранее систем обработки, в которых конечное решение принимал человек-оператор, в этих образцах конечное решение принимается автоматом [16].

Для разработки подобных систем потребовалось создание теории автоматической обработки радиолокационной информации (РЛИ). Существуют методы, способные осуществлять автоматическую обработку РЛИ [4-9]. Однако большинство данных методов базируются на положениях классической теории радиолокации, которая опирается на предположения, многие из которых не выполняются на практике. Необходимость использования этих предположений обусловлена трудностью формализации и математического описания всевозможных воздействий помех и других факторов, отсутствием единой методологии оценки систем обработки в различных воздушных и помеховых условиях. И оказалось, что при создании образцов РЛС с автоматической обработкой информации, классическая теория часто оказывается неприменима [9]. До сих пор нет универсальной теории для создания автоматической обработки РЛИ.

Действительно, современные радиолокационные средства в достаточно простых условиях успешно справляются со своими задачами, а в сложных условиях (нестационарных и негауссовых помех) их эффективность может резко снижаться. Снижается достоверность выдаваемой РЛС информации за счет появления большого числа ложных отметок и появления ложных траекторий. Их число может намного превышать число целей в зоне обзора, а использование недостоверной информации ведет к снижению эффективности, например, стрельбовых средств в несколько раз [7].

Непредсказуемость внешней обстановки и высокая динамика её изменения создают значительные трудности для формулировки алгоритмов и для обеспечения высокого качества их работы. Появляются и новые задачи, например, распознавание, кластеризация, классификация и анализ ситуаций. И классических методов решения таких задач нет [11].

На данный момент одно из актуальных и многообещающих направлений построения эффективных систем обработки информации является использование такой информационной технологии, как искусственные нейронные сети (ИНС). Аппарат нейронных сетей предлагает, там, где не существует алгоритмов или эти алгоритмы недостаточно эффективны, многообещающий подход к созданию систем обработки, обладающих столь привлекательными чертами как гибкость, способность адаптироваться к изменениям внешних условий, сохраняя устойчиво высокое качество работы.

Нейронные сети предусматривают применение принципиально нового подходу к синтезу методов обработки в алгоритмическом смысле. Данная технология предоставляет компьютерной модели нейронной сети возможность обучаться на примерах и получать решение для проблем, ранее считавшихся неразрешимыми без участия человека. При этом, как уже упоминалось ранее, достигается гибкость и активность работы, сохранение устойчиво высоких показателей работы при отличиях внешних условий, от рассматриваемых или создаваемых на этапе разработки, возможность построения эффективных систем без трудоемких, а зачастую и невыполнимых, построений аналитических описаний, способность оперирования нечеткими понятия и так далее.

ИНС в последнее время успешно применяются в широком спектре приложений от бытовых приборов, в которых с их помощью осуществляется выбор наиболее благоприятных режимов работы, до крупномасштабных систем финансового анализа и сложнейших вычислительных комплексов управления военными действиями.

Всё это свидетельствует о необходимости внедрения нейросетевой технологии в перспективные системы обработки информации [11]. И соответственно в системы, решающие радионавигационные задачи, так как они требуют высокой скорости обработки больших массивов данных для получения достоверных результатов анализа обстановки.

Следует сказать, что ИНС – это алгоритмический аппарат построения систем принятия решений на основе сетей формальных нейронов, которые реализуют элементарные арифметические функции. Данное понятие позволяет исследовать нейронные сети с помощью традиционного математического аппарата или развивая этот аппарат, но оставаясь в рамках традиционных математических парадигм. В итоге решения, получаемые с помощью нейросетевых технологий, позволяют приблизиться к сколь угодно близко к истинно оптимальным решениям для сложных и нетривиальных условий принятия решений.

Тогда, чтобы успешно применять аппарат нейронных сетей в процессах обработки РЛИ должны быть тщательно проанализированы условия использования, этапы преобразования входных данных, обоснованы принципы оценки качества обработки, определены наиболее перспективные места применения.

В настоящее время актуальными являются исследования по повышению качетсва обработки информации за счет использования нейросетевой технологии в каждом из подтипов подсистем обработки в зависимости от физической структуры входных и выходных сигналов, места их применения, характера изменения внешних условий. Объем исследований при такой постановке задачи оказывается очень широк, и охватить весь спектр возможных применений нейронных сетей в полном объеме не удается. В этих условиях встает задача разделения подобных между собой этапов обработки информации в разнородных системах, эффективность которых могла бы быть резко повышена за счет применения нейронных сетей.

В данной работе нас будет интересовать автоматизация одного из этапов обработки радиолокационной информации.

# Анализ этапов обработки радиолокационной информации

Обработка РЛИ представляет собой наиболее важный комплекс задач радиолокационных станций. Назначение обработки подготовить к выдаче в требуемом виде полную, достоверную и современную информацию для потребителя о состоянии воздушной обстановки, появлении и местоположении воздушных средств объектов, параметрах их движения, возможных вариантах развития динамики изменения воздушно-помеховой обстановки.

В ходе обработки РЛИ решается целый ряд задач, которые можно сгруппировать по близости объектов и применяемых методов в основные этапы.

К основным этапам обработки радиолокационной информации можно отнести [9,10].

* Формирование диаграммы направленности антенны для фазирования антенных решеток;
* Первичная (спектральная) обработка;
* Обнаружение радиолокационных объектов
* Вторичная (траекторная) обработка
* Распознавание объектов и ситуаций
* Анализ обстановки и адаптация

Решаемые на этих этапах задачи служат выполнению главной цели, для которой предназначен радиолокатор – выдаче полной, достоверной и своевременной информации потребителю о воздушных целях.

Большого внимания заслуживает такой этап обработки, как обнаружение отметок от целей и их кластеризация для дальнейшей вторичной (траекторной) обработки целей.

Обнаружение радиолокационных целей обычно включает в себя два подэтапа: обнаружение импульсов, отраженных от целей, и их последующего пакетирования в ходе которого объединяются решения, принятые по каждому отдельному импульсу. Обнаружение целей должно проводиться автоматически, сохраняя высокие характеристики при работе в различны тактических условиях, при воздействии разнотипных помех, в том числе нестационарных, т.е. в существенно неопределенных условиях.

Рассмотрим некоторые методы, которые применяются для автоматического обнаружения воздушных целей в РЛС, а именно использование адаптивного порога обнаружения [18], использование фазовых соотношений в многобазовых сигналах (знаковые методы) и ранговые методы [5].

Данным методам присущи крупные недостатки, которые определяют практическую полную непригодность для РЛС, ведущих работу в реальных условиях. Рассмотрим подробнее: в реальных условиях при воздействии на РЛС помех нестационарного типа, а также хотя и стационарных, но негауссовых помех таких, как отражения от подвижных и местных предметов, пассивные помехи, уровень ложных тревог может возрасти на 2…3 порядка, что недопустимо. Кроме того, при наличии в пределах опорной выборки сигналов, отраженных от соседних с обнаруживаемой целью, что характерно для обнаружения целей в группах, энергетические потери могут составлять 10…15 дБ и более [18]. Исключение составляют ранговые методы, обеспечивающие практически полную стабилизацию уровня ложных тревог, но эти методы при малой пачке имеют очень высокие потери даже при обнаружении одиночных целей, а при обнаружении групп целей эти потери катастрофически возрастают.

Для синтеза эффективных методов автоматического обнаружения необходимо привлечь статическую теорию различения гипотез. Синтезируемые методы должны эффективно работать в условиях воздействия основных помех. При этом необходимо определить совокупность параметров помех и обеспечить устойчивость и совокупность параметров.

Пакетирование импульсов проводится для формирования отметок от целей, на основе полученных нескольких импульсов от цели. Эта задача аналогична задаче кластеризации: необходимо из множества одиночных отметок выбрать несколько центров группирования, которые соответствуют обнаруживаемым целям. В итоге проведение группирования отметок в кластеры позволит улучшить алгоритмы вторичной обработки сигналов, а именно сопровождение траекторий групп целей. Группирование позволит сократить количество ложных траекторий примерно на порядок.

Задача кластеризации эффективна решается с помощью нейронных сетей. Подробнее про аппарат ИНС будет показано далее.

# Основные принципы нейросетевых технологий

Для анализа методов нейросетевых технологий определим основные содержательные элементы технологии: информационной и нейросетевой.

## *Определение нейронных сетей*

Воспользуемся определением нейронной сети автора нескольких классических трудов по теории нейронных сетей С. Хайкина [13,19]. Нейронная сеть – это процессор с массово-параллельной архитектурой, который имеет естественную склонность к записи эмпирических знаний и представления для их последующего использования. Этот процесс похож на мозг в двух отношениях:

1. Знания получаются сетью посредством процесса обучения.

2. Для запоминания знаний используются межнейронные веса, известные как синаптические связи.

В данной работе под нейросетевой технологией понимаются приемы, способы и методы обработки информации, основанные на использовании сетей, связанных между собой достаточно простых элементов (нейронов), реализующих методы совместной работы, близкие по содержанию к работе биологических нейронных систем. Данное определение рассматривает близость к биологическим нейронам лишь в самом широком смысле. Здесь не пытаемся осуществить полное моделирование биологической системы. Поэтому многие нейронные сети, не имеющие биологических аналогов, оказываются успешными и полезными в вычислительном отношении и в плане применения в различных приложениях [17,20]. Основные достижения нейросетевой технологии связаны с реальными успешными применениями математических алгоритмов, появившихся как развитие моделей биологических нейронных сетей.

В итоге теория нейронных сетей опирается на предположение о том, что с помощью, рассматриваемой в ее рамках методологически простой модели удастся получить достаточно сложное поведение, необходимое, для получения правильных решений в изменяющихся условиях.

## *Нейросетевые модели*

В основе нейросетевой технологии лежит концепция распределенных адаптивных нелинейных вычислений. Нейронные сети строятся из большого числа простых элементов, каждый из которых обрабатывает свою часть единой задачи. Принципы вычислений для стандартного элемента достаточно просты. Вся мощь нейронных сетей основывается на множестве взаимосвязей между этими простыми элементами, которые совместно участвуют в решении задачи, и в адаптивной настройке параметров (весов), связывающих элементы между собой.

Каждой нейросетевой архитектуре присущ собственный метод обучения, ориентированный на конкретную организацию взаимосвязей в ней. Поэтому обычно рассматриваются не просто архитектуры нейронных сетей, а комплекс из организации элементов сети, нейронов, в виде слоев и метода обучения или алгоритма достижения нейронной сетью состояния, обеспечивающего выполнению ею требуемых действий.

Рассмотрим подробнее две большие группы на которые разделены нейронные сети в соответствии с применяемыми методами обучения: обучение с учителем и обучение без учителя.

При обучении с учителем правильные результаты функционирования сети (желаемые выходы) считаются известными, и одновременно с каждым набором входных значений сети предъявляется информация о том, насколько правильно она функционирует или каким должен быть правильный отклик – референсом. Эта информация используется нейронной сетью для подстройки весов, которые изменяются таким образом, чтобы максимально приблизиться к задаваемым целевым значениям. После обучения нейронная сеть тестируется при различных входных значениях и оценивается, насколько ее выход оказывается близко к желаемому отклику. В итоге обучение с учителем требует на вход нейронной сети представление об окружающем пространстве в виде пары вход-выход, и для измерения качества обучения сети используются прямолинейные и понятные метрики.

При обучении без учителя нейронной сети не сообщаются правильные выходные значения. Сеть работает автономно: она исследует данные, которые ей представляются, находит в них некоторые характерные закономерности и обучается отображать эти закономерности в своем отклике. Она сама подстраивает веса синаптических связей, опираясь на внутреннюю структуру данных. Какие именно признаки способно отыскивать и отображать нейронная сеть, зависит от ее конкретной архитектуры и используемого метода обучения. Такие нейронные сети обычно выполняют задачи сжатия информации, снижения размерности или кластеризации.

Итак, в данной работе нас интересует задача кластеризации, которая решается способом обучения нейронной сети без учителя. Для метода обучения без учителя есть несколько особенностей – во-первых это возможность использования несопоставимо больших объемов данные, поскольку их не нужно размечать данные для обработки, что является плюсом, т.к. РЛИ поступает на радиолокатор в больших объемах и человек-оператор не всегда в силах разобрать информацию. Во-вторых, есть и минусы – это неясность в выборе измерения качества метода и отсутствие прямолинейных и интуитивно понятных метрик, как в задачах обучения с учителем.

## *Кластерный анализ*

Для начала дадим определение понятию кластеризация. Кластеризация или кластерный анализ — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы оказываются объекты с подобными параметрами, а объекты разных группы имеют максимальные отличия друг от друга. Существенное отличие задачи кластеризации от других задач, решаемых нейронными сетями, например, классификации, состоит в том, что перечень групп четко не заданы и требуется глубокий анализ входной базы данных и подбор параметров в процессе работы алгоритма кластеризации.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. Перед запуском алгоритма чаще всего требуется либо *нормализовать* данные, либо *стандартизировать*. Нормализация предполагает замену номинальных признаков так, чтобы каждых из них находился в диапазоне от нуля до единицы. Стандартизация же подразумевает предобработку данных, после которой каждый признак имеет среднее значение равное нулю и среднее отклонение равное единице. В итоге нормализация или стандартизация дает равную важность переменным во время кластеризации.
3. Вычисление значений *меры сходства* между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

### *Меры расстояний*

Требуется ответить на вопрос как оценивать качественно работу алгоритма кластеризации. Как было сказано ранее задача оценивания качества кластеризации является более сложной по сравнению с оценкой качества классификации. Во-первых, такие оценки не должны зависеть от самих значений меток, а только от самого разбиения выборки. Во-вторых, не всегда известны истинные метки объектов, поэтому также нужны оценки, позволяющие оценить качество кластеризации, используя только неразмеченную выборку.

Выделяют *внешние* и *внутренние* метрики качества. Внешние используют информацию об истинном разбиении на кластеры, в то время как внутренние метрики не используют никакой внешней информации и оценивают качество кластеризации, основываясь только на наборе данных. Оптимальное число кластеров обычно определяют с использованием внутренних метрик.

Приведем меры для измерения расстояний, наиболее часто встречающихся в кластерном анализе.

* **Евклидово расстояние**

Наиболее распространенная функция для определения расстояния. Представляет собой геометрическое расстояние в многомерном пространстве между двумя точками:

|  |  |
| --- | --- |
|  |  |

Где и точки в -мерном пространстве.

* **Квадрат евклидова расстояния**

Применяется для придания большего веса более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:

|  |  |
| --- | --- |
|  |  |

* **Расстояние городских кварталов (манхэттенское расстояние)**

Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния:

|  |  |
| --- | --- |
|  |  |

* **Расстояние Чебышева**

Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

|  |  |
| --- | --- |
|  |  |

* **Степенное расстояние**

Применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по следующей формуле:

|  |  |
| --- | --- |
|  |  |

где и – параметры, определяемые пользователем. Параметр ответственен за постепенное взвешивание разностей по отдельным координатам, параметр r ответственен за прогрессивное взвешивание больших расстояний между объектами. Если оба параметра – и — равны двум, то это расстояние совпадает с расстоянием Евклида.

В итоге выбор метрики полностью лежит на исследователе, поскольку результаты кластеризации могут существенно отличаться при использовании различных мер.

### *Обзор алгоритмов кластеризации*

Приведем обзор различных алгоритмов кластеризации, чтобы решить, какой из алгоритмов лучше всего подойдет для решения поставленной задачи.

* **Иерархическая кластеризация**

Иерархическая кластеризация группирует данные в различных масштабах, создавая дерево кластеров или дендрограмму. Дерево – это не единственный набор кластеров, а многоуровневая иерархия, в которой кластеры на одном уровне объединяются в кластеры на следующем. Это позволяет решить какой уровень кластеризации подходит для решения поставленной задачи.

Среди алгоритмов иерархической кластеризации выделяют два основных типа: восходящие и нисходящие алгоритмы. Нисходящие алгоритмы работают по принципу «сверху-вниз»: в начале все объекты помещаются в один кластер, который затем разбивается на все более мелкие кластеры. Восходящие же алгоритмы работают по обратному алгоритму «снизу-вверх»: в начале работы каждый объект помещается в отдельный кластер, а затем объединяются во все более крупные кластеры, пока все объекты выборки не будут содержатся в одном кластере. В итоге строится система вложенных разбиений.

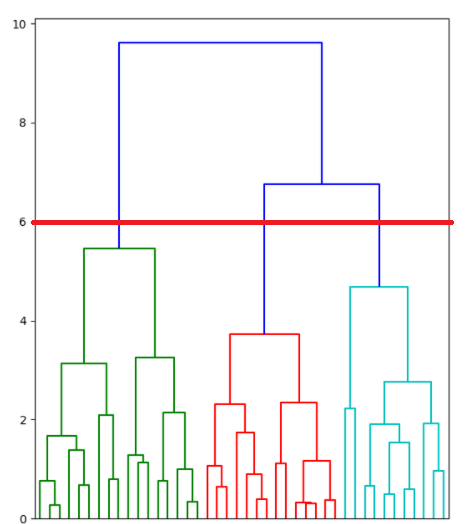


Рис. 3‑1. Пример иерархической кластеризации – дендрограмма. Уровнями можно задать количество кластеров разбиения.

* **Метод k-средних**

Данный метод кластеризации считается наиболее простым, но в то же время недостаточно точным. Суть данного алгоритма в следующем: метод разбивает множество элементов векторного пространства на заранее известное число кластеров k. Алгоритм стремится минимизировать среднеквадратическое отклонения на точках каждого кластера. На каждой итерации перевычисляется центр масс для каждого кластера, полученного на предыдущей итерации, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения внутрикластерного расстояния. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, а на каждом шаге суммарное квадратичное отклонение уменьшается, поэтому зацикливание невозможно.

* **Смешанная гауссовская модель**

Смешанная гауссовская модель формирует кластеры по принципу суперпозиции многомерных гауссовских законов распределения плотности вероятности. Для каждого наблюдения данный метод определяет апостериорные плотности вероятности, которые указывают, что наблюдение имеет некоторую вероятность принадлежности к каждому кластеру.

По смешанной гауссовской модели можно выполнить *hard-кластеризацию* (жесткая кластеризация), которая гарантирует высокую точность группировки, где все запросы привязываются друг к другу и в один кластер попадают только на сто процентов совместимые запросы, путем выбора компонента, который максимизирует апостериорную вероятность. Hard-кластеризация определяет точку из данных только в один кластер. И в итоге кластер образуется там, где наибольшая апостериорная вероятность.

Кроме hard-кластеризации можно использовать смешанную гауссовскую модель для так называемой *soft-кластеризации* (нечетной кластеризации). При данном методе присваивается вероятность попадания точки данных для каждого кластера, и эта вероятность указывает на силу связи точки данных с кластером. В отличие от методов hard-кластеризации, методы soft-кластеризации являются гибкими, поскольку они могут назначать точку данных нескольким кластерам.

* **Спектральная кластеризация**

Спектральная кластеризация использует спектр (собственных значений) матрицы сходства данных для осуществления снижения размерности перед кластеризацией в пространство меньших размерностей. Матрица сходства подается в качестве входа и состоит из количественных оценок относительно схожести каждой пары точек в данных. В низкой размерности данные в кластерах разделяются более широко, позволяя использовать далее алгоритмы кластеризации, например, k-средних. Низкая размерность данных основана на значимых собственных векторах матрицы Кирхгофа. Данная матрица является одним из способов представить график подобия, который отображает локальные отношения между точками данных, как неориентированный граф.

* **DBSCAN**

DBSCAN (Density-based spatial clustering of applications with noise, плотностной алгоритм пространственной кластеризации с присутствием шума) является основанным на плотности алгоритмом, который идентифицирует кластеры произвольной формы и выбросы (шум) в данных. Во время кластеризации DBSCAN определяет точки, которые не принадлежат ни одному из кластеров и относит к шуму. Для этого метода не требуется предварительное знание количества кластеров.

Рассмотрев данные методы кластеризации составим таблицу основных характеристик этих методов (таблица 1).

Таблица 3.3‑1. Характеристики основных методов кластеризации

|  |  |  |  |
| --- | --- | --- | --- |
| **Метод кластеризации** | **Описание алгоритма** | **Входные данные** | **Требование конкретного количества кластеров** |
| Иерархическая кластеризация | Расстояние между объектами | Попарные расстояния между наблюдениями | Нет |
| K-средних | Расстояние между объектами и центроидами | Фактические наблюдения | Да |
| Смешанные гауссовские модель | Смесь нормальных распределений | Фактические наблюдения | Да |
| Спектральная кластеризация | График, представляющий связи между точками данных | Фактические наблюдения или матрица подобия | Да, но алгоритм также обеспечивает способ оценить количество кластеров |
| DBSCAN | Плотность областей в данных | Физические наблюдения или попарные расстояния между наблюдениями | Нет |

Требуется определить какой метод кластеризации использовать для решения поставленной задачи. Напомним, что решается задача проведения группирования отметок в кластеры [6]. В свою очередь количество кластеров заранее не известно, поэтому подойдут алгоритмы, которые не учитывают количество кластеров, как входной параметр. Также радиолокационные отметки имеют зашумленные данные, поэтому нужно найти такой алгоритм, который бы учитывал случайные выбросы. Тогда учитывая все высказывания подходит алгоритм DBSCAN. Этим алгоритмом в данной работе будет реализована кластеризация отметок.

## *Описание алгоритма DBSCAN*

Имея дело с пространственными кластерами различной плотности, размера и формы, может быть сложно обнаружить группу точек. Задача может быть еще более сложной, если данные содержат шум и выбросы. Для работы с большими пространственными базами данных обычно применяется алгоритм DBSCAN [15]. Основных причины использования алгоритма:

1. Требует минимальных знаний предметной области.
2. Он может обнаружить кластеры произвольной формы.
3. Эффективен для большой базы данных, то есть размер выборки превышает несколько тысяч.

Рассматривая данные причины убеждаемся, что с помощью данного алгоритма возможно решить поставленную задачу.

Основная концепция алгоритма DBSCAN состоит в том, чтобы найти области высокой плотности, которые отделены друг от друга областями низкой плотности. И чтобы правильно применить этот алгоритм кластеризации для решения поставленной задачи, требуется подробнее рассмотреть его параметры и характеристики. На вход метод просит матрицу близости и два параметра – радиус -окрестности и -минимальное количество соседей.

Для определения и введем несколько определений. Пусть задана некоторая симметричная функция расстояния и константы и . Тогда:

1. Назовем область , для которой , где – окрестность объекта
2. Центральным объектом или ядерным объектом степени называется объект, -окрестность которого содержит не менее объектов:
3. Объект непосредственно плотно-достижим из объекта , если и – корневой объект
4. Объект p плотно-достижим из объекта , если , такие что непосредственно плотно-достижим из

Следуя определению плотной области, точка может быть классифицирована как основная точка если . Центральные точки, как следует из названия, обычно находятся внутри кластера. Пограничная точка (достижимая по плотности точка) имеет меньше, чем в своей области*,*  но лежит в окрестности другой центральной точки. Шум (выпадающая точка) – это любая точка данных, которая не является ни основной, ни пограничной.

Достижимость не является симметричным отношением, поскольку, по определению, никакая точка не может быть достигнута из неосновной точки, независимо от расстояния (так что неосновная точка может быть достижимой, но ничто не может быть достигнуто из неё). Поэтому дальнейшее понятие *связности* необходимо для формального определения области кластеров, найденных алгоритмом DBSCAN. Две точки  и *связаны по плотности*, если имеется точка , такая что и , и  достижимы из . Связность по плотности *является* симметричным отношением.

Тогда кластер удовлетворяет двум свойствам:

1. Все точки в кластере попарно связны по плотности.
2. Если точка достижима по плотности из какой-то точки кластера, она также принадлежит кластеру.

Этапы работы алгоритма DBSCAN.

Выше приведены определения и теперь можно описать шаги алгоритма DBSCAN:

1. Алгоритм начинается с произвольной точки, которая ещё не была просмотрена.
2. Выбирается -окрестность выбранной точки и, если она содержит точек, то начинается формирование кластера, в противном случае точка помечается как шум. Эта точка может быть позже найдена в -окрестности другой точки и, таким образом, может стать частью кластера.
3. Если точка найдена как основная точка, то точки в -окрестности также являются частью этого кластера. Таким образом, все точки найденные в -окрестность, добавляются вместе с их собственной -окрестностью, если они также являются основными точками.
4. Вышеописанный процесс продолжается, пока не будет найден связный по плотности кластер.
5. Процесс возобновляется. Выбирается и обрабатывается новая непосещённая точка, что ведёт к обнаружению следующего кластера или шума.

Введя определения и описание работы алгоритма DBSCAN перейдем к недостаткам. Приведем два самых больших недостатков алгоритма:

1. Если в базе данных есть точки, которые образуют кластеры различной плотности, то с помощью данного алгоритма не удается хорошо раскластеризовать точки, поскольку кластеризация зависит от параметров и , они не могут быть выбраны отдельно для всех кластеров.
2. Если данные и функции не так хорошо понятны специалисту в области, то может быть настроить параметры и . Тогда требуется сравнение нескольких итераций с различными значениями и .

В итоге сложность применения алгоритма DBSCAN – это выбор и .

Существуют эвристики для подбора и . Чаще всего применяется такой метод и его вариации:

1. Выбирается . Используются значения больше или равные порядку размерности входных данных. Чем более неоднородный датасет, тем соответственно больше уровень шума и тогда следует взять большим.
2. Для оценки значения требуется вычислить среднее расстояние по для каждой точки входных данных. Далее сортируются полученные значения по возрастанию и строится график.
3. Построится резко возрастающий график. Значение выбирается в полосе, где происходит сильный перегиб кривой, это область, где точки начинают затухать в область выбросов [15].

Итог. DBSCAN отлично работает на плотных, хорошо отделенных друг от друга кластерах. При этом их форма совершенно не важна. Алгоритм отлично обнаруживает кластеры малой размерности. Успешно применяется для большого датасета , причем сложность элементов датасета значения не имеет. Количество элементов в кластере может варьироваться, количество выбросов значения не имеет, если они рассеяны по большому объему, а также количество кластеров значения не имеет.

# Кластерный анализ

Чтобы решить поставленную задачу для начала требуется дать определение понятию кластеризация. *Кластеризация или кластерный анализ* — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы оказываются объекты с подобными параметрами, а объекты разных групп имеют максимальные отличия друг от друга.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. Перед запуском алгоритма чаще всего требуется *нормализовать* данные или по-другому *стандартизировать* данные. Нормализация предполагает замену номинальных признаков так, чтобы каждых из них находился в диапазоне от нуля до единицы. Стандартизация же подразумевает предобработку данных, после которой каждый признак имеет среднее значение равное нулю и среднеквадратичное отклонение (СКО) равное единице. В итоге нормализация или стандартизация дает равную важность переменным во время кластеризации.
3. Вычисление значений *меры сходства* между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

Кроме того, кластерный анализ является общим методом машинного обучения без учителя для поиска шаблонов в наборе данных. Обучение без учителя используется, чтобы формировать группы из наборов данных, которые обычно заранее не размечены. Например, используются кластерный анализ для исследовательского анализа данных, чтобы найти спрятанные шаблоны или группы объектов в непомеченных входных данных.

Кластерный анализ создает группы или по-другому *кластеры*, данных. Объекты, которые принадлежат тому же кластеру, похожи друг на друга и отличны от объектов, принадлежащих другим кластерам. Чтобы определить сколько таких "подобных" и "отличных", используют меру по несходству или по-другому метрику расстояний. Кроме того, в зависимости от имеющегося набора данных требуется использовать масштабирование (или стандартизацию) переменных, чтобы дать равную оценку во время кластеризации.

Категоризация точек данных на основе их расстояния до соседних точек в этом же наборе данных является простым и эффективным способом кластеризации точек.

Решение данной задачи будет проводится в среде Matlab, поэтому следует привести описание функций, с которыми будет проводиться работа.

В среде Matlab попарное расстояние между точками набора данных находится с помощью функции pdist2. Синтаксис данной функции:

D = pdist2(X,Y,Distance)- возвращает расстояние между каждой парой наблюдений в и  с используемой метрикой, заданной в Distance.

D = pdist2(X,Y,Distance,DistParameter) - возвращает расстояние с помощью метрики, заданной Distance и DistParameter. Можно задать DistParameter только, когда Distance 'seuclidean',

 'minkowski', или 'mahalanobis'.

D = pdist2(\_\_\_,Name,Value)- задает дополнительную опцию с помощью одного из аргументов пары "имя-значение" ['Smallest'](https://docs.exponenta.ru/stats/pdist2.html#mw_ff5fa47d-3482-45db-86f4-1448503cf7b8) или

['Largest'](https://docs.exponenta.ru/stats/pdist2.html#mw_96fe0732-1dc9-4494-9e9f-6383d6ea45ca) в дополнение к любому из аргументов в предыдущих синтаксисах.

Например,

D = pdist2(X,Y,Distance,'Smallest’,K) - вычисляет расстояние с помощью метрики, заданной [Distance](https://docs.exponenta.ru/stats/pdist2.html#d123e625005) и возвращает  наименьшие попарные расстояния до наблюдений в  для каждого наблюдения в  в порядке возрастания.

D = pdist2(X,Y,Distance,'Largest’,K вычисляет расстояние с помощью метрики, заданной Distance и [DistParameter](https://docs.exponenta.ru/stats/pdist2.html#bsc_12z_sep_shared-DistParameter) и возвращает самые большие попарные расстояния в порядке убывания.

[D,I] = pdist2(\_\_\_,Name,Value) - также возвращает матричный . Матричный  содержит индексы наблюдений в  соответствие расстояниям в .

Требуется определить такое понятие как метрика расстояний. Метрика расстояний является функцией, которая задает расстояние между двумя наблюдениями. В среде Matlab функция попарного-расстояния pdist2 поддерживает следующие метрики расстояния: Евклидово расстояние, квадратичное Евклидово расстояние, стандартизированное Евклидово расстояние, расстояние Махаланобиса, расстояние городского квартала (манхэттенское расстояние), расстояние Минковского, расстояние Чебышева, расстояние косинуса, корреляционное расстояние, расстояние Хемминга, расстояние Жаккара и расстояние Спирмена. Описание данных метрик сведено в Таблицу 1 и продолжение в Таблице 2.

Таблица 1. Метрики расстояний.

|  |  |  |
| --- | --- | --- |
| Значение | Формула | Описание |
| 'euclidean' | Где и точки в -мерном пространстве | Значение по умолчанию. Наиболее распространенная функция для определения расстояния. Представляет собой геометрическое расстояние в многомерном пространстве между двумя точками. Евклидово расстояние является особым случаем расстояния Минковкого, где |
| 'squaredeuclidean' |  | Применяется для придания большего вес более отдаленных друг от друга объектов. |
| 'seuclidean' |  | Каждое координатное различие между наблюдениями масштабируется путем деления на соответствующий элемент стандартного отклонения, |
| 'mahalanobis' |  | Предполагает, что точки множества эллипсоидально (частный случай – сферически) распределены вокруг центра масс – расстояние безразмерно и масштабно-инвариантно.  Ковариация – это численное выражение свойства ковариантности двух признаков точек. Свойство ковариантности означает, что признаки имеют тенденцию изменяться совместно. – ковариационная матрица |
| 'cityblock' |  | Это расстояние является средним разностей по координатам. Для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Расстояние городского квадрата является особым случаем расстояния Минковского, где |
| 'minkowski' |  | Параметрическая метрика на евклидовом пространстве, которую можно рассматривать как обобщение евклидова расстояния и расстояния городских кварталов. |
| 'chebychev' |  | Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние городского квадрата является особым случаем расстояния Минковского, где |
| 'cosine' |  | Метод сходства между двумя ненулевыми векторами пространства данных. Определен как косинус угла между ними, которых также совпадает с внутренним произведением тех же векторов, нормализованных к обоим, имеющим длину 1. |

Таблица 2. Продолжение таблицы 1. Метрики расстояний.

|  |  |
| --- | --- |
| Функция | Описание |
| 'correlation' | Ещё называют скорректированным косинусным сходством. Использование данного метода компенсирует недостаток предыдущего метода путем вычитания среднего значения соответствующе точки набора. |
| 'hamming' | Число позиций, в которых соответствующие точки двух наборов одинаковой длины различны. |
| 'jaccard' | Используется для калибрования сходства и разнообразия из образцов наборов. Основан на пересечении и объединении данных |
| 'spearman' | Является методом ранговой корреляции и используется для выявления и оценки тесноты связи между двумя рядами сопоставляемых количественных показателей. |

# 4. Алгоритмы кластеризации

4-1. Иерархическая кластеризация

Иерархическая кластеризация группирует данные в различных масштабах, создавая дерево кластеров или дендрограмму (*dendrogram*). Дерево не является ни одним из наборов кластеров, а является скорее многоуровневой иерархией, в которой кластеры на одном уровне объединяются в кластеры на следующем. Эта многоуровневая иерархия позволяет выбирать требуемые уровень или шкалу кластеризации, которые наиболее подходят для решения поставленной задачи. Иерархическая кластеризация присваивает каждую точку данных набора к кластеру.

Приведем описание встроенных функций Matlab. Использование [clusterdata](https://docs.exponenta.ru/stats/clusterdata.html) позволяет выполнить иерархическую кластеризацию на неразмеченных входных данных. Функция clusterdata соединяет в себе следующие функции: [pdist](https://docs.exponenta.ru/stats/pdist.html), [linkage](https://docs.exponenta.ru/stats/linkage.html), и [cluster](https://docs.exponenta.ru/stats/cluster.html), которые можно использовать отдельно для более детального анализа иерархической кластеризации. С помощью функции [dendrogram](https://docs.exponenta.ru/stats/dendrogram.html) можно построить график кластерного дерева.

***Описание алгоритма***

Чтобы реализовать иерархическую кластеризацию на наборе данных с помощью функций статистики и машинного обучения среды Matlab Statistics and Machine Learning Toolbox, требуется выполнить следующие шаги:

1. **Найти подобие или несходство между каждой парой объектов во входном наборе данных.** На этом шаге вычисляется расстояние между точками данных с помощью [pdist](https://docs.exponenta.ru/stats/pdist.html) функция. Данная функция поддерживает много различных способов нахождения расстояния, а именно, можно изменять метрику расстояния, для получения лучшего результата.
2. **Построить иерархическое кластерное дерево.** На этом шаге требуется соединить пары точек, которые находятся в непосредственной близости с помощью функции [linkage](https://docs.exponenta.ru/stats/linkage.html). Эта функция использует информацию о расстоянии, сгенерированную на шаге, чтобы определить близость объектов друг другу. Когда объекты соединяются в бинарные кластеры, недавно сформированные кластеры группируются в большие кластеры, пока иерархическое дерево не сформируется.
3. **Определить, где сократить иерархическое дерево в кластеры.** На этом шаге используется функция [cluster](https://docs.exponenta.ru/stats/cluster.html). Чтобы определить количество кластеров нужно обрезать иерархическое дерево по требуемому уровню, тогда нижняя часть иерархического дерева и все точки ниже этого уровня присваиваются одному кластеру. Данный метод позволяет создать кластеры путем обнаружения естественных групп в иерархическом дереве или путем сокращения иерархического дерева по произвольно выбранному уровню.

4-2. Метод k- средних значений

Данный метод кластеризации считается наиболее простым, но в то же время ему не хватает точности. Суть данного алгоритма в следующем: метод разбивает множество элементов векторного пространства данных на заранее известное число кластеров *k*. Т.е. для данного алгоритма число кластеров *k* является входным аргументом*.* Работа алгоритма заключается в том, что алгоритм стремиться минимизировать среднеквадратическое отклонение (или другую выбранную меру расстояний) для точек каждого кластера. На каждой итерации вычисляется заново центр масс для каждого кластера, полученного на предыдущей итерации, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения внутрикластерного расстояния. Происходит это за конечное число итераций, так как количество возможных разбиений конечного множества также конечно, а на каждом шаге суммарное квадратичное отклонение уменьшается, поэтому зацикливание невозможно.

Функции, используемые в среде Matlab: kmeans – *k-* кластеризация. Есть ещё виды данной кластеризации, а именно: kmedoids – *k-* *medoids* кластеризация; mahal – расстояния Махаланобиса. Например, расстояние Махаланобиса учитывает корреляцию между переменными и инвариантно к масштабу.

В отличие от иерархической кластеризации, этот метод работает с фактическими наблюдениями (т.е. с входными данными напрямую, а не с мерой по несходству) и создается один уровень кластеров. Поэтому кластеризация k- средних значений чаще подходит для больших объемов данных, чем иерархическая кластеризация.

Функция [kmeans](https://docs.exponenta.ru/stats/kmeans.html) получает данные и возвращает индекс кластера, которому присваивает каждое наблюдение, находит кластеры, в которых объекты максимально близки друг к другу по параметрам, и максимально далеки от объектов в других кластерах.

Каждый кластер в методе k- средних значений состоит из объектов члена и центроида (или по-обычному центра). В каждом кластере, kmeans минимизирует сумму расстояний между центроидом и всеми объектами члена кластера.

Можно варьировать аргументы пары "имя-значение" доступных для kmeans; например, можно задать начальные значения кластерных центроидов и максимальное количество итераций для алгоритма.

4-3. Основанная на плотности пространственная кластеризация с шумом (DBSCAN)

DBSCAN (*Density-based spatial clustering of applications with noise*, плотностной алгоритм пространственной кластеризации с присутствием шума) является основанным на плотности алгоритмом кластеризации, который идентифицирует кластеры произвольной формы и выбросы (шум) в данных. Во время кластеризации алгоритм DBSCAN определяет точки, которые не принадлежат ни одному из кластеров и относит к шуму. В отличие от метода k- средних значений, DBSCAN не требует предварительных знаний количества кластеров.

Функция в среде Matlab. [dbscan](https://docs.exponenta.ru/stats/dbscan.html) - основанная на плотности пространственная кластеризация приложений с шумом (DBSCAN). Точка набора данных, которая будет присвоена кластеру, лежит условие, что окружность вокруг этой точки радиуса эпсилон ([epsilon](https://docs.exponenta.ru/stats/dbscan.html#mw_8e9da8d6-2310-43e6-83ba-848b3f135fb7)) содержит, по крайней мере, минимальное количество соседей ([minpts](https://docs.exponenta.ru/stats/dbscan.html#mw_8f71b08a-185b-49d9-b224-3733d73ca4bd)). Или, точка может лечь в окружении эпсилона другой точки, которая удовлетворяет условиям epsilon и minpts.

Алгоритм DBSCAN идентифицирует три вида точек:

1. Базовая точка — точка в кластере, которая имеет, по крайней мере, minpts соседей в epsilon окрестности.
2. Пограничная точка — точка в кластере, которая имеет меньше, чем minpts соседей в epsilon окрестности.
3. Шумовая точка — выброс, который не принадлежит никакому кластеру.

***Описание алгоритма***

Для заданных значений радиуса [epsilon](https://docs.exponenta.ru/stats/dbscan.html#mw_8e9da8d6-2310-43e6-83ba-848b3f135fb7) и минимального количества соседей [minpts](https://docs.exponenta.ru/stats/dbscan.html#mw_8f71b08a-185b-49d9-b224-3733d73ca4bd) требуемых для базовой точки, функция [dbscan](https://docs.exponenta.ru/stats/dbscan.html) реализует метод DBSCAN можно следующим образом:

1. Алгоритм начинается с произвольной точки, которая ещё не была просмотрена.
2. Выбирается epsilon окрестность этой точки и, если она содержит minpts точек, то начинается формирование кластера, в противном случае точка помечается как шум. Эта точка может быть позже найдена в epsilon окрестности другой точки и, таким образом, может стать частью кластера.
3. Если точка найдена как основная точка, то точки в epsilon-окрестности также являются частью этого кластера. Таким образом, все точки, найденные в epsilon окрестность, добавляются вместе с их собственной epsilon окрестностью, если они также являются основными точками.
4. Вышеописанный процесс продолжается, пока не будет найден связанный по плотности кластер.
5. Процесс возобновляется. Выбирается и обрабатывается новая не рассмотренная точка, что ведёт к обнаружению следующего кластера или шума.

4-4. Смешанная гауссовская модель

Метод смешанной гауссовской модели (GMM) формирует кластеры по принципу суперпозиции многомерных гауссовских законов распределения плотности вероятности. Для каждого наблюдения данный метод определяет апостериорные плотности вероятности, которые указывают, что наблюдение имеет некоторую вероятность принадлежности к каждому кластеру.

По GMM можно выполнить *hard-кластеризацию* (жесткую кластеризацию), которая гарантирует высокую точность группировки, где все запросы привязываются друг к другу и в один кластер попадают только на сто процентов совместимые запросы, путем выбора компонента, который максимизирует апостериорную вероятность. *Hard*-кластеризация определяет точку из данных только в один кластер. И в итоге кластер образуется там, где существует наибольшая апостериорная вероятность.

Кроме *hard*-кластеризации можно использовать смешанную гауссовскую модель *soft-кластеризации* (нечетной или мягкой кластеризации). При данном методе присваивается вероятность попадания точки данных для каждого кластера, и эта вероятность указывает на силу связи точки данных с кластером. В отличие от методов *hard*-кластеризации, методы *soft*-кластеризации являются гибкими, поскольку они могут назначить точку данных нескольким кластерам.

GMM может быть более соответствующим методом, чем метод k- средних значений, когда кластеры имеют различные размеры и различные структуры корреляции в них.

***Описание алгоритма***

Смешанные гауссовские модели (GMMs) присваивают каждое наблюдение кластеру путем максимизации апостериорной вероятности, что точка данных принадлежит своему присвоенному кластеру. Создается объект GMM путем подбирания модели к данным ([fitgmdist](https://docs.exponenta.ru/stats/fitgmdist.html)) или настройкой значений параметров ([gmdistribution](https://docs.exponenta.ru/stats/gmdistribution.html)). Используются следующие функции в среде Matlab.

[GMModel](https://docs.exponenta.ru/stats/fitgmdist.html#bt8x9gq-GMModel) = fitgmdist([X](https://docs.exponenta.ru/stats/fitgmdist.html#bt8x9gq-X),[k](https://docs.exponenta.ru/stats/fitgmdist.html#bt8x9gq-k)) возвращает Гауссову модель распределения смеси с *k* компонентами адаптированными к данным *X*.

[GMModel](https://docs.exponenta.ru/stats/fitgmdist.html#bt8x9gq-GMModel) = fitgmdist([X](https://docs.exponenta.ru/stats/fitgmdist.html#bt8x9gq-X),[k](https://docs.exponenta.ru/stats/fitgmdist.html#bt8x9gq-k),[Name,Value](https://docs.exponenta.ru/stats/fitgmdist.html#namevaluepairarguments)) возвращает Гауссову модель распределения смеси с дополнительными опциями, заданными парой аргументов Name,Value.

gmdistribution объектно-ориентированная память Гауссово распределение смеси, которая является многомерным распределением, которое состоит из многомерных компонентов распределения Гаусса. Каждый компонент задан его средним значением и ковариацией. Смесь задана вектором из смешивания пропорций, где каждая пропорция смешивания представляет часть данных, описанных соответствующим компонентом.

gm=gmdistribution([mu](https://docs.exponenta.ru/stats/gmdistribution.html#mw_d549c5a8-3cd1-4514-91b6-a01da9651f83),[sigma](https://docs.exponenta.ru/stats/gmdistribution.html#mw_455d92f0-3b14-4484-9592-23ca36344e34)) создает gmdistribution  объект модели с помощью заданных средних значений mu и ковариации (СКО) sigma с равными пропорциями смешивания.

gm=gmdistribution([mu](https://docs.exponenta.ru/stats/gmdistribution.html#mw_d549c5a8-3cd1-4514-91b6-a01da9651f83),[sigma](https://docs.exponenta.ru/stats/gmdistribution.html#mw_455d92f0-3b14-4484-9592-23ca36344e34),[p](https://docs.exponenta.ru/stats/gmdistribution.html#mw_5518832d-e041-46fd-a8b6-4615e85d498a)) задает смесительные пропорции многомерных компонентов Распределения Гаусса.

Затем используются объектные функции, чтобы выполнить кластерный анализ ([cluster](https://docs.exponenta.ru/stats/gmdistribution.cluster.html), [posterior](https://docs.exponenta.ru/stats/gmdistribution.posterior.html), [mahal](https://docs.exponenta.ru/stats/gmdistribution.mahal.html)), оценивается модель ([cdf](https://docs.exponenta.ru/stats/gmdistribution.cdf.html), [pdf](https://docs.exponenta.ru/stats/gmdistribution.pdf.html)), и генерируются случайные варьируемые величины ([random](https://docs.exponenta.ru/stats/gmdistribution.random.html)).

4-5. Спектральная кластеризация

Спектральная кластеризация использует спектр собственных значений матрицы сходства данных для осуществления снижения размерности перед кластеризацией в пространство меньших размерностей. Матрица сходства подается в качестве входа и состоит из количественных оценок относительно схожести каждой пары точек в данных. В низкой размерности данные в кластерах разделяются более широко, позволяя использовать далее алгоритмы кластеризации, например, метод *k*-средних значений. Низкая размерность данных основана на значимых собственных векторах матрицы Лапласа. Данная матрица является одним из способов представить график подобия, который отображает локальные отношения между точками данных, как неориентированный граф.

Использование функции Matlab [spectralcluster](https://docs.exponenta.ru/stats/spectralcluster.html) позволяет выполнить спектральную кластеризацию на матрице входных данных или на матрице подобия. Функция [spectralcluster](https://docs.exponenta.ru/stats/spectralcluster.html) требует изначально заданного числа кластеров *k*.

Однако можно проверить, что оценка для числа кластеров k правильна при помощи одного из следующих методов: посчитать количество нулевых собственных значений Матрицы Лапласа. Кратность нулевых собственных значений является индикатором количества кластеров в исходных данных.

***Описание алгоритма***

Спектральная кластеризация является основанной на графике алгоритмом для нахождения k кластеров произвольной формы в данных.

С помощью функции spectralcluster кластеризация данных происходит следующим образом:

1. Для каждой точки данных в *X*, задается локальное окружение с помощью метода определения радиуса epsilon или с помощью метода самых близких соседей knn. Как задано в SimilarityGraph аргументом является пара "имя-значение". Затем находятся попарные расстояния  для всех точек  в этом окружении.
2. Преобразуются расстояния до мер по подобию с помощью преобразования ядра Матрица  является матрицей подобия, и

является масштабным коэффициентом для ядра. Для этого используется функция KernelScale аргументом является пара "имя-значение".

1. Вычисляется ненормированная Матрица Лапласа , нормированная Матрица Лапласа случайного обхода или нормированная симметричная Матрица Лапласа , в зависимости от значения LaplacianNormalization аргументом является пара "имя-значение".
2. Создается матрица , содержащая столбцы , где столбцы являются собственными векторами, которые соответствуют  самые маленькие собственные значения Матрицы Лапласа. При обработке каждой строки необходимо выполнить кластеризацию точки n, например, с помощью метода k-средних значений, ClusterMethod аргументом является пара "имя-значение".
3. Присваиваются исходные точки в наборе *X* к тем же кластерам, как и их соответствующие строки в V.

В данном разделе были рассмотрены основные алгоритмы кластеризации, которые есть в разделе статистики и машинного обучения в среде Matlab. Сведем характеристики данных алгоритмов в общую Таблицу 3.

Таблица 3. Характеристики основных методов кластеризации.

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм кластеризации** | **Описание алгоритма**  **кластеризации** | **Входные данные** | **Требование конкретного количества кластеров** |
| Иерархическая кластеризация | Расстояние между объектами | Попарные расстояния между наблюдениями | Нет |
| K-средних | Расстояние между объектами и центроидами | Фактические наблюдения | Да |
| Смешанные гауссовские модель | Смесь нормальных распределений | Фактические наблюдения | Да |
| Спектральная кластеризация | График, представляющий связи между точками данных | Фактические наблюдения или матрица подобия | Да, но алгоритм также обеспечивает способ оценить количество кластеров |
| DBSCAN | Плотность областей в данных | Физические наблюдения или попарные расстояния между наблюдениями | Нет |

# Глава 1. Обзор существующих решений

Обзор существующих алгоритмов из различных статей (моя + Олега), которые указаны

В настоящее время растет разнообразие работ, где решается задача кластеризации радиолокационных импульсов с помощью различных методов. Например, распознавание динамических объектов в радиолокационном пространстве с помощью алгоритмов кластеризации k-means и нечеткой кластеризации c-means по навигационным параметрам [3], модель масштабной смеси нормальных распределений для классификации и кластеризации радиолокационных излучателей [4]*,* метод автоматической классификации с использованием сети вычисления p-значений для проверки гипотез о типах излучателей, где алгоритм кластеризации основывается на методе обучаемой кластеризации векторов [5]*,* алгоритмы кластеризации с радарным сканирование на основе плотности [6]применяют и аппарат нейронных сетей, а именно глубокие рекуррентные нейронные сети (РНС) для классификации и грубой кластеризации различных групп импульсов иерархически в отношении их последовательных структур [7]***.***

# глава 2. синтез алгоритма комплексирования

Анализирую рассмотренные в Главе 1 примеры алгоритмов обработки для систем локальной навигации можно прийти к выводу, что самыми эффективными являются комплексные алгоритмы.

В данной работе предполагается синтезировать алгоритм комплексирования приемопередатчика СШП ЛНС и датчика угловой скорости смартфона.

## 2.1 Постановка задачи

Пусть имеется потребитель, позиционирующийся внутри помещения с помощью локальной навигационной системы, включающий в себя сверхширокополосный радиомодуль, координаты которого требуется определять, датчик угловых скоростей (оба измерителя встроены в смартфон потребителя) и систему опорных приемо-передающих СШП маяков, размещенных по периметру помещения.

Необходимо синтезировать алгоритм комплексной обработки радио- и инерциальных измерений – комплексный фильтр Калмана. Далее необходимо оценить работоспособность синтезированного фильтра: качество работы фильтра будем оценивать путем сравнения предельных ошибок фильтрации, полученных из оценок матрицы дисперсий ошибок, полученных в имитационном моделировании и экспериментально, и рассчитанных аналитически путем решения уравнения Рикатти.

Для синтеза комплексного фильтра будем использовать теорию оптимальной фильтрации [17-19]. В соответствии с этой теорией необходимо задать векторный марковский процесс  и описать его изменение во времени с помощью матричного уравнения. В общем случае, наблюдениями могут являться все или некоторые из компонент , а также функционально связанные с ними дополнительные физические величины. Наблюдение в случае с комплексным фильтром включает наблюдения от СШП-системы  и наблюдения от датчика угловых скоростей  [21].

Известно, что темп получения измерений от СШП радио- и инерциальной системы различен. Обычно для удобства расчетов вводится шкала дискретного времени с двойной нумерацией: в более редкие моменты времени поступают радиотехнические измерения и инерциальные, а в более частые моменты времени поступают только инерциальные измерения. При проектировании системы темпы измерений выбирают кратными, измерители тактируют от одного опорного генератора, а решение уравнений фильтрации привязывают к более “частой” шкале времени.

Учитывая, что в нашей системе потребитель – это человек, являющийся малодинамичным объектом, будем считать при моделировании, что угол рысканья меняется незначительно за период радиоизмерений. Таким образом, принимаем допущение, что в фильтр инерциальные и радиоизмерения поступают синхронизировано и с одним темпом. Также при моделировании не будем учитывать дрейфы нулей ДУС, погрешности масштабных коэффициентов и перекосы осей.

## 2.2 Синтез комплексного фильтра

Структурная схема синтезируемого фильтра представлена на рисунке 6. **-** измерения скорости угла курса, поступающий с ДУС; **-** измерения координат, поступающие из СШП ЛНС; – оценка вектора состояния на выходе фильтра Калмана.

Такой вариант комплексирования можно отнести к так называемым слабосвязанным схемам, так как приемопередатчик СШП ЛНС И ДУС вырабатывают независимые оценки вектора состояния, при этом между ними есть связующий блок – расширенный фильтр Калмана, который формируем комплексную оценку вектора состояния.

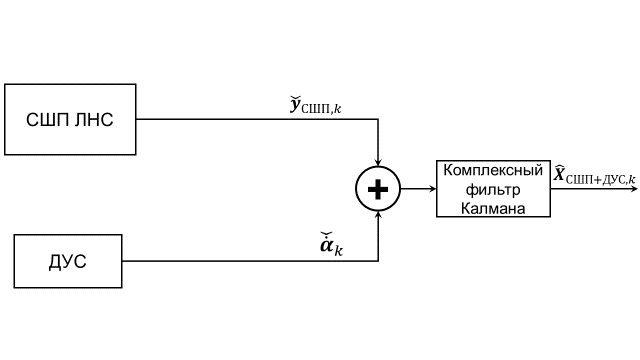


Рисунок 6 - Структурная схема синтезируемого комплексного фильтра

При синтезе комплексного алгоритма на основе фильтра Калмана наблюдениями являются радиоизмерения сверхширокополосной локальной навигационной системы и измерения датчика угловых скоростей в моменты времени :

|  |  |
| --- | --- |
| , |  |

В вектор состояния включим координаты потребителя, модуль вектора скорости потребителя, угол курса и скорость его изменения:

|  |  |
| --- | --- |
| , |  |

где  – координаты потребителя; – модуль вектора скорости потребителя; ,  – угол курса и его производная.

Любая математическая модель создается для описания какой-то реальной системы. Очевидно, что чем больше параметров учитывает модель, тем ближе она описывает действительность. Однако, большое количество параметров требует большого количества источников достоверной информации об этих параметрах. В случае математического моделирования фильтра Калмана, вектор состояния отвечает за связь всей модели с действительностью с точки зрения его элементов, которые и оцениваются фильтром. В данной задаче имеется ограниченный набор информации о параметрах движения потребителя. Во-первых, исходя из постановки задачи, от сверхширокополосной системы навигации поступают вторичные измерения, а именно координаты, то есть никакой первичной информации, например, о временах прихода сигналов метки на опорные маяки для нас не доступна в рамках поставленной задачи. Во-вторых, известно, что на борту смартфонов используются низкоклассные инерциальные МЭМС-датчики, а значит, их измерения тоже не будут отличаться высокой точностью. Так образом, нет смысла расширять вектор состояния за счет таких элементов как ошибки инерциальных датчиков различных типов или разностей дальностей между радиометками и опорными маяками, так как эта информация будет заведомо неточной и может негативно повлиять на качество работы фильтра. Поэтому, в рассмотренном векторе состояния фигурирует минимальное количество элементов, которые однозначно определяются через вектор измерений.

Опишем модель динамики потребителя:

|  |  |
| --- | --- |
| , |  |

где  – дискретный белый гауссовский шум с дисперсией ;  – дискретный белый гауссовский шум с дисперсией .

Таким образом, изменение вектора состояний можно записать как:

|  |  |
| --- | --- |
| , |  |

где  – нелинейная векторная функция, связывающая значения вектора состояния на *k* и *k-1* шаге;  – матрица формирующих шумов;  – вектор независимых белых гауссовых шумов с нулевыми математическими ожиданиями и единичными дисперсиями.

Вектор формирующих шумов выглядит следующим образом:

|  |  |
| --- | --- |
|  |  |

Матрица формирующих шумов записывается как показано ниже:

|  |  |
| --- | --- |
|  |  |

Алгоритм фильтрации состоит из двух этапов:

1. Шаг экстраполяции:

|  |  |
| --- | --- |
|  |  |
|  |  |

Производную нелинейной векторной функции, связывающей значения вектора состояния на текущем и предыдущим шагах, зададим исходя из уравнений динамики:

|  |  |
| --- | --- |
|  |  |

Матрица дисперсий формирующих шумов:

|  |  |
| --- | --- |
|  |  |

1. Шаг оценивания:

|  |  |
| --- | --- |
|  |  |

где  - матрица наблюдений:

|  |  |
| --- | --- |
|  |  |

 - матрица дисперсий шумов наблюдений:

|  |  |
| --- | --- |
|  |  |

Алгоритм (1) - (13) реализует комплексную обработку измерений СШП радио модуля смартфона и измерений ДУС смартфона в составе навигационной системы для позиционирования потребителя внутри помещений.

## 2.3 Аналитическая оценка ошибок фильтрации

Постановка задачи оценки точности фильтрации сводится к нахождению матрицы дисперсий ошибок оценок фильтрации **D** [19].

Изменение матрицы **D**(*t*) во времени описывается матричным нелинейным дифференциальным уравнением Рикатти:

|  |  |
| --- | --- |
|  |  |

Для решения данного уравнения опишем модель динамики потребителя в непрерывном времени и определим необходимые матрицы также в непрерывном времени:

|  |  |
| --- | --- |
|  |  |

Так как рассматриваемый алгоритм фильтра Калмана является нелинейным относительно функции ***f***, то уравнение Рикатти примет вид:

|  |  |
| --- | --- |
|  |  |

В связи с тем, что фильтруемые процессы и формирующие шумы являются стационарными случайными процессами, то дисперсии ошибок фильтрации достаточно определить в установившемся режиме, то есть при .

Тогда нелинейное дифференциальное уравнение преобразуется в матричное алгебраическое уравнение вида:

|  |  |
| --- | --- |
|  |  |

Матрица , описывающая изменение вектора состояния во времени, будет нелинейной:

|  |  |
| --- | --- |
|  |  |

Вектор  содержит 5 компонент, поэтому матрица дисперсий ошибок имеет размер 5×5. Представим ее в виде:

|  |  |
| --- | --- |
|  |  |

Остальные матрицы, необходимые для решения уравнения Рикатти будут постоянными скалярами, соответствующими матрицам (6), (10), (12), (13). С их учетом алгебраическое уравнение примет вид:

|  |  |
| --- | --- |
|  |  |

Тогда, с учетом симметричности матрицы дисперсии ошибок фильтрации получим систему нелинейных алгебраических уравнений:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

С точки зрения сложности модели, формирующий шум  для винеровского процесса  имеет малый порядок в рамках поставленной задачи навигации человека-потребителя, однако, сильно усложняет расчеты дисперсий ошибок фильтрации. Поэтому, на этапе расчета матрицы  в установившемся режиме будем считать этот шум нулевым [19]. Данное допущение не вызовет значительное влияние на конечный результат, так как известно, что при движении человека скорость изменения угла курса практически не имеет аномальных выбросов.

Тогда, система нелинейных алгебраических уравнений будет иметь следующий вид:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Решение упрощенной системы уравнений выглядит следующим образом:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Параметрические выражения (44) – (53) являются элементами матрицы дисперсий ошибок фильтрации. При имитационном моделировании и экспериментальном исследовании будем рассчитывать элементы главной диагонали этой матрицы при фиксированных СКО формирующих шумов, шумов наблюдения и фиксированных значениях угла курса.

# Глава 3. Моделирование комплексного фильтра

Для проверки работоспособности и эффективности синтезированного фильтра была смоделирована траектория движения потребителя, представленная на рисунке 7. Синим обозначена истинная (референсная) траектория, а черным – смоделированные измерения координат, полученные от СШП ЛНС.



Рисунок 7 – Модель траектории движения потребителя

Траектория длиной 15,7 м состоит из трех участков: первый участок длительностью 11,9 с, является режимом статики; второй участок длительностью 11,9 с представляет собой прямолинейное равномерное движение со средней скоростью 1,3 м/с; третий участок длительностью 6,1 с, также, как и первый, отражает режим статики.

Так как задача навигации поставлена для случая перемещения потребителя в определенном помещении, все расчеты координат будут производиться в локальной системе координат, связанной с СШП ЛНС.

При моделировании будем считать, что потребитель изначально смотрел в направлении своего движения и угол курса не изменялся по ходу его движения и пребывания в режимах статики. Это необходимо для сравнения предельных ошибок фильтрации, полученных в результате моделирования и с помощью аналитических параметрических выражений для дисперсий. Также известными являются углы крена и тангажа (телефон сориентирован параллельно плоскости земли).

В таблице 2 заданы значения СКО шумов наблюдения и формирующих шумов, используемых при моделировании синтезированного фильтра:

Таблица 2. Параметры моделирования

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| СКО формирующего шума угла курса | 0,2 рад |
| СКО формирующего шума модуля скорости движения | 0,3 м/с |
| СКО шума наблюдения координаты Х | 0,2 м |
| СКО шума наблюдения координаты У | 0,2 м |
| СКО шума наблюдения скорости угла курса | 0,01 рад/с |

Ниже на рисунках 8 – 13 приведены результаты моделирования работы синтезированного комплексного фильтра Калмана.

Исходя из постановки задачи, наблюдаемыми процессами являются изменение координат потребителя во времени и скорость изменения угла курса во времени. На рисунках 8 – 10 представлены эти процессы: синим обозначены референсные измерения, черным обозначены смоделированные измерения и красным обозначены оценки, формируемые на выходе синтезированного комплексного фильтра.



Рисунок 8 – Зависимость истинной скорости изменения угла курса, ее смоделированных измерений и ее оценки от времени, формируемой комплексным фильтром

На рисунке 8 представлены графики изменения истинной скорости угла курса, смоделированной скорости угла курса и ее оценка, формируемая фильтром. Этот элемент вектора состояния моделируется как винеровский процесс, а значит, его оценка также будет являться винеровским процессом. Это подтверждается на графике: оценка скорости курса имеет тот же характер, что и смоделированные измерения. Также из графика видно, что сформированная оценка не имеет систематической погрешности, которая обычно визуально отражается как смещения математического ожидания случайного процесса. В данном же случае видно, мат. ожиданием является истинный процесс, относительно которого и флуктуирует как моделируемые измерения, так и оценки фильтра.



Рисунок 9 – Зависимость истинной координаты Х, ее смоделированных измерений и ее оценки, формируемой комплексным фильтром



Рисунок 10 – Зависимость истинной координаты Y, ее смоделированных измерений и ее оценки, формируемой комплексным фильтром

На графиках изменения координат от времени (Рисунки 9 и 10) видно, что их оценки, сформированные фильтром, не имеют систематической ошибки, и так же, как и оценки скорости угла курса, колеблются относительно истинного процесса. Данное замечание тоже подтверждает адекватность работы фильтра. На этих графиках также можно заметить и инерционность оценок, которая возникает при смене режимов статики и динамики. Скорость этих переходных процессов зависит от величины СКО формирующих шумов координат, которые подбираются эмпирически, в зависимости от постановки задачи. В частности, в рассмотренном случае, эти переходные процессы не оказывают влияние на общую оценку траектории.

Для полноты картины, на рисунках 11 и 12 представлены зависимости элементов вектора состояния, не входящие в вектор измерений. Синим обозначены референсные истинные процессы, а красным – оценки, формируемые фильтром Калмана.

Из графиков на рисунке 11 видно, что угол курса оценивается адекватно, хотя и имеет некоторую систематическую ошибку.

Что касается временной зависимости скорости потребителя (рисунок 12), то так же, как и на графиках для оценок координат, наблюдается инерционность оценок скорости потребителя. Однако, при принятых СКО формирующего шума эти переходные процессы не оказывают значительного влияния на общую оценку скорости.

оценка курса

Рисунок 11 – Зависимость истинного угла курса и его оценка, формируемая комплексным фильтром



Рисунок 12 – Зависимость истинной скорости потребителя и ее оценка, формируемая комплексным фильтром

На рисунке 13 представлена итоговая оценка траектории движения потребителя. Несмотря на наличие инерционности оценок координат и скорости, а также наличие систематики оценки угла курса, оценка траектории, сформированная фильтром, соответствует референсной.



Рисунок 13 – Зависимость истинной траектории движения потребителя, ее смоделированных измерений и ее оценки, формируемой комплексным фильтром

Очевидно, что для наглядной оценки эффективности синтезированного алгоритма стоит привести статистические характеристики ошибок оценивания элементов вектора состояния. В таблице 3 приведены среднеквадратические отклонения и математические ожидания ошибок оценок координат и угла курса. Также приведена радиальная ошибка позиционирования (в англоязычной литературе обозначается как DRMS - Distance Root Mean Square), которая вычисляется как:

|  |  |
| --- | --- |
|  |  |

Таблица 3. Статистические характеристики ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Х, м** | **Y, м** | **α, ⁰** |
| СКО ошибок оценки в режиме статики до движения | 0,019 | 0,010 | 0,230 |
| Математическое ожидание ошибок оценки в режиме статики до движения | 0,006 | 0,003 | 0,057 |
| Радиальная ошибка позиционирования в режиме статики до движения | 0,045 | |  |
| СКО ошибок оценки в динамике | 0,170 | 0,090 | 0,290 |
| Математическое ожидание ошибок оценки в динамике | 0,075 | 0,040 | 0,220 |
| Радиальная ошибка оценки позиционирования в динамике | 0,400 | |  |
| СКО ошибок оценки в режиме статики после движения | 0,130 | 0,075 | 0,290 |
| Математическое ожидание ошибок оценки в режиме статики после движения | 0,210 | 0,090 | 0,086 |
| Радиальная ошибка позиционирования в режиме статики после движения | 0,300 | |  |

По данным таблицы 3 можно сделать следующие выводы:

1. СКО ошибок оценки координат в динамике увеличивается почти на порядок по сравнению с СКО в статических режимах. Это обусловлено тем, что в динамике среднее значение, в отличие от режима статики, постоянно меняется и фильтр подстраивается под эти изменения с запаздыванием.
2. Сравнивая СКО и математическое ожидание ошибок оценки координат, можно увидеть, что все значения для координаты Х в ⁓2 раза больше, чем для координаты Y. Это можно обосновать тем, что в математической модели фильтра оценивается абсолютное значение скорости, а не ее составляющие, которые могут значительно отличаться друг от друга. То есть, в данном случае формирующий шум распределяется между составляющими скорости пропорционально значению угла курса, а не пропорционально вкладу составляющих скорости в ее абсолютное значение.
3. Постоянство математического ожидания и СКО оценок угла курса подтверждает адекватность работы фильтра, так как угол курса моделировался постоянным значением.
4. Радиальная ошибка позиционирования показывает средний радиус области разброса оценок координат на плоскости относительно математического ожидания. Очевидно, что в режимах статики эти значения будут меньше, чем в режиме динамики. Однако, из таблицы видно, что в режиме статики после движения, это число в ⁓ 7 раз больше, чем в начальном режиме статики. Это, опять же, связано с инерционностью фильтра. Если искусственно исключить этот участок, то радиальные ошибки будут сравнимыми – 0.045 м для первого режима статики и 0.071 м для второго режима статики.

Проанализируем ошибки фильтрации, сравнив значения, полученные из аналитических выражений и полученные из оценок матрицы дисперсий синтезированного фильтра. На рисунках 14 – 17 приведены:

* временные зависимости для мгновенных ошибок оценок элементов вектора состояния, формируемых на выходе комплексного фильтра (красные кривые);
* временные зависимости предельных ошибок фильтрации элементов вектора состояния по уровню , полученные по оценкам матрицы дисперсии синтезированного фильтра (синие штриховые кривые);
* предельные ошибки фильтрации элементов вектора состояния по уровню , полученные из аналитических выражений (44), (48), (51) и (58);



Рисунок 14 – Зависимость ошибки оценки курса от времени

На рисунке 14 видно, что значение предельной ошибки фильтрации угла курса по уровню , вычисленное из аналитического выражения, имеет большой запас перед предельной ошибкой фильтрации по уровню , полученной из оценок матрицы дисперсий ошибок фильтрации. Это говорит о том, что величину СКО формирующего шума угла курса можно выбрать большей без потери качества фильтрации.



Рисунок 15 – Зависимость ошибки оценки координаты Х от времени



Рисунок 16 – Зависимость ошибки оценки координаты Y от времени



Рисунок 17 – Зависимость ошибки оценки скорости от времени

Для мгновенных ошибок оценок координат и скорости (Рисунки 15 – 17) потребителя наблюдается выход за уровни граничных дисперсий при смене режимов статики и динамики. Причем, ошибки оценки для координаты Х и ошибки оценки скорости превышают как предельные ошибки, полученные из матрицы дисперсии фильтра, так и значения, полученные из аналитических выражений. Это говорит о том, что выбранные уровни СКО шумов наблюдения координат и СКО формирующего шума скорости выбраны слишком высокими.

В таблице 4 приведены численные значения предельных ошибок фильтрации, полученные при моделировании и из аналитических выражений, приведенных в Главе 2.

Таблица 4. Оценки предельных ошибок фильтрации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Х, м** | **Y, м** | **V, м/с** | **α, ⁰** |
| Оценка предельной ошибки фильтрации, полученная аналитически | 0,28 | 0,28 | 0,5 | 16,6 |
| Оценка предельной ошибки фильтрации, полученная из матрицы дисперсий фильтра при имитационном моделировании | 0,22 | 0,16 | 0,3 | 2,0 |

По данным из таблицы 4 можно сделать вывод, что в пределах погрешностей, соизмеримых с заданными значениями СКО формирующих шумов и СКО шумов наблюдения из таблицы 2, теоретические значения предельных ошибок, которые можно считать референсными, так как они получены непосредственно из математической модели фильтра, в установившемся режиме совпадают с предельными ошибками фильтрации, полученными из оценок матрицы дисперсии при имитационном моделировании. Это говорит о сходимости синтезированного фильтра.

# глава 4. Экспериментальное исследование алгоритма комплексирования СШП ЛНС И ДУС

## 4.1 Архитектура экспериментального стенда

Рассматриваемый в данной работе алгоритм обработки измерений системы внутреннего позиционирования является комплексным, поскольку объединяет в себе измерения двух различных типов навигационных систем – измерения сверхширокополосной локальной системы навигации (СШП ЛНС) и измерения инерциальной навигационной системы (ИНС).

В связи с этим необходимо отдельно рассмотреть используемый тип архитектуры для обеих подсистем.

### **4.1.1. Сверхширокополосная локальная навигационная система**

СШП ЛНС состоят из подсистемы опорных маяков и подсистемы меток, являющихся навигационной аппаратурой потребителя (Рисунок 18).

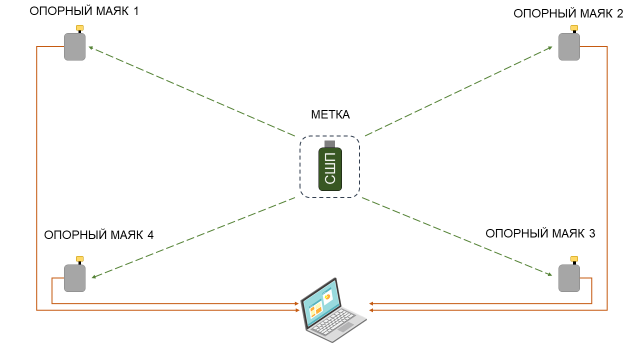


Рисунок 18 – Структурная схема СШП ЛНС

Опорные маяки – это СШП радиоустройства, которые образуют инфраструктуру навигационной системы и располагаются в точках с известными координатами в локальной системе координат. Как правило, опорные маяки размещают по периметру помещения – на стенах или под потолком.

Метки – это радиоустройства или их структурные компоненты, которыми оснащаются потребители, чье местоположение требуется определять в рабочей зоне локальной навигационной системы. В рамках поставленной задачи в качестве метки предполагается использование смартфона, у которого есть СШП-приемопередатчик на борту.

Опорные маяки и метки обмениваются сообщениями по физическому СШП радиоканалу и производят измерения радионавигационных параметров на разных уровнях – это могут быть первичная информация такая как, времена приходов сигналов или задержки их распространения, а также преобразованная из них, вторичная. Например, разности фаз, разности дальностей, амплитуды радиосигналов. Данные измерения передаются в узел обработки, соответствующий выбранной архитектуре, где реализуются алгоритмы расчета координат меток. Данные алгоритмы в зависимости от выбранной архитектуры могут быть реализованы на стороне потребителя (меток), на стороне инфраструктуры (маяков) или на отдельном устройстве.

Кроме того, системы различных архитектур различаются по минимальному числу опорных маяков необходимых для решения навигационной задачи, используемым навигационным алгоритмам, а также распределением геометрического фактора (ГФ). Геометрический фактор связывает точность определения координат метки с геометрией расположения всех опорных маяков и отслеживаемой метки: он показывает во сколько раз погрешность определения координат метки выше, чем погрешность первичных измерений.

В СШП ЛНС выделяют следующие виды архитектур:

1. Запросные

A. Дальномерные (ToF – Time of Flight);

B. Угломерно-дальномерные (AoA – angle of arrival);

2. Беззапросные

A. Разностно-дальномерные (TDoA – Time difference of arrival);

B. Инверсные разностно-дальномерные (RTDoA – Reverse Time difference of arrival).

В таблице 5 приведены основные характеристики вышеперечисленных систем, по которым можно оценить пригодность того или иного вида ЛНС для поставленной задачи.

Таблица 5. Сравнение характеристик архитектур СШП ЛНС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Запросные** | | **Беззапросные** | |
|  | **ToF** | **AoA** | **TDoA** | **RTDoA** |
| **Точность** | Единицы см | Десятки см | Десятки см | Десятки см |
| **Число одновременно отслеживаемых меток** | Десятки | Десятки | Сотни | Тысячи |
| **Решение навигационной задачи** | На маяках и на метке | На маяках | На маяках | На метке |

Сферы применения локальных навигационных систем на базе смартфонов различны – навигация пешеходов в общественных местах, мониторинг персонала на предприятиях, в том числе в зонах повышенной опасности. Для таких применений параметр «точности» не настолько критичен по сравнению с параметром «число одновременно отслеживаемых меток», поэтому для ЛНС на базе смартфонов наиболее предпочтительны беззапросные архитектуры.

Выбор между прямой и инверсной беззапросной архитектурой обуславливается на основе того, на стороне меток или инфраструктуры требуется решать навигационную задачу. В задаче мониторинга требуется наличие координат меток на стороне инфраструктуры, что приводит к прямой архитектуре. В задаче навигации пешеходов требуется иметь координаты в аппаратуре потребителя, что приводит к инверсной архитектуре.

### **4.1.2. Инерциальная навигационная система**

В современных смартфонах встроены датчик определения его ориентации в пространстве, также они могут подсчитать пройденный путь потребителя в шагах, километрах и даже калориях для оценки активности пользователя. К этим датчикам, в том числе, относятся инерциальные: акселерометры и датчики угловых скоростей. Эти датчики обычно работают в составе инерциальной навигационной системы, в которую также входят неинерциальные датчики, например, магнитометры, барометры и т.д.

Особенностью этих датчиков является гораздо более высокий темп измерений по сравнению с СШП ЛНС, характеризующийся значениями порядка десятков-сотен герц.

Таким образом, архитектура типичной ИНС приведена на рисунке 19 и содержит следующие функциональные блоки:

* трехосевой акселерометр, осуществляющий оценку линейных ускорений объекта;
* трехосевой датчик угловых скоростей (ДУС), осуществляющий оценку угловых скоростей поворота;
* трехосевой магнитометр, осуществляющий оценку направления вектора магнитного поля;
* центральный вычислитель (DMP – digital motion processor), осуществляющий первичную обработку измерений инерциальных датчиков, реализующий оценку ориентации объекта, подсчета шагов, расчета координат.

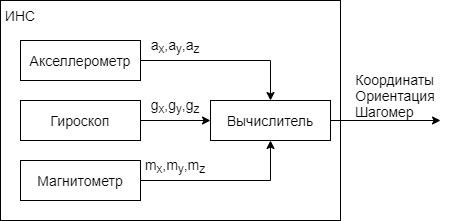


Рисунок 19 – Архитектура ИНС

Для макета навигационной системы в качестве инерциального измерителя предполагается использовать датчики смартфона на базе операционной системы Android 10.0. В данной версии операционной системы есть возможность получать измерения как от акселерометров, датчиков угловых скоростей, магнитометров, датчика расчета кватерниона в первичном, необработанном виде, так и от центрального вычислителя с учетом компенсации дрейфов нулевых значений датчиков, перекосов осей датчиков и их масштабных коэффициентов.

Согласно поставленной в данной диссертации задаче, погрешности датчиков не учитываются в векторе состояния комплексного фильтра, поэтому при экспериментальных исследованиях будем использовать информацию о компенсированных угловых скоростях, получаемых на выходе центрального вычислителя.

### **4.1.3. Макет комплексной локальной навигационной системы**

Итоговая структура макета комплексной системы навигации представлена на рисунке 20.

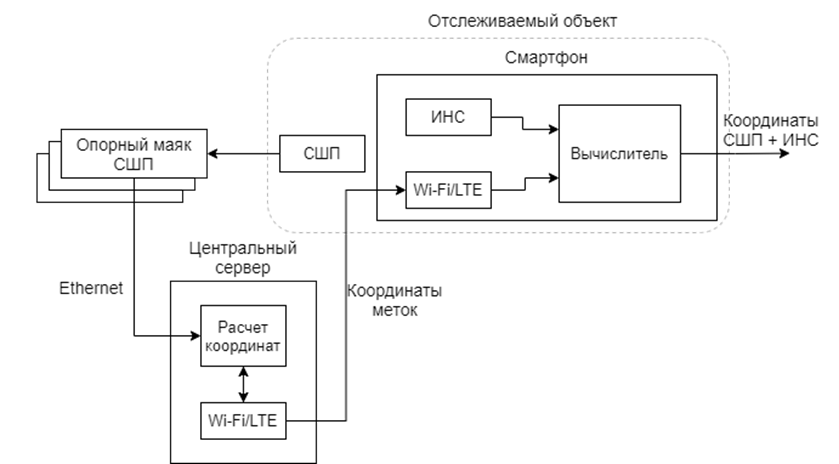


Рисунок 20 – Структурная схема архитектуры экспериментального макета

С позиции СШП ЛНС она является прямой беззапросной системой, за исключением того, что СШП приемо-передатчик не входит в состав смартфона. Это связано с тем, что на данный момент на мировом рынке не так много смартфонов, имеющих в своем составе СШП приемопередатчик, поскольку повсеместное внедрение данной технологии находится на начальном этапе, что обуславливает актуальность задач разработки архитектур систем позиционирования и алгоритмов обработки измерений для систем на базе данной технологии.

Также, для смартфонов, уже имеющих на борту СШП приемопередатчик, на данный момент нет открытых прикладных интерфейсов программирования (API – application programming interface) как для операционной системы Android, так и для Apple IOS, которые позволили бы осуществлять гибкую разработку и конфигурирование подобных систем. В связи с этим, в экспериментальном макете для проверки синтезированного алгоритма комплексирования измерений СШП ЛНС и ИНС, было принято решение использовать внешний СШП приемо-передатчик, поскольку с точки зрения алгоритмов, их входных и выходных данных это не имеет существенного значения.

В качестве СШП ЛНС в экспериментальном макете использовалась разностно-дальномерная беззапросная система TTK1000 производства Qorvo (ранее DecaWave), Ирландия.



Рисунок 21 – Внешний вид опорного маяка СШП ЛНС

В инфраструктурную часть этой системы входит четыре СШП опорных маяка, которые располагают по периметру рабочей зоны (Рисунок 21).



Рисунок 22 – Аппаратура потребителя: смартфон и СШП радиометка

Аппаратурой потребителя в составе макета комплексной системы навигации выступил смартфон с операционной системой Android 10.0, а также СШП радиометка, расположенная рядом со смартфоном (Рисунок 22).

## Результаты экспериментального исследования синтезированного алгоритма

Эксперимент проводился на закрытой площадке размером 9×18 м2 (Рисунок 23).



Рисунок 23 – Система СШП якорей

Для объективного сравнения результатов имитационного моделирования и экспериментального исследования алгоритма потребитель двигался по аналогичной смоделированной траектории.

В соответствии с постановкой задачи и с условиями имитационного моделирования все оценки элементов вектора состояния будут производиться в локальной системе координат, связанной с СШП ЛНС.

При проведении эксперимента изначально сориентируем смартфон потребителя параллельно плоскости пола и по углу курса, который не меняется по ходу его движения и пребывания в режимах статики. Это необходимо для сравнения экспериментальных предельных ошибок фильтрации с ошибками, полученными в результате имитационного моделирования и с помощью аналитических параметрических выражений для дисперсий.

При обработке экспериментальных измерений используются те же значения СКО шумов наблюдения и формирующих шумов, что использовались при моделировании синтезированного фильтра (Таблица 2).

На рисунках 24-27 показаны результаты обработки экспериментальных измерений синтезированным алгоритмом. Черной штриховой линией обозначены референсные измерения, синим обозначены экспериментальные измерения и красным обозначены оценки, формируемые на выходе синтезированного комплексного фильтра.



Рисунок 24 – Временная зависимость референсной скорости изменения угла курса, ее экспериментальных измерений и ее оценки, формируемой комплексным фильтром

На рисунке 24 представлены графики изменения истинной скорости угла курса, экспериментально измеренной скорости угла курса и ее оценки, формируемой фильтром. Оценка скорости курса имеет тот же характер, что и смоделированные измерения, однако, если сравнить с аналогичным графиком, полученным из имитационного моделирования (Рисунок 8), то видно, что уровень колебаний измерений относительно референса на порядок выше. Также из графика видно, что сформированная оценка не имеет систематической погрешности, также как оценка при моделировании.

На рисунках 25 и 26 представлены временные зависимости изменения истинных координат, их экспериментальных измерений и их оценок, сформированных синтезированным фильтром.

На этих графиках видно, что их оценки, сформированные фильтром, так же, как и в случае моделирования, не имеют систематической ошибки, и так же, как и оценки скорости угла курса, колеблются относительно референса. Это подтверждает адекватность работы фильтра в экспериментальных условиях. На этих графиках тоже сохраняется инерционность оценок, которая возникает при смене режимов статики и динамики, а также в начале поступления измерений в фильтр.



Рисунок 25 – Зависимость истинной координаты X, ее смоделированных измерений и ее оценки, формируемой комплексным фильтром



Рисунок 26 – Зависимость истинной координаты Y, ее смоделированных измерений и ее оценки, формируемой комплексным фильтром

Если сравнить измерения координат, то видно, что измерения Y имеют большее количество интенсивных выбросов, чем измерения Х. Поэтому, оценка для координаты Х более гладкая и близкая к референсу.



Рисунок 27 – Зависимость истинной траектории движения потребителя, ее экспериментальных измерений и ее оценки, формируемой комплексным фильтром

На рисунке 27 представлена итоговая оценка траектории движения потребителя на фоне измерений и истинной траектории. Эти графики наглядно показывают, что синтезированный фильтр справился со всеми выбросами радиоизмерений и оценил траекторию верно.

Это подтверждает и таблица 7, в которой приведены статистические характеристики ошибок оценок элементов вектора состояния в разных режимах на выходе фильтра.

Также для трех режимов приведена радиальная ошибка позиционирования, которая показывают средний радиус области ошибок оценки координат.

Таблица 6. Статистические характеристики ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Х, м** | **Y, м** | **α, ⁰** |
| СКО ошибок оценки в режиме статики до движения | 0,12 | 0,03 | 2,98 |
| Математическое ожидание ошибок оценки в режиме статики до движения | 0,08 | 0,03 | 1,38 |
| Радиальная ошибка позиционирования в режиме статики до движения | 0,25 | |  |
| СКО ошибок оценки в динамике | 0,31 | 0,15 | 5,84 |
| Математическое ожидание ошибок оценки в динамике | 0,32 | 0,14 | 0,29 |
| Радиальная ошибка оценки позиционирования в динамике | 0,68 | |  |
|  | **Х, м** | **Y, м** | **α, ⁰** |
| СКО ошибок оценки в режиме статики после движения | 0,09 | 0,11 | 4,01 |
| Математическое ожидание ошибок оценки в режиме статики после движения | 0,08 | 0,27 | 1,72 |
| Радиальная ошибка позиционирования в режиме статики после движения | 0,28 | |  |

Выводы из данных таблицы 6:

1. Статистические характеристики ошибок оценок координат при сравнении, как и для случая моделирования, имею значительные отличия. Причем, в режиме статике до движения и в динамическом режиме завышенными оказываются оценки ошибок для координаты Х, а в режиме статики после движения, наоборот, ошибки оценок координаты Y значительно выше. Это еще раз доказывает справедливость утверждения о непропорциональном распределении формирующего шума модуля скорости потребителя по его составляющим.
2. Стабильными в пределах погрешности остаются статистические характеристики ошибок оценки угла курса при смене режимов статики и динамики. Это оправдывает использование в качестве измерений скорости изменения угла курса.
3. По сравнению с результатами имитационного моделирования, радиальная ошибка позиционирования в обоих режимах статики практически одинаковая. Однако, инерционность фильтра повлияла в этом случае на эту характеристику в динамическом режиме, так как она превышает потенциальную точность СШП радиоизмерений.

Проанализируем ошибки фильтрации, сравнив значения, полученные из аналитических выражений и полученные при обработке экспериментальных измерений из оценок матрицы дисперсий синтезированного фильтра. На рисунках 28 – 31 приведены:

* временные зависимости для мгновенных ошибок оценок элементов вектора состояния, формируемых на выходе комплексного фильтра (красные кривые);
* временные зависимости экспериментальных предельных ошибок фильтрации элементов вектора состояния по уровню , полученные по оценкам матрицы дисперсии синтезированного фильтра (синие штриховые кривые);
* предельные ошибки фильтрации элементов вектора состояния по уровню , полученные из аналитических выражений (44), (48), (51) и (58);



Рисунок 28 – Зависимость ошибки оценки координаты Х от времени



Рисунок 29 – Зависимость ошибки оценки координаты Y от времени



Рисунок 30 – Зависимость ошибки оценки скорости от времени

Для мгновенных ошибок оценок координат и скорости потребителя (Рисунки 28 – 30) наблюдается аномальные выбросы при смене режимов статики и динамики. Аналогичная ситуация происходила и при имитационном моделировании из-за слишком высоких значений уровни СКО шумов наблюдения координат и СКО формирующего шума скорости. Но для возможности объективного сравнения ошибок фильтрации на всех этапах исследования, в эксперименте использовались те же значения СКО этих шумов. Тем не менее, итоговая оценка траектории говорит о том, что эти выбросы не оказывают значительного влияния на эффективность работы синтезированного фильтра.



Рисунок 31 – Зависимость ошибки оценки курса от времени

Как и предполагалось при имитационном моделировании, запас по уровню аналитически рассчитанной предельной ошибки фильтрации угла курса сыграл роль при экспериментальных исследованиях: низкоклассный МЭМС-датчик угловой скорости на борту смартфона выдает нестабильные измерения с большим разбросом. То есть, в данном случае имитационная модель оказалась далека от реальных измерений.

В таблице 4 приведены численные значения оценок предельных ошибок фильтрации координат, скорости потребителя и угла курса, полученные из аналитических выражений, приведенных в Главе 2, при имитационном моделировании и экспериментально.

Таблица 7. Оценки предельных ошибок фильтрации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Х, м** | **Y, м** | **V, м/с** | **α** |
| Оценка предельной ошибки фильтрации, полученная аналитически | 0,28 | 0,28 | 0,5 | 16,6 |
| Оценка предельной ошибки фильтрации, полученная из матрицы дисперсий фильтра при имитационном моделировании | 0,22 | 0,16 | 0,3 | 2,0 |
| Предельная ошибка фильтрации, полученная экспериментально | 0,35 | 0,36 | 0,6 | 9,2 |

По данным из таблицы 7 можно сделать вывод, что в пределах погрешностей, соизмеримых с заданными значениями СКО формирующих шумов и СКО шумов наблюдения из таблицы 2, все рассмотренные предельные ошибки фильтрации одинаковы на всех этапах проверки синтезированного фильтра. Таким образом, синтезированный комплексный фильтр Калмана применим и эффективен в планарной задаче позиционирования потребителя с помощью ЛНС на базе смартфонов.

# заключение

Целью данной магистерской диссертации является синтез и исследование алгоритма комплексирования сверхширокополосной системы локальной радионавигации и датчика угловых скоростей на борту смартфона.

Для достижения поставленной в работе цели были решены следующие задачи: а) Был проведен обзор существующих исследований по теме ВКР; б) был синтезирован алгоритм комплексирования приемопередатчика СШП ЛНС и датчика угловых скоростей; в) была оценена предельная ошибка фильтрации путем решения уравнения Рикатти для матрицы дисперсий; г) было проведено имитационное моделирование синтезированного алгоритма и оценена предельная ошибка фильтрации; д) был собран макет навигационной системы для проведения экспериментов; е) были проведены эксперименты и оценена предельная ошибка фильтрации; ж) было проведено сравнение оценок предельных ошибок фильтрации, полученных аналитически, на имитационном моделировании и экспериментально.

В результате исследования было выявлено полное соответствие в пределах погрешности между теоретически рассчитанными оценками предельных ошибок фильтрации и оценками предельных ошибок фильтрации, полученными по из матриц дисперсий ошибок в имитационной модели и по экспериментальным измерениям.

Научно-практическая ценность данной магистерской диссертации заключается в подтверждении эффективности синтезированного комплексного алгоритма и его применимости для реальных задач навигации внутри помещений.

Новизна результатов заключается: а) в новом синтезированном алгоритме обработки измерений; б) в применении данного алгоритма в перспективных комплексных системах навигации на базе смартфонов, в составе которых есть сверхширокополосный приемопередатчик, ведь такие датчики сейчас внедряются во все флагманские устройства.

# список литературы

1. Cheng-xiao WANG, Chong SHEN, Kun ZHANG, Han-wen LI, Qian GAO and Xiao-min FENG. Research on TDOA/AOA Fusion Algorithm Based on UWB Technology. 2018 International Conference on Communication, Network and Artificial Intelligence (CNAI 2018), ISBN: 978-1-60595-065-5
2. Р.С. Куликов, Д.В. Царегородцев, Н.А. Куковякина, А.А. Шамина, В.А. Лепетюха. Разностно-дальномерный метод определения координат потребителя применительно к построению локальных беззапросных систем навигации. Вестник МЭИ. №6. 2018 г.
3. Paolo Dabove, Vincenzo Di Pietra, Marco Piras, Ansar Abdul Jabbar, Syed Ali Kazim, “Indoor positioning using Ultra-Wide Band (UWB) mtechnologies: positioning accuracies and sensors’ performances”, 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), 2018
4. J. Ling, L. Wang, H. Ji, H. Xie, J. Ding and Q. Dai, "UWB-Based Real-Time Continuous Positioning System in NLOS Tunnel Environment," 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Zhengzhou, China, 2018, pp. 142-1424. doi: 10.1109/CyberC.2018.00037
5. T. Zhou, Y. Cheng, B. Lian and Y. Zhang, "Research on UWB Localization Based on TOA in Indoor NLOS Environment," 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, 2018, pp. 983-988. doi: 10.1109/ITME.2018.00219
6. W. Kang and Y. Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization," in IEEE Sensors Journal, vol. 15, no. 5, pp. 2906-2916, May 2015. doi: 10.1109/JSEN.2014.2382568
7. E. Foxlin, “Pedestrian tracking with shoe-mounted inertial sensors”, IEEE Computer Graphics and Applications, vol, 25, no. 6, pp. 38-46, Nov.-Dec. 2005.
8. Маринушкин П. С., Бахтина В. А., Подшивалов И. А., Стукач О. В. Вопросы разработки инерциальных пешеходных навигационных систем на основе МЭМС-датчиков // Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 06. С. 157–173.
9. Hai Yang et al., “Smartphone-Based Indoor Localization System Using Inertial Sensor and Acoustic Transmitter/Receiver”, IEEE Sensors Journal, vol. 16, no. 22 , pp. 8051 – 8061, Nov. 2016.
10. Чугунов А.А., Петухов Н.И., Митич А., Семенов В.Д., Захарова Е.В., Церегородцев Д.В., Болдырев А.Р. Комплексирование локальной сверхширокополосной угломерно-дальномерной и инерциальной навигационных систем. XXVII Санкт-Петербургская международная конференция по интегрированным навигационным системам. 2020. С. 61-68.
11. F. Zampella, A. De Angelis, I. Skog, D. Zachariah and A. Jiménez, "A constraint approach for UWB and PDR fusion," 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, NSW, 2012, pp. 1-9, doi: 10.1109/IPIN.2012.6418929.
12. A. Benini, A. Mancini, A. Marinelli, and S. Longhi, “A Biased Extended Kalman Filter for Indoor Localization of a Mobile Agent using Low-Cost IMU and UWB Wireless Sensor Network,” in IFAC Proceedings Volumes, vol. 45, no. 22, 2012, pp. 735–740
13. Daquan Feng, Chunqi Wang, Chunlong He, Yuan Zhuang, and Xiang-Gen Xia. «Kalman Filter Based Integration of IMU and UWB for High-Accuracy Indoor Positioning and Navigation», IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3133 - 3146, 2020
14. Pengzhan Chen, Ye Kuang, Xiaoyue Chen, “A UWB/Improved PDR Integration Algorithm Applied to Dynamic Indoor Positioning for Pedestrians,” Sensors, 17, 2065, 2017
15. IEEE Std. 802.15.4-2020, IEEE Standard for Information Technology – Telecommunications and Information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
16. Сверхширокополосные сигналы. URL: <https://ru.wikipedia.org/wiki/Сверхширокополосные_сигналы> (Дата обращения: 23.03.2021)
17. ГЛОНАСС. Принципы построения и функционирования/ под ред. А.И. Перова, В.Н. Харисова. Изд. 4-е, перераб. и доп. – М.: Радиотехника, 2010. 800 с., ил.
18. А.И. Перов. Статистическая теория радиотехнических систем. – М.: Радиотехника, 2003. - 398 с.
19. С.В. Первачев. Радиоавтоматика: учебник для вузов. – М.: Радио и связь, 1982, 296 с.
20. Решение ГКРЧ № 09-05-02 от 15 декабря 2009 г. URL: <https://digital.gov.ru/ru/documents/3927/> (Дата обращения: 20.06.2021)
21. И.А. Нагин. Комплексирование навигационной аппаратуры потребителя СРНС с нерадиотехническими датчиками. Диссертация на соискание ученой степени кандидата технических наук, 2013.

# Приложение А

Публикации на основе результатов ВКР

1. Т.А. Бровко, А.П. Малышев, А.А. Чугунов, И.В. Корогодин, "Разработка системы навигации в закрытых помещениях на базе смартфонов с использованием технологии сверхширокополосных сигналов" // Тезисы докладов IV национальной научно-технической конференции "Технологии будущего", 5-9 апреля 2021.
2. Т.А. Бровко, А.П. Малышев, А.Е. Боев, С.В. Солдаткин, Р.С. Куликов. Комплексная система навигации смартфонов внутри помещений // Тезисы докладов XXVII Международной научно-технической конференции студентов и аспирантов «Радиоэлектроника, электротехника и энергетика». 11-12 марта 2021 года.
3. Tatiana Brovko, Alexander Chugunov, Alexander Malyshev, Ilya Korogodin, Oleg Glukhov, Nikita Petuhov, “Complex Kalman Filter Algorithm For Smartphone-based Indoor UWB/INS Navigation Systems”, 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russia, 2021 (публикуется).

# Приложение Б

Листинг программы имитационной модели комплексного фильтра

close all

clear all

clc

N = 5000;

sigma\_ksi\_V = 0.3;

sigma\_ksi\_alpha = 0.015;

sigma\_n\_x = 0.2;

sigma\_n\_y = 0.2;

sigma\_n\_omega = 0.02;

Vx\_true\_0 = (18.33 - 4.2)/(24.02-12.02);

Vy\_true\_0 = (9.04 - 1.91)/(24.02-12.02);

T = 0.1;

N1 = fix(12.02/T);

N2 = fix(24.02/T);

N\_all = fix(30/T);

for k = 1:N\_all

if k >= 1 && k <= N1

x\_true(k) = 4.2;

y\_true(k) = 1.91;

%V\_true(k) = 0;

Vx\_true(k) = 0;

Vy\_true(k) = 0;

alpha\_true(k) = atan2((9.04 - 1.91),(18.33 - 4.2));

omega\_true(k) = 0;

if k == 1

t(k) = 0;

else

t(k) = t(k-1) + T;

end

if k == N1

Vx\_true(k) = Vx\_true\_0;

Vy\_true(k) = Vy\_true\_0;

end

elseif k >= N2

x\_true(k) = x\_true(k - 1);

y\_true(k) = y\_true(k - 1);

Vx\_true(k) = 0;

Vy\_true(k) = 0;

alpha\_true(k) = alpha\_true(k - 1);

omega\_true(k) = 0;

t(k) = t(k-1) + T;

else

Vx\_true(k) = Vx\_true(k - 1);

Vy\_true(k) = Vy\_true(k - 1);

x\_true(k) = x\_true(k - 1) + Vx\_true(k-1)\*T;

y\_true(k) = y\_true(k - 1) + Vy\_true(k-1)\*T;

alpha\_true(k) = alpha\_true(k-1);

omega\_true(k) = 0;

t(k) = t(k-1) + T;

end

y\_meas(:,k) = [x\_true(k) + sigma\_n\_x\*randn; y\_true(k) + sigma\_n\_y\*randn; omega\_true(k) + sigma\_n\_omega\*randn];

end

x\_est = [4.2; 1.91; 0; atan2((9.04 - 1.91),(18.33 - 4.2)); 0];

Dx\_est = eye(5);

D\_ksi = [sigma\_ksi\_V^2 0 ;

0 (sigma\_ksi\_alpha)^2];

Dn = [sigma\_n\_x^2 0 0; 0 sigma\_n\_y^2 0; 0 0 sigma\_n\_omega^2];

G = [0 0; 0 0; T 0; 0 0; 0 1];

C = [1 0 0 0 0;

0 1 0 0 0;

0 0 0 0 1];

for k = 2:N\_all

x\_ext(1,k) = x\_est(1,k-1) + x\_est(3,k-1)\*cos(x\_est(4,k-1))\*T;

x\_ext(2,k) = x\_est(2,k-1) + x\_est(3,k-1)\*sin(x\_est(4,k-1))\*T;

x\_ext(3,k) = x\_est(3,k-1);

x\_ext(4,k) = x\_est(4,k-1) + x\_est(5, k-1)\*T;

x\_ext(5,k) = x\_est(5,k-1);

dFdx = [1 0 cos(x\_ext(4,k))\*T -x\_ext(3,k)\*sin(x\_ext(4,k))\*T 0;

0 1 sin(x\_ext(4,k))\*T x\_ext(3,k)\*cos(x\_ext(4,k))\*T 0;

0 0 1 0 0;

0 0 0 1 T;

0 0 0 0 1];

Dx\_ext = dFdx\*Dx\_est\*dFdx' + G\*D\_ksi\*G';

% Dx\_est = (Dx\_ext^-1 + C'\*Dn\*C)^-1;

K = Dx\_ext\*C'/(C\*Dx\_ext\*C'+Dn);

x\_est(:,k) = x\_ext(:,k) + K\*(y\_meas(:,k) - C\*x\_ext(:,k));

Dx\_est = (eye(5)-K\*C)\*Dx\_ext;

Mx(1,k) = sqrt(Dx\_est(1,1));

My(1,k) = sqrt(Dx\_est(2,2));

MV(1,k) = sqrt(Dx\_est(3,3));

Malp(1,k) = sqrt(Dx\_est(4,4));

Momeg(1,k) = sqrt(Dx\_est(5,5));

end

for q = 1:length(omega\_true)

if q == 1

omega\_tr\_int(q) = omega\_true(q)\*T + alpha\_true(1);

omega\_ms\_int(q) = y\_meas(3,q)\*T + alpha\_true(1);

else

omega\_tr\_int(q) = omega\_tr\_int(q - 1) + omega\_true(q)\*T;

omega\_ms\_int(q) = omega\_ms\_int(q - 1) + y\_meas(3,q)\*T;

end

end

%% Static before moving

x\_error\_stat1 = x\_est(1,1:N1) - x\_true(1:N1);

y\_error\_stat1 = x\_est(2,1:N1) - y\_true(1:N1);

alpha\_error\_stat1 = x\_est(4,1:N1) - alpha\_true(1:N1);

mean\_x\_stat1 = mean(x\_error\_stat1);

mean\_y\_stat1 = mean(y\_error\_stat1);

mean\_alpha\_stat1 = mean(alpha\_error\_stat1);

std\_x\_stat1 = std(x\_error\_stat1);

std\_y\_stat1 = std(y\_error\_stat1);

std\_alpha\_stat1 = std(alpha\_error\_stat1);

%% Moving

x\_error\_mov = x\_est(1,N1:N2) - x\_true(N1:N2);

y\_error\_mov = x\_est(2,N1:N2) - y\_true(N1:N2);

alpha\_error\_mov = x\_est(4,N1:N2) - alpha\_true(N1:N2);

mean\_x\_mov = mean(x\_error\_mov);

mean\_y\_mov = mean(y\_error\_mov);

mean\_alpha\_mov = mean(alpha\_error\_mov);

std\_x\_mov = std(x\_error\_mov);

std\_y\_mov = std(y\_error\_mov);

std\_alpha\_mov = std(alpha\_error\_mov);

%% Static after moving

x\_error\_stat2 = x\_est(1,N2:N\_all) - x\_true(N2:N\_all);

y\_error\_stat2 = x\_est(2,N2:N\_all) - y\_true(N2:N\_all);

alpha\_error\_stat2 = x\_est(4,N2:N\_all) - alpha\_true(N2:N\_all);

mean\_x\_stat2 = mean(x\_error\_stat2);

mean\_y\_stat2 = mean(y\_error\_stat2);

mean\_alpha\_stat2 = mean(alpha\_error\_stat2);

std\_x\_stat2 = std(x\_error\_stat2);

std\_y\_stat2 = std(y\_error\_stat2);

std\_alpha\_stat2 = std(alpha\_error\_stat2);

%%

velocity\_error = x\_est(3,:) - sqrt(Vx\_true.^2 + Vy\_true.^2);

%%

%график траектории

figure(3)

plot(x\_true, y\_true,'b-', 'linewidth',2)

hold on

plot(y\_meas(1,:), y\_meas(2,:),'k-', 'linewidth',1)

grid on

xlabel('$$X, m$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$Y, m$$','Interpreter','LaTeX', 'FontSize',24)

legend({"Истинная траектория","Смоделированная траектория"},'FontSize',10)

set(gca, 'FontSize',17)

%Угол курса

figure(4)

plot(t, alpha\_true(1,:),'b-', 'linewidth',2)

hold on

plot(t, x\_est(4,:), 'r-','linewidth',2)

grid on

ylabel('$$ \alpha, rad $$', 'Interpreter','LaTeX','FontSize',24)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

%title('Истинный угол курса и его оценка','FontSize',18)

legend({"Истинный угол курса","Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

%производная угла курса

figure(7)

plot(t, omega\_true(1,:),'b-', 'linewidth',2)

hold on

plot(t, y\_meas(3,:),'k-', 'linewidth',0.5)

plot(t, x\_est(5,:), 'r-','linewidth',2)

grid on

ylabel('$$ \dot{\alpha}, rad/s $$', 'Interpreter','LaTeX','FontSize',24)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

%title('Истинная скорость угола курса, смоделированное измерение и оценка','FontSize',18)

legend({"Истинная скорость угла курса", "Cмоделированное измерение","Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

%Координата Х

figure(5)

% title('','FontSize',18)

plot(t, x\_true(1,:),'b-', 'linewidth',2)

hold on

grid on

plot(t, y\_meas(1,:),'k-', 'linewidth',0.5)

plot(t, x\_est(1,:),'r-', 'linewidth',2)

ylabel('$$X, m$$','Interpreter','LaTeX', 'FontSize',24)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

%title('Зависимость истинной координаты Х, смоделированного измерения и оценки от времени','FontSize',18)

legend({"Истинная координата Х", "Cмоделированное измерение", "Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

%Координата У

figure(6)

plot(t, y\_true(1,:),'b-', 'linewidth',2)

hold on

grid on

plot(t, y\_meas(2,:),'k-', 'linewidth',0.5)

plot(t, x\_est(2,:),'r-', 'linewidth',2)

ylabel('$$Y, m$$','Interpreter','LaTeX', 'FontSize',24)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

%title('Зависимость истинной координаты Y, смоделированного измерения и оценки от времени','FontSize',18)

legend({"Истинная координата Y","Cмоделированное измерение", "Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

%Оценка траектории

figure(9)

plot(x\_true, y\_true,'b-', 'linewidth',2)

hold on

plot(y\_meas(1,:), y\_meas(2,:),'k-', 'linewidth',0.5)

plot(x\_est(1,:),x\_est(2,:),'r-', 'linewidth',2)

grid on

%title('Истинная траектория, ее модель и еe оценка','FontSize',18)

xlabel('$$X, m$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$Y, m$$','Interpreter','LaTeX', 'FontSize',24)

legend({"Истинная траектория","Смоделированная траектория", "Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

%Ошибка оценки угла курса

figure(14)

plot(t(1:N1),alpha\_error\_stat1,'b','linewidth',2)

hold on

plot(t,3\*Malp,'r--','linewidth',2)

plot(t,3\*0.096,'m.')

plot(t(N1:N2),alpha\_error\_mov,'b','linewidth',2)

plot(t(N2:end),alpha\_error\_stat2,'b','linewidth',2)

plot(t,-3\*Malp,'r--','linewidth',2)

plot(t,-3\*0.096,'m.','linewidth',2)

grid on

title('Ошибка оценки угла курса','FontSize',18)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \widehat{\alpha} - \alpha, rad $$', 'Interpreter','LaTeX','FontSize',24)

set(gca, 'FontSize',17)

%legend("Ошибка оценки угла курса","Предельная граница ошибки фильтрации, рассчитанная по оценке матрицы дисперсии","Предельная граница ошибки фильтрации, рассчитанная аналитически",'FontSize',10)

%ошибка оценкии координаты Х

figure(15)

plot(t(1:N1),x\_error\_stat1,'b-','linewidth',2)

hold on

plot(t(N1:N2),x\_error\_mov,'b-','linewidth',2)

plot(t(N2:end),x\_error\_stat2,'b-','linewidth',2)

plot(t,3\*Mx,'r--','linewidth',2)

plot(t,-3\*Mx,'r--','linewidth',2)

plot(t,3\*0.0939,'m.','linewidth',2)

plot(t,-3\*0.0939,'m.','linewidth',2)

grid on

title('Ошибка оценкии координаты Х','FontSize',18)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \widehat{X} - X, m $$','Interpreter','LaTeX', 'FontSize',24)

set(gca, 'FontSize',17)

%legend("Ошибка оценки угла курса","Предельная граница ошибки фильтрации, рассчитанная по оценке матрицы дисперсии","Предельная граница ошибки фильтрации, рассчитанная аналитически",'FontSize',10)

%ошибка оценкии координаты Y

figure(16)

plot(t(1:N1),y\_error\_stat1,'b-','linewidth',2)

hold on

plot(t(N1:N2),y\_error\_mov,'b-','linewidth',2)

plot(t(N2:end),y\_error\_stat2,'b-','linewidth',2)

plot(t,3\*My,'r--','linewidth',2)

plot(t,-3\*My,'r--','linewidth',2)

plot(t,3\*0.0939,'m.','linewidth',2)

plot(t,-3\*0.0939,'m.','linewidth',2)

grid on

title('Ошибка оценкии координаты Y','FontSize',18)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \widehat{Y} - Y, m $$','Interpreter','LaTeX', 'FontSize',24)

set(gca, 'FontSize',17)

% legend("Смоделированное измерение","Оценка фильтра СШП+ИНС")

%Ошибка оценкии скорости

figure(17)

plot(t, velocity\_error,'b-','linewidth',2)

hold on

plot(t,3\*MV,'r--','linewidth',2)

plot(t,-3\*MV,'r--','linewidth',2)

plot(t,3\*0.1798,'m.','linewidth',2)

plot(t,-3\*0.1798,'m.','linewidth',2)

grid on

title('Ошибка оценкии скорости','FontSize',18)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \widehat{V} - V, m/s $$','Interpreter','LaTeX','FontSize',24)

set(gca, 'FontSize',17)

% legend("Смоделированное измерение","Оценка фильтра СШП+ИНС")

# Приложение В

Листинг программы расчета матрицы дисперсий ошибок фильтрации

close all

clear all

clc

syms A B C D E F G H I J x y z p q r s m

E1 = 2\*(x\*C-y\*z\*D) - (A^2\*s+B^2\*r);

E2 = (F\*x-y\*z\*G) + (C\*x+D\*y\*z)-(A\*B\*s+B\*E\*r);

E3 = (H\*x-y\*z\*I) - (A\*C\*s+B\*F\*r);

E4 = (I\*x-y\*z\*J) - (A\*D\*s+B\*G\*r);

E5 = 2\*(y\*F+x\*z\*G) - (B^2\*s+E^2\*r);

E6 = (H\*y+x\*z\*I) - (C\*B\*s+E\*F\*r);

E7 = (I\*y+x\*z\*J) - (B\*D\*s+E\*G\*r);

E8 = p - (C^2\*s+F^2\*r);

E9 = -(C\*D\*s+F\*G\*r);

E10 = q - (D^2\*s+G^2\*r);

vars = [A, B, C, D, E, F, G, H, I, J];

E = [E1, E2, E3, E4, E5, E6, E7, E8, E9, E10];

[S1, S2, S3, S4, S5, S6, S7, S8, S9, S10] = solve(E, vars)

# Приложение Г

Листинг программы расчета дисперсий ошибок фильтрации

close all

clear all

clc

sigma\_ksi\_V = 0.3;

sigma\_n\_x = 0.15;

sigma\_n\_y = 0.15;

alpha = atan2((9.04 - 1.91),(18.33 - 4.2));

x = cos(alpha);

y = sin(alpha);

p = sigma\_n\_x;

q = sigma\_n\_y;

s = sigma\_ksi\_V;

D11 = (y^2\*((4\*s^3)/(p\*x^2 + q\*y^2))^(3/4) + y^2\*(((4\*p\*s^3\*x^2)/(p\*x^2 + q\*y^2) - 4\*s^3 + (4\*q\*s^3\*y^2)/(p\*x^2 + q\*y^2))/(p\*x^2 + q\*y^2))^(1/2)\*((4\*s^3)/(p\*x^2 + q\*y^2))^(1/4))/(2\*s^2);

D22 = (y^2\*((4\*s^3)/(p\*x^2 + q\*y^2))^(3/4) - y^2\*(((4\*p\*s^3\*x^2)/(p\*x^2 + q\*y^2) - 4\*s^3 + (4\*q\*s^3\*y^2)/(p\*x^2 + q\*y^2))/(p\*x^2 + q\*y^2))^(1/2)\*((4\*s^3)/(p\*x^2 + q\*y^2))^(1/4))/(2\*s^2);

D33 = (y\*((4\*s^3)/(p\*x^2 + q\*y^2))^(1/2))/(2\*s);

D44 = ((4\*s^3)/(p\*x^2 + q\*y^2))^(1/4);

SKO\_COORD = sqrt(D11)

SKO\_ALPHA = sqrt(D33)

SKO\_V = sqrt(D44)

# Приложение Д

Листинги программ обработки экспериментальных измерений

**main.m**

close all

clear all

clc

config = Config();

config.sigma\_ksi = 10;

[uwb, gyro, acc, mag, euler] = log\_reader\_1\_0();

gyro\_new = interp1(gyro(1,:),gyro(4,:),uwb(1,:));

config.sigma\_n = 0.15;

config.sigma\_phi = 0.02;

config.sigma\_ksi\_v = 0.4;

config.sigma\_ksi\_phi = 0.3;

config.T = 0.1;

y = [uwb(2,:);uwb(3,:);gyro\_new];

X\_true = [y(1,1);y(2,1);1.5;atan2((9.04 - 1.91),(18.33 - 4.2));0];

X2(:,1) = X\_true(:,1);

Fil2 = EKF\_3(X2(:,1), y(:,1), config);

D\_x(:,1) = Fil2.Dx(1,1);

D\_y(:,1) = Fil2.Dx(2,2);

D\_V(:,1) = Fil2.Dx(3,3);

D\_alp(:,1) = Fil2.Dx(4,4);

for i = 1:length(y)

if i > 1

Fil2 = Fil2.Update(y(:,i), config.T, config);

X2(:,i) = Fil2.X;

M\_x(:,i) = sqrt(Fil2.Dx(1,1));

M\_y(:,i) = sqrt(Fil2.Dx(2,2));

M\_V(:,i) = sqrt(Fil2.Dx(3,3));

M\_alp(:,i) = sqrt(Fil2.Dx(4,4));

M\_omeg(:,i) = sqrt(Fil2.Dx(4,4));

end

end

%% Reference

N1 = 63;

N2 = 175;

N\_all = length(uwb(1,:));

Vx\_true\_0 = (18.33 - 4.2)/(24.02-12.02);

Vy\_true\_0 = (9.04 - 1.91)/(24.02-12.02);

% for k = 1:N + 100

for k = 1:N\_all

if k >= 1 && k <= N1

x\_true(k) = 3.0802;

y\_true(k) = 1.3319;

%V\_true(k) = 0;

Vx\_true(k) = 0;

Vy\_true(k) = 0;

alpha\_true(k) = atan2((9.04 - 1.91),(18.33 - 4.2));

omega\_true(k) = 0;

if k == 1

T = 0.1;

else

T = uwb(1,k) - uwb(1,k-1);

end

if k == N1

Vx\_true(k) = Vx\_true\_0;

Vy\_true(k) = Vy\_true\_0;

end

elseif k >= N2

x\_true(k) = x\_true(k - 1);

y\_true(k) = y\_true(k - 1);

Vx\_true(k) = 0;

Vy\_true(k) = 0;

alpha\_true(k) = alpha\_true(k - 1);

omega\_true(k) = 0;

else

T = uwb(1,k) - uwb(1,k-1);

Vx\_true(k) = Vx\_true(k - 1);

Vy\_true(k) = Vy\_true(k - 1);

x\_true(k) = x\_true(k - 1) + Vx\_true(k-1)\*T;

y\_true(k) = y\_true(k - 1) + Vy\_true(k-1)\*T;

alpha\_true(k) = alpha\_true(k-1);

omega\_true(k) = 0;

end

end

%% Static before moving

x\_error\_stat1 = X2(1,1:N1) - x\_true(1:N1);

y\_error\_stat1 = X2(2,1:N1) - y\_true(1:N1);

alpha\_error\_stat1 = X2(4,1:N1) - alpha\_true(1:N1);

mean\_x\_stat1 = mean(x\_error\_stat1);

mean\_y\_stat1 = mean(y\_error\_stat1);

mean\_alpha\_stat1 = mean(alpha\_error\_stat1);

std\_x\_stat1 = std(x\_error\_stat1);

std\_y\_stat1 = std(y\_error\_stat1);

std\_alpha\_stat1 = std(alpha\_error\_stat1);

drmse\_stat1 = 2\*sqrt(std\_x\_stat1^2 + std\_y\_stat1^2);

%% Moving

x\_error\_mov = X2(1,N1:N2) - x\_true(N1:N2);

y\_error\_mov = X2(2,N1:N2) - y\_true(N1:N2);

alpha\_error\_mov = X2(4,N1:N2) - alpha\_true(N1:N2);

mean\_x\_mov = mean(x\_error\_mov);

mean\_y\_mov = mean(y\_error\_mov);

mean\_alpha\_mov = mean(alpha\_error\_mov);

std\_x\_mov = std(x\_error\_mov);

std\_y\_mov = std(y\_error\_mov);

std\_alpha\_mov = std(alpha\_error\_mov);

drmse\_mov = 2\*sqrt(std\_x\_mov^2 + std\_y\_mov^2);

%% Static after moving

x\_error\_stat2 = X2(1,N2:200) - x\_true(N2:200);

y\_error\_stat2 = X2(2,N2:200) - y\_true(N2:200);

alpha\_error\_stat2 = X2(4,N2:200) - alpha\_true(N2:200);

mean\_x\_stat2 = mean(x\_error\_stat2);

mean\_y\_stat2 = mean(y\_error\_stat2);

mean\_alpha\_stat2 = mean(alpha\_error\_stat2);

std\_x\_stat2 = std(x\_error\_stat2);

std\_y\_stat2 = std(y\_error\_stat2);

std\_alpha\_stat2 = std(alpha\_error\_stat2);

velocity\_error = X2(3,:) - sqrt(Vx\_true.^2 + Vy\_true.^2);

drmse\_st2 = 2\*sqrt(std\_x\_stat2^2 + std\_y\_stat2^2);

D\_coord = 0.0939;

%%

figure(2)

% title("x")

plot(uwb(1,:),x\_true(1,:),'k--','linewidth',2)

hold on

plot(uwb(1,:),y(1,:),'b','linewidth',2)

grid on

% plot(X1(1,:),'b','linewidth',2)

plot(uwb(1,:),X2(1,:),'r','linewidth',2)

ylabel('$$X, m$$','Interpreter','LaTeX', 'FontSize',24)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

%title('Зависимость истинной координаты Х и ее оценки от времени','FontSize',18)

legend({"Референс", "Измерение", "Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

figure(3)

plot(uwb(1,:),y\_true(1,:),'k--','linewidth',2)

hold on

plot(uwb(1,:), y(2,:),'b','linewidth',2)

% plot(X1(2,:),'b','linewidth',2)

plot(uwb(1,:),X2(2,:),'r','linewidth',2)

grid on

ylabel('$$Y, m$$','Interpreter','LaTeX', 'FontSize',24)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

legend({"Референс", "Измерение", "Оценка фильтра СШП+ИНС"},'FontSize',10)

set(gca, 'FontSize',17)

figure(4)

plot(uwb(1,:),omega\_true(1,:),'k--','linewidth',2)

hold on

plot(uwb(1,:),y(3,:),'b','linewidth',2)

grid on

% plot(X1(5,:),'b','linewidth',2)

plot(uwb(1,:),X2(5,:),'r','linewidth',2)

xlim([uwb(1,1) uwb(1,end)])

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \dot{\alpha}, rad/s $$', 'Interpreter','LaTeX', 'FontSize',24)

legend({"Референс", "Измерение", "Оценка фильтра СШП+ИНС"},'FontSize',10)

%title('оценка производной курса от времени','FontSize',18)

set(gca, 'FontSize',17)

figure(5)

plot(uwb(1,:),alpha\_true(1,:),'k--','linewidth',2)

grid on

hold on

% plot(X1(5,:),'b','linewidth',2)

plot(uwb(1,:),X2(4,:),'r','linewidth',2)

xlim([uwb(1,1) uwb(1,end)])

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \alpha, rad $$', 'Interpreter','LaTeX', 'FontSize',24)

legend({"Референс", "Оценка фильтра СШП+ИНС"},'FontSize',10)

%title('оценка курса от времени','FontSize',18)

set(gca, 'FontSize',17)

figure(6)

plot(x\_true(1,:),y\_true(1,:),'k--','linewidth',2)

hold on

plot(y(1,:),y(2,:),'b','linewidth',2)

grid on

% plot(X1(1,:),X1(2,:),'b','linewidth',1)

plot(X2(1,:),X2(2,:),'r','linewidth',2)

%title('Истинная траектория и еe оценка','FontSize',18)

xlabel('$$X, m$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$Y, m$$','Interpreter','LaTeX', 'FontSize',24)

legend({"Референс", "Измерение", "Оценка фильтра СШП+ИНС"},'FontSize',10)

%title('оценка производной курса от времени','FontSize',18)

set(gca, 'FontSize',17)

%%

figure(7)

%plot(y(1,:) - uwb(2,:),'k')

%plot(X1(1,:) - uwb(2,:),'b','linewidth',2)

plot(uwb(1,:),X2(1,:) - uwb(2,:),'r','linewidth',2)

hold on

plot(uwb(1,:),3\*D\_coord,'m.','linewidth',8)

plot(uwb(1,:),-3\*D\_coord,'m.','linewidth',8)

plot(uwb(1,:),3\*M\_x,'b--','linewidth',2)

plot(uwb(1,:),-3\*M\_x,'b--','linewidth',2)

grid on

xlim([uwb(1,1) uwb(1,end)])

title('Ошибка оценки координаты Х','FontSize',18)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \widehat{X} - X, m $$','Interpreter','LaTeX', 'FontSize',24)

set(gca, 'FontSize',17)

figure(8)

title("y")

% plot(y(2,:) - uwb(3,:),'k')

%plot(X1(2,:) - uwb(3,:),'b','linewidth',2)

plot(uwb(1,:),X2(2,:) - uwb(3,:),'r','linewidth',2)

grid on

hold on

plot(uwb(1,:),3\*M\_y,'b--','linewidth',2)

plot(uwb(1,:),-3\*M\_y,'b--','linewidth',2)

plot(uwb(1,:),3\*0.0939,'m.','linewidth',4)

plot(uwb(1,:),-3\*0.0939,'m.','linewidth',4)

xlim([uwb(1,1) uwb(1,end)])

title('Ошибка оценки координаты Y','FontSize',18)

xlabel('$$t, s$$','Interpreter','LaTeX', 'FontSize',24)

ylabel('$$ \widehat{Y} - Y, m $$','Interpreter','LaTeX', 'FontSize',24)

set(gca, 'FontSize',17)

**EKF\_3.m**

classdef EKF\_3

properties (Access = public) % атрибуты класса

X; % вектор состояния

Dx; % матрица дисперсий ошибок фильтрации

sigma\_n; % СКО шума наблюдения ToA отметок

sigma\_phi;

sigma\_ksi\_v; % СКО формирующего шума по скорости

sigma\_ksi\_phi;

c; % скорость света

D\_ksi; % матрица формирующих шумов

end

methods (Access = public)

function obj = EKF\_3(X0, y, config) % конструктор

obj.X = X0;

obj.Dx = eye(length(X0));

obj.sigma\_n = config.sigma\_n; % м

obj.sigma\_phi = config.sigma\_phi; % м

obj.sigma\_ksi\_v = config.sigma\_ksi\_v; % м/с

obj.sigma\_ksi\_phi = config.sigma\_ksi\_phi; % м/с

obj.D\_ksi = eye(2);

obj.D\_ksi(1,1) = obj.sigma\_ksi\_v^2;

obj.D\_ksi(2,2) = obj.sigma\_ksi\_phi^2;

end

function obj = Update(obj, y, dt, config) % загрузка новых измерений = обновление фильтра

X\_prev = obj.X;

Dx = obj.Dx;

G = [0 0; 0 0; dt 0; 0 0; 0 dt];

D\_n = config.sigma\_n^2 \* eye(size(y,1));

D\_n(3,3) = config.sigma\_phi^2;

F = obj.make\_F(X\_prev, config, dt);

dF = obj.make\_dF(X\_prev, config, dt);

X\_ext = F;

D\_x\_ext = dF \* Dx \* dF' + G \* obj.D\_ksi \* G';

dS = obj.make\_dS(X\_ext, config);

S = dS\*X\_ext;

K = D\_x\_ext \* dS' \* inv(dS\*D\_x\_ext\*dS' + D\_n);

Dx = D\_x\_ext - K \* dS \* D\_x\_ext;

X\_prev = X\_ext + K\*(y - S);

obj.X = X\_prev;

obj.Dx = Dx;

end

function [ S ] = make\_S( obj, X, config )

for i = 1:size(config.posts,2)

S(i,1) = sqrt((X(1,1) - config.posts(1,i))^2 + (X(3,1) - config.posts(2,i))^2 + (config.hei - config.posts(3,i))^2) + X(5,1);

end

end

function [ F ] = make\_F( obj, X, config, dt )

F(1,1) = X(1,1) + X(3,1) \* cos(X(4,1))\*dt;

F(2,1) = X(2,1) + X(3,1) \* sin(X(4,1))\*dt;

F(3,1) = X(3,1);

F(4,1) = X(4,1) + X(5,1)\*dt;

F(5,1) = X(5,1);

end

function [ dF ] = make\_dF( obj, X, config, dt )

dF(1,:) = [1 0 cos(X(4,1))\*dt -X(3,1) \* sin(X(4,1))\*dt 0];

dF(2,:) = [0 1 sin(X(4,1))\*dt X(3,1) \* cos(X(4,1))\*dt 0];

dF(3,:) = [0 0 1 0 0];

dF(4,:) = [0 0 0 1 dt];

dF(5,:) = [0 0 0 0 1];

end

function [ dS ] = make\_dS( obj, X, config )

dS = [1 0 0 0 0;

0 1 0 0 0;

0 0 0 0 1];

end

function [ delta ] = make\_delta( obj, y, X, config )

X = obj.X;

for i = 1:length(y)

x = config.posts(:,i);

delta(i,1) = y(i)/(1 + (X(2)\*(X(1) - x(1)) + X(4)\*(X(3) - x(2)))/(config.c\*sqrt((X(1) - x(1))^2 + (X(3) - x(2))^2 + (config.hei - x(3))^2)));

end

end

end

end

**config.m**

function [ config ] = Config( )

config.c = 299792458;

config.T = 1;

config.sigma\_t = 1\*1e-9;

config.sigma\_n = config.sigma\_t \* config.c;

config.sigma\_phi = 0.1;

config.sigma\_ksi\_v = 0.01;

config.sigma\_ksi\_phi = 1;

config.posts = [0 0 0; 10 0 0; 0 10 0; 10 10 0]';

config.posts = [-5076.25700228714 12312.9342219411 -8241.14728795100 -1.11097161563963;-11487.2777145567 3509.94201846127 5345.24999149516 -0.445039107080202;160.134190817221 106.173298698501 188.845909395533 124.399999890768];

config.posts = [-5e3 12e3 -8e3 0; -11e3 3.5e3 5e3 0; 10 10 10 10];

config.posts\_number = size(config.posts, 2);

config.hei = 1\*10e3;

config.sigma\_ksi = 100;

config.sigma\_h = 0;

config.max\_coord = 100e3;

config.max\_V = 600;

config.max\_acc = 100;

end

log\_reader.m

function [uwb, gyro, acc, mag, euler] = log\_reader\_1\_0(filename)

if nargin == 0

[file, path] = uigetfile('\*.\*');

filename = fullfile(path,file);

end

f = fopen(filename);

uwb = [];

gyro = [];

acc = [];

mag = [];

euler = [];

while feof(f)==0

s=fgetl(f);

if contains(s,"GYR")

k = size(gyro,2) + 1;

S = split(s);

gyro(:,k) = [str2num(S{2,1})/1e9; str2num(S{3,1}); str2num(S{4,1}); str2num(S{5,1})];

end

if contains(s,"EYR")

k = size(euler,2) + 1;

S = split(s);

euler(:,k) = [str2num(S{2,1})/1e9; str2num(S{3,1}); str2num(S{4,1}); str2num(S{5,1})];

end

if contains(s,"ACC")

k = size(acc,2) + 1;

S = split(s);

acc(:,k) = [str2num(S{2,1})/1e9; str2num(S{3,1}); str2num(S{4,1}); str2num(S{5,1})];

end

if contains(s,"MAG")

k = size(mag,2) + 1;

S = split(s);

mag(:,k) = [str2num(S{2,1})/1e9; str2num(S{3,1}); str2num(S{4,1}); str2num(S{5,1})];

end

if contains(s,"server")

k = size(uwb,2) + 1;

S = split(s);

uwb(:,k) = [str2num(S{2,1})/1e9; str2num(S{4,1}); str2num(S{5,1}); str2num(S{6,1})];

end

end

if size(uwb,2) > 0

T\_min = min([acc(1,1) gyro(1,1) mag(1,1) uwb(1,1) euler(1,1)]);

else

T\_min = min([acc(1,1) gyro(1,1) mag(1,1) euler(1,1)]);

end

acc(1,:) = acc(1,:) - T\_min;

gyro(1,:) = gyro(1,:) - T\_min;

mag(1,:) = mag(1,:) - T\_min;

euler(1,:) = euler(1,:) - T\_min;

if size(uwb,2) > 0

uwb(1,:) = uwb(1,:) - T\_min;

subplot(411)

title("UWB")

plot(uwb(1,:),uwb(2:3,:),'linewidth',2)

grid on

xlabel('t, sec')

ylabel('meter')

end

subplot(412)

title("Euler")

plot(euler(1,:),euler(2:4,:),'linewidth',2)

grid on

xlabel('t, sec')

ylabel('rad')

subplot(413)

title("acc")

plot(acc(1,:),acc(2:4,:),'linewidth',2)

grid on

xlabel('t, sec')

ylabel('С…Р·')

subplot(414)

title("gyro")

plot(gyro(1,:),gyro(2:4,:),'linewidth',2)

grid on

xlabel('t, sec')

ylabel('С…Р·')

fclose(f);

end