

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Кафедра радиофизики и цифровых медиа технологий

ЯНЦЕВИЧ

Юлия Андреевна

АНАЛИЗ МЕТОДОВ ТРИЛАТЕРАЦИИ ДЛЯ ОПРЕДЕЛЕНИЯ
ПОЛОЖЕНИЯ МОБИЛЬНОГО УСТРОЙСТВА

Дипломная работа

Научный руководитель:
старший преподаватель
В.А.Пахомов

Допущена к защите

«___» _____ 2018 г.

Зав. кафедрой радиофизики и цифровых медиа технологий
И.Э. Хейдоров

Минск, 2018

Оглавление

<u>ОГЛАВЛЕНИЕ</u>	2
<u>РЕФЕРАТ</u>	3
<u>ВВЕДЕНИЕ</u>	6
<u>ГЛАВА 1. ТЕХНОЛОГИИ ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ</u>	9
<u>ГЛАВА 2. ТРИЛАТЕРАЦИЯ</u>	24
<u>ГЛАВА 3. ПРОБЛЕМА ПСЕВДОДАЛЬНОСТИ В ТРИЛАТЕРАЦИИ</u>	27
<u>ГЛАВА 4. ПРАКТИЧЕСКАЯ ЧАСТЬ</u>	32
<u>ЗАКЛЮЧЕНИЕ</u>	44
<u>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</u>	45
<u>ПРИЛОЖЕНИЕ 1</u>	47
<u>ПРИЛОЖЕНИЕ 2</u>	51
<u>ПРИЛОЖЕНИЕ 3</u>	54

Реферат

Дипломная работа 57 с., 22 рис., 2 табл., 17 источников

Трилатерация, GPS, LBS, навигация, позиционирование, местоположение, псевдодальность, координаты объекта.

Объект исследования: трилатерационные алгоритмы для получения координат местоположения объекта по 3 передатчикам.

Цель: изучить различные трилатерационные методы позиционирования, выбрать несколько из них и реализовать в среде Matlab, сравнить их погрешности определения координат и выбрать метод, который будет давать максимально точные результаты нахождения местоположения.

Методы исследования: реализация методов в программной среде Matlab, а также использование программных возможностей данной программы для проведения статистического анализа и вывода необходимых графиков.

В результате проведенного исследования были изучены все существующие трилатерационные алгоритмы и из них были выбраны 3 наиболее популярных метода для определения наиболее точных координат объекта, среди которых: обычный трилатерационный метод, метод Бэнкрофта и итерационный метод. Все эти 3 метода были реализованы в среде Matlab. Для анализа данных методов было взято помещение размерами 100 x 100 (метров). Проверка методов сначала проводилась для частных случаев исходных данных с координатами объекта для получения графиков с результатами измерений в двухмерном пространстве. Затем была взята выборка данных, состоящая из 10 000 комплектов случайных значений. Для сравнения методов координаты 3 передатчиков были неизменны для всех измерений. Таким образом на вход программе у нас подавались настоящие координаты объектов выбранные случайно в рамках исследуемого помещения, а также случайные данные со значениями погрешностей. Из них составлялись так называемые псевдодальности, которые и использовались для дальнейших исследований трилатерационных алгоритмов. Исследования показали, что наиболее точные результаты в поиске координат нужного нам объекта можно получить с методом Бэнкрофта, где среднее значение погрешностей составило 1.0275 м, а среднеквадратичная погрешность данного метода составила 2.6073 м. Немного хуже результаты оказались у итерационного метода, где средняя и среднеквадратичная ошибки составили соответственно 2.1965 м и 1.4423 м. И самые отдаленные результаты от настоящих координат показал обычный трилатерационный метод со значениями погрешностей средней - 3.9259 м, среднеквадратичной – 2.8526 м.

Рэферат

Дыпломная работа 57 с., 22 мал., 2 табл., 17 крыніц.

Трылатэрацыя, GPS, LBS, рух, пазіцыянаванне, месцазнаходжанне, псеўдадавальнасць, каардынаты аб'екта.

Аб'ект даследавання: трылатэрацыйныя алгарытмы для атрымання каардынатаў месцазнаходжання аб'екта па 3 перадатчыкам.

Мэта: даследаваць розныя трылатэрацыйныя метады пазіцыянавання, выбраць некалькі з іх і рэалізаваць у асяроддзі Matlab, параўнаць іх хібнасці вызначэння каардынатаў і выбраць метады, які будзе даваць максімальна дакладныя вынікі знаходжання месцазнаходжання.

Метады даследавання: рэалізацыя метадаў у праграмах асяроддзі Matlab, а таксама выкарыстанне праграмных магчымасцяў гэтай праграмы для правядзення статыстычнага аналізу і вывадзення неабходных графікаў.

У выніку праведзенага даследавання былі вывучаны ўсе існуючыя трылатэрацыйныя алгарытмы і з іх былі выбраны 3 найбольш папулярных метады для вызначэння найбольш дакладных каардынатаў аб'екта, сярод якіх: звычайны трылатэрацыйны метады, метады Бэнкрофта і ітэрацыйны метады. Усе гэтыя 3 метады былі рэалізаваны ў асяроддзі Matlab. Для аналізу дадзеных метадаў было ўзята памяшканне памерамі 100 x 100 метраў. Праверка метадаў спачатку праводзілася для прыватных выпадкаў зыходных дадзеных з каардынатамі аб'екта для атрымання графікаў з вынікамі вымярэнняў у двухмерным прасторы. Затым была ўзята выбарка дадзеных, якая складаецца з 10 000 камплектаў выпадковых значэнняў. Для параўнання метадаў каардынаты 3 перадатчыкаў былі нязменныя для ўсіх вымярэнняў. Такім чынам на ўваход праграме ў нас падаваліся сапраўдныя каардынаты аб'ектаў выбраныя выпадкова ў рамках доследнага памяшкання, а таксама выпадковыя дадзеныя са значэннямі хібаў. З іх складаліся так званыя псеўдадавальнасці, якія і выкарыстоўваліся для далейшых даследаванняў трылатэрацыйных алгарытмаў. Даследаванні паказалі, што найбольш дакладныя вынікі ў пошуку каардынатаў патрэбнага нам аб'екта можна атрымаць з метадамі Бэнкрофта, дзе сярэдняе значэнне хібаў складала 1.0275 м, а сярэднеквадратычная хібнасць дадзенага метада складала 2.6073 м. Крыху горш вынікі апынуліся ў ітэрацыйным метады, дзе сярэдняя і сярэднеквадратычная памылкі складалі адпаведна 2.1965 м і 1.4423 м. І самыя аддаленыя вынікі ад сапраўдных каардынатаў паказаў звычайны трылатэрацыйны метады са значэннямі хібаў сярэдняй - 3.9259 м, сярэднеквадратычнай – 2.8526 м.

Abstract

Diploma work 57 p., 22 fig., 2 tables, 17 sources.

Trilateration, GPS, LBS, navigation, positioning, location, pseudo-range, object coordinates.

Research object: trilateration algorithms for obtaining coordinates of location of an object on 3 transmitters.

Purpose: to study various trilateration methods for positioning, choose some of them and implement them in the Matlab environment, compare their errors of determination of coordinates and choose a method which will give the most exact results of finding the location.

Research methods: implementation of methods in the program Matlab environment and also use the program opportunities of this program for carrying out the statistical analysis and derivation of necessary graphs.

In the result of the conducted research, all existing trilateration algorithms have been studied and 3 most popular methods for determination of the most exact coordinates of an object have been chosen, those are: the usual trilateration method, Bancroft's method and iterative method. All these 3 methods have been implemented in Matlab environment. For the analysis of these methods a 100 x 100 (meters) room has been taken. Testing of the methods was first performed for special cases of basic data with object coordinates for obtaining graphs with results of measurements in two-dimensional space. After that, the selection of data consisting of 10 000 sets of random values has been taken. For comparison of methods, the coordinates of 3 transmitters were unchanged for all measurements. Thus, on the entrance program we were given the real coordinates of objects chosen randomly within the analysed room and also the random data with values of errors. So-called pseudo-ranges were formed from them, which were used for further researches of trilateration algorithms. Researches have shown that the most exact results in search of coordinates of a needed object can be achieved with Bancroft's method where the average value of errors was 1.0275 m, while the mean square error of this method was 2.6073 m. The iterative method has given a little worse results, where average and mean square errors were 2.1965 m and 1.4423 m respectively. The most distant results from the real coordinates were shown by the usual trilateration method with values of average error - 3.9259 m, mean square error – 2.8526 m.

Введение

В настоящее время наблюдается интерес к возможности получения информации о местоположении того или иного объекта. Для этого используются системы позиционирования, как в глобальных координатах, так и в локальных, которые подробнее будут рассмотрены в главе 1. К системам позиционирования в глобальных координатах мы относим такие системы, как российская ГЛОНАСС, европейская Galileo и, конечно же, самая популярная на сегодняшний день глобальная система навигации Global Positioning System (GPS), которая имеет широкий охват и достаточно большую точность определения местоположения. Однако данные системы имеют и ряд недостатков, в числе которых зависимость от погодных условий, которая не даёт определять местоположение со стопроцентной точностью.

Это всё привело к созданию различных систем позиционирования в локальных координатах. Среди них можно выделить и позиционирование при помощи беспроводных сетей: Wi-Fi, GSM, Bluetooth. Все беспроводные сети имеют схожие методики позиционирования и заключаются они в анализе параметров передающих станций на приёмнике. В качестве передатчика в GSM-сетях выступает станция цифровой сети, в Bluetooth – специальные маячки (так называемые “биконы”), а в Wi-Fi сетях – точка доступа. А в качестве приёмника во всех из вышеперечисленных беспроводных сетях зачастую используется датчик на смартфоне.

Измерение расстояния беспроводным способом – это базовая операция в системах локального позиционирования.

Обычно данная задача выполняется радиолокационными станциями, которые необходимы для радиообнаружения наземных, морских и воздушных объектов с помощью анализа характеристик отраженных от объекта радиоволн. В традиционных радарах при этом используется пассивное отражение радиосигналов.

В современных беспроводных технологиях используются частотные диапазоны технологий используются частотные диапазоны, с помощью которых можно работать без получения специальных разрешений. В этих случаях на объектах наблюдения размещают цифровые беспроводные приёмопередатчики-метки.

Эти приёмопередатчики реализуют измерения расстояния беспроводным способом по следующим методам:

- по времени распространения сигнала
- по уровню сигнала
- по фазе сигнала [14].

Основным принципом большинства систем позиционирования является определение положения объекта на основе анализа измеренных тем или иным образом расстояний от объекта до точек с известными координатами (спутников, базовых станций, маяков). Существует множество методов решения этой задачи. Наиболее точные результаты, однако, показывают методы, в основе которых лежит трилатерационный метод и в главах 2 и 3 будут подробнее расписаны основы трилатерации, а также проблемы возникающие при решении навигационной задачи с помощью трилатерации. В данной работе реализовано несколько алгоритмов трилатерации и выполняется их сравнительный анализ. В главе 4 можно будет подробнее ознакомиться с данными методами, а также увидеть сравнительную таблицу для реализованных в данной работе алгоритмов.

Цель работы: реализация и исследование алгоритмов трилатерации для определения местоположения мобильного устройства.

Задачи данной дипломной работы: проанализировать существующие алгоритмы трилатерации, реализовать выбранные трилатерационные алгоритмы в среде Matlab, решающие проблему псевдодальности, и при помощи численного моделирования определить погрешности выбранных методов.

Глава 1

Технологии определения местоположения

1.1. Спутниковые системы

Спутниковые навигационные системы, например, GPS, ГЛОНАСС и Galileo, создавались специально для навигационных целей и позволяют человеку получить данные о его местоположении в пространстве, а именно: узнать его географические координаты, а также узнать высоту над уровнем моря, на которой он в данный момент находится. [17]

GPS, ГЛОНАСС и Galileo – всё это аналоги из разных стран. Соответственно, и рассматривать все эти системы позиционирования отдельно – не имеет смысла. Принципы их работы рассмотрим на примере GPS.

Базовую идею технологии GPS составляет использование динамических опорных ориентиров, образованных системой искусственных спутников Земли. Эти спутники вращаются вокруг Земли по шести орбитам с высотой порядка двадцати тысяч километров и периодом обращения около 12 часов. Угол, под которым расположены орбиты, составляет примерно 60 градусов, примерно так, как это показано на рис 1.1. Спутники непрерывно излучают специальные навигационные радиосигналы в диапазоне около 1,5 ГГц. Именно благодаря такой структуре достигается относительно равномерное покрытие всей поверхности Земли искомыми радиосигналами.[14]

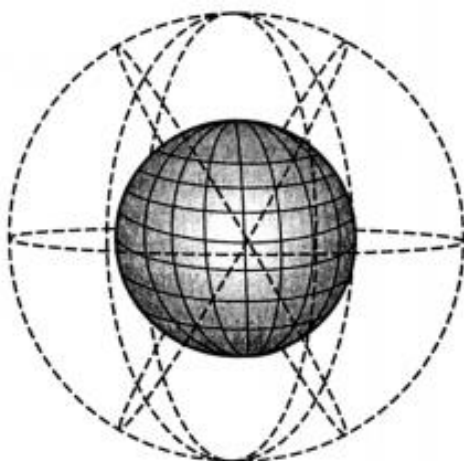


Рис.1.1.1

Возможность использования спутников в качестве опорных ориентиров (реперов) основано на том, что орбиты достаточно устойчивы, вследствие чего местоположение спутников известно и предсказуемо в высокой степени.

Каждая спутниковая система навигации состоит из трёх основных сегментов:

- Пользовательский сегмент;
- Наземный сегмент управления и контроля;
- Космический сегмент или орбитальная группировка;

В качестве наземного сегмента управления и контроля выступает ряд наземных станций, производящих постоянный мониторинг орбит всех спутников и корректирующих информацию об их текущем местоположении.

Пользовательский сегмент – это высокочувствительные приемники, объединенные с вычислительным блоком и различными модулями отображения информации. Принимая радиосигналы от находящихся в зоне прямой видимости спутников, эти устройства пересчитывают полученные данные и в режиме реального времени определяют координаты своего местоположения.[14]

Если у нас есть постоянная скорость, значение которой нам известно, то мы можем измерять пройденный путь временными интервалами. Исходя из этого, можно прийти к выводу, что и расстояние от спутника до GPS-приемника может измеряться временем прохождения радиоимпульса от точки передачи до точки приема. Но, однако, если мы будем измерять расстояние таким методом, то необходимо будет установить жесткие ограничения на синхронность излучения радиосигналов всеми навигационными спутниками.

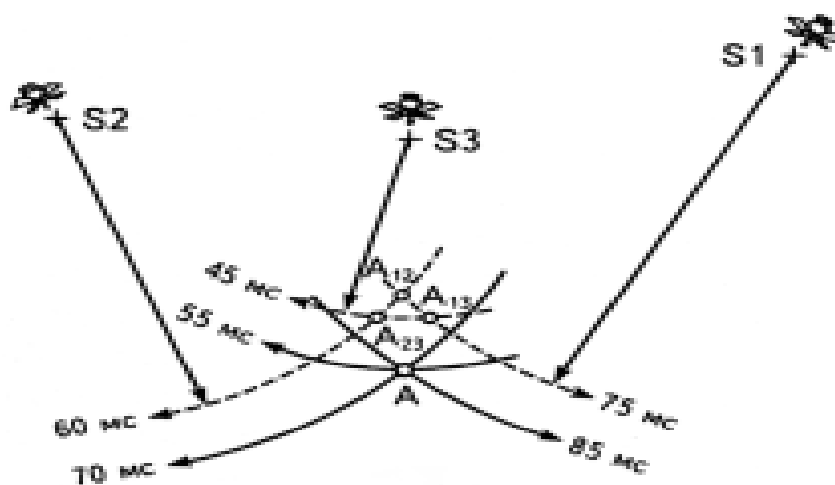


Рис. 1.1.2.

Именно поэтому на каждом спутнике устанавливаются сверхточные атомные часы. Однако, так как данное оборудование стоит не малых денег, для того, чтобы избежать необходимости его установки на абонентских устройствах, используют сигналы третьего спутника и специальные алгоритмы определения временных поправок.

За устранение противоречий отвечает вычислительный блок. Он вносит поправку в понимание времени приемником и затем повторяет вычисления. Если расстояние между рассчитанными координатами точек увеличилось, то изменяется знак поправки, и все пересчитывается. В случае же, если расстояние между координатами точек уменьшилось, то процесс продолжается с некоторым переменным шагом до тех пор, пока координаты точек не совпадут с заданной точностью.

Из всего вышесказанного можно прийти к выводу, что для определения верных координат местоположения GPS-приемника на поверхности необходимо иметь в наличии не менее трех спутников, а при неизвестной высоте над уровнем моря – не менее четырех спутников.

Что касается навигационных данных, передаваемых спутником, они могут быть 3 видов:

- Данные о параметрах орбит и местоположении всех спутников;
- Специальные “дальномерные” сигналы, которые используются для изменения временных интервалов;
- Высokоточные данные о корректировке параметров орбиты и системных часов именно этого спутника.

Кроме навигационных данных, спутниками передается и набор служебной информации: свои номера, данные внутренней диагностики, значения системного времени и т.д.

Точные координаты могут быть вычислены для места на поверхности Земли по измерениям расстояний от группы спутников (если их положение в космосе известно). В этом случае спутники являются пунктами с известными координатами. Предположим, что расстояние от одного спутника известно и мы можем описать сферу заданного радиуса вокруг него.

Если мы знаем также расстояние и до второго спутника, то определяемое местоположение будет расположено где-то в круге, задаваемом пересечением двух сфер.

Третий спутник определяет две точки на окружности. Теперь остаётся только выбрать правильную точку. Однако одна из точек всегда может быть отброшена, так как она имеет высокую скорость перемещения или находится на или под поверхностью Земли. Таким образом, зная расстояние до трёх спутников, можно вычислить координаты определяемой точки

Что касается точности результатов, при использовании всех возможностей GPS-технологий, применение атомных стандартов синхронизации системного времени с погрешностью несколько наносекунд обеспечивает точность определения координат на уровне нескольких сантиметров.

Однако на практике точность вычислений оказывается намного ниже по ряду причин:

- Измерение расстояний через задержки времени предполагает постоянную скорость распространения радиоволн. Однако при входе в атмосферу скорость их распространения меняется. Эти изменения зависят от погоды (влажность, давление и т.п.) и от степени ионизации ее верхних слоев.
- Следует иметь в виду возможные отражения радиоволн. При этом их путь прохождения уже не является прямолинейным и не соответствует кратчайшему расстоянию между двумя точками. В таком случае в точке приема присутствуют однотипные радиоволны, дошедшие до нее с разной задержкой времени.
- Итоговая погрешность измерений зависит как от числа принимаемых спутников, так и от геометрии их расположения.

Кроме этих причин, можно также выделить и субъективные причины. Способы кодирования измерительных сигналов, используемые в настоящее время, накладывают ограничения на точность измерений. Обработка данных вычислительным блоком также производится с конечным числом разрядов. Кроме этого количество проводимых итераций при коррекции часов приемника может ограничиваться как системными, так и программными средствами. И наконец, системное время несложно специально исказить по командам специальной станции. [14]

1.2 Location-Based Service

Location-based service (LBS) – программный сервис, использующий данные о локации для управления какими-либо функциями. Под «поиском» в сотовой LBS подразумевается не определение местоположения непосредственно мобильного телефона, а привязка его местоположения к ориентирам, нанесённым на электронную карту LBS-системы оператором сотовой сети или поставщиком услуги.

По принципу работы LBS-мониторинг напоминает GPS, но источником сигнала выступает не спутник, а ближайшие GSM-станции оператора сотовой связи. С использованием LBS точное местоположение определить сложнее, чем при помощи GPS. Точность определения зависит от густоты покрытия и сети базовой станции, какие существуют текущие местные радиоусловия и конфигурации сот. [12]

В отличие от спутниковой системы Global Positioning System (GPS), технология LBS позволяет определять местоположение активного терминала, используя собственные ресурсы беспроводных сетей на основе программных решений, которые поддерживают как новые, так и старые модели телефонов. Сервисы, которые дают возможность определять местоположение конкретного терминала, могут быть полезны, к примеру, для пожарной службы, скорой помощи, правоохранительных органов, оперативных отрядов реагирования МЧС. Кроме всего этого, их в полной мере могут оценить абоненты, которым услуги LBS смогут помочь сориентироваться на незнакомой местности, получить информацию о местонахождении ближайшего кафе, заправочной станции, аптеки, либо же узнать, где в данный момент находится другой абонент, пользующийся услугой (при его согласии и с условием, что оба абонента находятся в зоне покрытия оператора).[16]

В качестве способов определения координат мобильного терминала могут использоваться самые разнообразные комбинации программных либо аппаратных решений. Последние, безусловно, можно считать более эффективными, поскольку при таком подходе есть возможность применения телефонов со встроенными GPS-приемниками спутниковой навигации. Кроме этого, возможна установка дополнительного оборудования на базовых станциях сотового оператора.[16]

Программные решения наиболее экономически обоснованы, но при этом они отличаются низкой точностью позиционирования. Тем не менее, они

позволяют выполнять необходимые задачи с использованием уже существующего стандарта связи GSM без применения дополнительного оборудования. Следует понимать, что в данном случае речь идет не о вычислении точных координат, а об идентификации положения владельца мобильного телефона на местности либо на электронной карте. В случае использования программных решений местоположение сотового терминала просчитывается на основе колебаний трех основных параметров радиосигнала: амплитуды в месте приема, направления прихода сигнала и времени его задержки при распространении. Амплитуда радиосигналов указывает на расстояние между передатчиком и приемником, однако из-за влияния множества факторов не обеспечивает необходимой точности и служит в качестве вспомогательного способа позиционирования.[16]

Направление прихода сигнала может быть определено автоматически по различию фаз сигналов на элементах антенной решетки, установленной на базовой станции сети оператора. Чтобы определить местоположение мобильного аппарата, требуется пересечение как минимум двух точек пеленга. Если известен момент радиопередачи и приемник базовой станции зафиксировал время регистрации сигнала, то вычислить расстояние между приемником и передатчиком достаточно просто. Но точность измерений в этом случае можно гарантировать только при условии жесткой временной синхронизации всех сетевых элементов. [16]

Основные методы поиска в LBS:

- Cell of Origin
- TOA
- A-GPS
- OTD

Общее для всех способов - то, что они опираются на использование ресурсов сети сотовой связи, при этом сокращая объем ресурса, который можно было бы задействовать для передачи трафика. Кроме того, на сети должно быть установлено дополнительное программное обеспечение и оборудование, а точность определения местоположения остается ниже, чем при использовании спутниковых или комбинированных способов.

1.2.1 Технология Cell of Origin

Cell of Origin (COO) – это технология мобильного позиционирования для поиска абонента сотовой связи. Она вычисляет координаты абонента по таблице расположения и радиуса ячеек сотовой сети в которой мобильный телефон определяется определенной базовой станцией. (Рис.1.2.1.1)

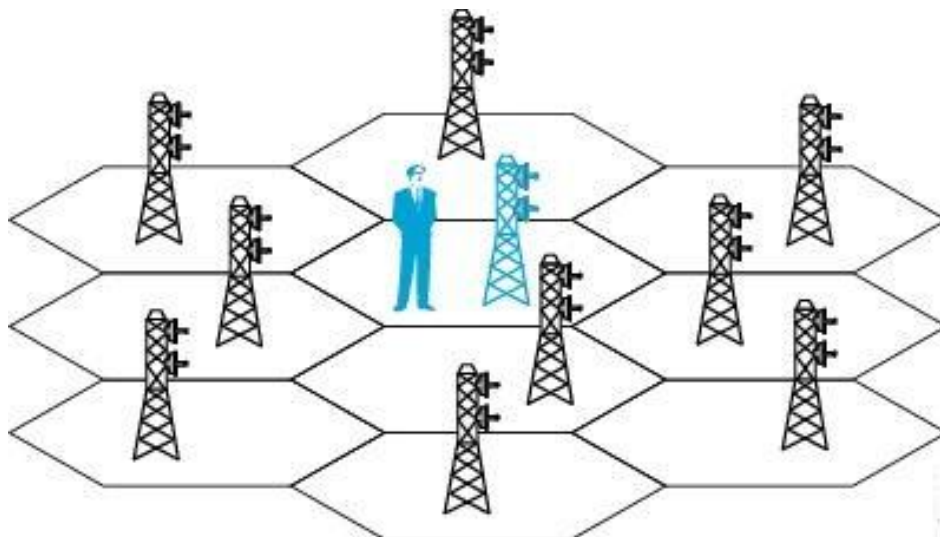


Рис. 1.2.1.1

Принцип действия данной технологии заключается в том, что сотовый телефон (или модуль сотовой связи) знает, каким приемопередатчиком базовой станции он обслуживается. Затем имея базу данных координат передатчиков базовой станции можно приблизительно определить своё местоположение.

Точность определения местоположения в данной технологии зависит от густоты сети базовых станции, конфигураций сот и от количества радиопомех. Если же в центре в центре крупных городов точность будет достигать несколько сотен метров – то на окраинах, небольших городах – около километра. За пределами же зоны покрытия сотовой сети данный метод не будет работать вовсе. [14]

1.2.2 Метод TOA

TOA (Time of Arrival – по времени прибытия сигнала) – метод определения местоположения GSM-терминала, который измеряет и сравнивает временные интервалы прохождения сигнала от мобильного телефона до нескольких базовых станций, используя абсолютные значения установленного времени. Затем, высчитывая разность времени прибытия служебного сигнала на «видимые» соты, сетевой узел рассчитывает местоположение абонента при помощи триангуляционного алгоритма. (Рис.1.2.2.1) [14]

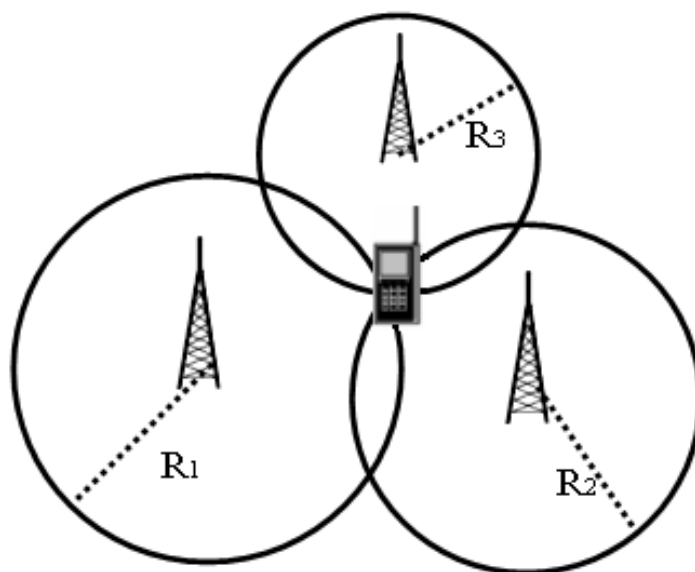


Рис. 1.2.2.1

Методы TOA обладают некоторыми преимуществами по сравнению с другими методами, в числе которых низкая стоимость и простота использования. Данным TOA доступна синхронизация времени на разных базовых станциях, при этом, не требуя сложного оборудования.

Точность позиционирования данного метода может достигать в крупных городах лишь до 125 м, а на окраинах до километра.

1.2.3 Технология OTD

OTD (Observed Time Difference – наблюдаемая разница во времени) – технология, схожая с TOA. В данной технологии измеряют и сопоставляют интервалы времени прохождения сигнала от нескольких базовых станций до мобильного аппарата, при этом используя относительные значения по времени.

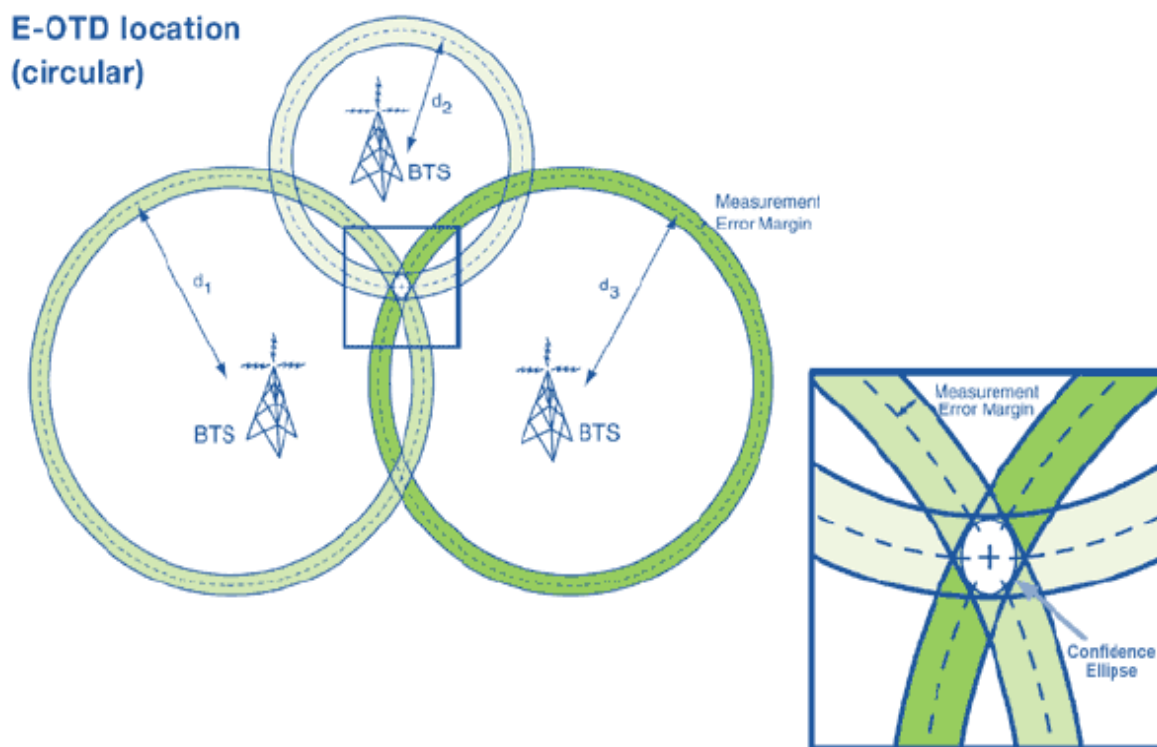


Рис.1. 2.3.1

В сетях GSM более популярен метод E-OTD, когда при анализе результатов достижения сервисных команд сотового телефона учитываются временные интервалы прохождения сигналов ещё до вспомогательных точек LMU. Это позволяет снизить погрешность определения местоположения абонента до нескольких десятков метров в зависимости от числа LMU-узлов вокруг. [15]

1.2.4 Технология A-GPS

A-GPS (Assisted Global Positioning System – Вспомогательная Глобальная система позиционирования) – технология, основанная на встраивании в мобильные телефоны GPS-модуля и переносе части вычислительных функций на Mobile Location Center для ускорения определения местоположения устройства и снижения энергопотребления. (Рис 1.2.4.1)

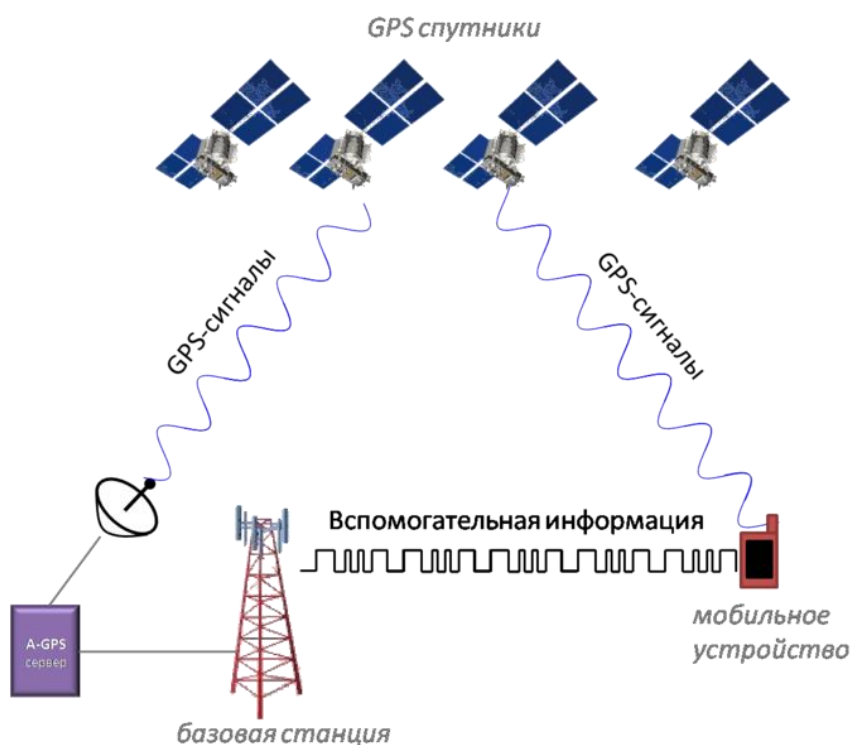


Рис 1.2.4.1

Метод использует систему глобального позиционирования с поддержкой от сетевой инфраструктуры. В мобильный телефон должен быть встроен модуль GPS, а часть вычислительных функции переносится на MLC для снижения энергопотребления и ускорения определения местоположения [13].

Расчет координат происходит следующим образом. GPS-сигнал поступает на мобильный телефон, затем он передается на БС, которая рассчитывает координаты телефона и передает их обратно. Базовые станции, принимающие сигнал радиотелефона, должны быть оснащены блоками определения местоположения LMU. Процесс позиционирования инициируется абонентом и

заканчивается передачей абоненту данных о месте его нахождения. При этом сеть упрощает терминалу поиск спутников, указывая, какие именно надо искать.

Более того, в некоторых сетях, (к примеру, CDMA) базовые станции могут быть оснащены своими GPS-антеннами и, соответственно, приемниками. Они могут выступать в качестве ретранслятора сигнала от спутников, при этом по понятным причинам возникает дополнительная погрешность, и точность определения координат уменьшается (от 5 до 400 м). Таким образом, достигается работа сервиса даже там, где в прямой видимости нет спутников или сигнал достаточно слабый (помещения в том числе). Время расчета текущих координат составляет от 20 до 40 секунд, в среднем это все-таки 20 секунд.

Перекладывание части функций на сетевое оборудование позволяет уменьшить размеры телефонов, увеличить их время работы — сегодня они практически ничем не отличаются от обычных сотовых аппаратов. Правда, это модели высшего ценового диапазона, так как требуется не только изменение самой трубки, но и сетевой инфраструктуры. [14]

Среди преимуществ данной технологий, отличительных от GPS — быстрое определение координат сразу после включения, а также повышение чувствительности приёма слабых сигналов в «мёртвых зонах».

Среди недостатков — не работает вне зоны покрытия сотовой сети, а также потребление трафика для синхронизации.

Примерная точность определения местоположения в центрах городов — от 20 до 500 метров. При ухудшении условий и увеличении расстояния до вышек точность снижается и может составлять около 1,5 – 2 км. [14]

1.3 Wi-Fi Positioning System (WPS)

Wi-Fi Positioning System(WPS) — навигационная система определения местоположения, основанная на определении координат объекта по Wi-Fi точкам. В связи с тем, что растёт количество Wi-Fi точек в городах — растёт и точность определения местоположения. При этом подключение к самой сети не обязательно. Смартфон сканирует пространство на наличие Wi-Fi-сетей и вносит данные о своём местоположении в их базу данных. Информация из БД распространяется между всеми устройствами, что и помогает обнаружить своё местоположение.

По сравнению со спутниковой навигацией поиск местоположения при помощи беспроводных сетей менее точный на открытой местности. Однако в закрытых помещениях данный метод показывает себя лучше, чем GPS.

Выбор беспроводной сети для целей позиционирования в первую очередь зависит от требуемых точностей и финансовых возможностей заказчика. Так как спрос на сотовую связь и Интернет постепенно ведет к повсеместному внедрению цифровой сотовой сети и сети Wi-Fi, финансовые затраты на эти две технологии минимальны. Из них Wi-Fi сети дают более высокую (метровую) точность, в связи с чем по праву занимают лидирующую позицию по популярности среди всех LPS. Хотя LAN сети заметно уступают всем беспроводным сетям по точности (от сотен до десятков метров), они незаменимы на обширных пространствах, где нет необходимости в точном определении местоположения.

Беспроводное позиционирование может быть реализовано разными способами:

1. Метод распознавания шаблона.

В основу данного метода положено то, что в каждой точке устройство видит уникальную радио картину. Мобильное устройство сканирует радио обстановку – точки доступа и уровень их сигналов, затем сверяет получившуюся схему радиосигналов со списком шаблонов и находит координату устройства. Для настройки всей сети проводится длительный процесс сканирование эфира всего помещения, а также регулировка данных, т.к. в реальности происходят постоянные изменения.



Рис.1.3.1

2. По точке доступа, к которой присоединен клиент.

Из преимуществ данного метода – простота. Однако точность оставляет желать лучшего. Метод лишь позволяет определить присутствие объекта, чем спозиционировать его.

3. Триангуляция.

Данный метод определяет силу сигнала от клиента на трех – четырех точках доступа Wi-Fi в зоне пересечения возможного расположения клиента относительно каждой точки спозиционировать устройство. (Рис 1.3.1)

К недостаткам данного подхода можно отнести то, что препятствия на пути радиосигнала будут мешать точности определения координаты.

Точность определения координат может вычисляться до 5-7м.

4. Ангуляция или позиционирование с определением угла входящего сигнала.

В данном методе внешний модуль точного позиционирования, подключенный к модульной точке доступа, со специальной антенной позволяет дополнительно определить угол, под которым пришел сигнал и сузить сегмент возможного нахождения Wi-Fi клиента до луча. Физически устройство представляет из себя точку доступа с включенным модулем точного позиционирования и специальной антенной из 32 антенн, каждая из которых получает сигнал иначе, чем соседняя. Алгоритм позволяет из собранных данных рассчитать угол, под которым пришёл сигнал.

Примерная точность определения местоположения – около метра.

1.4 iBeacon, Eddystone

Рассмотрим принцип работы технологии iBeacon. В основе его работы лежит взаимодействие маячка и мобильного устройства. Сначала маячок (передатчик) периодически (с интервалом от долей секунды до нескольких секунд) передает на мобильное устройство пакеты установки соединения, не устанавливая само соединение, стандарта Bluetooth LE, которые содержат

помимо заголовка полезную информацию: UUID (128-битный уникальный идентификатор группы маяков, определяющий их тип или принадлежность одной организации), Major и Minor (16-битное беззнаковое значение, с помощью которого можно группировать маяки с одинаковым UUID/UUID и Major) и уровень сигнала в 1 метре от передатчика (8-битное знаковое целое – значение индикации уровня принимаемого сигнала (RSSI), откалиброванное на расстоянии 1 метр от приемника, используемое для определения близости маяка к приёмнику (мобильному устройству).

Для того, чтобы определить расстояние до маяка в данной технологии используется измеренное на приёмнике реальное значение RSSI, с объявленным маяком значением уровня сигнала в метре от передатчика. В зависимости расстояния между значением уровня сигнала на расстоянии метра и значением RSSI.

Для определения местоположения в пространстве необходимы несколько таких маяков и использование метода трилатерации или же метода «отпечатков». Под отпечатками тут понимается предварительная запись векторов из RSSI от идентифицированных маяков для некоторого набора точек в помещении. При навигации измерения сравниваются с этим набором, и выбирается самая подходящая по «отпечатку» точка.

Уровень сигнала на таких приемниках зависит от расстояния лишь в случае беспрепятственного распространения радиосигнала. В реальных помещениях на RSSI влияет многолучевое распространение радиоволн, что может приводить как к заниженным, так и к завышенным уровням сигнала. Для получения более точных оценок требуется применение алгоритмов сглаживания. [14]

Eddystone имеет те же принципы работы, что и iBeacon. Основное его отличие – это работа на любой операционной системе с поддержкой технологии BLE (Bluetooth Low Energy), в то время как iBeacon работает преимущественно на устройствах Apple. Также EddyStone в отличие от iBeacon позволяет отсылать уведомления пользователю смартфона даже в случае, если у пользователя не установлено специальное мобильное приложение.

Технология Bluetooth Low Energy по сравнению с обычной Bluetooth-технологией позволяет меньше затрачивать заряд батареи. При этом радиус действия ограничивается 10 м, а скорость передачи данных — 1 Мбит/с. (Радиус действия Bluetooth составляет примерно 100м). [7]

1.5 Сравнительная таблица

В представленной ниже Таблице 1 были выделены все основные характеристики технологий определения местоположения мобильных устройств: их преимущества и недостатки.

Технология		Потребляемая энергия	Поддержка на ОС	Необходимость установки доп. приложений	ПРИМЕНЕНИЕ	РАДИУС ДЕЙСТВИЯ	Точность позицион., м
GPS		много	Все ОС (поддержка GPS)	-	НА МЕСТНОСТИ		≈13
LBS	COO	мало	Все ОС	-	НА МЕСТНОСТИ	В ЗОНЕ ДЕЙСТВИЯ БС	≈1000-35000
	TOA	мало	Все ОС	-	НА МЕСТНОСТИ	В ЗОНЕ ДЕЙСТВИЯ БС	≈125-1000
	OTD	мало	Все ОС	-	НА МЕСТНОСТИ	В ЗОНЕ ДЕЙСТВИЯ БС	≈1000
	A-GPS	среднее	Все ОС (поддержка GPS)	-	НА МЕСТНОСТИ	В ЗОНЕ ДЕЙСТВИЯ БС	≈1500-2000
WPS		мало	Все ОС (поддержка Wi-Fi)	-	ВНУТРИ ПОМЕЩЕНИЙ	30-200м	≈5-10
iBeacon		Мало	APPLE	-	ВНУТРИ ПОМЕЩЕНИЙ	10м	≈3м
			ANDROID4+	+			
EddyStone		Мало	Все ОС	-	ВНУТРИ ПОМЕЩЕНИЙ	10м	≈3м

Таблица 1

В данной таблице (Таблица 1) приведены основные характеристики технологий позиционирования мобильных устройств. Можно увидеть, что наиболее точные результаты на местности даёт технология GPS. Если же рассматривать позиционирование внутри помещений – среди перечисленных технологий точность позиционирования лучше будет у технологий-аналогов iBeacon и EddyStone.

Глава 2

Трилатерация

Существует несколько основных способов определения текущей позиции какого-либо передвигающегося объекта.

Навигационное счисление (пути) — текущее положение объекта рассчитывается относительно предыдущего.

В расчете используются:

- предыдущая позиция;
- скорость движения;
- направление движения;
- интервал времени;
- дистанция;
- траектория движения.
- Расчеты местоположения по характеристикам сигналов, получаемых этим объектом от известных источников. [6]

Позиция объекта считается такой же, как у ближайшего источника сигнала. Используются методы трилатерации и триангуляции. Расчет позиции объекта производят по характеристикам сигналов, получаемых объектом от источников с известными координатами. [9]

Во многих существующих методах определения местоположения довольно часто используются трилатерационные алгоритмы.

Трилатерация бывает следующих видов:

- Простая трилатерация

заключается в вычислении позиции объекта как точки, равноудаленной от источников сигнала.

- Трилатерация с учетом силы сигнала.
- Расстояние от объекта до источника сигнала считается пропорциональным силе приходящего от нее сигнала.

- Трилатерация с учетом задержки сигнала (TOA — Time of Arrival). Расстояние рассчитывается по времени задержки получения сигнала.
- Мультилатерация (TDOA — Time Difference of Arrival).
- Известна также как гиперболическая трилатерация. Позиция рассчитывается по разнице во времени получения сигналов. Этот метод аналогичен TOA, но не нуждается в синхронизации часов. [6]

Сам принцип работы можно рассмотреть на примере спутниковой трилатерации.

Спутниковая трилатерация предполагает, что точные координаты любой точки на поверхности Земли могут быть вычислены путем измерений расстояний от группы спутников, если их положение в космосе известно.

В этом случае спутники являются пунктами с известными координатами. Предположим, что расстояние от одного спутника известно, и вокруг него можно описать сферу заданного радиуса. Если известно также расстояние до второго спутника, то определяемое местоположение будет расположено где-то в круге, задаваемом пересечением двух сфер. Третий спутник определяет две точки на окружности. Четвертый спутник позволяет окончательно точно определить местоположение точки. Таким образом, зная расстояние до четырех спутников, можно вычислить координаты определяемой точки. (Рисунок 2.1.) [2]

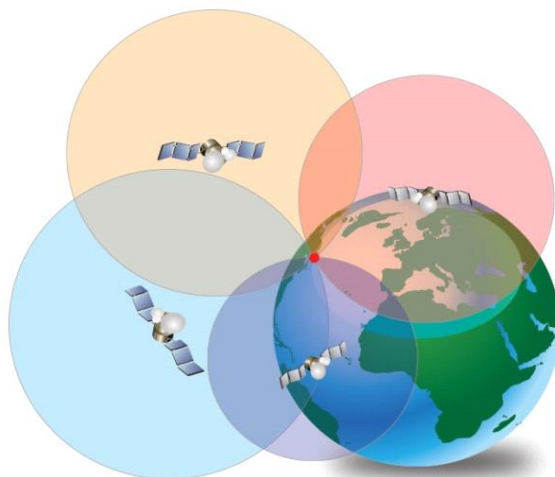


Рисунок 2.1.

Расстояние до спутников определяется по измерениям времени прохождения радиосигнала от космического аппарата до приемника, умноженным на скорость света. Для того чтобы определить время

распространения сигнала, нам необходимо знать, когда он был передан со спутника. Для этого на спутнике и в приемнике одновременно генерируется одинаковый псевдослучайный код. Каждый спутник системы GPS передает два радиосигнала: на частоте $L1=1575,42$ МГц и $L2=1227,60$ МГц. Сигнал $L1$ имеет два дальномерных кода с псевдослучайным шумом (PRN), P-код и C/A-код. «Точный», или P-код, может быть зашифрован для военных целей. «Грубый», или C/A-код, не зашифрован. Сигнал частоты $L2$ модулируется только с P-кодом.

Большинство гражданских пользователей при работе с GPS-системами используют C/A-код. Некоторые приемники Trimble, применяемые в геодезии, работают с P-кодом. Приемник проверяет входящий сигнал со спутника и определяет, когда он сам генерировал такой же код. Полученная разница, умноженная на скорость света, и дает искомое расстояние. Использование кода позволяет приемнику определить временную задержку в любое время. Кроме того, спутники могут излучать сигнал на одной и той же частоте, так как каждый спутник идентифицируется по своему PRN.

Как видно из вышесказанного, вычисления напрямую зависят от точности хода часов на спутниках и в приемниках. Код должен генерироваться на спутнике и в приемнике строго в одно и то же время. На спутниках установлены атомные часы, имеющие точность около одной наносекунды. Однако это решение является слишком дорогим, чтобы использовать его в приемниках GPS. Поэтому для устранения ошибок хода часов приемника используются результаты измерения сигналов от четвертого спутника. Их можно использовать для устранения ошибок, которые возникают, если часы на спутнике и в приемнике не синхронизированы. [4]

Глава 3

Проблема псевдодальности в трилатерации

Трилатерация / Мультилатерация – фундаментальная основа для большинства алгоритмов GPS позиционирования. Для наилучшего решения проблемы псевдодальности и нахождения местоположения приёмника сигнала в трёхмерном пространстве используют 4 передатчика. Таким образом – это аналитический подход, который находит позицию получателя, решая необходимое количество линейных / квадратных уравнений. Когда эти уравнения отлично сформулированы – мы можем получить максимально точное местоположение приёмника. Однако определение точного диапазона почти невозможно в реальной жизни из-за многих внешних факторов, таких как шумовые помехи, исчезновение сигнала, проблемы синхронизации часов и т.д. Следовательно, трилатерации не удаётся достигнуть достаточной точности в условиях реального мира.

Аналитические подходы поиска местоположения основываются на измерении точного расстояния до передатчиков. Трилатерация – это основа этих методов, где измерения диапазона от $n+1$ передатчика используются для оценки n -мерного местоположения. В идеальном случае, когда мы можем измерить точные оценки диапазона приёмника – мы можем определить псевдодальности для нашего приёмника. Основная геометрия позиционирования с использованием трёх спутников, помещённых в гипотетическое двухмерное пространство показана на рисунке 2.1.

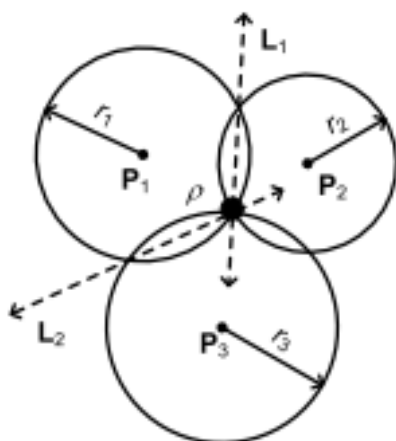


Рисунок.3.1. Идеальный случай 2D-трилатерации

Круги, окружающие передатчики с известными положениями $p_1(x_1, y_1)$, $p_2(x_2, y_2)$ и $p_3(x_3, y_3)$ обозначают расстояния, полученные из отдельных измерений диапазона для каждого передатчика. Расстояние вычисляется по формуле:

$$r_i = \|p_i - p\|^2 = ((x - x_i)^2 + (y - y_i)^2)$$

В двумерном пространстве можно рассчитать положение приёмника используя всего лишь три измеряемых диапазона. Два измерения таких диапазонов могут привести к двум решениям, которые будут соответствовать точкам пересечения двух кругов с радиусами равными расстояниям до датчика. Третье измерение устраняет эту двусмысленность, оставляя лишь одно верное расположение датчика.[11]

В связи с тем, что между передачей и приёмом сигнала у нас будет временная задержка, на которую влияют также и различные помехи – возникает погрешность в измерениях дальности объекта.

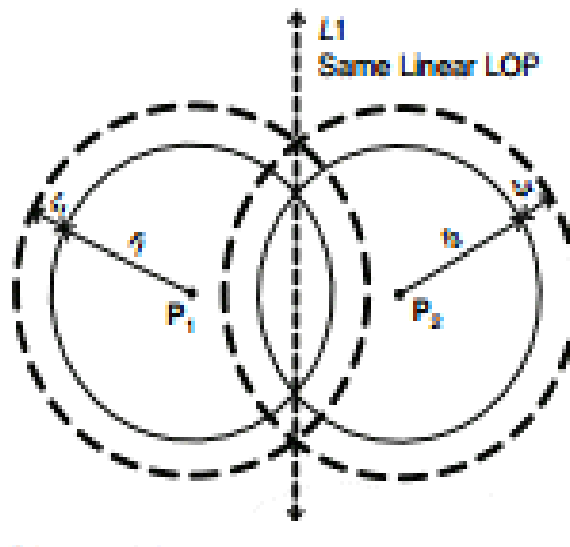


Рисунок.3.2. Изображение псевдодальности.

На рисунке 3.2. L_1 и L_2 показывают погрешность, которая возникает в ходе таких измерений.

Как показано на рисунках 3.1 и 3.2. позиционирование с помощью одной лишь геометрии работает правильно для случая, когда у нас измерены точные диапазоны. Однако на самом деле при измерении диапазонов возникает

необходимость учитывать как внешние воздействия шума, так и внутренние ошибки часов приёмников и передатчика. Также при позиционировании следует учитывать, что наиболее точные оценки будут получены, если шумовой эффект абсолютно одинаковый для всех приёмников. Несмотря на то, что трилатерация является математическим подходом и, в идеале, она может точно найти приёмник, при наличии шума положение того же приёмника будет найдено уже с некоторыми погрешностями.

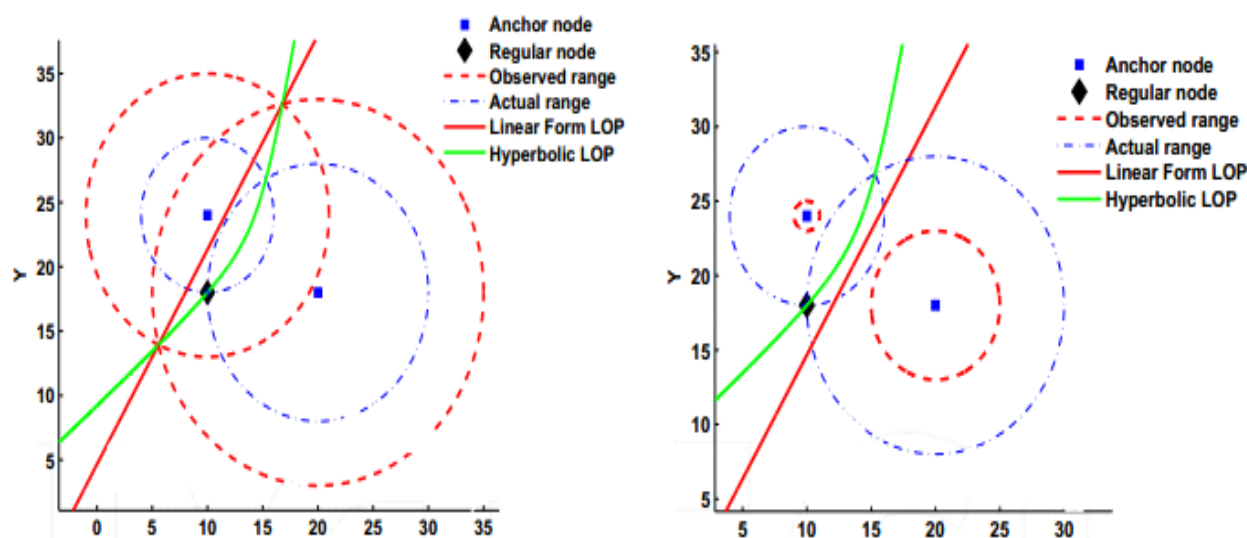


Рисунок.3.3

На рисунке 3.3 у нас представлены гиперболическая и линейная формы оценок псевдодальностей приёмника при равном шуме. На этом рисунке слева мы видим общий случай: когда два наблюдаемых диапазона пересекаются, а на рисунке справа – случай, когда круговые изображения псевдодальностей диапазонов не пересекаются из-за шума и недооценки действительных диапазонов.

Для решения проблемы псевдодальности используют следующие методы:

- Обычный трилатерационный метод
- Метод Бэнкрофта (Метод наименьших квадратов)
- Итерационный метод наименьших квадратов
- Алгоритм Клеусберга

Обычный трилатерационный метод использует следующую формулу:

$$(x_2^i - x_1^i)x + (y_2^i - y_1^i)y + (z_2^i - z_1^i)z = \frac{1}{2} (\|P_2^i\|^2 - \|P_1^i\|^2 + (r_1^i)^2 - (r_2^i)^2 + 2\xi(r_1^i - r_2^i))$$

где x, y, z – координаты спутников, i – номер спутника. В данном методе предполагается, что шум для всех спутников будет одинаковым и постоянным.

Алгоритм Клеусберга – это векторное алгебраическое решение для 3D-позиционирования. Векторное представление данного алгоритма изображено на рисунке 3.4.

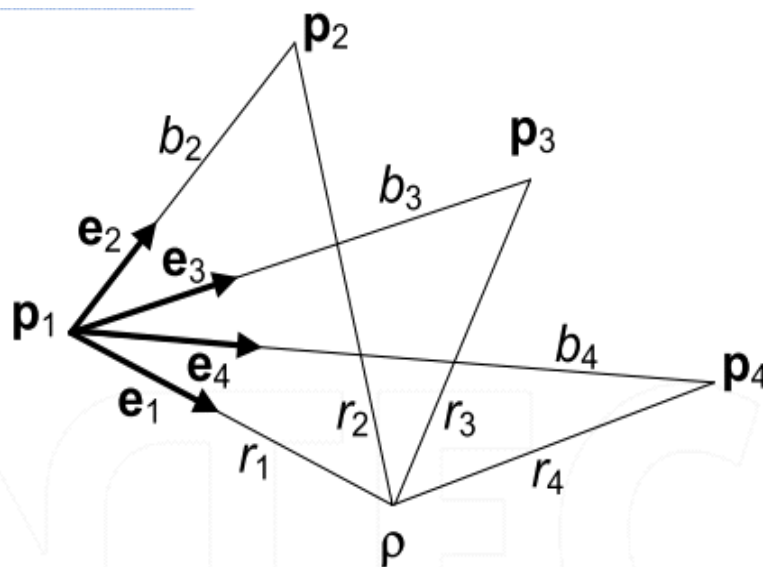


Рисунок 3.4

Алгоритм Клеусберга основан на следующей формуле:

$$\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} = (\bar{r}_i - \bar{r}_1) = d_i$$

Основная особенность итерационного метода – это то, что данный метод использует начальное приближение и циклически с каждой новой итерацией приближается к верному решению.

$$\sum_{i=1}^N \{p_i(x, y, z, \Delta_T) - p_i\}^2 \rightarrow \min$$

Метод Бэнкрофта – это метод решения проблемы псевдодальности с помощью линейной алгебры. В данном алгоритме решение уравнений сводится к проблеме наименьших квадратов.

$$\lambda^2 \langle \mathbf{B}^+ \mathbf{e}, \mathbf{B}^+ \mathbf{e} \rangle + 2\lambda (\langle \mathbf{B}^+ \tilde{\mathbf{e}}, \mathbf{B}^+ \mathbf{a} \rangle - 1) + \langle \mathbf{B}^+ \mathbf{a}, \mathbf{B}^+ \mathbf{a} \rangle = 0$$

Глава 4

Практическая часть

В трилатерационных алгоритмах, как уже написано в теоретической части - одна из главных проблем в определении наиболее точного местоположения объекта – это проблема псевдодальности. Для избавления от данной погрешности, в программе Matlab был реализовано 3 алгоритма, которые располагаются в Приложении 1, Приложении 2 и Приложении 3 соответственно.

4.1. Алгоритм Бэнкрофта

Для начала рассмотрим первый алгоритм – алгоритм Бэнкрофта. В качестве начальных данных мы задаём координаты датчиков и псевдодальности. На выходе мы получаем координаты приёмника, близкие к его реальным координатам.

Допустим:

x – это координаты нашего приёмника.

$\{s_i : 1 \leq i \leq n\}$ – координаты датчиков.

$\{t_i : 1 \leq i \leq n\}$ – псевдодальности.

n – количество датчиков.

Тогда расстояние от датчика до приёмника будет находиться следующим образом:

$$t_i = d(x, s_i) + b,$$

где $d(x, y)$ – это расстояние от x к y и b – временная задержка.

Создадим вектор 1×4 с данными

$$a_i = (s_i^T \ t_i)^T, \ 1 \leq i \leq n,$$

где T обозначает транспонирование.

Определим функцию Минковского для трехмерного пространства

$$\langle a, b \rangle = a_1 b_1 + a_2 b_2 - a_3 b_3$$

Определим вектора A , i_0 , r :

$$A = (a_1 \ a_2 \ \dots \ a_n)^T$$

$$i_0 = (1 \ 1 \ \dots \ 1)^T$$

$$r = (r_1 \ r_2 \ \dots \ r_n)^T$$

где r_i , $1 \leq i \leq n$ высчитывается по формуле:

$$r_i = \langle a_i \ a_i \rangle / 2$$

Посчитаем обобщенную инверсию

$$B = (A^T W A)^{-1} A^T W,$$

где W – положительно определенная весовая матрица. Матрица AB в данном случае ортогональной проекцией оператора.

Посчитаем вектора 1×4 u и v по следующим формулам:

$$u = B i_0; \quad v = B r$$

Скалярные коэффициенты E , F , G тогда будут определяться следующим образом:

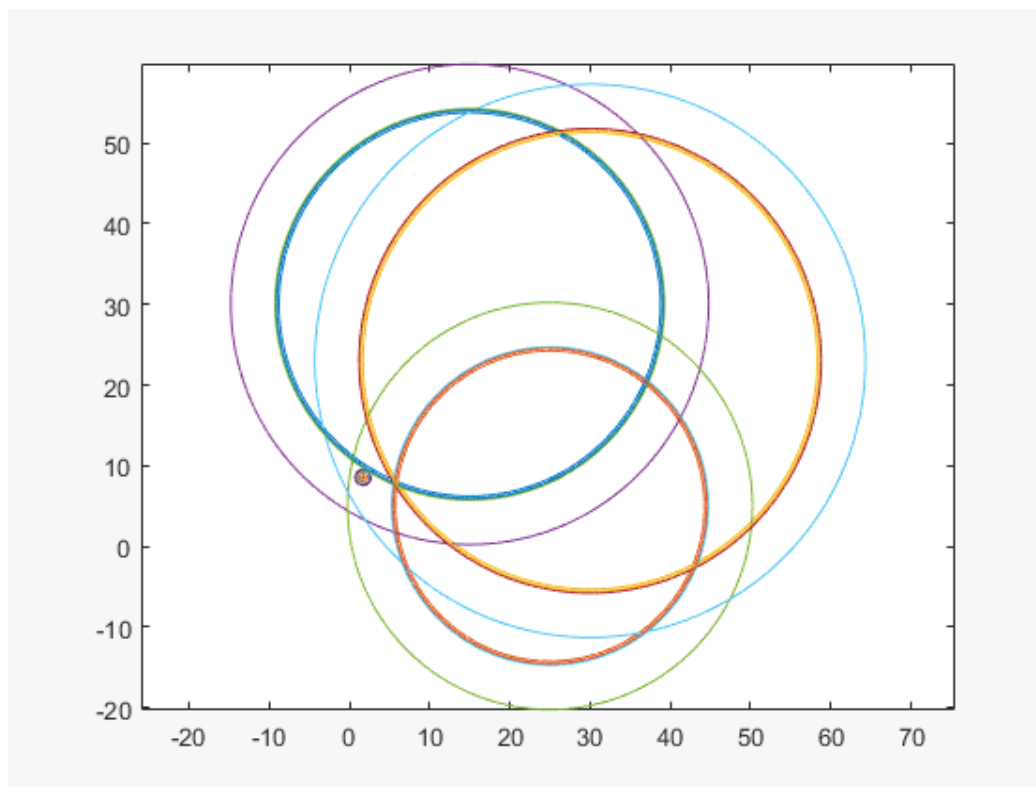


Рисунок. 4.1.1

$$E = \langle u, u \rangle ; \quad F = \langle u, v \rangle ; \quad G = \langle v, v \rangle$$

Затем решив квадратичное уравнение:

$$E \lambda^2 + 2F\lambda + G = 0 \text{ относительно пары неизвестных } \lambda_{1,2}$$

Посчитав 1×4 вектор $y_{1,2} = \lambda_{1,2} u + v$ получим $y^T = (x^T - b)^T$, который и содержит в себе необходимые нам координаты и временную задержку. [5]

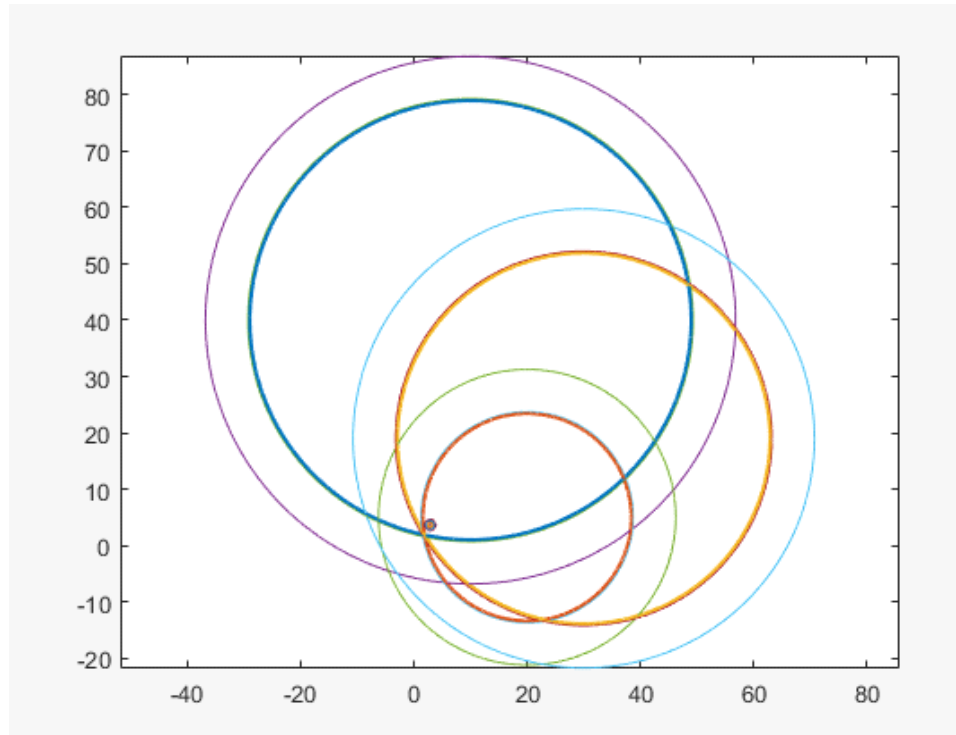


Рисунок. 4.1.2

На данных графиках (Рисунок. 4.1.1, Рисунок. 4.1.2) жирными линиями выделены настоящие дальности от датчика (центра круга) до объекта. Их пересечение – реальные координаты объекта. Дополнительные круги, отличающиеся от дальностей большим радиусом – будут псевдодальностями. Точкой на графике показано найденное местоположение объекта.

На данных графиках видно, что найденное местоположение будет близко к реальному местоположению объекта. К примеру, можно рассмотреть случай, показанный на Рисунок. 4.4. При координатах спутников $s1 = [10 \ 40]$, $s2 = [20 \ 5]$, $s3 = [30 \ 19]$, координате объекта $kx = [2 \ 2]$ и временной задержке $kb = 8$ м у нас выходят следующие результаты: координаты объекта $x = [2.8580 \text{ м} \ 3.6059 \text{ м}]$, временная задержка $b = 7.5478$ м. Соответственно вычисленная погрешность

определения координат составила [0.858 м 1.6059 м], а погрешность определения временной задержки 0.4522 м.

4.2. Итерационный метод

Теперь рассмотрим второй метод - итерационный. Данный метод основан на методе наименьших квадратов, который заключается в минимизации суммы погрешности, получаемой при измерении точных координат.

$$\sum_{i=1}^N \{p_i(x, y, z, \Delta_T) - p_i\}^2 \rightarrow \min$$

Для минимизации $f(x)$ необходимо найти экстремум

$$\frac{d}{dx} f(x) = 0$$

В рассматриваемой задаче функция и параметр – векторные.

Вектор параметров:

$$x = (x \ y \ z \ c)^T$$

Вектор наблюдаемой функции:

$$F(\vec{x}) = \begin{pmatrix} \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} + \Delta T \cdot c \\ \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} + \Delta T \cdot c \\ \dots \\ \sqrt{(x - x_N)^2 + (y - y_N)^2 + (z - z_N)^2} + \Delta T \cdot c \end{pmatrix}$$

Так как непосредственно эту задачу решить сложно на практике и применяется итеративный метод решения, иначе называемый методом Ньютона. Записывается в следующем виде:

$$\vec{x}^k = \vec{x}^{k-1} + \frac{d}{d\vec{x}} F(\vec{x}) (\vec{\rho} - \vec{\rho}(\vec{x})).$$

В данной формуле буквой k обозначается номер итерации.

Далее необходимо определить производную $\frac{d}{dx}F(x)$, для этого составляется так называемая «матрица направляющих косинусов», которая записывается в следующем виде:

$$\mathbf{H} = \frac{d}{d\vec{x}}\mathbf{F}(\vec{x}) = \begin{pmatrix} \frac{\partial}{\partial x}\rho_1(\vec{x}) & \frac{\partial}{\partial y}\rho_1(\vec{x}) & \frac{\partial}{\partial z}\rho_1(\vec{x}) & 1 \\ \frac{\partial}{\partial x}\rho_2(\vec{x}) & \frac{\partial}{\partial y}\rho_2(\vec{x}) & \frac{\partial}{\partial z}\rho_2(\vec{x}) & 1 \\ \dots & \dots & \dots & \dots \\ \frac{\partial}{\partial x}\rho_N(\vec{x}) & \frac{\partial}{\partial y}\rho_N(\vec{x}) & \frac{\partial}{\partial z}\rho_N(\vec{x}) & 1 \end{pmatrix}$$

Сам алгоритм можно разделить на несколько этапов. Для начала задаём начальные условия: $x^{(k)}=0, y^{(k)}=0, z^{(k)}=0$ ($k=0$ – номер итерации). $k=0$ – номер итерации.

Далее проводим измерения и рассчитываем псевдодальность по формуле:

$$\tilde{\tau} = t_{\text{прм,ШВП}} - t_{\text{изл,БШВ}}$$

После этого рассчитываем оценки псевдодальностей:

$$\hat{\rho}_i^{(k-1)} = \sqrt{(x^{(k-1)} - x_1)^2 + (y^{(k-1)} - y_1)^2 + (z^{(k-1)} - z_1)^2} + \Delta\tau \cdot c$$

Ошибка оценок псевдодальностей:

$$\Delta\rho_i^{(k-1)} = \rho_i^{(k-1)} - \hat{\rho}_i^{(k-1)}$$

Рассчитываем матрицу направляющих косинусов

$$\mathbf{H} = \begin{pmatrix} \frac{x^{(k-1)}-x_1}{R_1} & \frac{y^{(k-1)}-y_1}{R_1} & \frac{z^{(k-1)}-z_1}{R_1} & 1 \\ \frac{x^{(k-1)}-x_2}{R_2} & \frac{y^{(k-1)}-y_2}{R_2} & \frac{z^{(k-1)}-z_2}{R_2} & 1 \\ \dots & \dots & \dots & \dots \\ \frac{x^{(k-1)}-x_N}{R_N} & \frac{y^{(k-1)}-y_N}{R_N} & \frac{z^{(k-1)}-z_N}{R_N} & 1 \end{pmatrix}$$

$$R_i = \sqrt{(x^{(k-1)} - x_i)^2 + (y^{(k-1)} - y_i)^2 + (z^{(k-1)} - z_i)^2} - \text{расстояние до } i\text{-го НС.}$$

Дальше нам необходимо скорректировать текущее решение с помощью следующей формулы:

$$\vec{x}^{(k)} = \vec{x}^{(k-1)} + \mathbf{H} \cdot \vec{\Delta\rho}^{(k-1)}$$

На финальном этапе мы проверяем, насколько близко мы приблизились к нашему решению. Если смещение оценки на последней итерации достаточно мало, то мы завершаем цикл, если нет – то возвращаемся ко второму этапу, где мы рассчитываем нашу псевдодальность, но уже с новым начальным приближением. [1]

$$\sqrt{(x^{(k)} - x^{(k-1)})^2 + (y^{(k)} - y^{(k-1)})^2 + (z^{(k)} - z^{(k-1)})^2}$$

В итоге мы получаем графики, которые доказывают работоспособность нашего метода. Два из экспериментальных примеров данных графиков можно увидеть на рисунке 4.2.1 и 4.2.2.

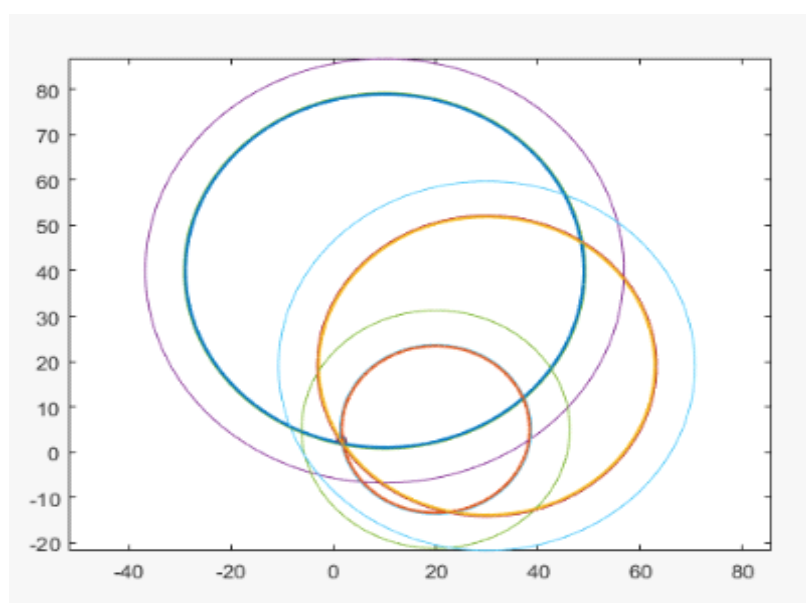


Рисунок.4.2.1

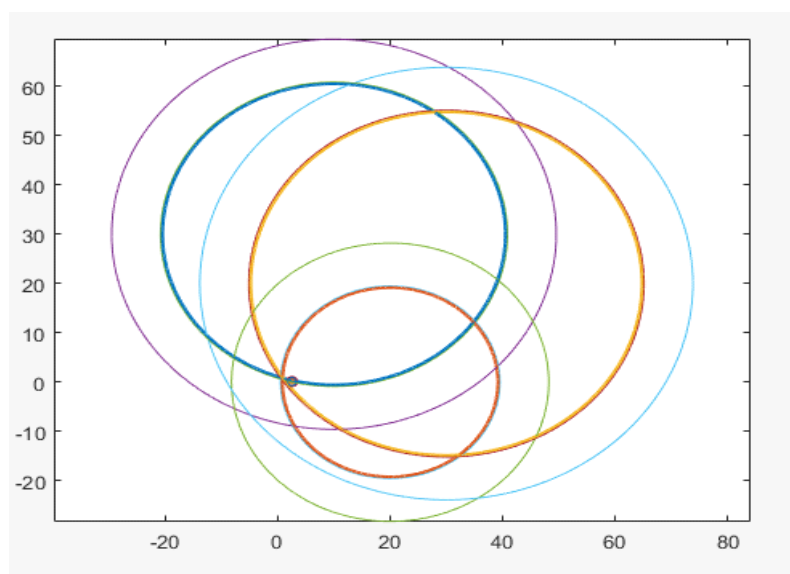


Рисунок.4.2.2

На рисунке 4.2.1 показаны результаты, где полученная погрешность полученных координат составила 0.67646 м. При оригинальных координатах в точке (2,2) мы получили координаты в точке (1,5547 ; 2, 5003).

4.3. Обычный трилатерационный метод

Третьим из рассмотренных методов будет так называемый обычный трилатерационный метод. Допустим $P_i = (p_1^i, p_2^i)$ будет произвольной спутниковой парой, где $p_1^i = (x_1^i, y_1^i, z_1^i)$ и $p_2^i = (x_2^i, y_2^i, z_2^i)$ будут соответствовать местоположению i -той пары спутников. Аналогично двумерному пространству составляется и уравнение для нахождения псевдодальности и в трёхмерном пространстве:

$$(x_2^i - x_1^i)x + (y_2^i - y_1^i)y + (z_2^i - z_1^i)z = \frac{1}{2}(\|p_2^i\|^2 - \|p_1^i\|^2 + (r_1^i) - (r_2^i) + 2\varepsilon(r_1^i - r_2^i))$$

Здесь полагается, что шум для каждой спутниковой пары будет постоянный и одинаковый, т.е. $\varepsilon_1 = \varepsilon_2 = \varepsilon$.

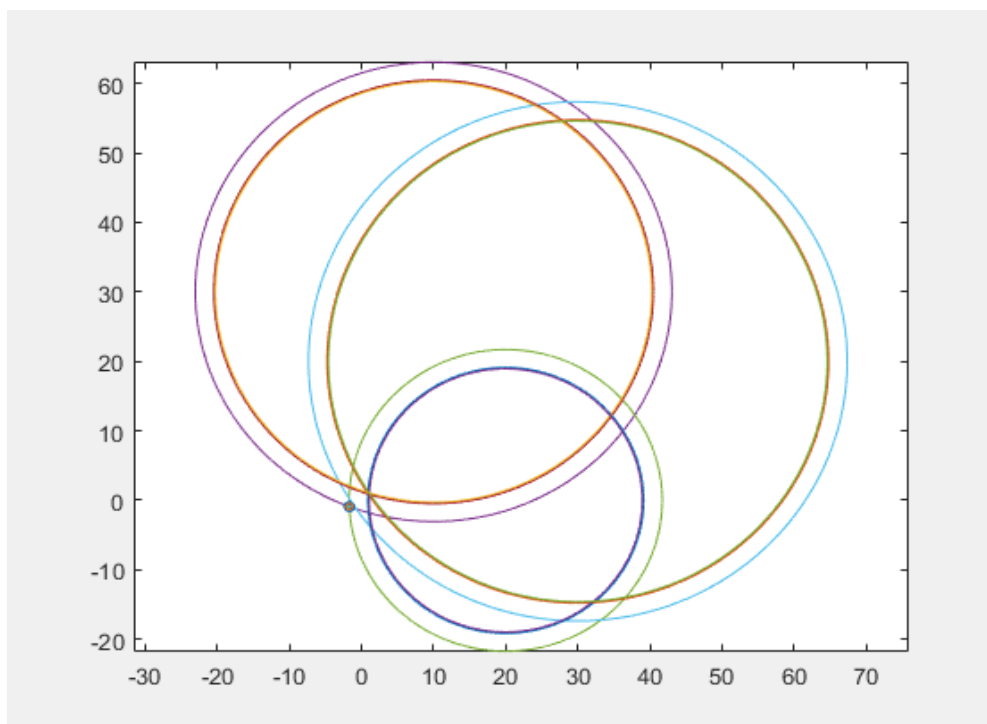


Рис. 4.3.1

Уравнение становится линейным в точках с координатами x, y, z и с шумом ε , если шум представлен единственным параметром ε для всех пар спутников. В

этом случае есть четыре неизвестных в этом уравнении, и поэтому для нахождения неизвестных нам понадобится четыре уравнения.

На рисунке 4.3.1 можно увидеть один из примеров работы алгоритма, где результаты, показывающие, что погрешность измерений будет больше, чем у представленных выше итерационного метода и метода Бэнкрофта.

4.4. Сравнение алгоритмов

После реализации данных методов была взята выборка данных из 10000 пар элементов, заполненных случайными данными с координатами объектов. Все эти данные поступали на вход алгоритму, в качестве итоговых данных мы считали полученную погрешность для каждого значения для трех различных методов.

Размеры исследуемого помещения мы взяли 50 х 50 метров. Координаты трёх неизменных датчиков для испытаний – M1(50, 20), M2(20, 0) и M3(0,30). Данную схему расположения можно увидеть на рисунке 4.4.1.

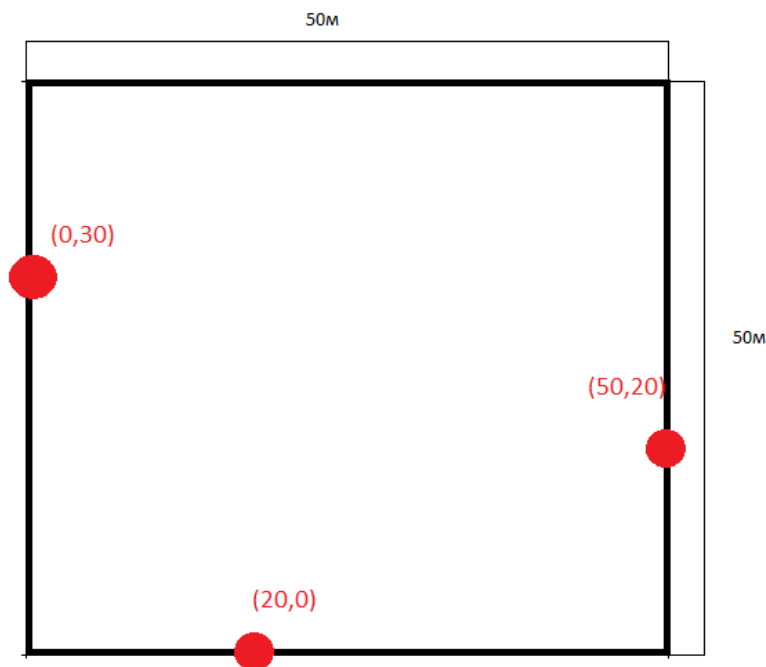


Рисунок 4.4.1.

На рисунке 4.4.2 изображено распределение погрешностей при определении точной координаты первым методом.

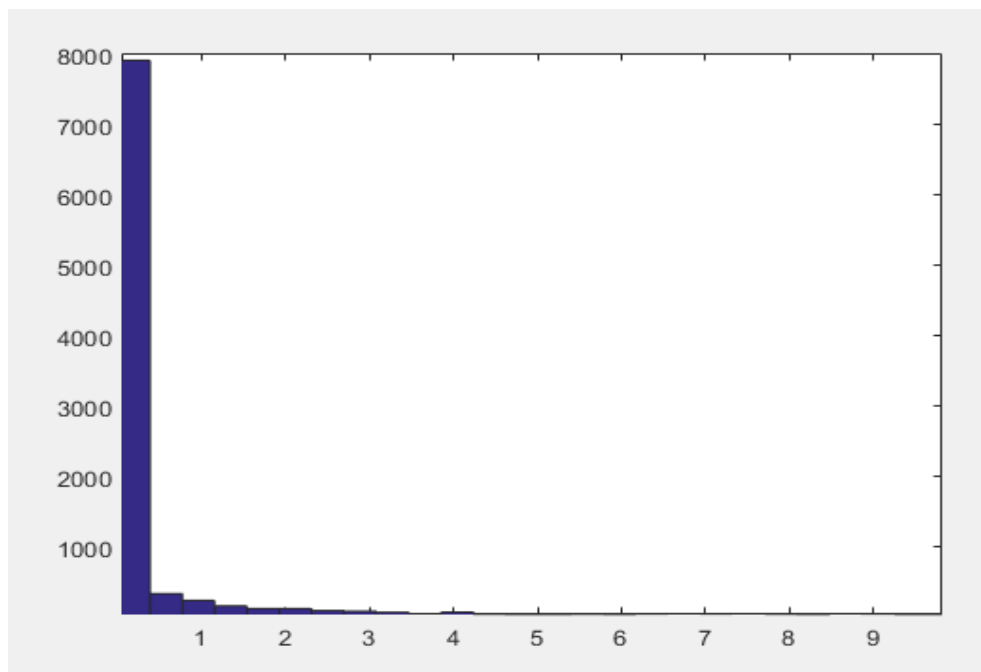


Рисунок.4.4.2

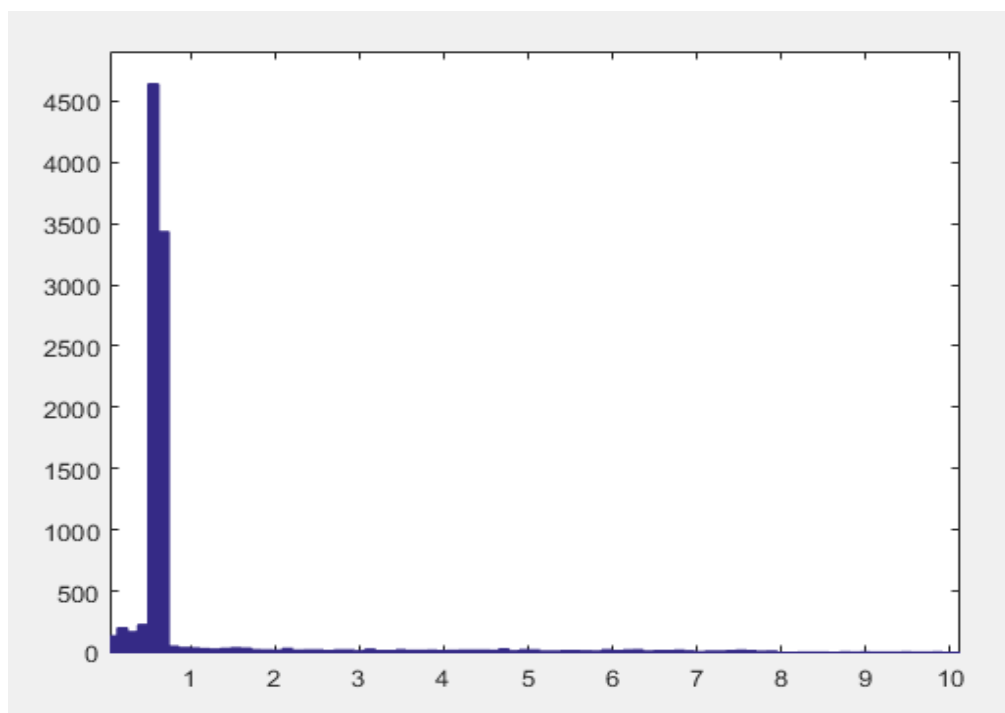


Рисунок.4.4.3

На следующем рисунке 4.4.3 изображено распределение погрешностей при определении точной временной задержки первым методом.

Среднее значение погрешности составило 1.0275 м.

Среднеквадратичное отклонение составило 2.6073 м.

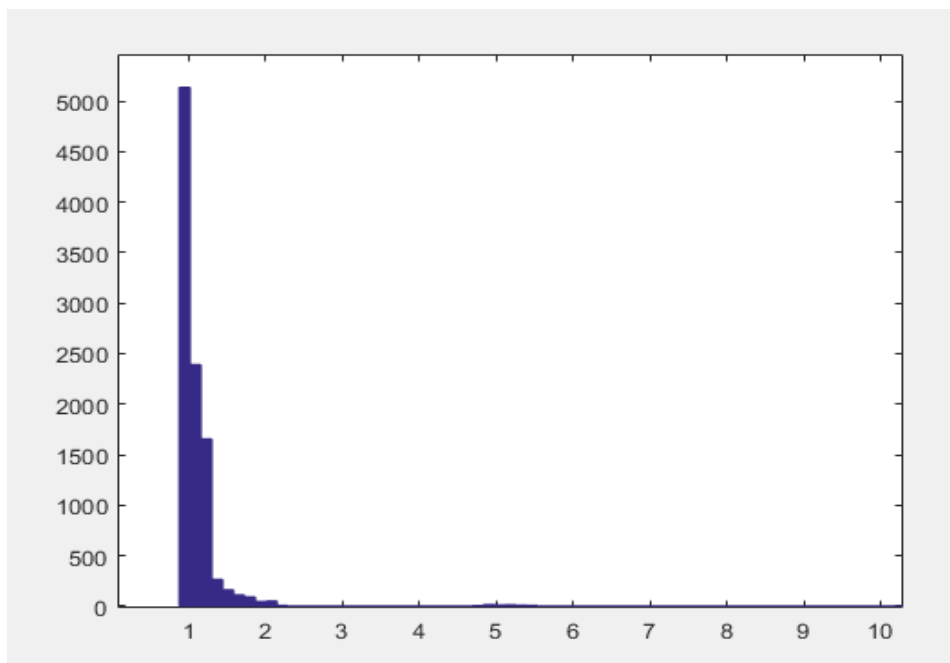


Рисунок.4.4.4

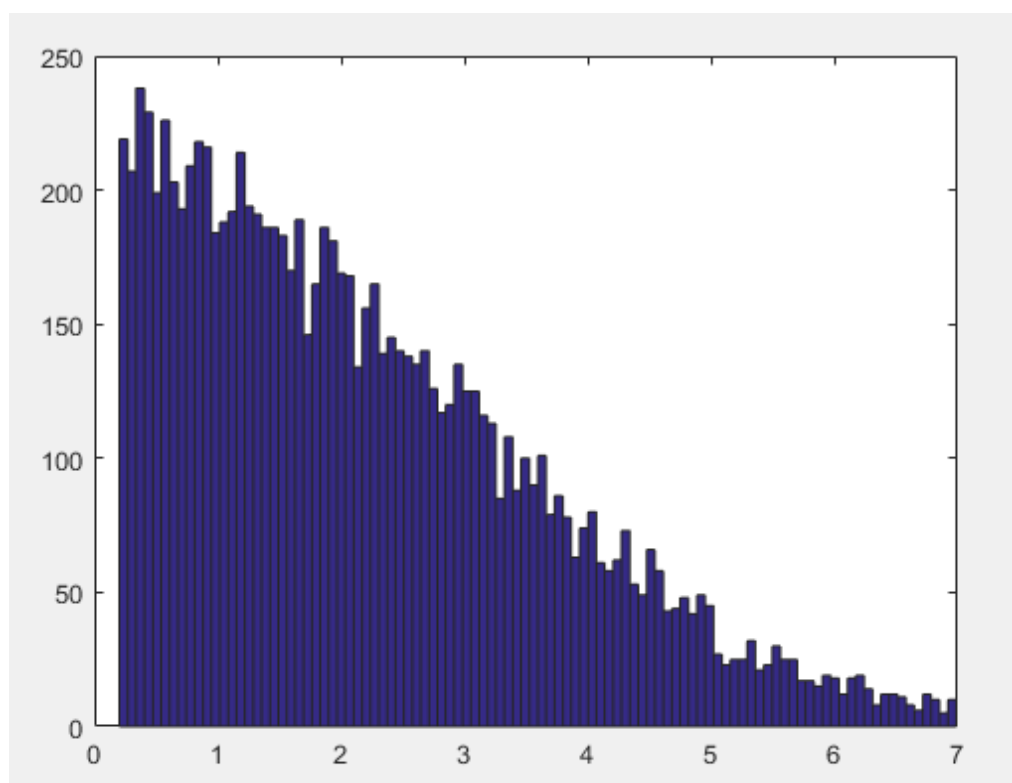


Рис.4.4.5

На рисунке 4.4.4 изображено распределение погрешностей при определении точной координаты итерационным методом.

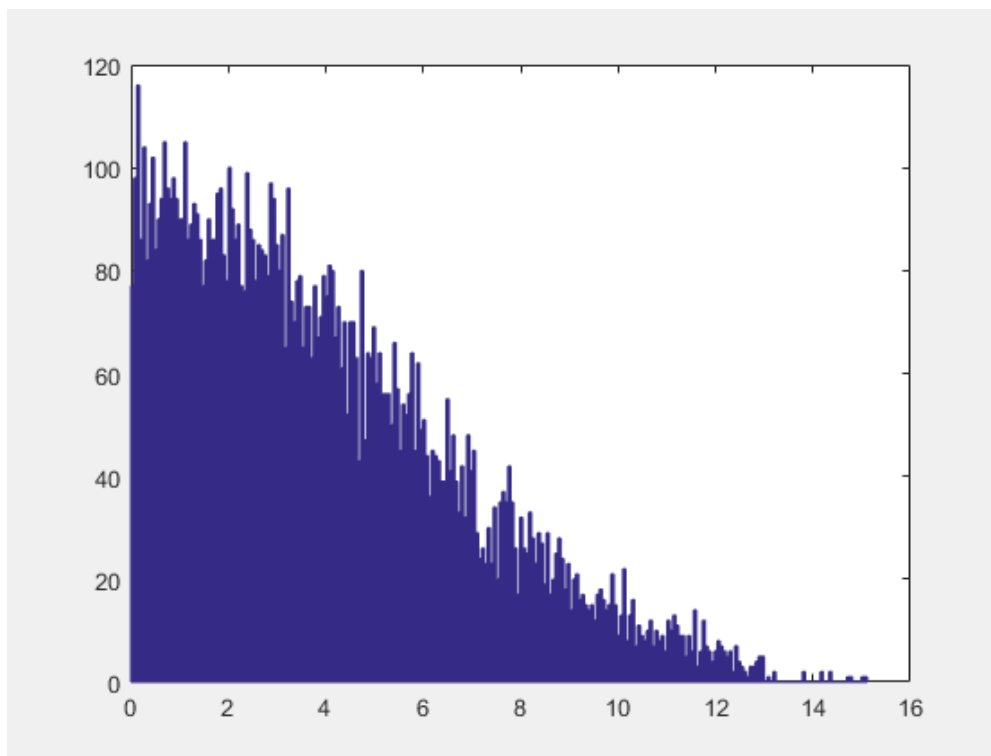


Рис.4.4.6

На следующем рисунке 4.4.5 изображено распределение погрешностей при определении точной временной задержки итерационным методом.

Среднее значение погрешности составило 2.1965 м.

Среднеквадратичное отклонение составило 1.4423 м.

На рисунке 4.4.6 изображено распределение погрешностей при определении точной координаты обычным трилатерационным методом.

Среднее значение погрешности составило 3.9259 м.

Среднеквадратичное отклонение составило 2.8526 м.

Название метода	Метод Бэнкрофта	Итерационный метод	Обычный трилатерационный метод
Среднее значение погрешности (м)	1.0275	2.1965	3.9259

Среднеквадратичное значение погрешности (м)	2.6073	1.4423	2.8526
---	--------	--------	--------

Таблица 2

В таблице 2 можно увидеть значения погрешностей для каждого из реализованных методов для выборки из 10 000 элементов и помещения с размерами 50 x 50 м. Посмотрев на эти данные можно прийти к выводу, что метод Бэнкрофта будет работать лучше и показывать наиболее точные результаты.

Заключение

Технологии позиционирования в настоящее время уже рассматриваются, не только как средство для создания систем обеспечения большей безопасности для абонентов сотовых систем, но и как возможность оказания целого ряда дополнительных услуг, основанных на знании местоположения абонента.

Отсюда можно сделать вывод, что развитие технологий в данной сфере просто необходимо. К этому выводу можно прийти и по причине того, что идеальной точности в методах позиционирования пока добиться не удалось.

В ходе работы мы проанализировали существующие алгоритмы трилатерации, решающие проблему псевдодальности, реализовали выбранные алгоритмы в среде Matlab. В ходе тестирования данных алгоритмов мы вычисляли координаты объекта, близкие к его реальным координатам, при помощи численного моделирования вычисляли погрешности вычислений и проводили сравнение данных алгоритмов по ряду характеристик, рассматривали их преимущества и недостатки.

Имея полученные данные со средними и среднеквадратичными погрешностями можно прийти к выводу, что из реализованных в данной курсовой работе алгоритмов – алгоритм Бэнкрофта (метод наименьших квадратов) даёт лучшие результаты при вычислении положения объекта. Исследования показали, что наиболее точные результаты в поиске координат нужного нам объекта можно получить с методом Бэнкрофта, где среднее значение погрешностей составило 1.0275 м, а среднеквадратичная погрешность данного метода составила 2.6073 м. Немного хуже результаты оказались у итерационного метода, где средняя и среднеквадратичная ошибки составили соответственно 2.1965 м и 1.4423 м. И самые отдаленные результаты от настоящих координат показал обычный трилатерационный метод со значениями погрешностей средней - 3.9259 м, среднеквадратичной – 2.8526 м.

Список использованных источников

1. Trilateration: The Mathematics Behind a Local Positioning System / Department of Mathematical and Computer Sciences Colorado School of Mines Golden - Colorado, U.S.A, 2011. – 37 с.
2. Kaplan E.D. Understanding GPS principles and applications / E.D. Kaplan, C.J. Hegarty – 2-е изд. - Massachusetts, 2006. – 146 с.
3. Гончаров И.А. Основы любительской GPS-навигации / И.А. Гончаров - М.: Горячая Линия - Телеком, 2007. – 128 с.
4. Rahman M.Z. Beyond Trilateration: GPS Positioning Geometry and Analytical Accuracy / M. Z. Rahman - Rijeka, Croatia, 2012. – 17 с.
5. Bancroft S. An Algebraic Solution of the GPS Equations / S. Bancroft – Olathe, Kansas, 1985 – 4 с.
6. Трилатерация [Электронный ресурс]. – 2017. – Режим доступа: <https://ru.wikipedia.org/wiki/Трилатерация> – Дата доступа: 15.05.2018.
7. iBeacon Insider: Your Guide to iBeacon Technology [Электронный ресурс]. – 2015. – Режим доступа: <http://www.ibeacon.com/> – Дата доступа: 11.05.2018.
8. GPS: The Global Positioning System [Электронный ресурс]. – 2018. – Режим доступа: <http://www.gps.gov/> – Дата доступа: 10.04.2018.
9. Russell C. The Surveying Handbook / Brinker, Russell C., Minnick, Roy – Luxembourg, 1995. – 971 с.

10. Triangulation, Trilateration or Multilateration [Электронный ресурс]. – 2014. – Режим доступа: <http://circuitcellar.com/ee-tips/triangulation-trilateration-or-multilateration-ee-tip-125/> – Дата доступа: 10.04.2018.
11. Cook B. Indoor Location Using Trilateration Characteristics / B. Cook1, G. Buckberry, I. Scowcroft – London, 2005 – 4 с.
12. Schiller J. Location-Based Services / J. Schiller, A.Voisard - Amsterdam, 2004 – 266 с.
13. Дворкин Н.Б. Мобильные навигационные сервисы и применение технологии OpenCellID для определения местоположения / Н.Б. Дворкин, Д.Е. Намиот - Москва, 2010 – 10 с.
14. GPS: The Global Positioning System [Электронный ресурс]. – 2018. – Режим доступа: <https://ru.wikipedia.org/wiki/GPS> – Дата доступа: 12.04.2018.
15. E-OTD [Электронный ресурс]. – 2013. – Режим доступа: <https://en.wikipedia.org/wiki/E-OTD> – Дата доступа: 15.04.2018.
16. Бербатов А. Мобильный поводырь: Технология LBS для вашего телефона [Электронный ресурс]. – 2007. – Режим доступа: <http://media.mabila.ua/ru/articles/lbs/> – Дата доступа: 10.05.2018.
17. Москаленко Т.А. Методы позиционирования робототехнических систем внутри помещения на базе телекоммуникационных технологий / Т. А. Москаленко, Р. В. Киричек - Санкт-Петербург, 2016. – 9с.

Приложение 1

```

clc;
close all
% x      - user position coordinates
% si     - sensor position coordinates
% ti     - pseudorange measurements (ti=d(x,si)+clockOffset)
s1 = [10 30];
s2 = [20 0];
s3 = [30 20];
%kX - original user position coordinates
kX = [1 1];
%=====
%random chisla s normalnim raspredeleniem
chisloElementov = 10000;
MU =1;
SIGMA =10;
RandKb = abs(normrnd(MU,SIGMA,[1 chisloElementov]));

MU =1;
SIGMA =50;
RandkX = abs(normrnd(MU,SIGMA,[1 chisloElementov]));

%=====
%vektori pogreshnostei
%zanulyaem
vectPogreshnostX = zeros(1,chisloElementov);
vectPogreshnostB = zeros(1,chisloElementov);
%=====
%ispolzuem fiksirovannoe znachenie nachalnoi clock offset
%kb - original time offset
%kb = 8;
%=====
%ispolzuem randomnoe znachenie clock offset
schetchik = 1;
while schetchik <= chisloElementov
kb = RandKb(schetchik);
%=====
kX= RandkX(schetchik);
t1 = (((s1(1)- kX(1))^2+(s1(2)-kX(2))^2)^0.5)+kb;
t2 = (((s2(1)- kX(1))^2+(s2(2)-kX(2))^2)^0.5)+kb;
t3 = (((s3(1)- kX(1))^2+(s3(2)-kX(2))^2)^0.5)+kb;
% ai      - (si^t ti)^t
a1 = [s1'; t1]';
a2 = [s2'; t2]';
a3 = [s3'; t3]';
% A      - (a1 a2 a3 a4)^t
A = [a1;a2;a3]';
i0 = [1 1 1]';
%ri      - <ai,ai>/2
% <a,b> = a1b1 + a2b2 + a3b3 - a4b4
r1 = (a1(1)^2 +a1(2)^2 - a1(3)^2)/2;
r2 = (a2(1)^2 +a2(2)^2 - a2(3)^2)/2;
r3 = (a3(1)^2 +a3(2)^2 - a3(3)^2)/2;
r = [r1 r2 r3]';
W = [1 1 1; 1 1 1; 1 1 0];
%B = (A^t*W*A)^-1 *A^t *W
% W - positive definite weighting matrix
B = inv(A);

```

```

%B = inv(A' * W * A) * A' * W;
% u & v - 1 x 4 vectors
u = B * i0;
v = B * r;
% E, F, G - scalar coefficient
E = (u(1)^2 + u(2)^2 - u(3)^2);
F = (u(1)*v(1) + u(2)*v(2) - u(3)*v(3)) -1;
G = (v(1)^2 + v(2)^2 - v(3)^2);
%E 1^2 +2F 1 + G = 0
sqrtD = sqrt((2*F)^2 - 4*E*G);
lyambda1 = (-2*F - sqrtD)/(-2*E);
lyambda2 = (-2*F + sqrtD)/(-2*E);
% y1,2 - column vectors 1 x 3
y1 = lyambda1 * u + v;
y2 = lyambda2 * u + v;
%y = (x^t-b)^t
x1 = [y1(1) y1(2)];
x2 = [y2(1) y2(2)];
% b - clock offset
b1 = -y1(3);
b2 = -y2(3);

tempSum1 = abs(abs(x1-s1)+b1 - t1) + abs(abs(x1-s2)+b1 - t2) + abs(abs(x1-s3)+b1
- t1);
tempSum2 = abs(abs(x2-s1)+b2 - t1) + abs(abs(x2-s2)+b2 - t2) + abs(abs(x2-s3)+b2
- t1);

if(tempSum1(1)+tempSum1(2)<=tempSum2(1)+tempSum2(2))
    x=x1;
    b=b1;
else
    x=x2;
    b=b2;
end

%pogreshnosti
PogreshnostB = (kb - b);
PogreshnostB = abs(real(PogreshnostB));
PogreshnostXx = (kX - x);

PogreshnostX = ((PogreshnostXx(1))^2+(PogreshnostXx(2))^2)^0.5;
PogreshnostX = abs(real(PogreshnostX));
%=====
vectPogreshnostX(schetchik) = PogreshnostX;
vectPogreshnostB(schetchik) = PogreshnostB;
schetchik=schetchik + 1 ;
end

%grafiki
%bez pogreshnosti
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb);
hold on
plotCircle(s2(1), s2(2), t2-kb);
hold on
plotCircle(s3(1), s3(2), t3-kb);
axis equal
%bez pogreshnosti

```



```

plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1);
hold on
plotCircle(s2(1), s2(2), t2);
hold on
plotCircle(s3(1), s3(2), t3);
axis equal

%psevdodalnost
plotCircle(x(1), x(2), 0.6);
plotCircle(x(1), x(2), 0.7);
plotCircle(x(1), x(2), 0.5);
plotCircle(x(1), x(2), 0.4);
plotCircle(x(1), x(2), 0.3);
plotCircle(x(1), x(2), 0.2);
plotCircle(x(1), x(2), 0.1);
%plotCircle(x(1), x(2), 0.8);
%plotCircle(x(1), x(2), 0.5);
%plotCircle(x(1), x(2), 0.3);
%plotCircle(x(1), x(2), 0.1);
%axis equal
%=====
%plotCircle(s1(1), s1(2), t1-kb+0.5);
%hold on
%plotCircle(s2(1), s2(2), t2-kb+0.5);
%hold on
%plotCircle(s3(1), s3(2), t3-kb+0.5);
%axis equal
%=====
%videlenie original range
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb+0.1);
hold on
plotCircle(s2(1), s2(2), t2-kb+0.1);
hold on
plotCircle(s3(1), s3(2), t3-kb+0.1);
axis equal
%=====
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb-0.1);
hold on
plotCircle(s2(1), s2(2), t2-kb-0.1);
hold on
plotCircle(s3(1), s3(2), t3-kb-0.1);
axis equal
%=====
vectPogreshnostX
vectPogreshnostB
%=====

%=====
figure;hist(vectPogreshnostB,100);
figure;hist(vectPogreshnostX,100);
%=====
%srednee
srednee = 0;
schet1 = 1;
while schet1 < chisloElementov
srednee = srednee + vectPogreshnostB(schet1) ;
schet1 = schet1 +1;

```

```

end
srednee = srednee / chisloElementov;
srednee
%srednekvadratichoe
schet1 = 1;
srednekvadratichoe = 0;
while schet1 < chisloElementov
srednekvadratichoe = srednekvadratichoe +(vectPogreshnostB(schet1) -
srednee)^2 ;
schet1 = schet1 +1;
end
srednekvadratichoe = (srednekvadratichoe / chisloElementov)^0.5;
srednekvadratichoe

```

Приложение 2

```
% Iterative least squares estimate
clc;
close all
% x      - user position coordinates
% si     - sensor position coordinates
% ti     - pseudorange measurements (ti=d(x,si)+clockOffset)
s1 = [10 30];
s2 = [20 0];
s3 = [30 20];
%kX - original user position coordinates
kX = [1 1];
%=====
%random chisla s normalnim raspredeleniem
%viborka
chisloElementov = 10000;
MU =1;
SIGMA =50;
RandKb = abs(normrnd(MU,SIGMA,[1 chisloElementov]));

MU =1;
SIGMA =50;
RandKx = abs(normrnd(MU,SIGMA,[1 chisloElementov]));

%=====
%vektori pogreshnostei
vectPogreshnostX = zeros(1,chisloElementov);
vectPogreshnostB = zeros(1,chisloElementov);
%=====
%ispolzuem fiksirovannoe znachenie nachalnoi clock offset
%kb - original time offset
%kb = 8;
%=====
%ispolzuem randomnoe znachenie clock offset
schetchik = 1;
while schetchik <= chisloElementov
kb = RandKb(schetchik);
kX = RandKx (schetchik);
%=====
t1 = (((s1(1)- kX(1))^2+(s1(2)-kX(2))^2)^0.5)+kb;
t2 = (((s2(1)- kX(1))^2+(s2(2)-kX(2))^2)^0.5)+kb;
t3 = (((s3(1)- kX(1))^2+(s3(2)-kX(2))^2)^0.5)+kb;
%t4 = (((s4(1)- kX(1))^2+(s4(2)-kX(2))^2)^0.5)+kb;
%vector t
t = [t1 t2 t3];
%Пусть nachPriblX - нач. приближенные координаты нашего объекта
nachPriblX2 = [0 0 0];
%Пусть nachPriblClockOffset - нач. приближенные временной задержки
%nachPriblClockOffset = 0;
proverka = 900;
while proverka > 2
nachPriblX = nachPriblX2;
%Ri - расстояние до i датчика
R1 = ((nachPriblX(1) - s1(1))^2+(nachPriblX(2)-s1(2))^2)^0.5;
R2 = ((nachPriblX(1) - s2(1))^2+(nachPriblX(2)-s2(2))^2)^0.5;
R3 = ((nachPriblX(1) - s3(1))^2+(nachPriblX(2)-s3(2))^2)^0.5;
%H - матрица направляющих косинусов
H=[(nachPriblX(1)-s1(1))/R1 (nachPriblX(2)-s1(2))/R1 1 ;...
(nachPriblX(1)-s2(1))/R2 (nachPriblX(2)-s2(2))/R2 1 ;...
(nachPriblX(1)-s3(1))/R3 (nachPriblX(2)-s3(2))/R3 1];
%?????
```

```

pseudorange = [((nachPriblX(1)- s1(1))^2+(nachPriblX(2) - s1(2))^2)^0.5 +
nachPriblX(3) ...
((nachPriblX(1)- s2(1))^2+(nachPriblX(2) - s2(2))^2)^0.5 + nachPriblX(3) ...
((nachPriblX(1)- s3(1))^2+(nachPriblX(2) - s3(2))^2)^0.5 + nachPriblX(3) ];
%dp - ошибка оценок псевдодальностей
dp = pseudorange - t;
dp = dp';
%коррекция текущего решения
z=(H*dp)';
nachPriblX2 = nachPriblX + z;

%proverka
proverka = ((nachPriblX2(1) - nachPriblX(1))^2 + (nachPriblX2(2)-
nachPriblX(2))^2)^0.5;
end

%nachPriblX2 - ответ
x = [nachPriblX2(1) nachPriblX2(2)];
b=nachPriblX2(3);
PogreshnostB = (kb - b);
PogreshnostB = abs(PogreshnostB);
PogreshnostXx = (kX - x);
PogreshnostX = ((PogreshnostXx(1))^2+(PogreshnostXx(2))^2)^0.5;
PogreshnostX = abs(PogreshnostX);
%=====
vectPogreshnostX(schetchik) = PogreshnostX;
vectPogreshnostB(schetchik) = PogreshnostB;
schetchik=schetchik + 1 ;
end
%grafiki

%bez pogreshnosti
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb);
hold on
plotCircle(s2(1), s2(2), t2-kb);
hold on
plotCircle(s3(1), s3(2), t3-kb);
axis equal
%bez pogreshnosti
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1);
hold on
plotCircle(s2(1), s2(2), t2);
hold on
plotCircle(s3(1), s3(2), t3);
axis equal

%psevdodalnost
plotCircle(x(1), x(2), 1);
plotCircle(x(1), x(2), 0.8);
plotCircle(x(1), x(2), 0.5);
plotCircle(x(1), x(2), 0.3);
plotCircle(x(1), x(2), 0.1);
axis equal
%=====
plotCircle(s1(1), s1(2), t1-kb+0.5);
hold on
plotCircle(s2(1), s2(2), t2-kb+0.5);
hold on
plotCircle(s3(1), s3(2), t3-kb+0.5);

```

```

axis equal
%=====
plotCircle(s1(1), s1(2), t1-kb+0.3);
hold on
plotCircle(s2(1), s2(2), t2-kb+0.3);
hold on
plotCircle(s3(1), s3(2), t3-kb+0.3);
axis equal
PogreshnostX
PogreshnostB
%=====
vectPogreshnostX
vectPogreshnostB
figure;hist(vectPogreshnostB);
figure;hist(vectPogreshnostX);
%=====
%srednee
srednee = 0;
schet1 = 1;
while schet1 < chisloElementov
srednee = srednee + vectPogreshnostB(schet1) ;
schet1 = schet1 +1;
end
srednee = srednee / chisloElementov;
srednee
%srednekvadratichnoe
schet1 = 1;
srednekvadratichnoe = 0;
while schet1 < chisloElementov

srednekvadratichnoe = srednekvadratichnoe +(vectPogreshnostB(schet1) -
srednee)^2 ;
schet1 = schet1 +1;
end
srednekvadratichnoe = (srednekvadratichnoe / chisloElementov)^0.5;
srednekvadratichnoe

```

Приложение 3

```
clc;
close all
% x      - user position coordinates
% si     - sensor position coordinates
% ti     - pseudorange measurements (ti=d(x,si)+clockOffset)
s1 = [10 30];
s2 = [20 0];
s3 = [30 20];
%kX - original user position coordinates
kX = [1 1];
%=====
%random chisla s normalnim raspredeleniem
chisloElementov = 10000;
MU =1;
SIGMA =50;
RandKb = abs(normrnd(MU,SIGMA,[1 chisloElementov]));

MU =1;
SIGMA =50;
RandkX = abs(normrnd(MU,SIGMA,[1 chisloElementov]));

%=====
%vektori pogreshnostei
%zanulyaem
vectPogreshnostX = zeros(1,chisloElementov);
vectPogreshnostB = zeros(1,chisloElementov);
%=====
%ispolzuem fiksirovannoe znachenie nachalnoi clock offset
%kb - original time offset
%kb = 8;
%=====
%ispolzuem randomnoe znachenie clock offset
schetchik = 1;
while schetchik <= chisloElementov
kb = RandKb(schetchik);
kX = RandkX(schetchik);
%=====
%t - psevdodalnosti
t1 = (((s1(1)- kX(1))^2+(s1(2)-kX(2))^2)^0.5)+kb;
t2 = (((s2(1)- kX(1))^2+(s2(2)-kX(2))^2)^0.5)+kb;
t3 = (((s3(1)- kX(1))^2+(s3(2)-kX(2))^2)^0.5)+kb;
%=====
if t1 > t2
    leastdist1 = t2;
    leastdist2 = t1;
    koordinat1=s2;
    koordinat2=s1;
else
    leastdist1 = t1;
    leastdist2 = t2;
    koordinat1=s1;
    koordinat2=s2;
end
if t3 < leastdist2
    if t3 < leastdist1
        tempDist=leastdist1;
        leastdist1=t3;
        leastdist2=tempDist;
```

```

        tempKoord=koordinat1;
        koordinat1=s3;
        koordinat2=tempKoord;
    else
        leastdist2=t3;
        koordinat2=s3;
    end
end

thirdPart = ((koordinat2(1)- koordinat1(1))^2+(koordinat2(2)-
koordinat1(2))^2)^0.5;
%
x1=koordinat1(1);
x2=koordinat2(1);
y1=koordinat1(2);
y2=koordinat2(2);
a=leastdist2;
b=thirdPart;
c=leastdist1;
%
xx1=(1/2)*((y1-y2)*sqrt(-(-x1^2+2*x2*x1-x2^2+(-c+a-y1+y2)*(-c+a+y1-y2))*(-
x1^2+2*x2*x1-x2^2+(c+a-y1+y2)*(c+a+y1-y2))*(x1-x2)^2)+(x1^3-x1^2*x2+(y2^2-
2*y1*y2-c^2+y1^2+a^2-x2^2)*x1-x2*(a^2-c^2-x2^2-y2^2+2*y1*y2-y1^2))*(x1-
x2))/((x1-x2)*(x1^2-2*x2*x1+x2^2+(y1-y2)^2));
yy1=(-sqrt(-(-x1^2+2*x2*x1-x2^2+(-c+a-y1+y2)*(-c+a+y1-y2))*(-x1^2+2*x2*x1-
x2^2+(c+a-y1+y2)*(c+a+y1-y2))*(x1-x2)^2)+y1^3-y1^2*y2+(a^2+x1^2-c^2+x2^2-
2*x2*x1-y2^2)*y1+y2^3+(x2^2-2*x2*x1+c^2-a^2+x1^2)*y2)/(2*y1^2-
4*y1*y2+2*y2^2+2*(x1-x2)^2);
xx2=(1/2)*((-y1+y2)*sqrt(-(-x1^2+2*x2*x1-x2^2+(-c+a-y1+y2)*(-c+a+y1-y2))*(x1-
x2)^2*(-x1^2+2*x2*x1-x2^2+(c+a-y1+y2)*(c+a+y1-y2)))+(x1-x2)*(x1^3-x1^2*x2+(y1^2-
2*y1*y2+y2^2+a^2-c^2-x2^2)*x1-x2*(-c^2-x2^2+a^2-y1^2+2*y1*y2-y2^2)))/((x1^2-
2*x2*x1+x2^2+(y1-y2)^2)*(x1-x2));
yy2=(sqrt(-(x1-x2)^2*(-x1^2+2*x2*x1-x2^2+(c+a-y1-y2)*(c+a-y1+y2))*(-
x1^2+2*x2*x1-x2^2+(-c+a+y1-y2)*(-c+a-y1+y2)))+y1^3-y1^2*y2+(a^2+x1^2-c^2+x2^2-
2*x2*x1-y2^2)*y1+y2^3+(x2^2-2*x2*x1+c^2-a^2+x1^2)*y2)/(2*y1^2-
4*y1*y2+2*y2^2+2*(x1-x2)^2);
%
x1=[xx1 yy1];
x2=[xx2 yy2];
%
tempSum1 = abs(abs(x1-s1) - t1) + abs(abs(x1-s2) - t2) + abs(abs(x1-s3)- t1);
tempSum2 = abs(abs(x2-s1) - t1) + abs(abs(x2-s2) - t2) + abs(abs(x2-s3)- t1);

if(tempSum1(1)+tempSum1(2)<=tempSum2(1)+tempSum2(2))
    x=x1;
else
    x=x2;
end

%=====
%pogreshnosti
PogreshnostXx = (kX - x);

PogreshnostX = ((PogreshnostXx(1))^2+(PogreshnostXx(2))^2)^0.5;
PogreshnostX = abs(real(PogreshnostX));
%del

%
%=====
vectPogreshnostX(schetchik) = PogreshnostX;
schetchik=schetchik + 1 ;
end

```

```

%grafiki
%bez pogreshnosti
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb);
hold on
plotCircle(s2(1), s2(2), t2-kb);
hold on
plotCircle(s3(1), s3(2), t3-kb);
axis equal
%bez pogreshnosti
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1);
hold on
plotCircle(s2(1), s2(2), t2);
hold on
plotCircle(s3(1), s3(2), t3);
axis equal

%psevdodalnost
plotCircle(x(1), x(2), 0.6);
plotCircle(x(1), x(2), 0.7);
plotCircle(x(1), x(2), 0.5);
plotCircle(x(1), x(2), 0.4);
plotCircle(x(1), x(2), 0.3);
plotCircle(x(1), x(2), 0.2);
plotCircle(x(1), x(2), 0.1);
%plotCircle(x(1), x(2), 0.8);
%plotCircle(x(1), x(2), 0.5);
%plotCircle(x(1), x(2), 0.3);
%plotCircle(x(1), x(2), 0.1);
%axis equal
%=====
%plotCircle(s1(1), s1(2), t1-kb+0.5);
%hold on
%plotCircle(s2(1), s2(2), t2-kb+0.5);
%hold on
%plotCircle(s3(1), s3(2), t3-kb+0.5);
%axis equal
%=====
%videlenie original range
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb+0.1);
hold on
plotCircle(s2(1), s2(2), t2-kb+0.1);
hold on
plotCircle(s3(1), s3(2), t3-kb+0.1);
axis equal
%=====
plotCircle = @(xc, yc, R) plot(xc + R * cos(0:0.001:2*pi), yc + R *
sin(0:0.001:2*pi));
plotCircle(s1(1), s1(2), t1-kb-0.1);
hold on
plotCircle(s2(1), s2(2), t2-kb-0.1);
hold on
plotCircle(s3(1), s3(2), t3-kb-0.1);
axis equal
%=====
vectPogreshnostX
%=====

```



```

%=====
figure;hist(vectPogreshnostX,250);
%=====
%srednee
srednee = 0;
schet1 = 1;
while schet1 < chisloElementov
srednee = srednee + vectPogreshnostX(schet1);
schet1 = schet1 +1;
end
srednee = srednee / chisloElementov;
srednee
%srednekvadratichnoe
schet1 = 1;
srednekvadratichnoe = 0;
while schet1 < chisloElementov
srednekvadratichnoe = srednekvadratichnoe +(vectPogreshnostX(schet1) -
srednee)^2 ;
schet1 = schet1 +1;
end
srednekvadratichnoe = (srednekvadratichnoe / chisloElementov)^0.5;
srednekvadratichnoe

```