

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет радіотехнічних систем літальних апаратів

Кафедра проектування радіоелектронних систем літальних апаратів

**Пояснювальна записка
до дипломної роботи
бакалавра**

(освітньо-кваліфікаційний рівень)

на тему ***«Прийом і декодування сигналів літакових відповідачів
ADS-B з використанням SDR-приймачів RTL-SDR і AIRSPY.
Порівняння якісних показників»***

XAI.501.546к.050901.1405279.16В

Виконав: студент 4 курсу групи № 546к
напряму підготовки (спеціальності)
6.050901 – Радіотехніка

(шифр і назва напряму підготовки (спеціальності))

Чжан Даочен

(прізвище й ініціали студента)

Керівник: Шульгін В.І.

(прізвище й ініціали)

Рецензент: Олійник В.П.

(прізвище й ініціали)

Харків – 2016

НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ
ім. М.Є. ЖУКОВСЬКОГО
«Харківський авіаційний інститут»

Факультет _____ радіотехнічних систем літальних апаратів
Кафедра _____ проектування радіоелектронних систем літальних апаратів
Освітньо-кваліфікаційний рівень _____ бакалавр
Напрямок підготовки _____ 6.050901 «Радіотехніка»

ЗАТВЕРДЖУЮ
Завідувач кафедри проектування
радіоелектронних систем
літальних апаратів, д.т.н., с.н.с.
_____ Павліков В.В.
“ _____ ” _____ 2016 року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

Чжан Даочен

(прізвище, ім'я, по батькові)

1. Тема випускної роботи ***«Прийом і декодування сигналів літакових відповідачів ADS-B з використанням SDR-приймачів RTL-SDR і AIRSPY. Порівняння якісних показників»***

керівник випускної роботи _____ Шульгін В.І., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 17.05.16 р. № 555-уч

2. Строк подання студентом дипломної роботи _____ 10 червня 2016 р. _____

3. Вихідні дані до дипломної роботи: разработать алгоритмы и МАТЛАБ-программы для приема и декодирования сигналов самолетных ответчиков ADS-B с использованием SDR-приемников RTL-SDR и AIRSPY, сравнить качествен-ные показатели приема и декодирования и их зависимости от шумовых свойств, полосы пропускания, частоты дискретизации и других характеристик приемников. Частота сигнала 1090 МГц. Способ модуляции ПИМ. Частота дискретизации 10 (3) МГц. Разрядность данных 12 (8 бит). Полоса приемника 10 (3 МГц).

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити: Структура системы ADS-B и ее компоненты; Способ передачи информации, метод модуляции и кодирования, параметры передаваемых сигналов; Структура передаваемых данных; Типы сообщений ADS-B; Прием

данных ADS-B с использованием программно-определяемого радио (SDR); Прием сигнала с использованием SDR-приемников RTL-SDR и AIRSPY Обнаружение пакетов данных (сквиттеров DF11 и DF17); Идентификация пакетов DF11 и DF17; Декодирование данных ADS-B; Оценка помехоустойчивости процедур обнаружения и декодирования ADS-B-данных; Сравнение качества приема и декодирования данных ADS-B при использовании приемников RTL-SDR и ARSPY.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентация по материалам выпускной работы объемом 25-30 слайдов.

6. Консультанти частин випускної роботи

Частина	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна	Шульгін В.І.	11.05.2016	11.05.2016
Додаткова	Шульгін В.І.	11.05.2016	11.05.2016
Економічна	Хмель С.М., к.е.н., доц. к. 601		

7. Дата видачі завдання 11 травня 2016 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання та аналіз завдання	11.05.15	
2	Робота з інформаційними джерелами	20.05.15	
3	Узгодження з керівником обраних технічних рішень	21.05.15	
4	Виконання моделювання в МАТЛАБ	25.05.15	
5	Обговорення з керівником результатів розробки	25.05.15	
6	Усунення зауважень та уточнення кількісних оцінок	27.05.15	
7	Виконання завдання з питань безпеки	29.05.15	
9	Оформлення пояснювальної записки	5.06.15	
8	Виконання графічної частини роботи	7.06.15	
9	Попередній захист роботи	10.06.15	
10	Подання роботи на рецензування	13.06.15	
11	Усунення зауважень рецензента	13.06.15	
12	Захист роботи на державній комісії	15.06.15	

Студент _____ Чжан Даочен
(підпис) (прізвище та ініціали)

Керівник дипломної роботи _____ Шульгін В.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка складається з 103 сторінок, 39 малюнків, 25 таблиць, 4 додатків, 19 джерел.

Об'єкт дослідження-програмний декодер ADS-B сигналу.

Предмет дослідження – алгоритми і програмне забезпечення для детектування, ідентифікації і декодування ADS-B сигналу.

Мета роботи – дослідження можливості використання RTL-SDR і AIRSPY приймачів для прийому і обробки ADS-B сигналу. Порівняння якісних показників цих приймачів.

Метод дослідження – отримання, детектування, ідентифікація і декодування сигналу, отриманого за допомогою приймачів RTL-SDR і AIRSPY в середовищі математичного моделювання MATLAB.

Для детектування сигналу запропоновано використовувати оптимальний виявитель сигналу – енергетичний детектор. Зроблено аналіз отриманого сквіттера, за яким було встановлено оптимальний селектор для демодуляції сигналу. Різні способи можуть бути використані для демодуляції сигналу (імпульсний метод, метод мінімального СКО і метод за допомогою кореляційної функції). Для декодування різних типів ADS-B повідомлень потрібно використовувати відповідні алгоритми для ідентифікації літального апарату, визначення координат і висоти ЛА, визначення вектора швидкості ЛА і т.д.

ADS-B, СКВІТТЕР, ПРИЙМАЧ RTL-SDR, AIRSPY ПРИЙМАЧ, ПРИЙОМ, ВИЯВЛЕННЯ, ІДЕНТИФІКАЦІЯ, ДЕКОДУВАННЯ, MATLAB.

РЕФЕРАТ

Пояснительная записка состоит из 103 страниц, 39 рисунков, 25 таблиц, 4 приложений, 19 источников.

Объект исследования– программный декодер ADS-B сигнала.

Предмет исследования– алгоритмы и программное обеспечение для детектирования,идентификации и декодирования ADS-B сигнала.

Цель работы – исследование возможности использования RTL-SDR и AIRSPY приемники для приема и обработки ADS-B сигнала. Сравнение качественных показателей этих приемников.

Метод исследования – получение, детектирование,идентификация и декодирование сигнала, полученного при помощи приемники RTL-SDR и AIRSPY в среде математического моделирования MATLAB.

Для детектирования принимаемого сигнала предложено использовать оптимальный обнаружитель сигнала – энергетический обнаружитель. Произведен анализ полученного сквиттера, по которому был установлен оптимальный селектор для демодуляции сигнала. Различные способы могут быть использованы для демодуляции сигнала (импульсный метод, метод минимального СКО и метод с помощью корреляционной функции). Для декодирования различных типов ADS-B сообщений нужно использовать соответствующие алгоритмы для идентификации летательного аппарата, определения координат и высоты ЛА, определения вектора скорости ЛА и т.д.

ADS-B, СКВИТТЕР, ПРИЕМНИК RTL-SDR, ПРИЕМНИК AIRSPY, ПРИЕМ,ОБНАРУЖЕНИЕ, ИДЕНТИФИКАЦИЯ, ДЕКОДИРОВАНИЕ, MATLAB.

ABSTRACT

Final work: 103 pages, 39 figures, 25 tables, 4 appendix 19 sources.

Object of research - a software detector ADS-B signal.

Item of research - algorithms and software for the detection, identification and decoding ADS-B signal.

Purpose of the work - using RTL-SDR and AIRSPY receivers for reception and processing of ADS-B signal. Comparison of quality indicators.

Methods of research - receiving, detecting, and decoding the ADS-B signal received by the receivers RTL-SDR and AIRSPY in Matlab mathematical modeling environment.

For detecting the received signal, the optimal signal detector—the energy detector may be used. According to analysis of the received squitter, the optimal selector for the signal demodulation can be set. Various methods can be used for the signal demodulation (the pulsed method, the minimum-standard-deviation method and the method with the help of the correlation function).

ADS-B, SQUITTER, RTL-SDR RECEIVER, AIRSPY RECEIVER, RECEPTION, DETECTION, IDENTIFICATION, DECODING, MATLAB.

摘要

论文内容：103 页，39 幅图，25 个表格，4 项附录，19 条参考文献。

研究对象—ADS-B 信号解码器

研究课题—ADS-B 信号检测，识别，解调和解码的处理算法

研究目的—探讨使用 RTL-SDR 和 AIRSPY 接收器接收和处理 ADS-B 信号以及两种接收器的质量指标比较。

研究方法—数学建模(在 Matlab 里处理和加工 RTL-SDRAIRSPY 接收器接收的 ADS-B 信号)

为了更好地检测所接收的信号，我们使用最合适的信号检测器：能量检测器。通过接收到的报文信号分析，我们可以为解调选择设置最合适的选择器。多种方法可以被用于解调所接收到的 ADS-B 脉冲信号（脉冲间隔法，最小方差法以及相关函数法）。对于不同类型的 ADS-B 信号应用不同的算法进行解码：飞机编号的鉴定，坐标和高度的确定，飞机的速度矢量的确定等。

关键词：ADS-B，断续发射，RTL-SDR 接收器，接收器 AIRSPY，接收，检测，识别，解码，MATLAB。

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ	9
ВВЕДЕНИЕ	10
1 НАЗНАЧЕНИЕ, ПРИНЦИП ДЕЙСТВИЯ И СТРУКТУРА ADS-B	12
1.1 Назначение ADS-B.....	12
1.2 Структура системы ADS-B и ее компоненты	12
1.3 Способ передачи информации, метод модуляции и кодирования, параметры передаваемых сигналов.....	15
1.4 Сквиттеры	16
1.5 Структура передаваемых данных.....	16
1.6 Содержание сообщений ADS-B.....	18
1.7 Типы сообщений ADS-B	19
1.8 Метод кодирования местоположения (CPR).	19
2 ПРИЕМ СИГНАЛОВ ADS-B	23
2.1 Прием данных ADS-B с использованием программно-определяемого радио (SDR).....	23
2.2 SDR приемники RTL-SDR и AIRSPY, структура и технические характеристики приемников.	24
2.2.1 SDR приемники RTL-SDR	24
2.2.2 SDR приемник AIRSPY	27
2.2.3 Сравнение SDR приёмников (RTL-SDR и AIRSPY).....	28
3 ЭКСПЕРИМЕНТ ПО ПРИЕМУ И ОБРАБОТКЕ СИГНАЛОВ ADS-B	31
3.1 Прием сигнала с использованием SDR-приемников RTL-SDR и AIRSPY.....	31
3.2 Обнаружение пакетов данных (сквиттеров DF11 и DF17).....	33
3.2.1 Детектирование сквиттеров ADS-B на фоне помех	34
3.2.2 Энергетический обнаружитель.....	40
3.3 Идентификация пакетов DF11 и DF17	43
3.4 Обнаружение преамбулы. Оценка временного положения (начала) сквиттера	45

3.5	Декодирование двоичных последовательностей с ПИМ (кода Манчестера)	47
3.5.1	Декодирование импульсным методом.....	48
3.5.2	Метод декодирования по минимуму СКО	50
3.5.3	Метод декодирования по максимуму корреляции	51
3.6	Декодирование с обнаружением ошибок (CRC)	52
3.7	Декодирование данных ADS-B	54
3.7.1	Определение DF и TC	54
3.7.2	Идентификация летательного аппарата.....	55
3.7.3	Определение координат и высоты ЛА.....	57
3.7.4	Определение вектора скорости ЛА	62
3.8	Оценка помехоустойчивости процедур обнаружения и декодирования ADS-B-данных.....	66
4	СРАВНЕНИЕ КАЧЕСТВА ПРИЕМА И ДЕКОДИРОВАНИЯ ДАННЫХ ADS-B ПРИ ИСПОЛЬЗОВАНИИ ПРИЕМНИКОВ RTL-SDR и ARSPY	70
5	ЭКОНОМИЧЕСКАЯ ЧАСТЬ	72
5.1	Обзорная часть	72
5.2	Расчёт себестоимости программного продукта	73
	ЗАКЛЮЧЕНИЕ	85
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	86
	ПРИЛОЖЕНИЕ А. Код программы приема сигналов	87
	ПРИЛОЖЕНИЕ Б. Код программы декодирования сигналов (в двоичном формате)	88
	ПРИЛОЖЕНИЕ В. Оценка помехоустойчивости процедур декодирования ads-b-данных	92
	ПРИЛОЖЕНИЕ Г. Сравнение качества декодирования данных ads-b при использовании приемников rtl-sdr и airspy	98

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ADS-B – Автоматическое зависимое наблюдение в режиме радиовещания;

GPS – спутниковая система глобального позиционирования (США);

ГЛОНАСС – глобальная навигационная система (Россия);

СУВД – система управления воздушным движением;

ВС – воздушное судно;

FMS – Flight Management System, бортовая система управления полетом воздушного судна;

SDR (Software Defined Radio) – программно-определяемое радио;

ПК – персональный компьютер;

INS – инерциальная навигационная система;

IRS – инерциальная система отсчета;

ADS-B IN – класс устройств, который принимает данные ADS-B;

ADS-B OUT – класс устройств, который передает данные ADS-B;

TIS-B – Traffic Information Service Broadcast,

FIS-B Flight Information Services-Broadcast

Identification Friend or Foe system (IFF)

SSR – Secondary Surveillance Radar, вторичный радар;

PPM – Pulse-position modulation, фазово-импульсная модуляция (ФИМ);

CPR – Compact Position Reporting, метод компактного донесения о местоположении;

ICAO – International Civil Aviation Organization, Международная организация гражданской авиации;

FAA – Federal Aviation Administration, Федеральное управление гражданской авиации США;

ТСХО – Temperature Compensated Crystal Oscillator, Термокомпенсированные кварцевые генераторы;

ОСХО – Oven Controlled Crystal Oscillator, Термостатированные кварцевые генераторы;

СКО – Среднеквадратическое отклонение;

ВВЕДЕНИЕ

Обеспечение безопасности полетов воздушных судов представляет собой важнейшую научно-техническую и инженерную проблему. Эта проблема является особенно актуальной в условиях постоянного роста плотности трафика, как в зонах аэропортов, так и в глобальном воздушном пространстве и решается системами управления воздушным движением (СУВД), которые являются неотъемлемой частью современной авиации.

Работа СУВД основана на определении параметров траектории наблюдаемых воздушных судов (ВС) по внешним измерениям (на решении навигационных задач). Технической базой современных СУВД служат обзорные (первичные) радиолокаторы, дополняемые приемопередающими устройствами (транспондерами, вторичными радиолокаторами), использующими для наблюдения за воздушным движением и решения навигационных задач данные спутниковых навигационных систем (GNSS, Глонасс, и т.д.). Данные о движении обрабатываются специальными алгоритмами, позволяющими распознавать опасное движение самолёта: выход за пределы регламентированных воздушных трасс, уход с предписанного эшелона, недопустимое сближение, полёт в запрещённых зонах и т.д.

Одной из таких, современных систем наблюдения за воздушным движением является система ADS-B (сокращение от Automatic Dependent Surveillance - Broadcast - Автоматическое зависимое наблюдение в режиме радиовещания). Это технология, внедряемая в настоящее время по всему миру и позволяющая пилотам в кабине самолета, а также диспетчерам на наземном пункте, видеть с высокой точностью и высокой частотой обновления движение воздушных судов на экране компьютера без использования традиционных обзорных радиолокаторов.

Сокращение ADS-B расшифровывается следующим образом:

1. Automatic - работает автоматически и не требует вмешательства оператора;
2. Dependent - зависит от системы GPS и от системы управления полетом (Flight Management System - FMS);
3. Surveillance - обеспечивает наблюдение за самолетом подобно радарным системам;
4. Broadcast - широковещательная непрерывная радиотрансляция данных всем самолетам и наземным станциям.

Основа работы системы базируется на принципе вторичной радиолокации и состоит в следующем: оборудованный ADS-B самолет примерно каждую секунду передает по радиоканалу свою точную позицию, высоту и вектор скорости, определяемые с высокой точностью в течение всего полета по данным GPS и бортовых навигационных средств.

Приемники ADS-B, установленные на борту другого самолета, в диспетчерском пункте, или в любой доступной точке обеспечивают прием всех этих данных, а компьютер обрабатывает и отображает их в удобной форме, например на карте. Система ADS-B реализует принцип "каждый видит каждого", что позволяет предотвращать конфликтные ситуации.

Преимуществами технологии ADS-B являются:

1. Низкая стоимость, высокая точность и большая скорость обновления данных.
2. Система ADS-B состоит из сети относительно простых и дешевых радиоприемников, которые просты в установке и использовании по сравнению с обзорными радиолокаторами.
3. ADS-B обеспечивает гораздо более точное определение координат самолета и выдает информацию самолета примерно каждую секунду.
4. Поскольку ADS-B имеет более высокую точность определения координат, это позволяет уплотнить трафик и сделать его более эффективным в районах с высокой интенсивностью полетов.
5. Полное покрытие территории еще одно преимущество системы. ADS-B оборудование можно устанавливать в районах, где использование радарного оборудования не представляется возможным.
6. Повышение безопасности в кабине пилотов - ADS-B позволяет отображать в кабине пилотов на экране монитора другие самолеты, оборудованные ADS-B.

Все эти преимущества ADS-B могут быть реализованы, если обеспечивается надежный прием сигналов самолетных ответчиков в пунктах наземного наблюдения за ВС. Одним из способов организации недорогого и повсеместного приема ADS-B является использование SDR-приемников, работающих под управлением персональных компьютеров (ПК), снабженных соответствующим программным обеспечением. В настоящее время на рынке радиотехнических устройств представлено достаточно много SDR-приемников различных производителей, обладающих различными техническими характеристиками и стоимостью. К наиболее популярным устройствам можно отнести недорогие приемники с USB подключением к ПК – RTL-SDR и более дорогие, но обладающие более высокими техническими характеристиками – приемники AIRSPY.

Целью настоящей работы является разработка алгоритмов и МАТЛАБ-программ для приема и декодирования сигналов самолетных ответчиков ADS-B с использованием SDR-приемников RTL-SDR и AIRSPY, а также сравнения качественных показателей приема в этих случаях, их зависимости от шумовых характеристик, полосы пропускания, частоты дискретизации и других характеристик приемников.

1 Назначение, принцип действия и структура ADS-B

Рассмотрим кратко назначение, принцип действия, структуру системы ADS-B и ее технические характеристики.

1.1 Назначение ADS-B

Система ADS-B (Automatic dependent surveillance-broadcast), в русской транскрипции – АЗН-В (автоматическое зависимое наблюдение-вещание) – технология, позволяющая и пилотам в кабине самолета, авиадиспетчерам на наземном пункте, а также любому “желающему” наблюдать на экране ПК трафик движения воздушных судов с большой точностью и получать различную аэронавигационную информацию. Оборудованное ADS-B транспондером воздушное судно передает всем свои координаты месторасположения в течение всего полёта, вместе с другими данными, такими как курс, высота, горизонтальная и вертикальная скорость. Приёмники ADS-B, интегрированные в системы контроля воздушного трафика авиадиспетчеров, а также установленные на борту воздушного судна, обеспечивают точное отображение на экране радара движения воздушных судов, оборудованных ADS-B, как в небе, так на земле. ADS-B также передает пилотам в реальном времени погодную информацию. Эта информация значительно расширяет осведомленность пилота о ситуации и повышает безопасность полетов. Доступ к ADS-B информации бесплатен и свободен для всех [1].

1.2 Структура системы ADS-B и ее компоненты

Система ADS-B представляет собой набор средств связи и наблюдения информации в одной системе. Она состоит из трех частей (источника информации, канала передачи, устройств обработки и отображения информации).

– Источник информации. Основная информация в ADS-B представляет собой 4-мерное пространство данных о местоположении (долгота, широта, высота и время) и, возможно, другими дополнительными сведениями (конфликт оповещения информации, путевые углы и другая информация о маршрутах), а также информация идентификации и категории самолетов. Она может также включать некоторую другую дополнительную информацию, как курс самолёта, воздушную скорость, скорость ветра, направление ветра и температуру окружающей среды. Эта информация может быть получена из следующих видов радиоэлектронного бортового и внешнего оборудования: 1) Глобальная навигационная спутниковая система (GPS); 2) инерциальная навигационная система (INS); 3) инерциальная система отсчета (IRS); 4) другие бортовые датчики, а также от диспетчера или руководителя полетов. Информация получается и накапливается на борту ЛА и передается получателям в радиовещательном режиме или по запросу.

– Канал передачи данных. Канал передачи данных в виде ADS-B сообщений по воздуху – воздушным и воздушно - наземным линиям передачи данных в широковещательном режиме, рисунок 1.1[2].

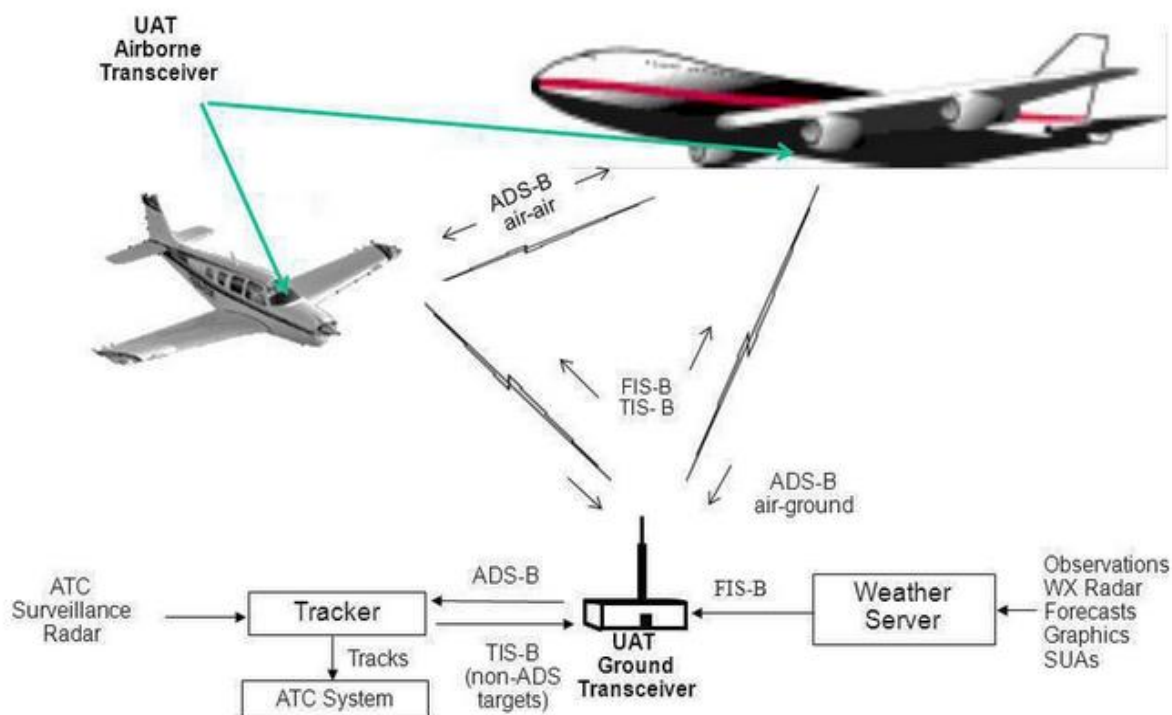


Рисунок 1.1 – Каналы передачи данных ADS-B

ADS-B устройства делятся на две категории: излучения (OUT) и приема (IN). ADS-B приемник (ADS-B IN) – этот класс устройств, который принимает данные ADS-B, но не может передавать ADS-B данные другим воздушным судам или наземным станциям. Приемник получает как информацию о движении судов, так и погодную информацию. В настоящее время FAA ограничило реализацию ADS-B в том, что наземные станции будут передавать данные трафика (включая данные радарного трафика) только тогда, когда примут информацию как минимум от одного ADS-B OUT устройства на воздушном судне, переданную в радиусе обслуживания. Таким образом, воздушное судно, оборудованное ADS-B приемником, может и не «видеть» данные трафика, если в зоне обслуживания отсутствуют воздушные суда с ADS-B OUT транзмиттерами и не передают сигнал. Погодная информация же всегда передается наземными станциями и всегда доступна для приема через ADS-B приемник.

ADS-B транзмиттеры (ADS-B OUT) — этот класс устройств, который может передавать ADS-B данные. Транзмиттеры также могут «пробуждать» ADS-B наземные станции и переключать их в режим передачи трафика воздушным судам в зоне их действия. ADS-B устройства, способные передавать

положение воздушного судна и путь пролета, должны быть сертифицированы FAA.

ADS-B трансиверы — это класс устройств, способных, как передавать так и принимать ADS-B данные. ADS-B трансиверы могут «пробуждать» FAA ADS-B наземные станции и переключать их в режим передачи трафика воздушным судам в зоне их действия. ADS-B устройства, способные передавать положение воздушного судна и путь пролёта, должны быть сертифицированы FAA. В данной работе мы будем рассматривать только устройства класса ADS-B IN, и только вопросы приема сигналов ADS-B.

— Устройства обработки и отображения информации *ADS-B*. Обработка информации ADS-B, включает прием аналогового сигнала, преобразование его в цифровую форму, обнаружение пакетов данных (сквиттеров), демодуляцию и декодирование данных с исправлением ошибок, декодирование данных о местоположении и других параметрах ЛА, определение степени достоверности полученных данных, отображение полученной информации на дисплее ПК на фоне карты региона, над которым выполняется полет ЛА - рисунок 1.2[3].

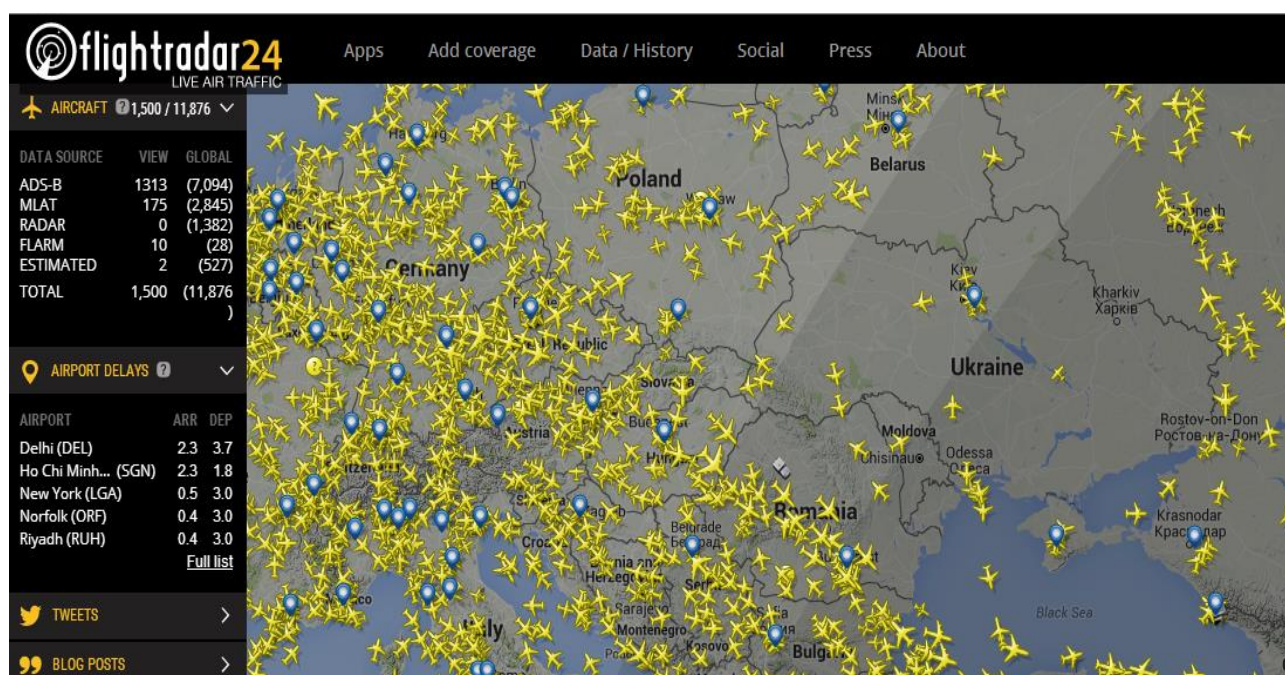


Рисунок 1.2 – Отображение воздушной информации в системе ADS-B

В работе будет рассмотрен прием данных ADS-B с использованием программно-определяемых SDR-приемников, а также алгоритмы и программы обработки принимаемых сигналов, выполненные в МАТЛАБ.

1.3 Способ передачи информации, метод модуляции и кодирования, параметры передаваемых сигналов

Исторически ADS-B развивалась из следующих систем. Режим Mode-A – так называемый вторичный радар (Secondary Surveillance Radar (SSR)), используется как гражданскими, так и военными воздушными судами, обеспечивает до 4096 идентификационных кодов и является наиболее широко распространенным и используемым режимом. Работает на частоте запроса 1030 МГц. В режиме Mode-A/C передаются данные о высоте, сквоке (коде самолетного ответчика - транспондера, выдаваемый диспетчером пилоту, который вводит его перед началом полета или по требованию диспетчера) и ICAO-коде самолета. Координаты ЛА в этом режиме не передаются.

В режиме Mode-S приемник «слушает» на частоте запроса 1030 МГц, и отвечает на запрос на частоте 1090 МГц, модулируя несущую сигнала методом DPSK (PPM) для минимизации помех другим запросам Mode A/C. Когда транспондер ВС получает запрос от наземного радара, он подтверждает получение запроса излучением фазово-импульсно модулированного (ФИМ) (англ. Pulse-position modulation, PPM) сигнала на частоте 1090 МГц. Кроме этого, вне зависимости от поступления запроса от наземного радара, примерно каждую секунду самолетом передается расширенный сквиттер (extended squitter), этот сквиттер содержит координаты месторасположения воздушного судна. Координаты воздушного судна передаются только в режиме Mode-S.

Для определения точных координат воздушного судна в пространстве ADS-B базируется на системе GPS. Эта информация о положении ВС комбинируется с другой информацией, такой как тип воздушного судна, скорость, его номер, рейс, курс, вертикальная скорость и затем широковещательно передается каждую секунду. Другие воздушные суда и наземные станции, оборудованные ADS-B, в радиусе примерно 300 километров принимают эту информацию. Наземные станции комбинируют полученную с разных точек информацию с дополнительной информацией, полученной от наземных радаров для не-ADS-B устройств, и ретранслируют данные для всех воздушных судов в радиусе обслуживания. Авиатрафик и погодная информация может отображаться в кабине пилотов.

Адресно-навигационная информация с борта ЛА в виде последовательность битов включается в пакет, который излучается бортом как широковещательная (сквиттер) рассылка. Пакет состоит из преамбулы, состоящей из неизменной фиксированной и заранее известной последовательности из 8 бит длительностью 8 мкс и следующих за ней 56 или 112 блоков данных, каждый из которых занимает 1 мкс. Биты данных кодируются кодом Манчестера, который показывает переходы из 0 в 1 или наоборот в исходных данных.

Такое кодирование дает следующие преимущества:

- в потоке данных отсутствует постоянная составляющая, если в исходной комбинации преобладают нули или единицы, выходная последовательность все равно будет периодичной и состоять из перемежающихся нулей и единиц;
- перемежающаяся последовательность содержит информацию о синхронизации, то есть является самосинхронизирующейся.

Использование кода Манчестера приводит к изменению местоположения импульса, и, соответственно к кодированию PPM (Pulse Position Modulation). При скорости передачи 1 Мбит/с на несущей 1090 МГц во временном интервале 1 мкс, выделенном для каждого бита, единица будет представлена импульсом 0,5 мкс находящимся в начале этого интервала, нуль будет представлен импульсом 0,5 мкс который будет находиться в конце интервала.

1.4 Сквиттеры

Термин «сквиттер» взял свое название в старых системах дальномерного оборудования где, наземная станция излучала запросы или «сквиттеры». Когда запросчик самолета находился в зоне «радиовидимости» и принимал сквиттер, то только тогда, в ответ он передавал запрос расстояния и получал информацию от наземной станции. Это позволяло уменьшить загрузку эфира из-за уменьшения количества ненужных запросов в эфире, что оптимизировало пропускную способность наземных станций.

Технология Mode S имеет два типа сквиттеров: короткий 56 битный, называемый DF11, и расширенный 112 битный – типа DF17. Сквиттер – это пакет данных с идентификатором. В основе системы ADS-B лежит использование расширенного сквиттера (extended squitter DF17) для ADS-B пакета данных. Сквиттер DF17 передается воздушным судном вне зависимости от того, запрашивает ли самолет наземная станция или другой самолет. Он выдается каждую секунду и несет в себе информацию о положении самолета в воздушном пространстве, идентификации самолета, категории, а также скорости, высоты ВС.

1.5 Структура передаваемых данных

На рисунке 1.3 показан пример типичного сквиттера ADS-B – пакета Mode-S, состоящего из 2-х отдельных частей: преамбулы и следующего за ней блока данных. Преамбула используется для «пробуждения» приемника запросов и декодера и состоит из 4 импульсов каждый длительностью 0.5 мкс, начинающимися в T_0 , T_0+1 мкс, $T_0+3.5$ мкс и $T_0+4.5$ мкс.

Полный блок (пакет) данных состоит из 56-ти или 112-ти бит общей длительностью 56 или 112 мкс, в зависимости от типа сквиттера.

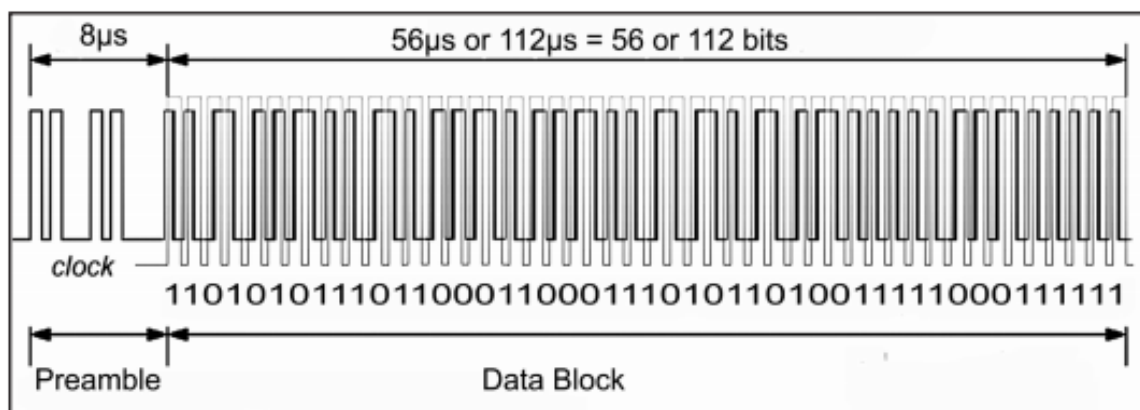


Рисунок 1.3 – Пример пакета Mode-S

Преамбула используется для распознавания начала пакета, поскольку ее битовая структура известна, и к тому же она (как и любая другая правильная синхронизирующая последовательность) не может встретиться в потоке данных, поскольку содержит пропуски в импульсах. По преамбуле в реальном времени идет настройка максимального и минимального уровня сигнала, чтобы адаптироваться под конкретные характеристики радиолинии борт – земля. Кроме этого, контролируется целостность преамбулы, которая может быть разрушена наложением сквиттера с другого борта.

На рисунке 1.4 показано содержимое первого типа ADS-B пакета – “короткого” Mode-S пакета (short message reply). Здесь также есть 8 мкс. преамбула, но 56-ти битный блок данных разделен на три части. Первые 5 бит содержат идентификатор формата, следующие 27 бит – информация слежения и контроля, остальные 24 бит содержат данные четности для обеспечения качества данных.

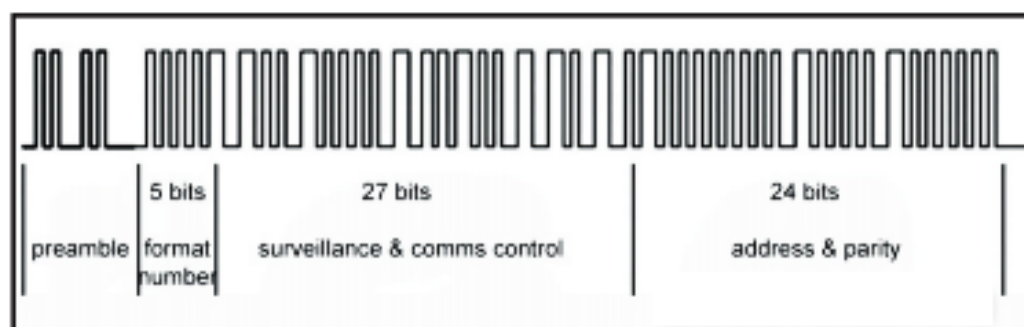


Рисунок 1.4 – Содержимое “короткого” пакета Mode-S

На рисунках 1.5 и 1.6 показано содержание типичного “длинного” Mode-S пакета (long message reply), и “расширенного” Mode-S пакета. Как и в коротком ответе есть 8 мкс преамбула, но блок данных содержит 112 бит, разделенных на 4 части. Снова первые 5 бит содержат идентификатор формата, следующие 27 бит содержат информацию слежения и контроля, далее 56 бит сообщения и снова 24 бита четности[4].

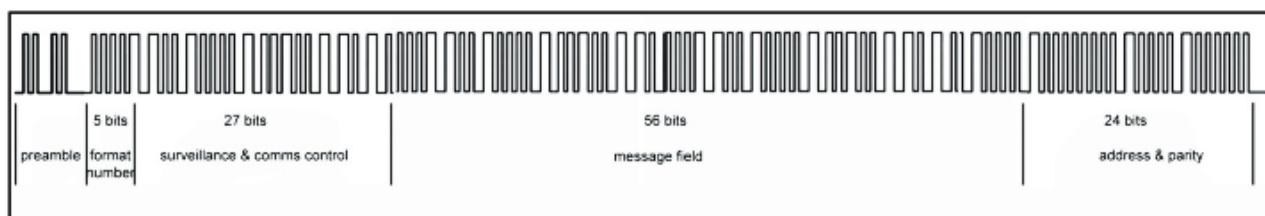


Рисунок 1.5 – Содержимое “длинного” пакета Mode-S

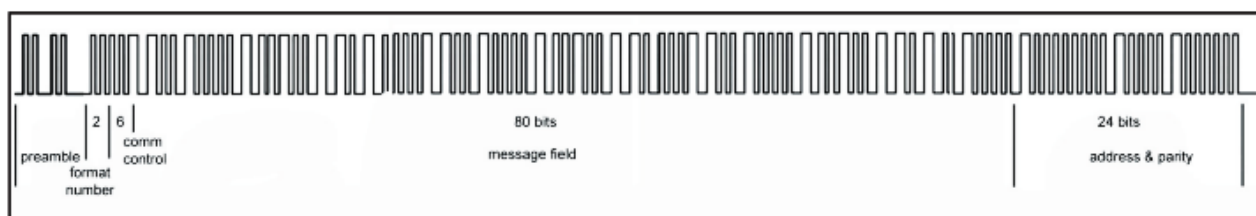


Рисунок 1.6 – Содержимое “расширенного” пакета Mode-S

Содержание пакетов ADS-B зависит от его типа.

1.6 Содержание сообщений ADS-B

В качестве примера рассмотрим типичную посылку ADS-B длиной в 112 бит, которая в двоичном формате выглядит так:

10001101010010000100000011010110001000000010110011000011

01110001110000110010110011100000010101110110000010011000

В таблице 1.1 приведены ключевые биты этой посылки.

Таблице 1.1 – Ключевые биты сообщений ADS-B

Начальный бит	Конечный бит	Аббревиатура	Название
1	5	DF	Формат сообщения
6	8	CA	Подтип сообщения
9	32	ICAO24	ICAO адрес ЛА
33	88	DATA	Пакет данных
89	112	PC	Контрольная сумма

Таким образом, данное сообщение может быть идентифицировано, как пакет DF = 17 (в десятичном формате), или 10001 (в двоичном).

Внутри пакета данных в свою очередь важными являются биты с 33 по 37 (5 бит) – это тип кода (Type Code)[5].

Таблица 1.2 – Тип кода

Начальный бит	Конечный бит	Аббревиатура	Название
33	37	ТС	Type Code

1.7 Типы сообщений ADS-B

Естественно, что различные виды навигационной и другой информации, которые могут передаваться с борта ЛА не могут поместиться в одном в 56-ти или 112-ти битовом пакете ADS-B. Поэтому информация распределяется по нескольким пакетам, с указанием DF и ТС (Формата сообщения и Типа кода). По этим значениям легко понять – какая информация содержится в пакете данных.

Таблица 1.3 –Тип сообщения

DF	ТС	Содержимое
17	1 to 4	Идентификатор ЛА
17	5 to 8	Пространственное положение
17	9 to 18	Барометрическая высота
17	19	Вектор скорости
17	20 to 22	Высота по GNSS
17	23	Тестовое сообщение
17	24	Статус системы
17	25 to 27	Резерв
17	28	Расширенный сквиттер (АС статус)
17	29	Target state and status (V.2)
17	30	Резерв
17	31	Статус ЛА

1.8 Метод кодирования местоположения (CPR).

В отличие от большей части навигационных и информационных данных, передача которых в ADS-B производится в не кодированном виде, информация о местоположении ЛА кодируется с целью уменьшения объема передаваемой

посылки. Поэтому, для того, чтобы были понятными действия, которые будут выполняться при определении географических координат ЛА, рассмотрим способ их кодирования.

Метод CPR или “компактного донесения о местоположении” был разработан для более экономного способа представления географических координат. Этот метод используется в ADS-B сообщениях и позволяет уменьшить количество бит кода, необходимых для передачи широты и долготы для каждой позиции самолета.

У ADS-B сообщений точность определения местоположения (или позиционное разрешение координат) составляет ~ 5.1 м при нахождении самолета в воздухе и ~ 1.3 метра на земле. Окружность земного шара составляет порядка 40 000 км. Определим число возможных значений для позиции: разделим $40\,000\,000\text{ м} / 5.1\text{ м} = \sim 7\,800\,000$. Чтобы закодировать такую величину потребуется выделить 23 бита в каждом ADS-B сообщении с координатами.

Долгота изменяется в диапазоне $-180^\circ/180^\circ$, то есть как раз использует полную окружность (360°), значит потребует для своей передачи все 23 бита. Широта измеряется в диапазоне $-90^\circ/90^\circ$, то есть использует половину окружности (180°), и может принимать 3 900 000 значений для позиции, что потребует 22 битной длины в коде. Аналогично для координат самолета, который находится на земле, потребуется 25 бит кода для передачи долготы и 24 бита - для широты.

CPR же передает позицию, используя по 17 бит для широты и долготы, плюс 1 служебный бит “CPR формата”. То есть, этот метод экономит 10 бит кода в координатном сообщении при нахождении самолета в воздухе и 14 бит, когда он на земле. Поскольку сообщения с координатами обычно передаются в эфир два раза в секунду, то CPR сэкономит 20 бит/сек для воздуха и 28 бит/сек для земли.

Для ADS-B транспондера максимальная скорость передачи ограничена стандартом и составляет 6.2 сообщения в сек. Из 112 бит в каждом сообщении, только 56 доступны для ADS-B данных. Причем первые 5 бит используются для передачи типа сообщения. Остаются свободными только 51 бит, которые и доступны для размещения данных. Если умножить 51 бит/сообщение на 6.2 сообщения/сек, то получим скорость передачи «полезной нагрузки» - 316 бит/сек. Таким образом метод CPR с экономит 6% - 9% доступных бит в каждом сообщении для других целей. Но за эту экономию придется заплатить сложными расчетами в программе.

CPR можно рассматривать как систему координат и набор алгоритмов для их координатных преобразований. У метода система координат - сферическая и сравнима с аналогичной «широта/долгота», обычно используемой для навигации. Алгоритмы координатного преобразования переводят значения широты/долготы в координаты CPR и обратно. Обычно это называют «CPR кодирование» и «CPR декодирование».

Земной глобус в CPR системе координат поделен на зоны. «Зоны широты» начинаются от экватора и идут к обоим полюсам. «Зоны долготы» стартуют от нулевого меридиана (Гринвич) и следуют вокруг глобуса на Восток. Размер каждой зона широты составляют ~ 360 морских миль (м. миль) в высоту, в направлении Север-Юг. Зона долготы имеет ~ 360 м. миль в ширину, если измерять в направлении Восток-Запад. То есть можно сказать, что они квадратные, с оговоркой на кривизну шарообразной Земли. Поэтому, для сохранения у зон долготы примерно постоянной ширины их количество уменьшается по мере продвижения от экватора к полюсам.

Сами зоны широты и долготы делятся на два типа, слегка различающихся размерами. Один тип зон называется «четные», а другой – «нечетные». рисунок 1.7а и рисунок 1.7б показывают зоны широты и долготы для всей Земли. Красные линии представляют границы четных зон, синие линии - нечетных.

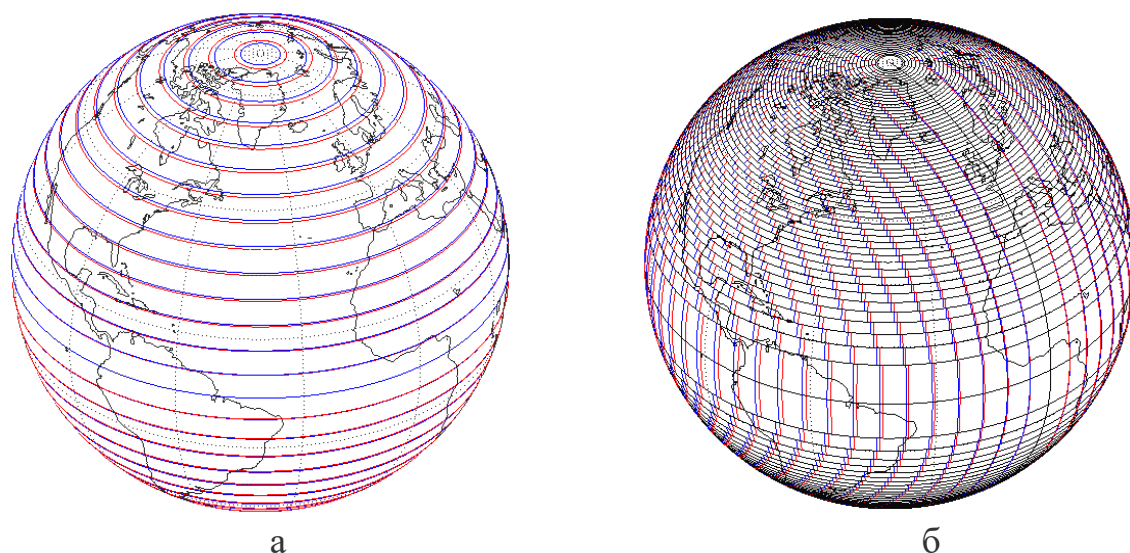


Рисунок 1.7 – Границы зон широты и долготы представленные в случае а и б

Каждая зона характеризуется номером ее индекса. Это можно сравнить с рулеткой, которая размечена четными и нечетными зонами широты, как показано на рисунке 1.8а. Рулетка имеет $4NZ = 60$ четных зон и $4NZ-1 = 59$ нечетных зон. Они пронумерованы, начиная с зоны с индексом 0. Границы зон таковы, что окажутся выровнены в начале и в конце этой рулетки, но больше - нигде. Сама рулетка начинается с индексной зоны 0 и идет на Север от экватора, то есть, как бы обернута вокруг Земного шара. Она проходит через Северный и Южный полюсы и возвращается к экватору, как показано на рисунке 1.8б.

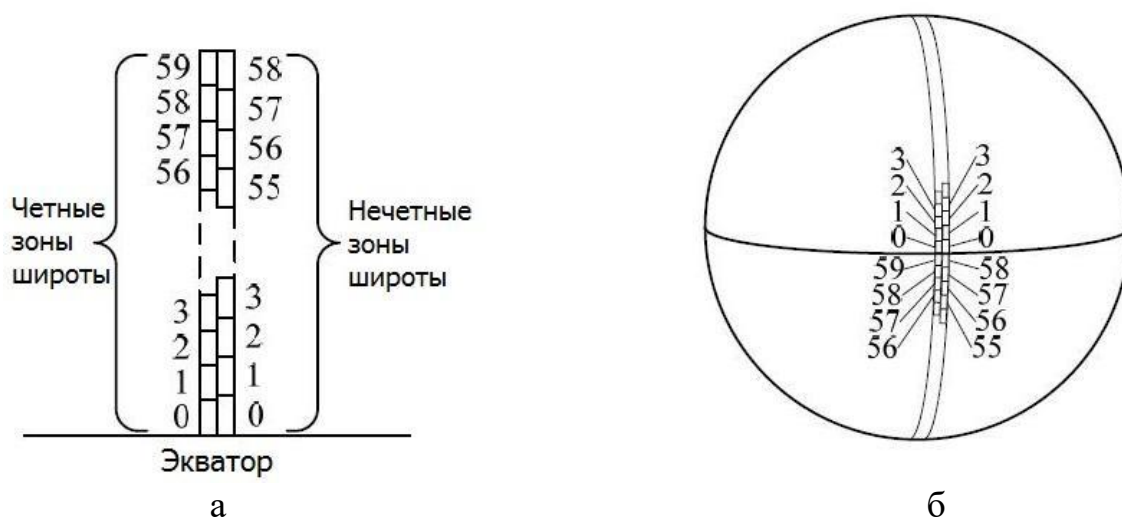


Рисунок 1.8 – Индексация зон широты и долготы в CPR
представленные в случае а и б

Индексация у зон долготы такая же, как и у широтных. Отличие только в том, что зона долготы с индексом 0 находится к Востоку от нулевого меридиана, а номера зон увеличиваются в восточном направлении. Выше отмечалось, что количество зон долготы варьируется в зависимости от их широты и это отражено в количестве зональных индексов. В свою очередь, зоны широты и зоны долготы разделены на ячейки. Каждая зона содержит $2N_b$ ячеек одинакового размера. Значение степени N_b меняется в зависимости от типа кодирования позиции. Ячейки нумеруются, начиная с 0-й на южной границе зон широты и западной границе зон долготы.

Таким образом, в итоге каждая точка на Земном шаре определяется в CPR системе координат индексом широтной зоны, номером широтной ячейки, индексом зоны долготы, номером ячейки долготы и CPR форматом (четной или нечетной зоны)[6].

2 ПРИЕМ СИГНАЛОВ ADS-B

Главными элементами наземного пункта, выполняющего прием и обработку сигналов ADS-B, являются антенная система, радиоприемные устройства, компьютер с программным обеспечением и устройства отображения данных. Этот раздел работы посвящен анализу возможности применения различных вариантов программно-определяемого радио для приема реальных сигналов ADS-B.

2.1 Прием данных ADS-B с использованием программно-определяемого радио (SDR)

Программно-определяемая радиосистема (англ. Software-defined radio, SDR) — радиопередатчик и/или радиоприемник, использующий технологию, позволяющую с помощью программного обеспечения устанавливать или изменять рабочие радиочастотные параметры, включая, в частности, диапазон частот, тип модуляции или выходную мощность, за исключением изменения рабочих параметров, используемых в ходе обычной предварительно определенной работы с предварительными установками радиоустройства, согласно той или иной спецификации или системы.

Программно-зависимые приёмники SDR достаточно несложны и малогабаритны. Размером от спичечного коробка до пачки сигарет. Но, при всей своей простоте, с компьютером и соответствующей программой, подобный приёмник превращается в достаточно серьёзное приёмное устройство, и вполне может использоваться как по прямому назначению, так и служить в качестве анализатора спектра. Для работы с такими приемниками создано несколько программ (например, SDRsharp, Rocky, SDRradio, KGKSDR) – рисунок 2.1[7].

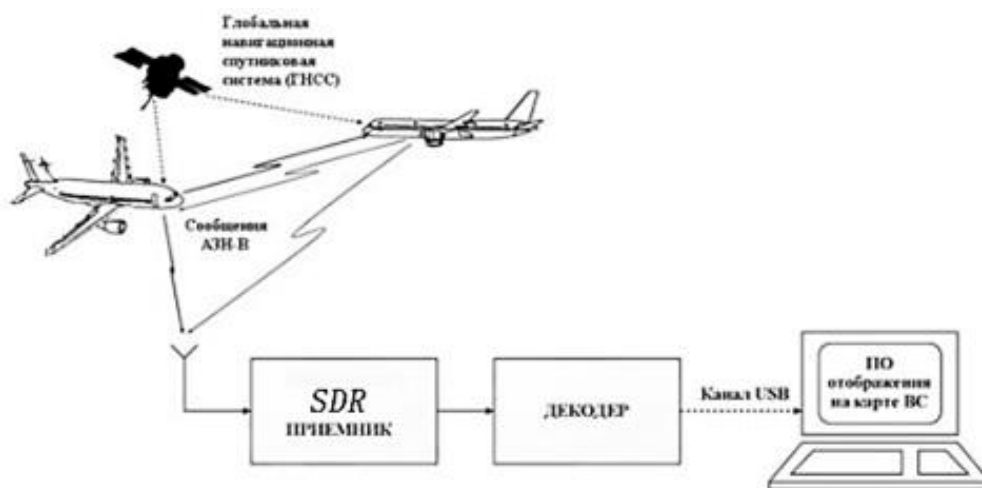


Рисунок 2.1 – SDR-система для приема сигналов ADS-B

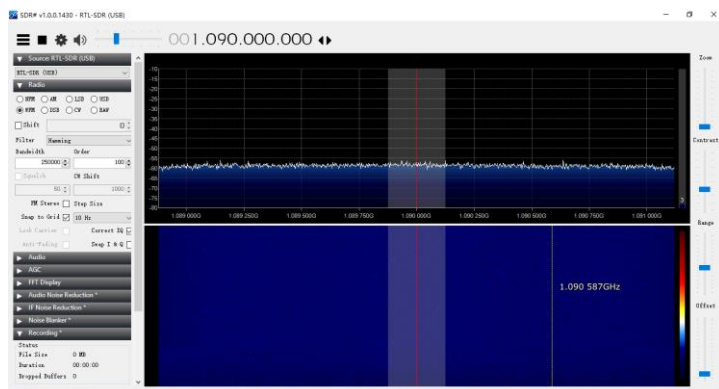


Рисунок 2.2 – Комплект аппаратуры и ПО для приема и обработки данных ADS-B (RTL-SDR, ПК и SDRSharp)

2.2 SDR приемники RTL-SDR и AIRSPY, структура и технические характеристики приемников.

Сегодня на рынке SDR оборудования достаточно широкий выбор радиоприемников, предназначенных как для профессиональной работы, и стоящих несколько тысяч долларов, так и для любительского использования, ценой от нескольких десятков до 200-400 долларов. Мы в своей работе будем использовать радиоприемные устройства двух нижних классов - наиболее простые и дешевые устройства RTL-SDR и устройства следующего по уровню класса, обладающие более высокими техническими характеристиками – приемники AIRSPY.

2.2.1 SDR приемники RTL-SDR

Внешний вид комплекта приемника RTL-SDR приведен на рисунке 2.3.



Рисунок 2.3 – Приемник RTL-SDR с антенной

Приемник предназначен для приема и обработки сигналов цифрового телевидения DVB-T, однако микросхема RTL2832U обладает недокументированными возможностями, которые позволяют переводить её в режим, в котором она передает через высокоскоростной USB интерфейс необработанный восьмиразрядный квадратурный сигнал с частотой дискретизации, достигающей 2.8 MSPS. В связи с этим было разработано программное обеспечение RTL-SDR, которое позволяет превратить тюнер на базе чипа RTL2832U в полноценный SDR приёмник. Контроллер RTL2832U был специально разработан для реализации концепции SDR. RTL2832U является высокопроизводительным DVB-T COFDM демодулятором, который поддерживает интерфейс USB 2.0, ширину полосы частот в 6.7 и 8 МГц. RTL2832U поддерживает тюнеры в том числе, с нулевой промежуточной частотой и промежуточными частотами 36.125 МГц и 4.57 МГц, и включает в себя FM / DAB / DAB + радио поддержку. Встроенный АЦП микросхемы RTL2832U обладает высокой стабильностью и низкое напряжение питания 3.5-5В. Аналоговый тюнер Elonic E4000 способен приматы частоты в диапазоне от 52 до 2200 МГц. Структурная схема приемной части RTL-SDR, выполненной на интегральной микросхеме E4000 приведена на рисунке 2.4.

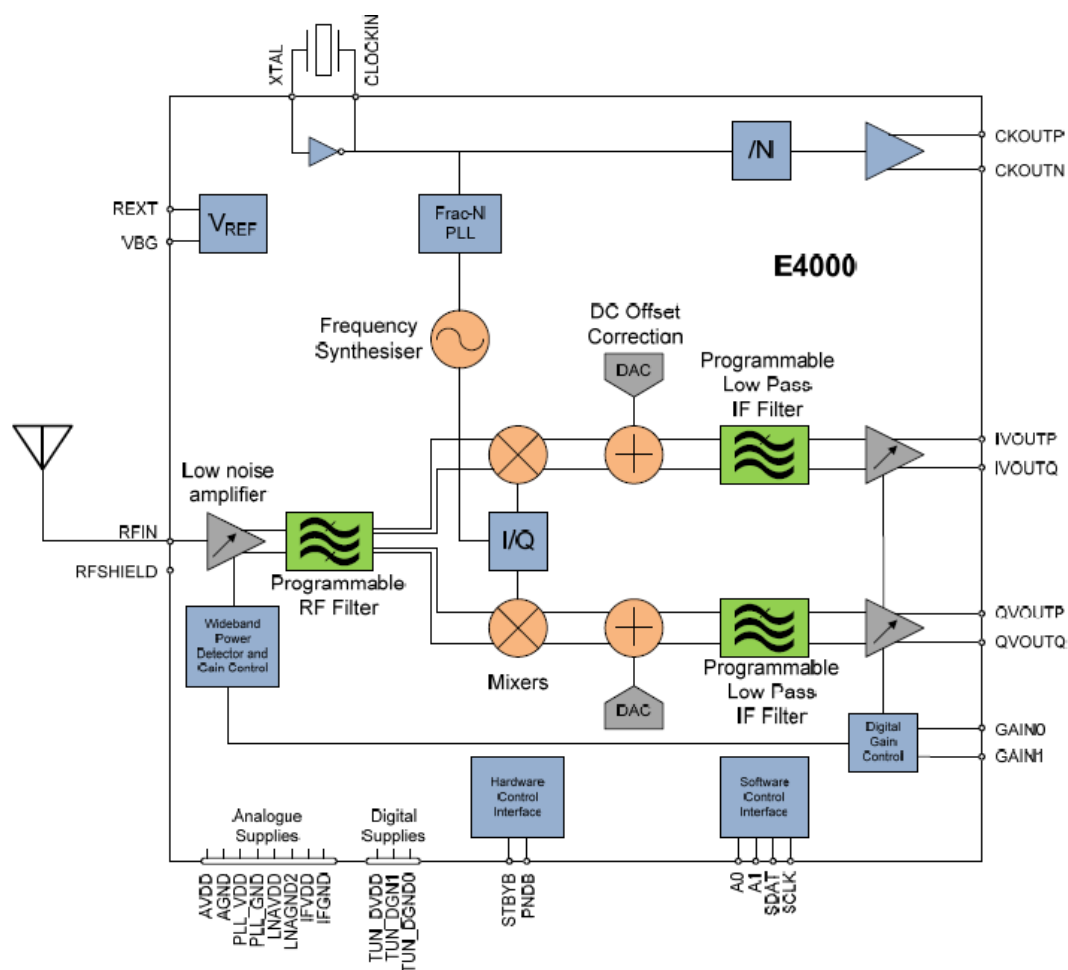


Рисунок 2.4 – Структурная схема E4000

Функциональная схема входной радиочастотной части приемника приведена на рисунке 2.5.

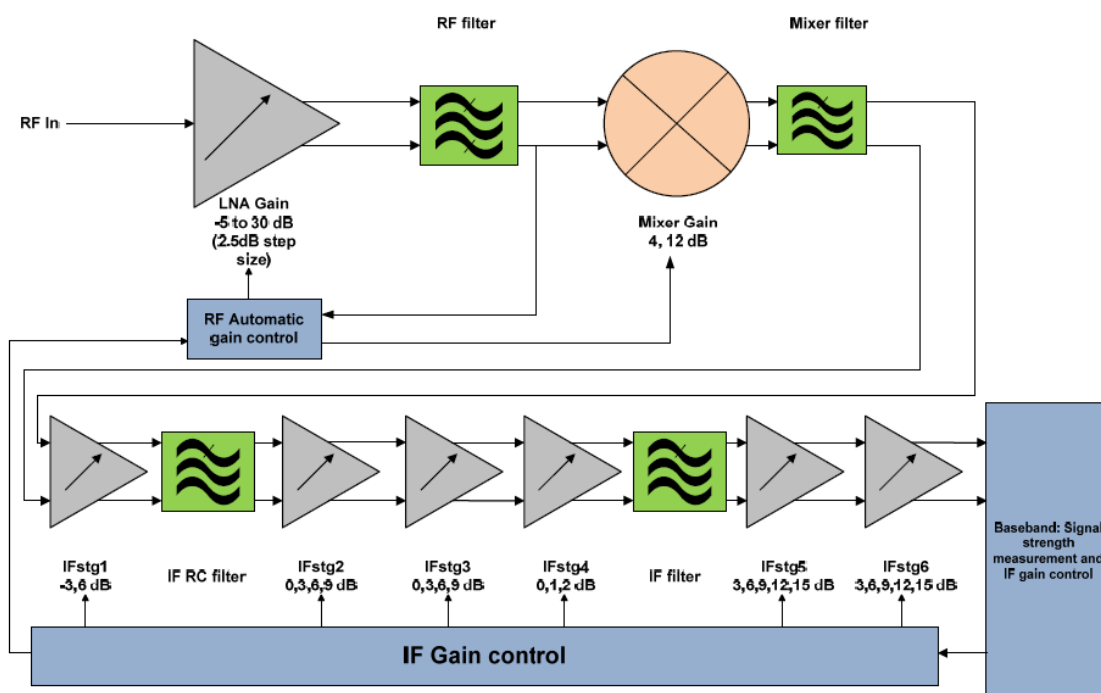


Рисунок 2.5 – Функциональная схема усилителя радиоприемника

Первый элемент - программируемый малошумящий усилитель. Усилитель обеспечивает усиление до 30 дБ при коэффициенте шума примерно 3,5дБ. Для высокочувствительных приемников это конечно довольно большой уровень шума, но с учетом низкой цены и интегрального исполнения – вполне допустимый. При необходимости приема слабых сигналов можно применять антенные усилители с низким уровнем шумов. Далее стоит программируемый полосовой фильтр с полосой в 200МГц. Он позволяет снизить уровень возможных перекрестных помех при приеме сигналов в широком динамическом диапазоне и диапазоне частот. Следующим стоит квадратурный смеситель, который обеспечивает усиление до 12дБ. Выходом смесителя являются два квадратурных аналоговых сигнала I и Q. Далее производится аналоговая НЧ фильтрация и усиление этих двух сигналов. Схемы обработки синфазного и квадратурного сигналов абсолютно одинаковы, поэтому на схеме рисунке 3.3 они изображены в виде одного канала. После фильтрации и усиления аналоговый сигнал с приемника передается дальше для дискретизации модулем RTL2832U[8].

Характеристики приемника RTL2832U E4000 (R820T)[9]

Диапазон частот: 24 - 1750МГц

Полоса обзора: от 250кГц до 3МГц

Чувствительность: 0.22 мкВ

Входное сопротивление приёмника: 75ом

Диапазонные фильтры: только внешние

Динамический диапазон: 50дБ (в режиме CW)

Задержка принимаемого сигнала: 340 мсек.

Интерфейс: USB 2.0

Операционная система: Windows, Linux, Android

2.2.2 SDR приемник AIRSPY

SDR-приемник AirSpy появился на рынке недавно. В отличие от обычного RTL-SDR он имеет АЦП производительностью 20МГц и разрядностью 12бит, и ширина захватываемой полосы радиочастот - 10 MHz, (против 2.4МГц и 8-бит у обычного RTL-SDR). Также установлен термокомпенсированный кварцевый генератор. Упрощенная функциональная схема приемника AIRSPY и его конструкция приведены на рисунках 2.6 и 2.7[10].

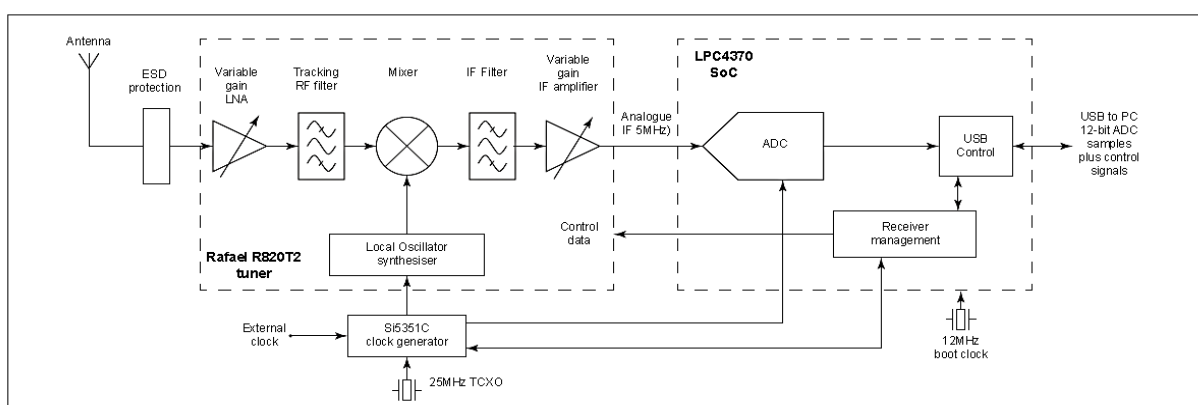


Рисунок 2.6 – Функциональная схема AIRSPY

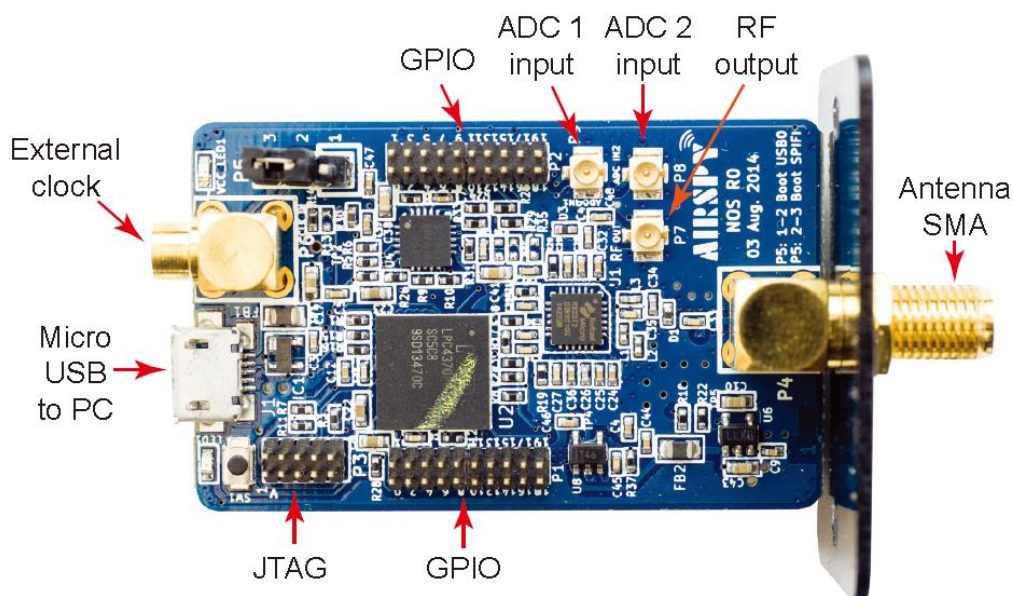


Рисунок 2.7 – Конструкция приемника AIRSPY

2.2.3 Сравнение SDR приёмников (RTL-SDR и AIRSPY)

Какие параметры отличают хороший SDR приёмник? В этой части мы рассмотрим приёмники RTL-SDR и Airspy.

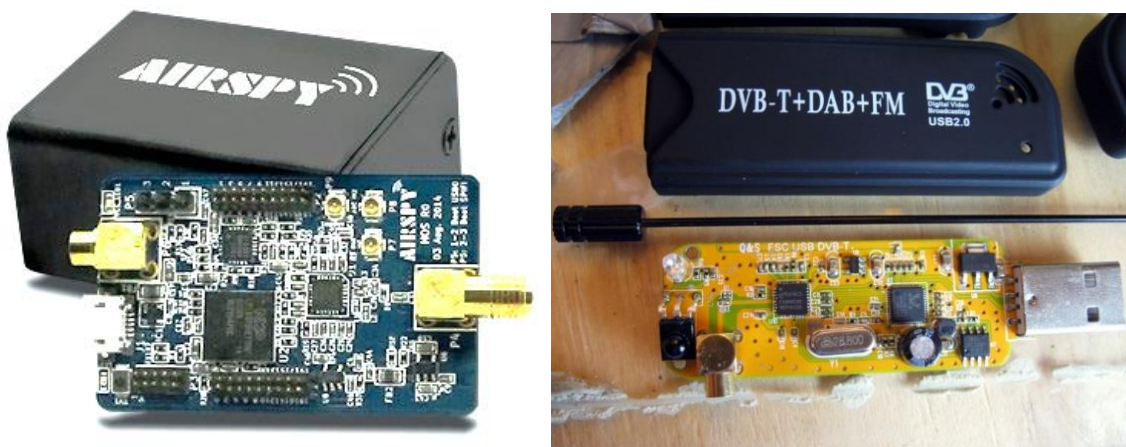


Рисунок 2.8 – Приёмники AIRSPY и RTL-SDR

Сначала мы определим некоторую терминологию, особенности и спецификации, которые требуются для качественного SDR приёмника.

- **Отношение сигнал/шум (SNR).** - При приёме сигнала основной параметр, который мы будем измерять – это отношение сигнала к шуму. Отношение сигнал/шум — это отношение пиковой мощности полезного сигнала к мощности шума.
- **Ширина полосы обзора (Bandwidth)** – Высокая ширина полосы обзора, означает больше сигналов на экране сразу, а также возможна дополнительная программная децимация сигнала (лучше SNR). Недостатком является то, что для высокой ширины полосы обзора – необходима большая мощность процессора.
- **Отсутствие фантомов в полосе обзора** – Полоса обзора на дисплеях SDR имеет тенденцию скатываться на края, а также показывать фантомные изображения других сигналов. Ширина полосы обзора без завалов на краях, и без фантомных сигналов является фактической полосой обзора и обычно меньше заявленной в спецификации.
- **Чувствительность** – Высокая чувствительность позволяет принимать слабые удалённые сигналы, и получать высокие значения SNR.
- **АЦП (ADC)** – Аналого-цифровой преобразователь. Основной компонент в SDR. АЦП преобразует аналоговый сигнал в цифровые биты. Высокая разрядность АЦП обеспечивает высокий динамический диапазон, и большое отношение сигнал/шум.
- **Перегрузка** – Перегрузка возникает при очень сильном входном сигнале, при этом происходит насыщение АЦП в итоге невозможно различить слабые

сигналы. На экране это выглядит как резкое снижение чувствительности и появление фантомных сигналов.

- Динамический диапазон – Непосредственно зависит от разрядности АЦП, но также зависит и от обработки программным обеспечением DSP. Высокий динамический диапазон позволяет SDR слышать слабые сигналы, при наличии мощных соседних сигналов. При низком динамическом диапазоне мощный входной сигнал насыщает АЦП, и приём слабых сигналов становится невозможным. Динамический диапазон может быть увеличен путём применения входных диапазонных фильтров.
- Фантомы – Плохие SDR приёмники, вероятнее всего перегрузятся и покажут изображения мощных сигналов там, где их не должно быть. Это может быть исправлено с помощью фильтрации, либо с помощью увеличения динамического диапазона путём повышения разрядности АЦП.
- Помехозащищённость – Хороший SDR приёмник ничего не должен принимать без подключённой антенны. Если он принимает сигналы без антенны, то сигналы помехи могут проходить непосредственно через монтажную плату, лишая возможности фильтровать их. Хороший SDR также хорошо справится с помехами, приходящими через USB.
- Предварительная селекция сигнала – высокоэффективный SDR приёмник, должен иметь на входе диапазонные фильтры для предварительной селекции сигнала, которые переключаются в зависимости от принимаемой частоты.
- Центральный DC Шип - Хорошие СДР приёмники должны иметь хорошую I/Q балансировку, чтобы предотвратить появление шипа в центре.
- Фазовый шум – Уровень фазового шума, определяется качеством используемых кварцевых генераторов. Низкие фазовые шумы кварцевого генератора означают лучший SNR для узкополосных сигналов и меньшее взаимное влияние близких сигналов. Взаимное влияние состоит в том, что слабый сигнал теряется в шумах фазы соседнего мощного сигнала.
- Стабильность частоты – Нам нужно что бы частота не изменялась в зависимости от температуры. Для достижения этого, должны использоваться ТСХО или подобные стабильные кварцевые генераторы.
- Структура приёмного тракта - Общий дизайн системы. Какие потери вносят компоненты, например, коммутаторы, используемые в РЧ канале. По мере увеличения сложности конструкции обычно больше компонентов добавляют в РЧ канал, это может уменьшить производительность RX.
- Программное обеспечение – аппаратные средства являются только половиной SDR приёмника. Программное обеспечение может обеспечить определённое преимущество перед соперниками.

Таблица 2.1 – Сравнение SDR приёмников RTL-SDR и AIRSPY

SDR приёмники Характеристики	RTL-SDR	Airspy
Цена (UAH/USD)	300 грн / \$12 USD	6000 грн / \$ 249 USD
Размеры	8.7x2.7x1.2 См	5.3 x 2.5 x 3.9 см
Вес	19 г	65 g
USB разъём	Micro USB	Micro USB
Антенный разъём	разъём MCX	SMA female
Корпус	Пластик	Алюминий
Частотный диапазон (MHz)	24 - 1750МГц	24 – 1800 МГц(0 – 1800 с применением Spyverter)
Разрядность АЦП	12бит	8бит
Ширина полосы обзора (MHz)	от 250кГц до 3МГц	10 (9MHz рабочий)
Динамический диапазон (dB)	50дБ (в режиме CW)	80дБ
Стабильность частоты (PPM)	ОСХО	0.5 PPM низкий фазовый шум TCXO
АЦП, RF чип	RTL2832U E4000(R820T)	LPC4370 ARM, R820T2
Задержка принимаемого сигнала	---	340мсек.
Интерфейс	USB 2.0	USB 2.0
Операционная система:	Windows, Linux, Android	Windows, Linux, Android, Mac и т.д

Из таблицы 1 видно, приёмник AIRSPY явно выигрывает практически по всем параметрам кроме цены. В следующем разделе работы мы проведем эксперименты по приему и декодированию реальных сигналов ADS-B, которые помогут нам подтвердить или опровергнуть заключение о преимуществах SDR-приемника AIRSPY перед значительно более дешевым RTL-SDR приемником, имеющим более узкий динамический диапазон, меньшую частоту дискретизации и разрядность АЦП[11].

3 ЭКСПЕРИМЕНТ ПО ПРИЕМУ И ОБРАБОТКЕ СИГНАЛОВ ADS-B

Основными элементами наземного пункта приема и обработки сигналов ADS-B, являются антенная система, радиоприемные устройства, компьютер с программным обеспечением и устройства отображения данных. При приеме сигнала, после того, как антенна и приемник подключены, необходимо выполнить настройку программного обеспечения для приема сигнала ADS-B (SDRsharp).

3.1 Прием сигнала с использованием SDR-приемников RTL-SDR и AIRSPY

Для приема и записи сигналов ADS-B нами будет использована программа SDRsharp, поэтому её следует настроить на работу с ADS-B сигналом.

После запуска программы SDRsharp мы получим окно примерно со следующим содержанием:

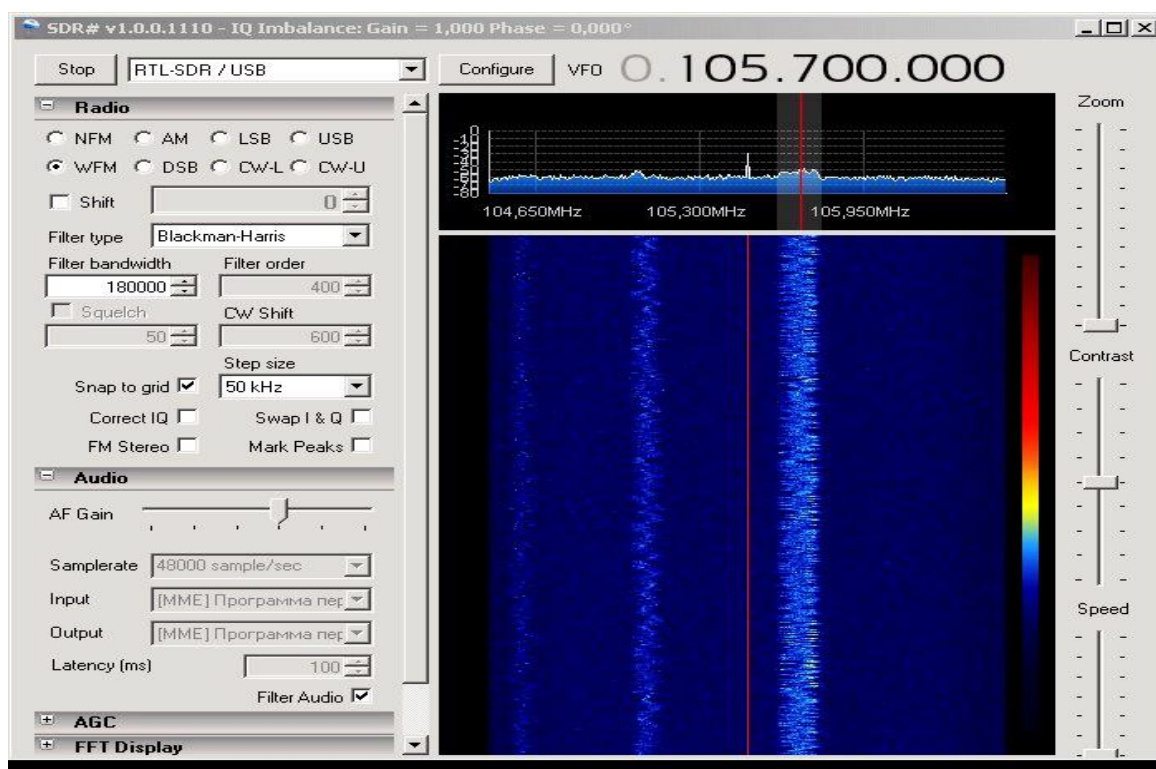


Рисунок 3.1 – Диалоговое окно программы SDRsharp

В нем необходимо включить опцию “Correct IQ”, это уберет лишнюю линию в середине спектра, выбрать приемник RTL-SDR/USB, а также установить модуляцию вида “AM”. Далее мы устанавливаем нужную нам частоту 1090 МГц, заходим в конфигурацию “Configure”, в появившемся окне нужно включить опцию “Tuner AGC”.

Настройка программы SDRSharp для работы с приемником Airspy выполняется аналогичным образом, но нужно учесть, что он может работать в трёх режимах: чувствительность (sensitivity), линейность (linearity) или произвольный (free). Режим чувствительность, максимально использует МШУ,

но за счёт понижения линейности и, следовательно, может увеличить интермодуляцию. Режим Линейность, использует в меньшей степени коэффициент усиления МШУ, и в большей IF/Mixer который может уменьшить интермодуляцию, но за счёт уменьшения отношения Сигнал/Шум (SNR). Произвольный режим позволяет регулировать усиление для IF/Mixer/LNA, в произвольном порядке[12].

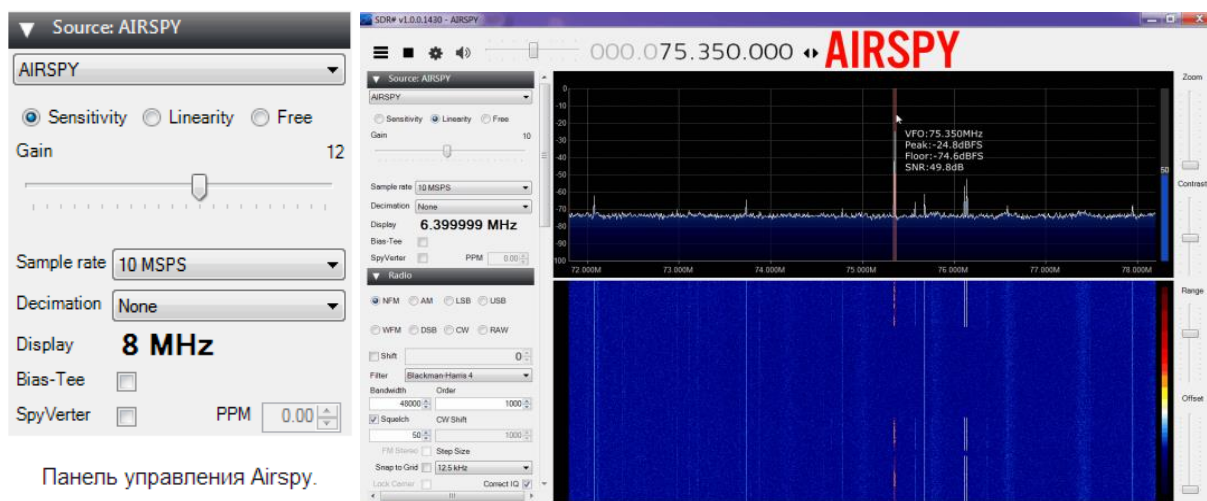


Рисунок 3.2 – Окна настроек SDRSharp при работе с AIRSPY

Программа управления приемниками RTL-SDR и AIRSPY позволяет только принять и записать принятые сигналы в файл, вся дальнейшая обработка выполняется в разработанном в ходе работы над проектом пакете МАТЛАБ-программ. Фрагмент программы, предназначенной для чтения принятого сигнала из файла и его отображения приведен ниже.

Для приемника RTL-SDR:

```
clear all;
close all;
N1=2800000;
[y,Fs] =
wavread('SDRSharp_20160506_11240
5Z_1090000kHz_IQ',[1 N1]);
Ta = 1;
dt = Ta/Fs;
Ns = 1;
Len = 2800000;
Nf = N1-1;
T = Len*dt;
```

Для приемника AIRSPY

```
clear all;
close all;
N1=10000000;
```

```
y = y';
y1 = y(1:Ns:Nf);
y2 = y(2:Ns:Nf);
Y = y1 + 1i*y2;
Y=abs(Y);
[b,a]=butter(2,500/1000,'low');
Y=filtfilt(b,a,Y);
t=0:1/Fs:(length(Y)-1)/Fs);
plot(t,Y);
ylabel('AMP');xlabel('t,c')
title('принятый сигнал RTL-SDR');
```

```
[y,Fs] =
wavread('SDRSharp_20160506_11241
2Z_1090379kHz_IQ',[1 N1]);
```

```

Ta = 1;
dt = Ta/Fs;
Ns = 1;
Len = 10000000;
Nf = N1-1;
T = Len*dt;

y = y';
y1 = y(1,Ns:Nf);

y2 = y(2,Ns:Nf);
Y = y1 + 1i*y2;
Y = abs(Y);
[b,a] = butter(2,500/1000,'low');
Y = filtfilt(b,a,Y);
t = 0:1/Fs:(length(Y)-1)/Fs;
plot(t,Y);
ylabel('AMP'); xlabel('t,c')
title('принятый сигнал AIRSPY ');

```

На следующих рисунках 3.3 и 3.4 приведены примеры отрезков сигналов ADS-B, принятых с использованием SDR-приемников RTL-SDR (частота дискретизации 2.8 МГц), и AIRSPY (частота дискретизации 10 МГц):

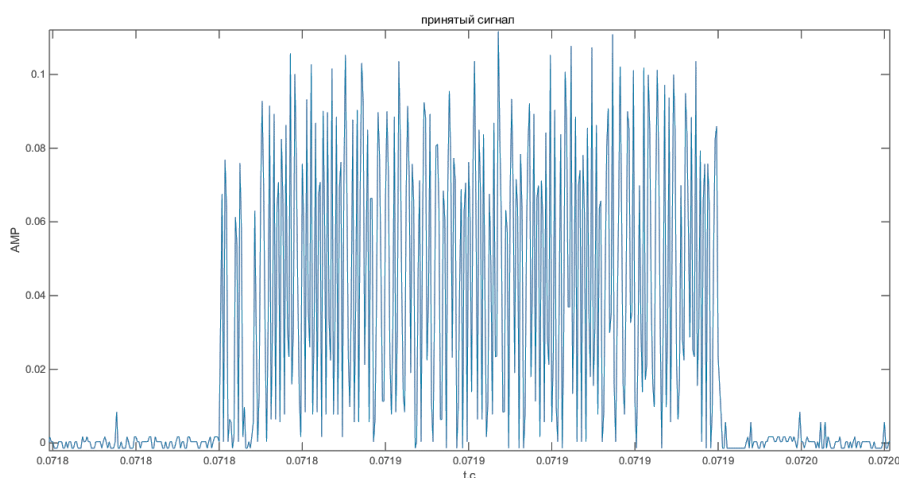


Рисунок 3.3 – Сквиттер DF17, принятый с использованием приемника RTL-SDR

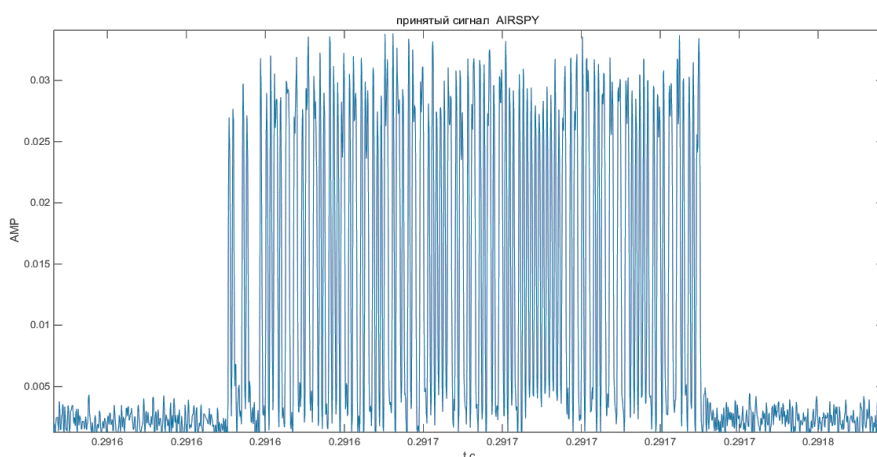


Рисунок 3.4 – Сквиттер DF17, принятый с использованием приемника AIRSPY

3.2 Обнаружение пакетов данных (сквиттеров DF11 и DF17)

Первой операцией, которая должна выполняться над принятым сигналом, является обнаружение и идентификация сквиттеров DF17, содержащих

информацию о координатах и векторе скорости ЛА, а также его ИКАО адресе и номере рейса.

3.2.1 Детектирование сквиттеров ADS-B на фоне помех

Обнаружение сквиттеров - это определение – присутствуют ли в принятом сигнале посылки DF11 и DF17, или их нет. Поскольку в решаемой нами задаче приема и декодирования сигналов ADS-B эта задача присутствует в явном виде, рассмотрим ее решение.

Задачи обнаружения сигналов должны решаться методами статистической теории решений. Эти методы основаны на статистическом анализе принимаемых приемником колебаний, полученных на определенном интервале наблюдения. В результате анализа принимается решение о наличии или отсутствии нужного сигнала от цели в составе этого колебания.

Рассмотрим, как ставится и решается задача обнаружения. Поскольку обнаружение сигнала производится на фоне шума, то это статистическая задача, и решается она вероятностными методами. При обнаружении сигнала можно определить две группы событий. В первую группу входят два события, состоящие в наличии или отсутствии интересующего нас сигнала в принятом колебании: «сигнал есть» (событие $A1$) и «сигнала нет» (событие $A0$). Каждое из этих событий имеет свою вероятность: $P(A1)$ и $P(A0)$.

Во вторую группу входят два других события, которые отражают ситуацию на выходе системы обработки после анализа полученного колебания и принятия решения: «принято решение, что сигнал есть» (событие $A'1$) и «принято решение, что сигнала нет» (событие $A'0$). Вероятности появления этих событий: $P(A'1)$ и $P(A'0)$.

Совершенно понятно, что события в этих двух группах зависимы. То есть, к примеру, решение "сигнал есть", принимается тогда, когда в принимаемом колебании наблюдаются какие-то признаки этого сигнала, а не "просто так захотелось". В результате возникнет один из четырех вариантов одновременного наступления двух зависимых событий. Два из этих вариантов дадут безошибочное решение: $A1$ и $A'1$ – правильное обнаружение сигнала (сигнал есть и мы приняли решение, что он есть) и $A0$ и $A'0$ – правильное необнаружение сигнала (сигнала нет, и мы приняли решение, что его нет). А два варианта дадут ошибочные решения: $A1$ и $A'0$ – пропуск цели ("сигнал есть", но мы сказали, что его нет) и $A0$ и $A'1$ – ложная тревога ("сигнала нет", но мы приняли решение, что он есть). Ошибочные варианты появятся из-за наличия помех. Если бы помехи отсутствовали, мы бы всегда принимали правильные решения.

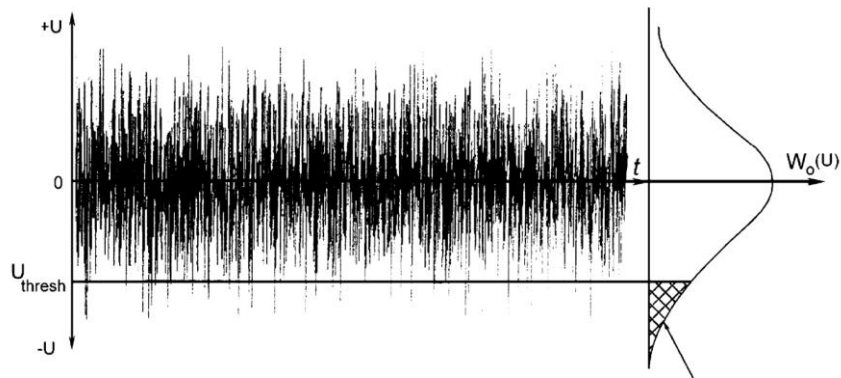


Рисунок 3.5 – Принятое колебание при отсутствии в нем сигнала

Известно, что вероятность одновременного наступления двух совместимых и зависимых событий $P(A_n + A'_k)$ определяется по правилу умножения вероятностей. Она равна произведению вероятности одного из этих событий $P(A_n)$ на условную вероятность появления второго, вычисленную в предположении, что первое событие совершилось $P(A'_k / A_n)$:

$$P(A_n + A'_k) = P(A_n) * P\left(\frac{A'_k}{A_n}\right) \quad (3.1)$$

Условная вероятность ложной тревоги (при условии, что “сигнала нет”), то есть вероятность того, что напряжение шума $u(t)$ превысит некоторое пороговое значение u_0 (рис.1) будет равна:

$$P(A_1 | A_0) = P[U(t) \geq U_0] = \int_{u_0}^{\infty} W_0(u) du \quad (3.2)$$

Тогда вероятность ложной тревоги:

$$P(A_0 + A_1) = P(A_0) * P(A_1 | A_0) = P(A_0) \int_{u_0}^{\infty} W_0(u) du \quad (3.3)$$

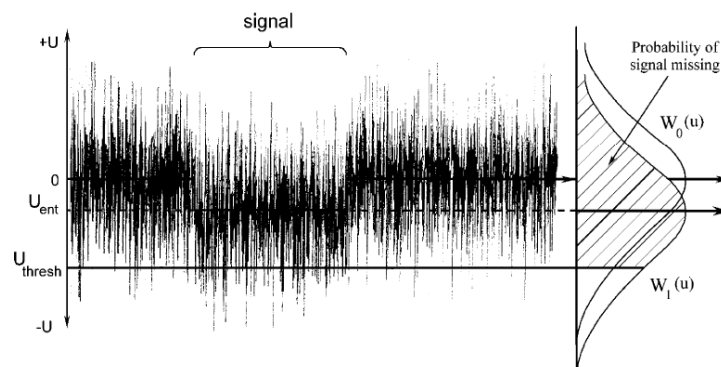


Рисунок 3.6 – Принятое колебание при наличии в нем сигнала

Условная вероятность пропуска сигнала (при условии, что “сигнал есть”), то есть вероятность того, что напряжение суммы сигнала и шума не превысит уровень u_0 (рис.3.6) будет равна:

$$P(A_0 | A_1) = P[U(t) \leq U_0] = \int_0^{u_0} W_1(u) du \quad (3.4)$$

Соответственно, вероятность пропуска сигнала:

$$P(A_1 + A_0) = P(A_1) * P(A_0 | A_1) = P(A_1) \int_0^{u_0} W_1(u) du \quad (3.5)$$

События $(A_0 + A_1')$ и $(A_1 + A_0')$ несовместимы. Тогда вероятность принятия одного из двух ошибочных решений в соответствии с правилом сложения вероятностей будет равна:

$$P[(A_0 + A_1') \text{ или } (A_1 + A_0')] = P(A_0 + A_1') + P(A_1 + A_0') = \\ P(A_0) \int_{u_0}^{\infty} W_0(u) du + P(A_1) \int_0^{u_0} W_1(u) du \quad (3.6)$$

выражение можно представить также в виде:

$$P[(A_0 + A_1') \text{ или } (A_1 + A_0')] = \\ 1 - [P(A_0) \int_0^{u_0} W_0(u) du + P(A_1) \int_{u_0}^{\infty} W_1(u) du] \quad (3.7)$$

Вероятность принятия правильного решения будет равна:

$$P[(A_0 + A_0') \text{ или } (A_1 + A_1')] = 1 - P[(A_0 + A_1') \text{ или } (A_1 + A_0')] = \\ P(A_0) \int_0^{u_0} W_0(u) du + P(A_1) \int_{u_0}^{\infty} W_1(u) du \quad (3.8)$$

Задача приемника – принимать решения с минимальной вероятностью ошибочных решений. Эта вероятность, при прочих равных условиях, зависит от порога u_0 . Если порог выбран очень высоким, то мы будем часто ”пропускать” сигнал, если слишком низким – будут часто возникать “ложные тревоги”.

Для отыскания оптимального уровня порога u_0 необходимо определить его значение, при котором вероятность правильного решения будет максимальна. Для этого вычислим производную:

$$\frac{dP[(A_0 + A_0') + (A_1 + A_1')]}{du_0} \quad (3.9)$$

и приравняем ее нулю. В результате получим: $P(A_0)W_0(u_0)=P(A_1) W_1(u_0)$ или

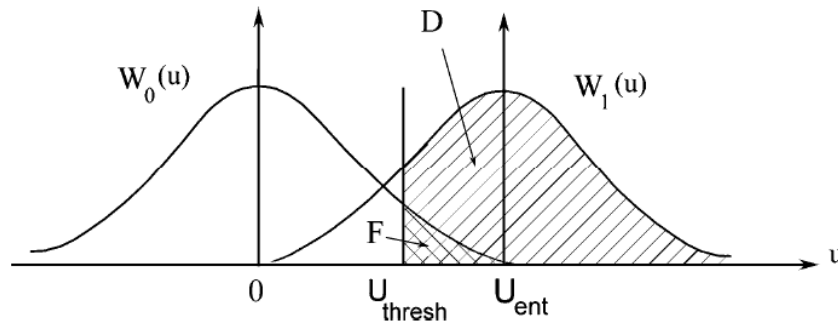
$$\frac{W_1(u_0)}{W_0(u_0)} = \frac{P(A_0)}{P(A_1)} \quad (3.10)$$


Рисунок 3.7 – Функции распределения шума $W_0(u)$ и суммы сигнала и шума $W_1(u)$.

Из рисунка видно, что уровень порога u_0 должен быть тем выше, чем больше амплитуда сигнала. При $P(A_0)=P(A_1)=0,5$ оптимальный уровень порога определяется точкой пересечения функций распределения $W_0(u)$ и $W_1(u)$. Для принятия решения о наличии сигнала необходимо, чтобы:

$$\frac{W_1(u)}{W_0(u)} \geq \frac{P(A_0)}{P(A_1)} \quad (3.11)$$

При обратном неравенстве принимается решение об отсутствии сигнала. Это неравенство справедливо для значения напряжения шума или смеси сигнала и шума в один момент времени и поэтому в него входят одномерные функции распределения W_0 и W_1 . Его можно распространить на случай, когда решение принимается по n отсчетам этого напряжения, полученным в интервале наблюдения.

$$\frac{W_1(u_1, u_2, u_3, \dots, u_n)}{W_0(u_1, u_2, u_3, \dots, u_n)} \geq \frac{P(A_0)}{P(A_1)} \quad (3.12)$$

Полученный статистический критерий является наиболее простым. Он называется критерием идеального наблюдателя. Его основной недостаток – на практике априорные вероятности наличия $P(A_1)$ или отсутствия $P(A_0)$ сигнала нам неизвестны, поэтому правильно установить порог принятия решения мы не можем. Приходится делать это эмпирически. Кроме того, критерий идеального наблюдателя не учитывает цену ошибочных решений. А она может быть разной – более нежелательными, например, могут быть “пропуски сигнала”.

Для устранения этого недостатка в уравнение для оценки вероятности ошибочного решения вводятся весовые коэффициенты B и C , характеризующие потери, связанные с “ложной тревогой” и “пропуском цели”: $P[(A_0+A'_1)]$ или

$(A1+A'0)] = B \cdot P(A0+A'1) + C \cdot P(A1+A'0)$. В этом случае для принятия решения о наличии цели необходимо выполнение неравенства:

$$\frac{W_1(u_1, u_2, u_3, \dots, u_n)}{W_0(u_1, u_2, u_3, \dots, u_n)} \geq \frac{B \cdot P(A_0)}{C \cdot P(A_1)} \quad (3.13)$$

Этот статистический критерий называется критерием минимального риска. Однако, и его использование затруднено на практике, потому что мы не знаем априорных вероятностей $P(A1)$ и $P(A0)$, а кроме того мы не знаем степени важности весовых коэффициентов B и C . Этот критерий, также как и критерий идеального наблюдателя относится к так называемым байесовским критериям.

Еще одним критерием является критерий максимального правдоподобия. Приведенная выше функция распределения n случайных значений колебания на выходе приемника $W(u_1, u_2, u_3, \dots, u_n)$ называется функцией правдоподобия. Для того, чтобы найти максимум функции правдоподобия, производную функции правдоподобия по искомому сигналу нужно приравнять нулю. Решение полученного уравнения позволяет найти максимально правдоподобную оценку значения сигнала.

$$\frac{W_1(u_1, u_2, u_3, \dots, u_n)}{W_0(u_1, u_2, u_3, \dots, u_n)} \geq 1 \quad (3.14)$$

Обычно же априорные вероятности, необходимые для принятия решения о наличии сигнала, на практике неизвестны. Поэтому наиболее широко используется еще один критерий, не зависящий от этих вероятностей. Это критерий Неймана-Пирсона, который обеспечивает максимальную вероятность правильного обнаружения $P(A1+A'1)$ (в дальнейшем обозначается D) при заданной вероятности ложной тревоги $P(A0+A'1)$ (в дальнейшем обозначается F). В соответствии с этим критерием величина порога u_0 в правой части отношения правдоподобия выбирается из заданной условной вероятности ложной тревоги:

$$P[U(t) \geq U_0] = \int_{u_0}^{\infty} W_0(u) du \quad (3.15)$$

Таким образом, независимо от выбранного критерия, решение задачи обнаружения сигнала сводится к вычислению отношения:

$$\Lambda = \frac{W_1(u_1, u_2, u_3, \dots, u_n)}{W_0(u_1, u_2, u_3, \dots, u_n)} \quad (3.16)$$

которое называется отношением правдоподобия. Решение о наличии сигнала принимается в том случае, когда это отношение превосходит некоторый фиксированный уровень u_0 , установленный заранее в зависимости от принятого

критерия. Конкретное же правило принятия решения – алгоритм обработки принимаемого колебания, зависит от характера сигнала и шума, на фоне которого принимается сигнал.

Если шум на входе приемника является «белым шумом», имеющим равномерную спектральную плотность мощности N_0 Вт/Гц в полосе частот приемника f и нормальную функцию распределения напряжения во времени вида:

$$W_0(u) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \quad (3.17)$$

Отсчеты напряжения шума, отстающие друг от друга на интервал дискретизации являются статистически независимыми. Тогда функция правдоподобия для N отсчетов напряжения шума является произведением N сомножителей:

$$W_0(u_1, u_2, u_3, \dots, u_n) = \prod_{i=0}^n W_0(u_i) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n u_i^2\right) \quad (3.18)$$

Функция распределения суммы сигнала и шума зависит от вида сигнала, и отличается от функции распределения шума только тем, что среднее значение этой суммы отлично от нуля и равно амплитуде сигнала s :

$$W_1(u) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(u-s)^2}{2\sigma^2}\right) \quad (3.19)$$

При полностью известном сигнале функция правдоподобия суммы сигнала и шума будет равна:

$$W_1(u_1, u_2, u_3, \dots, u_n) = \prod_{i=0}^n W_0(u_i) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (u_i - s_i)^2\right) \quad (3.20)$$

Тогда отношение правдоподобия для полностью известного сигнала будет равно:

$$\Lambda = \frac{W_1(u_1, u_2, u_3, \dots, u_n)}{W_0(u_1, u_2, u_3, \dots, u_n)} = \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n s_i^2\right) * \exp\left(-\frac{2}{\sigma^2} \sum_{i=1}^n u_i * s_i\right) \quad (3.21)$$

Или

$$\Lambda = \exp\left(-\frac{1}{N_0} \sum_{i=1}^n s_i^2 \Delta t\right) * \exp\left(\frac{2}{N_0} \sum_{i=1}^n u_i s_i \Delta t\right) \quad (3.22)$$

Если перейти от суммирования к интегрированию на отрезке времени от 0 до T , где располагаются n случайных значений напряжения на выходе приемника $u_1, u_2, u_3, \dots, u_n$:

$$\Lambda = \exp\left(-\frac{1}{N_0} \int_0^T s^2(t) dt\right) * \exp\left(\frac{2}{N_0} \int_0^T u(t) * s(t) dt\right) \quad (3.23)$$

Поскольку нас интересует только факт превышения порога, а не абсолютное значение отношения правдоподобия, к обеим сторонам неравенства можно применить любое однозначное преобразование, например – логарифм. Это позволит избавиться от экспоненты в этом выражении и упростить структурную схему оптимального приемника:

$$\ln \Lambda = \frac{2}{N_0} \int_0^T u(t) * s(t) dt - \frac{1}{N_0} \int_0^T s^2(t) dt \quad (3.24)$$

Для известного сигнала и заданной плотности мощности шума второе слагаемое является постоянной величиной, которая может быть учтена при выборе порога u_0 (или включена в его состав). И тогда для получения оптимального алгоритма обнаружения цели необходимо вычислить интеграл:

$$\frac{2}{N_0} \int_0^T u(t) * s(t) dt \quad (3.25)$$

и сравнить полученную величину с порогом [13].

К сожалению, в реальных условиях часть параметров сигнала – начальная фаза, амплитуда, временное положение сигнала, могут носить случайный характер. В частности, при решении задачи обнаружения сквиттеров, все эти параметры сигнала нам неизвестны. Нам неизвестна также форма сигнала, кроме его начального участка – преамбулы. Информационная же часть посылки – двоичная последовательность, закодированная Манчестерским кодом, нам неизвестна. Известна лишь длительность – 56 мкс для DF11 и 112 мкс для DF17.

3.2.2 Энергетический обнаружитель

При неизвестной (случайной) форме сигнала оптимальным является так называемый энергетический обнаружитель. Структурная схема энергетического обнаружителя изображена на рис. 3.8

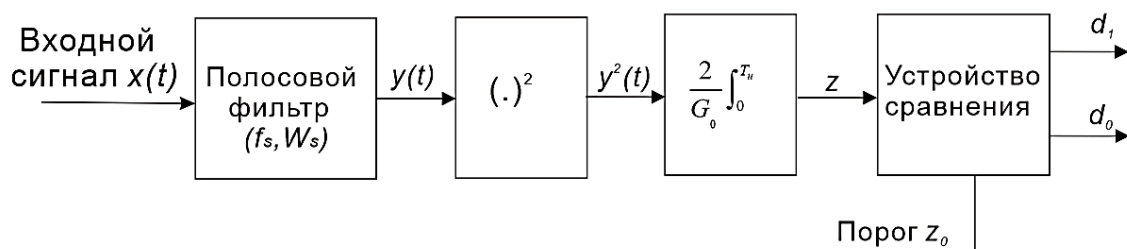


Рисунок 3.8 – Схема энергетического обнаружителя

Обнаружитель содержит: полосовой фильтр с полосой пропускания W_3 , равной полосе частот сигнала, квадратичный детектор $(\bullet)^2$, интегратор и устройство сравнения. Такой обнаружитель измеряет энергию принятой реализации в пределах конечного времени интегрирования $T_{\text{и}}$ и сравнивает выходной сигнал интегратора z с порогом Z_0 для принятия решения.

Алгоритм обнаружения имеет вид: принимается решение d_1 о наличии сигнала $s(t)$, если статистика $z \geq Z_0$, и решение d_0 об отсутствии сигнала, если $z < Z_0$, т.е.:

$$z = \frac{2}{G_0} \int_0^{T_{\text{и}}} y^2(t) dt \begin{cases} \geq Z_0 \Rightarrow d_1 & , \text{ сигнал есть} \\ < Z_0 \Rightarrow d_0 & , \text{ сигнала нет} \end{cases} \quad (3.26)$$

где $y(t)$ - выходной сигнал полосового фильтра.

Предлагаемый метод обнаружения сквиттеров отличается от используемого в ADS-B на практике метода – обнаружение по преамбуле. Он может оказаться более эффективным при низкой частоте дискретизации принимаемого колебания, сравнимой с полосой сигнала, когда на один двоичный символ сквиттера приходится 1-2 отсчета. В этом случае энергетическое отношение сигнал/шум на преамбуле, определяющее вероятность ошибки при ее обнаружении, может оказаться очень небольшим (на символы преамбулы будет приходиться всего 5-10 отсчетов). Соответственно, вероятность ошибки обнаружения может быть довольно большой. Если же сигнал обнаруживать по информационной части сквиттера, длительность которой составляет 56 (112 мкс), половина которой составляют единицы, и имеет гораздо большую энергию (примерно в 15 – 20 раз больше, чем энергия преамбулы), то вероятность ее правильного обнаружения может быть гораздо ближе к единице[14].

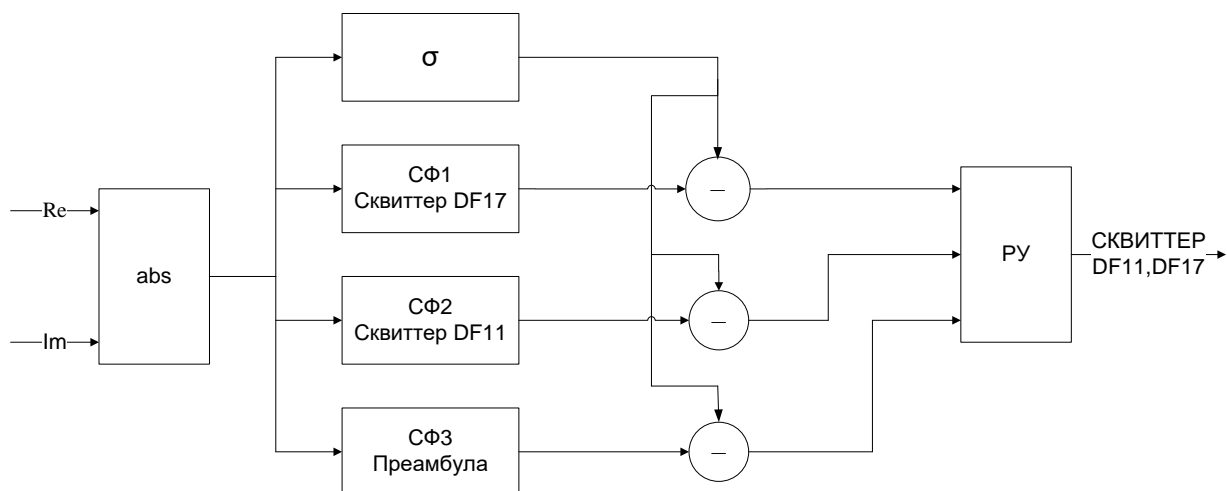


Рисунок 3.9 – Функциональная схема алгоритма энергетического обнаружения

Обнаружение происходит следующим образом: после прихода сигнала на радиоприемное устройство сигнал будет состоять из синфазной и квадратурной составляющих, эти составляющие нам необходимо объединить и взять по модулю. Далее нам необходимо вычислить дисперсию данного сигнала. В это же время проводится свертка сигнала с моделью сквиттера DF11, DF17 и преамбулой. После этого мы вычитаем вычисленную дисперсию из полученной свертки. На следующем этапе решающее устройство принимает решение о том, какой из возможных сигналов приходит на вход радиоприемного устройства.

С тем чтобы облегчить обработку сигналов приемника RTL-SDR, повышать частоты до 10 МГц с помощью способа интерполяции или передискретизации (в МАТЛАБ, с помощью функции *resample*). Результат работы обнаружителя показан на рисунках 3.10 и 3.11.

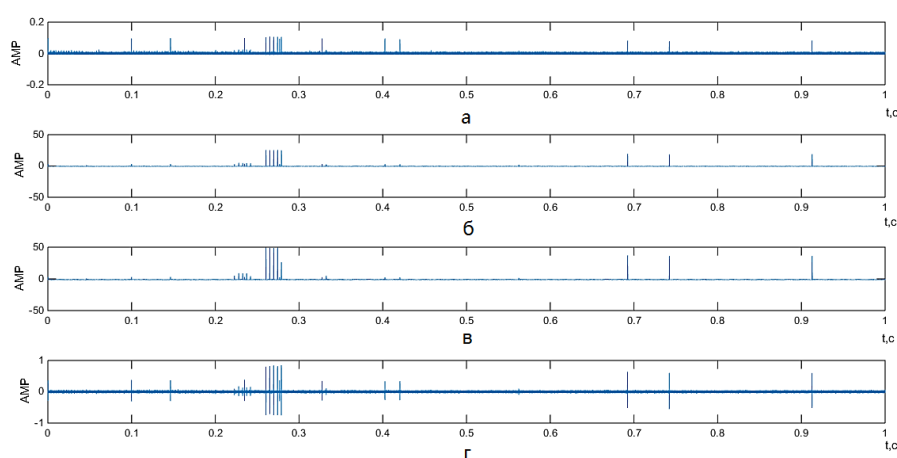


Рисунок 3.10 – Отклики согласованных фильтров на DF11 и DF17:
а – Сигнал - среднее и дисперсия; б – Отклик СФ на сквиттер DF11; в – Отклик СФ на сквиттер DF17; г – Отклик СФ на преамбулу

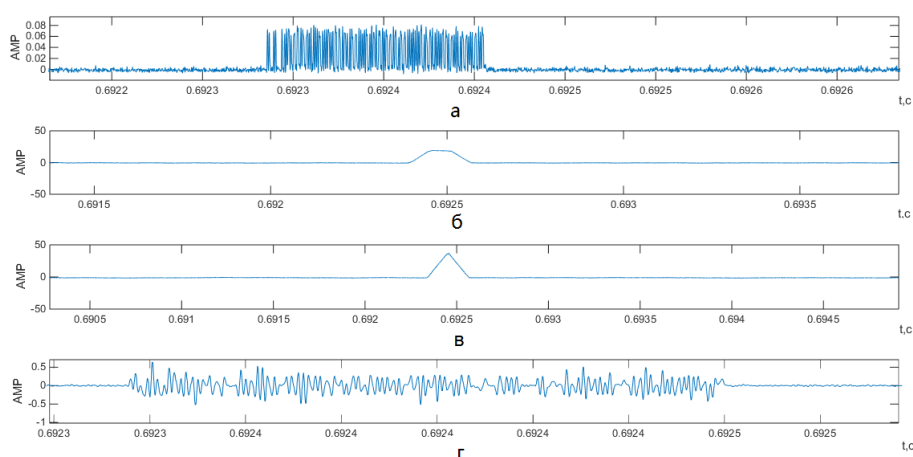


Рисунок 3.11 – То же, для одного сквиттера DF17:
а – Сигнал - среднее и дисперсия; б – Отклик СФ на сквиттер DF11; в – Отклик СФ на сквиттер DF17; г – Отклик СФ на преамбулу

Фрагмент МАТЛАБ-программы, выполняющей обнаружение сквиттеров DF11 и DF17 приведен ниже

```
t_d=0:1/Fs:((length(Y)-1)/Fs);
DF_11(560:1120)=1;
DF_11(1,560)=0;
DF_17(1:1120)=1;
Y_1=conv(Y,transpose(DF_11));
Y_2=conv(Y,transpose(DF_17));
fprintf('Выполняется свертка.... :
\n');
t_t1=0:1/Fs:((length(Y_1)-1)/Fs);
t_t2=0:1/Fs:((length(Y_2)-1)/Fs);

subplot(4,1,2);
plot(t_t1,Y_1);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на сквиттер
DF11');
subplot(4,1,3);

plot(t_t2,Y_2);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на сквиттер
DF17');
h_t=[1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1 1];
h_t=h_t-mean(h_t);

Z=conv(Y,transpose(h_t));
t_kor=0:1/Fs:((length(Z)-1)/Fs);
subplot(4,1,4);
plot(t_kor,Z);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на преамбулу')
```

3.3 Идентификация пакетов DF11 и DF17

После обнаружения всех сквиттеров типа DF11 и DF17 нужно выделить из них только посылки типа DF17, содержащие навигационные и информационные данные ВС. Более короткие посылки типа DF11 мы будем игнорировать. В работе предложен и разработан алгоритм идентификации расширенного сквиттера DF17. Идея алгоритма состоит в сравнении амплитуд откликов СФ настроенных на сигналы DF11 и DF17. При приеме DF17 отклик СФ, настроенного на DF17 будет всегда примерно в два раза (с учетом различия из-за действия шумов) превышать отклик СФ, настроенного на DF11. Блок схема алгоритма приведена на рисунке 3.12.

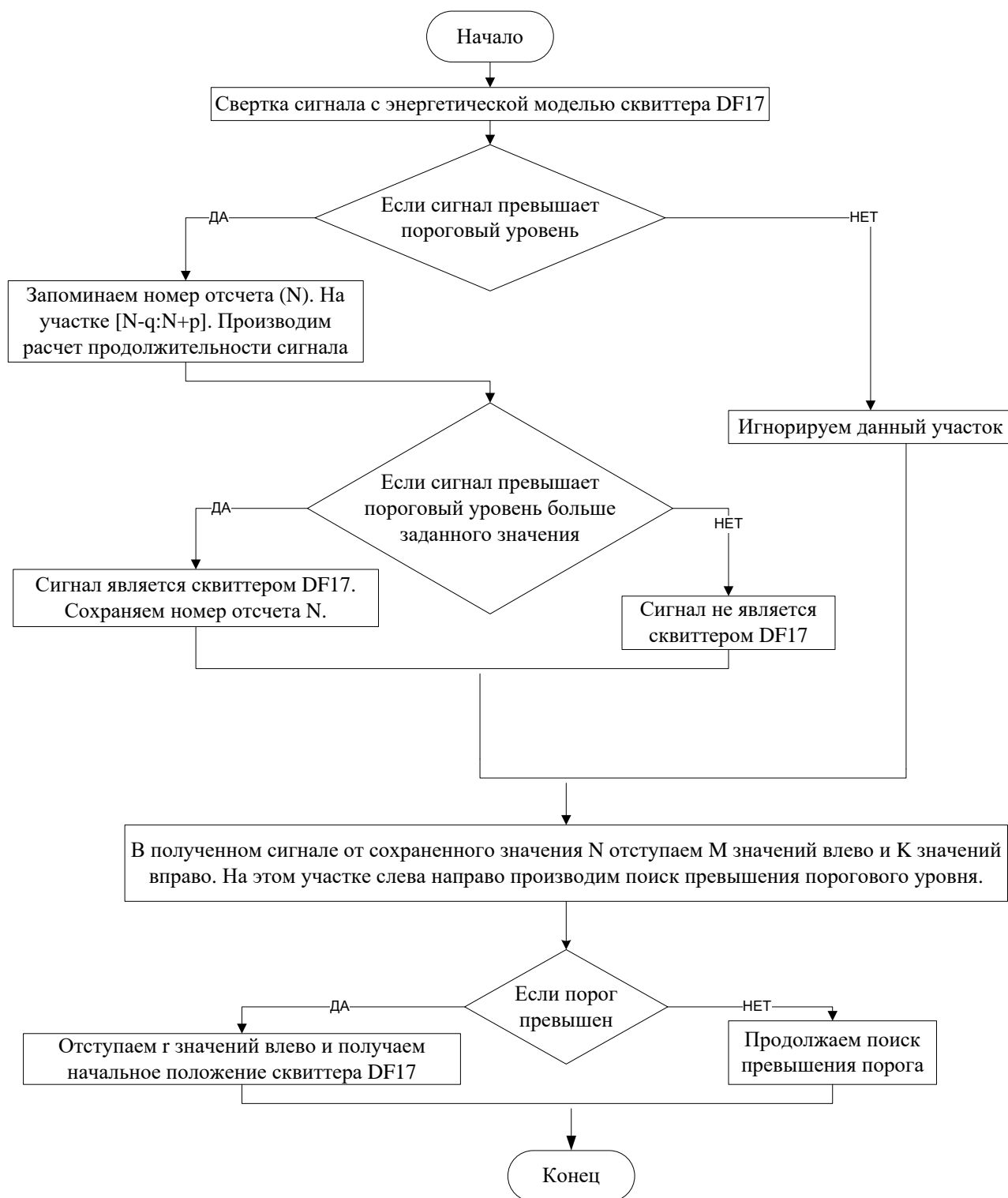


Рисунок 3.12 – Блок схема различения сквиттера DF17

Алгоритм работает следующим образом. На первом этапе выполняется фильтрация (свертка) принятого сигнала с ИПХ согласованных фильтров, настроенных на сквиттеры DF11 и DF17. В полученной свертке производится поиск максимального значения, превышающего пороговый уровень. Далее мы сравним максимальное значения отклика СФ на сквиттеры DF11 и DF17. Если

максимальное значение отклика СФ DF17 примерно в два раза превышает значение отклика СФ DF11, мы будем считать принятый сигнал сквиттером DF17. Если значение максимальное отклика DF17 почти равно значению DF11 то участок отбрасывается. Таким образом из всех посылок будет осуществлена селекция сквиттеров DF17. Кроме того, мы находим положение максимумов сигнала свертки для каждой из посылок. Это конец сквиттера DF17. Номер отсчета (N) сохраняется. Фрагмент МАТЛАБ-программы, выполняющих селекцию сквиттеров DF17 из всех посылок ADS-B приведен ниже

```
DF17=Y_2;
DF11=Y_1;
k=1;kk=1;
while k<length(Y)
    k=k+1;
    if DF11(k)>10
        DF11C=DF11((k-
0.0002*Fs):(k+0.0002*Fs));
        DF17C=DF17((k-
0.0002*Fs):(k+0.0002*Fs));
        s1=find(DF11C==max(DF11C));
        s2=find(DF17C==max(DF17C));
        detaDF=max(DF17C)-
1.5*max(DF11C);
        if mean(mean(detaDF)>0)
            display('сквиттер DF17 ');
            k=min(s2+k-0.0002*Fs);k;
            % display(k/Fs);
        else display('сквиттер DF11');
            k=min(s1+k-0.0002*Fs);
        end
        num(kk)=k;
        kk=kk+1;
        k=k+0.0002*Fs;
    end
end
display(num);
display(num/Fs);
```

Результат идентификации и определения временного положения сквиттеров DF17 приведен ниже:

```
Выполняется свертка.... :
сквиттер DF17
сквиттер DF17
сквиттер DF17
сквиттер DF17
сквиттер DF11
сквиттер DF17
сквиттер DF17
сквиттер DF17

num =

    2605310    2651502    2697715    2744048    2789664    6924557    7423648    9127829

ans =

    0.2605    0.2652    0.2698    0.2744    0.2790    0.6925    0.7424    0.9128
```

3.4 Обнаружение преамбулы. Оценка временного положения (начала) сквиттера

В посылке ADS-B сообщения имеется участок неизменной для всех типов сквиттеров формы – преамбула. С нее начинается посылка. Преамбула представляет собой последовательность нулей и единиц длительностью 8 мкс следующего вида: 1010000101000000. Преамбула позволяет точно определить

начало информационной части сквиттера для того, чтобы декодировать код Манчестера.

Обнаружение и определение временного положения преамбулы производится путем формирования корреляционной функции принятого сигнала с сигналом, имеющим форму преамбулы, и определении временного положения максимума этой корреляционной функции. Это и будет положение преамбулы. Результат этой операции для одного сквиттера DF17 приведен на рисунке 3.11, результат работы программы, определяющей положения всех посылок типа DF17 – на рисунке 3.13.

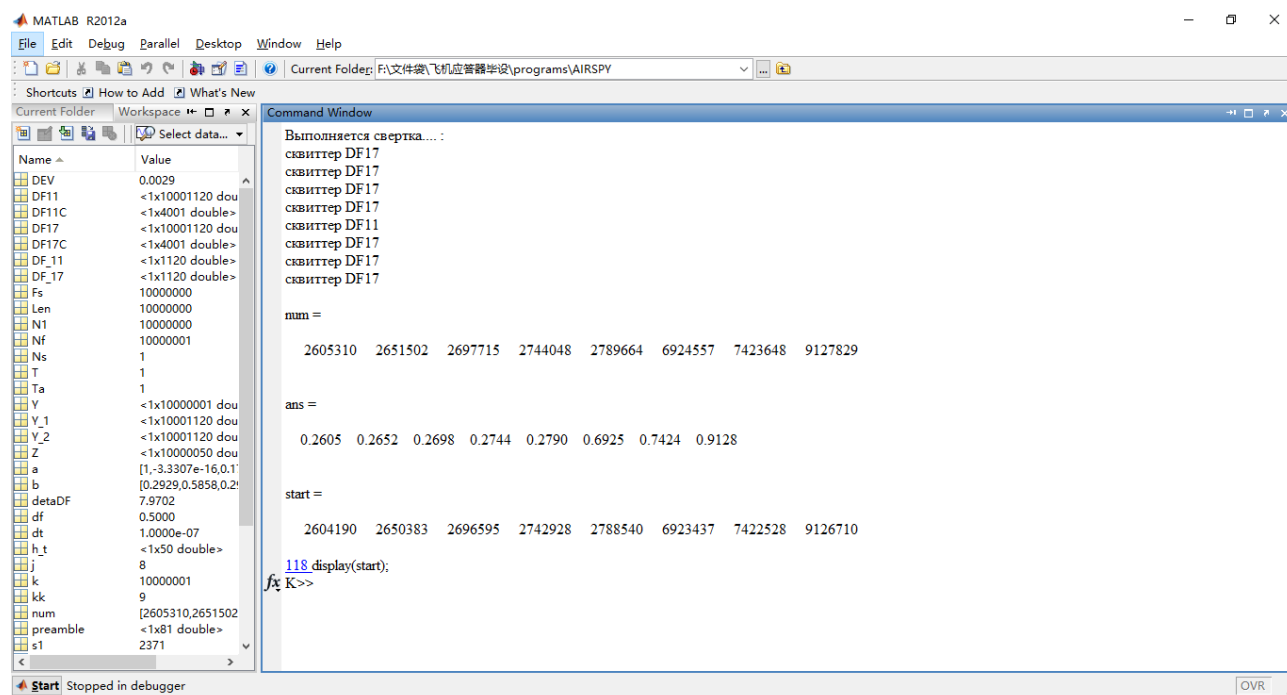


Рисунок 3.13 – Определение временного положения посылок DF17

Для данной записи сигнала ADS-B номера отсчетов, с которых начинается пакет DF17:

2604190 2650383 2696595 2742928 6923437 7422528 9126710

На следующих двух рисунках – рис. 3.14 и рис. 3.15 в качестве примера изображены обнаруженные и идентифицированные сквиттеры типа DF17 DF11.

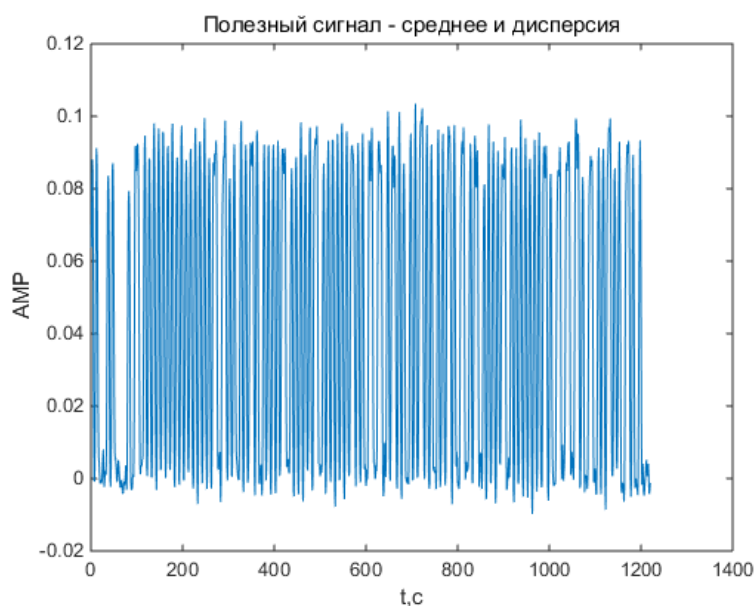


Рисунок 3.14 – Расширенный сквиттер DF17

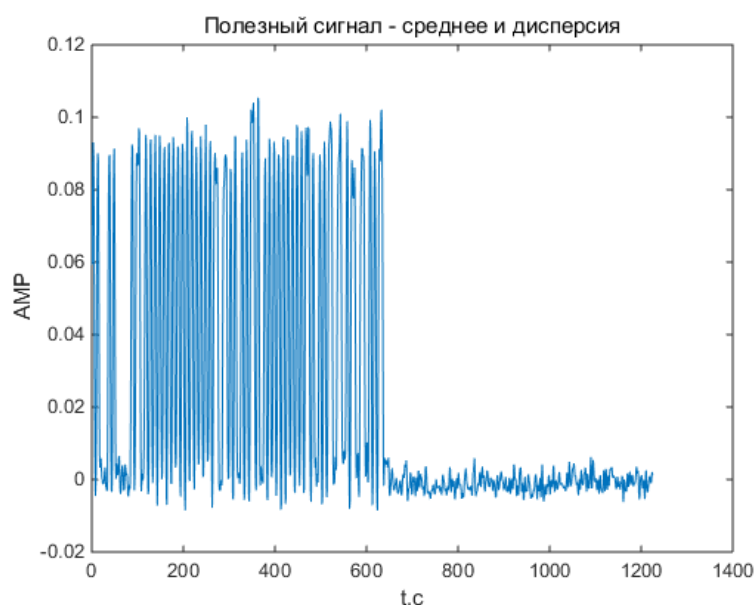


Рисунок 3.15 – Короткий сквиттер DF11

3.5 Декодирование двоичных последовательностей с ПИМ (кода Манчестера)

Кодирование с использованием кода Манчестера состоит в том, что биты двоичной последовательности 0 и 1 кодируются, как переход сигнала из нуля в единицу, или переход сигнала из единицы в ноль. Существует несколько распространенных методов декодирования кода Манчестера, в том числе – импульсный метод, и оптимальная процедура, которая состоит в том, чтобы определить, какой тип перехода (из “1” в “0” или из “0” в “1”) имеет место на интервале текущего бита. Она состоит в том, что начиная с первого бита данных осуществляется корреляция сигнала с двумя шаблонами вида $S1=[1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0]$ и $S0=[0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1]$, соответствующими двум типам переходов, и

решение принимается в пользу большего. Разновидностью оптимальной корреляционной процедуры является алгоритм по минимуму СКО. Рассмотрим реализацию этих трех методов декодирования.

3.5.1 Декодирование импульсным методом

Для того, чтобы декодирование было достоверным нужно, чтобы частота дискретизации сигнала и длительности битов двоичной последовательности (скорость передачи) соответствовали заданным. Поэтому первым действием процедуры декодирования является *проверка соответствия частоты дискретизации F_s и скорости передачи R мбит/с.*

1) От первой единицы (её начала) преамбулы до конца последовательности должно быть $5 \text{ мкс} \pm 0.2$ (50 ± 1 отчётов при $F_s = 10 \text{ МГц}$).

2) Найти положение последнего отсчёта преамбулы = 1^* . Прибавить к этому значению $(30 - 1)$ отчётов. Палагаем, что это начало данных ADS-B.

3) Проверяем эту точку на $= 1^*$, если да, то отступаем назад на один отсчет. Это точка $i_0 = 0$.

4) Начало собственно декодирования: проверяем на переход из “0” в “1”, $i \in 0 \dots 7$. Можно быть следующие варианты – рисунок 3.16:

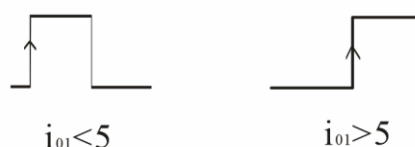


Рисунок 3.16 – Варианты перехода из “0” в “1”

Если значение точки перехода (в отсчетах) $i_{01} < 3$, то $i_0 = i_{01}$ и повторяем.

Если $i_{01} > 5$, то $i_0 = i_{01} - 5$ и повторяем.

То есть мы находим положение первого перехода из “0” в “1” и соответственно, уточняем положение начала пакета ADS-B DATA.

Если $i_{01 \text{ переход}} = 0 \pm 1$, то бит №1 = 1;

если $i_{01 \text{ переход}} = 5 \pm 1$, то бит №1 = 0.

Переходим к следующему биту, ищем переход из одного состояния в другое. Для этого сначала для этого используем состояние текущего (уже найденного) бита. Если его значение равно 1, то ищем переход из “0” в “1”; если значение текущего бита равно 0, то ищем переход из “1” в “0” – рисунок 3.17.

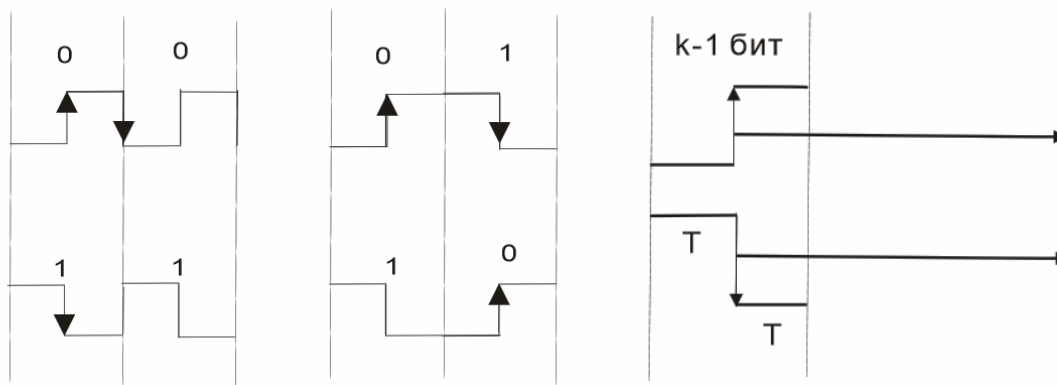


Рисунок 3.17 – Определение значения перехода при разных значениях текущего бита

Если $k-1$ -й бит = 0, то смотрим переход из “1” в “0”. Если $T=1$, то оценка значения бита = 0*; если $T=2$, то 1*;

Если $k-1$ бит = 1, то смотрим переход из “0” в “1”. Если $T=1$, то 1*; если $T=2$, то 0*.

Для анализа конца: Смотрим переход из “1” в “0” или из “0” в “1”, $T > 3$, END.

Полученный по данным пакета рисунок 3.14 результат выглядит так:

```
10001111010010111000010011101111010110000001001110110100101101100010
10110110100010110011011100100111111010010001
```

Фрагмент МАТЛАБ-программы, выполняющей декодирование кода Манчестера импульсным методом приведен ниже:

```
clear bit; bit0 = bit0 + 1;
for i = bit0 : 10
    delay = yy(i) - yy(i-1);
    if delay > 0.5
        if i - bit0 < 2
            bit(1) = 1;
        else bit(1) = 0;
        end
    end
end
%
T = 5; a = 2;
for i = bit0 : 10 : (ed-9)
    for j = i + 7 : i + 19
        delay = yy(j+1) - yy(j);
        if abs(delay) > 0.5
            if bit(a-1) == 1
                if ((j-i-5)/T) < 1.5
                    bit(a) = 1;
                else
                    bit(a) = 0;
                end
            elseif bit(a-1) == 0
                if ((j-i-5)/T) < 1.5
                    bit(a) = 0;
                else
                    bit(a) = 1;
                end
            end
            break;
        end
        break;
    end
    end
    a = a + 1;
```

```
end                                display(bit);
bit=num2str(bit);
```

3.5.2 Метод декодирования по минимуму СКО

Среднеквадратическое отклонение измеряется в единицах измерения самой случайной величины и используется при расчёте стандартной ошибки среднего арифметического, при построении доверительных интервалов, при статистической проверке гипотез, при измерении линейной взаимосвязи между случайными величинами. Определяется как квадратный корень из дисперсии случайной величины.

Среднеквадратическое отклонение определяется по формуле[15]:

$$\delta = \sqrt{\frac{1}{n}(x - \bar{x})^2}. \quad (3.27)$$

Процедура декодирования:

- 1) Находим начало и конец пакета данных (точно так же, как в импульсном методе);
- 2) Делим пакет на 112 частей.
- 3) Берем интервал данных, соответствующих первому биту (длительностью в 10 отсчетов при частоте дискретизации $F_s=10$ МГц) и определяем его похожесть на шаблоны:

$$Y_1 = \sum_{i=1}^{10} (u_i - x_{1i})^2, \quad (3.28)$$

$$Y_0 = \sum_{i=1}^{10} (u_i - x_{0i})^2.$$

Решение принимается по меньшему значению: текущий бит = 1, если $Y_1 < Y_0$, и текущий бит = 0, если $Y_0 < Y_1$.

Полученный по данным пакета рисунок 3.14 результат выглядит так:

```
10001111010010111000010011101111010110000001001110110100101101100010
10110110100010110011011100100111111010010001
```

Фрагмент МАТЛАБ-программы, выполняющей декодирование кода Манчестера методом минимума СКО приведен ниже:

```
clear bit; bit0=bit0+1;
S1=[1 1 1 1 1 0 0 0 0 0]; S0=[0 0 0 0 0 1 1 1 1 1];
for i=bit0:10:ed-9
    sum11=0; sum01=0; sum12=0; sum02=0; sum13=0; sum03=0;
    for j=i:i+9
        sum01=power((yy(j)-S0(j-i+1)),2)+sum01; sum02=power((yy(j-1)-S0(j-
i+1)),2)+sum02; sum03=power((yy(j+1)-S0(j-i+1)),2)+sum03;
```

```

    sum11=power((yy(j)-S1(j-i+1)),2)+sum11;sum12=power((yy(j-1)-S1(j-
i+1)),2)+sum12;sum13=power((yy(j+1)-S1(j-i+1)),2)+sum13;
end
aa=[sum01,sum02,sum03];bb=[sum11,sum12,sum13];
aaa=min(aa);bbb=min(bb);
if aaa<bbb
    bit((i+10-bit0)/10)=0;
else
    bit((i+10-bit0)/10)=1;
end
end
bit=num2str(bit);
display(bit)

```

3.5.3 Метод декодирования по максимуму корреляции

Метод аналогичен предыдущему, но решение принимается по большему значению корреляции между шаблонами и текущим фрагментом данных длиной в 10 отсчетов.

Расчет (i,j) элемента матрицы R осуществляется по формуле[16]

$$R(i, j) = \frac{C(i, j)}{\sqrt{C(i, j) \cdot C(j, j)}}. \quad (3.29)$$

где $C=\text{cov}(X)$ - матрица ковариаций. $R=\text{corrcoef}(x,y)$ функция предназначена для расчета матрицы парных коэффициентов корреляции R векторов x и y. Тот же результат можно получить при использовании $\text{corrcoef}([x \ y])$.

Полученный по данным пакета рисунок 3.14 результат выглядит так:

```

10001111010010111000010011101111010110000001001110110100101101100010
10110110100010110011011100100111111010010001

```

Фрагмент МАТЛАБ-программы, выполняющей декодирование кода Манчестера методом максимума корреляции приведен ниже:

```

S1=[1 1 1 1 1 0 0 0 0 0];S0=[0 0 0 0 0 1 1 1 1 1];
clear bit;
for i=bit0:10:ed-9
    if
        sum(sum(corrcoef([yy(i:i+9)',S1'])))>sum(sum(corrcoef([yy(i:i+9)',S0'])))
        bit((i+10-bit0)/10)=1;
    else
        bit((i+10-bit0)/10)=0;
    end
end

```

```

end
bit=num2str(bit);
display(bit)

```

3.6 Декодирование с обнаружением ошибок (CRC)

Как только все биты будут приняты, проводится проверка на ошибку, с использованием 24-бит алгоритма и CRC, содержащегося в принятом пакете данных. Если ошибки не обнаружены, пакет передается далее для декодирования. Если ошибки обнаружены (получается ненулевой результат вычисления по алгоритму CRC), то применяется алгоритм коррекции ошибки.

Циклический избыточный код (англ. Cyclic redundancy check, CRC) — алгоритм нахождения контрольной суммы, предназначенный для проверки целостности данных. CRC является практическим применением помехоустойчивого кодирования, основанном на определённых математических свойствах циклического кода.

Алгоритм CRC базируется на свойствах деления с остатком двоичных многочленов, то есть многочленов над конечным полем. Значение CRC является остатком от деления многочлена, соответствующего входным данным, на некий фиксированный порождающий многочлен.

Каждой конечной последовательности битов взаимно однозначно сопоставляется двоичный полином, последовательность коэффициентов которого представляет собой исходную последовательность. Например, последовательность битов 1011010 соответствует многочлену:

$$P(x) = 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0 = x^6 + x^4 + x^3 + x^1$$

Количество различных многочленов степени меньшей N равно 2^N , что совпадает с числом всех двоичных последовательностей длины N .

Значение контрольной суммы в алгоритме с порождающим многочленом $G(x)$ степени N определяется как битовая последовательность длины N , представляющая многочлен $R(x)$, получившийся в остатке при делении многочлена $P(x)$, представляющего входной поток бит, на многочлен $G(x)$:

$$R(x) = P(x) \cdot x^N \bmod G(x) \quad (3.30)$$

где

$R(x)$ — многочлен, представляющий значение CRC;

$P(x)$ — многочлен, коэффициенты которого представляют входные данные;

$G(x)$ — порождающий многочлен;

N — степень порождающего многочлена.

Умножение осуществляется приписыванием нулевых битов к входной последовательности[17].

При делении с остатком исходного многочлена на порождающий полином $G(x)$ степени N можно получить $2N$ различных остатков от деления. $G(x)$ является неприводимым многочленом. Существует множество стандартизированных образующих многочленов, обладающих хорошими математическими и корреляционными свойствами. В ADS-B сообщении используется порождающий многочлен:

$$G(x) = x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{10} + x^3 + 1$$

Дальше посмотрим декодирование по схеме рисунок 3.18[18].

Декодер работает следующим образом. Перед началом работы содержимое всех разрядов регистров равно нулю. Принимаемая последовательность r в течение первых n тактов вводится в буферный регистр и одновременно с этим в $(n - k)$ -разрядном сдвиговом регистре с обратными связями производится ее деление на порождающий полином $g(x)$. Через n тактов в буферном регистре оказывается принятое слово r , а в регистре синдрома – остаток от деления вектора ошибки на порождающий полином.



Рисунок 3.18 – Структурная схема декодера

Принцип работы декодера заключается в том, что независимо от того, в какой позиции произошла ошибка, осуществляется ее сдвиг в последнюю ячейку буферного регистра с соответствующим пересчетом синдрома и ее исправление в этой позиции.

МАТЛАБ-программа, выполняющая декодирование приведена ниже:

```

%% Do CRC computation, only analyze DF 17, since their CRC are
crc_temp = '000000000000000000000000';
generator = [1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 0 1];
[divid, remainder]=deconv(bit_bin,generator);
remainder=mod(remainder(end-23:end),2);
if isequal(remainder,[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]);
    disp(['CRC right! ',num2str(bit_bin(1:88)),']);
else disp(['CRC error: ',num2str(remainder),']);
end

```

Последовательность после CRC декодирования:

```

>> 1 0 0 0 1 1 0 1 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0
1 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 1

```

1.7 Декодирование данных ADS-B

Следующими шагами обработки данных ADS-B является декодирование содержащихся в принятом пакете данных.

3.7.1 Определение DF и TC

В ADS-B сообщении первые 5 бит пакета содержат формат пакета (Downlink Format (DF)). Первый байт 8D это 10001 и 101 что является DF=17 и CA=5. DF 17 указывает нам, что этот пакет является расширенным сквиттером 112 бит длиной. Не все расширенные сквиттеры содержат координаты. Мы еще должны проверить тип кода (Type Code (TC)). Пятый байт- это первый байт расширенного сквиттера, который содержит на 56 бит (7 байт) больше чем в коротком сквиттере (short squitter). Последние 3 байта не входят в сквиттер и являются проверочными битами на ошибку CRC. Тип кода (Type Code - TC) содержится в первых 5 битах пакета. Значение пакета приведены в таблице 3.1

Таблица 3.1 –Тип кода

DF	Type Code(TC)	Содержание	DF	Type Code(TC)	Содержание
17	1 to 4	Идентификация ВС	17	24	Статус
17	5 to 8	Положение ВС	17	25 to 27	Зарезервировано
17	9 to 18	Высота (Baro Alt)	17	28	Расширенный сквиттер (AC status)
17	19	Скорость ВС	17	29	Target state and status (V.2)
17	20 to 22	Положение по GPS (GNSS Height)	17	30	Зарезервировано
17	23	Тестовое сообщение	17	31	Aircraft Operation status

Полученные по пакету рисунок 3.14 результаты:

```

Command Window
DF =
1 0 0 0 1
CA =
1 1 1
ICAO =
0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1
DATA =
0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 0 1 1 0 1
TC =
0 1 0 1 1
PC =
0 1 1 1 0 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 0 0 0 1
F =
1
CRC right! [1 0 0 0 1 1 1 1 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 0

```

Фрагмент МАТЛАБ-программы, выполняющий данные действия, приведен ниже:

```

bit_bin=bit;
bit_bin=str2num(bit_bin);
for i=1:4:length(bit_bin)

    bit_dec=bin2dec(num2str(bit_bin(i:(i+3))));
    bit_hex((i+3)/4)=dec2hex(bit_dec);
end

DF=bit_bin(1:5); % Downlink Format
disp(['DF' ' =']);
disp(num2str(DF));

CA=bit_bin(6:8); % Message Subtype
disp(['CA' ' =']);
disp(num2str(CA));

ICAO=bit_bin(9:32); % ICAO aircraft address 24 bits
disp(['ICAO' ' =']);
disp(num2str(ICAO));

DATA=bit_bin(33:88); % Data frame
disp(['DATA' ' =']);
disp(num2str(DATA));

TC=bit_bin(33:37); % Type code
disp(['TC' ' =']);
disp(num2str(TC));

```

3.7.2 Идентификация летательного аппарата

Процедура идентификации ЛА и использованные при этом данные показаны в таблицах 3.2 (в 16-ричном) и 3.3 (в двоичном) форматах:

Таблица 3.2 – Данные идентификации ЛА в 16-ричном формате

DF	ICAO24	DATA	Parity check (PC)
8F	4B84EF	2054A632D71820	26E7BF

Таблица 3.3 – Данные идентификации ЛА в двоичном формате

DF	CA	ICAO24 ADDRESS	TC	
10001	111	010010111000010011101111	00100	000
Data				Parity check
010101001010011000110010110101110001100000100000				001001101110011110111111

DF: 10001 > 17

TC: 00100 > 4

Эти два значения подтверждают, что сообщение является хорошим для декодирования опознавательный индекс воздушного судна.

Далее, мы декодуем кадр данных, содержащий номер воздушного судна (идентификация). Для того, чтобы получить номер воздушного судна, необходима таблица для расчета индексов отображение букв:

```
#ABCDEFGHIJKLMNOPQRSTUVWXYZ#####_#####0123456789##
####
```

Результат идентификации ЛА приведен ниже:

HEX: 2054A632D71820

BIN: 00100 000 | 010101 001010 011000 110010 110101 110001 100000 100000

DEC: | 21 10 24 50 53 49 32 32

LTR: | U J X 2 5 1 _ _

Так что теперь у нас есть ID самолета:

aircraft ID = UJX251__

Фрагмент МАТЛАБ-программы, выполняющей идентификацию ЛА, приведен ниже

```
%% Decode plane ID
```

```
table='ABCDEFGHIJKLMNOPQRSTUVWXYZ
UVWXYZ#####_#####012
3456789#####';%look-up table is
needed for mapping index numbers to
letters:
```

```
ID=[];
```

```
for j=1:8
```

```
planeId(j)=bin2dec(num2str(bit_bin(4
1+(j-1)*6:(41+j*6-1) ) ) );
```

```
ID=strcat(ID,table(planeId(j)));
```

```
end
```

```
disp(['aircraft ID=',ID]);
```

3.7.3 Определение координат и высоты ЛА

При декодировании позиции самолета, два различных типа сообщений позиций (нечетных и четных кадров) необходимы, чтобы выяснить долготу и широта самолета. Положение описывается в формате CPR (“компактного донесения о местоположении”). Преимуществом CPR является то, что оно использует меньше битов для кодирования информации о местоположении. Недостатком является сложность декодирования.

Во первых, нужно определить, является ли пакет "нечетным" или "четным".

Получены два следующие сообщения (в 16-ричном виде)-Таблица 3.4, в двоичном формате – Таблицы 3.5 – 3.7:

8F4B84EF5813B4B62B68B3727E91

8F4B84EF5813A1443D9C1B4F2330

Таблица 3.4 –Сообщения в 16-ричном виде

DF	ICAO24	DATA	Parity check(PC)
8F	4B84EF	5813B4B62B68B3	727E91
8F	4B84EF	5813A1443D9C1B	4F2330

Таблица 3.5 – Сообщения в двоичном виде

DF	CA	ICAO24 ADDRESS	TC(DATA)
10001	111	010010111000010011101111	01011
10001	111	010010111000010011101111	01011

Таблица 3.6 – Сообщения в двоичном виде

SS	NICsb	ALT	T	F
00	0	000100111011	0	1
00	0	000100111010	0	0

Таблица 3.7 – Сообщения в двоичном виде

LAT-CPR	LON-CPR	CRC
00101101100010101	11011010001011001	011100100111111010010001
01010001000011110	11001110000011011	010011110010001100110000

Значения компонент сообщений приведены в таблице 3.8.

Таблица 3.8 – Компоненты сообщений:

Биты сообщения	#Биты	Аббревиатура	Название
1-5	5	DF	Формат сообщения
33-37	5	TC	Тип кода
38-39	2	SS	Статус наблюдения
40	1	NICsb	NIC дополнения-B
41-52	12	ALT	Высота
53	1	T	Время
54	1	F	CPR чет/нечет флаг кадра
55-71	17	LAT-CPR	Широта в формате CPR
72-88	17	LON-CPR	Долгота в формате CPR

В обоих сообщениях мы можем найти $DF = 17$ и $Tc = 11$, с тем же адресом ICAO24 4B84EF. Таким образом, эти два кадра действительны для декодирования позиций этого самолета.

Для каждого кадра, бит 54 определяет, является ли это "нечетный" или "четный" кадр: 0 > четный кадр; 1 > нечетный кадр.

Давайте сначала отделить CPR широты и долготы биты в обоих сообщениях. Начнем со следующих шагов:

F	CPR Широта	CPR Долгота	
0	01010001000011110	11001110000011011	> последний принятый кадр
1	00101101100010101	11011010001011001	

Шаг 1: Преобразование двоичной строки в десятичное значение:

поскольку CPR широта и долгота кодируются в 17 битах, 131072 (2^{17}) представляет собой максимальное значение. Полученные значения из расчетов представляют собой проценты от этого максимального значения.

Шаг 2: Вычисляем latitude индекс j lookup-таблицы:

$$j = \text{floor}(59 * \text{LatCPR} - E - 60 * \text{LatCPR} - O + 0.5)$$

Шаг 3: Вычислим относительные широты

Во-первых, будут использоваться две константы:

$$\begin{aligned} DLat_EVEN &= 360.0 / 60 \\ DLat_ODD &= 360.0 / 59 \end{aligned}$$

Тогда мы можем использовать следующие уравнения для расчета относительных широт:

$$\begin{aligned} LatE &= DLatE * (\text{mod}(j, 60) + LatCPR - E) \\ LatE &= LatE - 360 \text{ if } (LatE \geq 270) \\ LatO &= DLatO * (\text{mod}(j, 59) + LatCPR - O) \\ LatO &= LatO - 360 \text{ if } (LatO \geq 270) \end{aligned} \quad (3.31)$$

Если относительный результат широты больше, чем 270, то это означает, что воздушное судно находится в южном полушарии. Таким образом, вычтем 360 из значения.

Затем нам нужно проверить, (с помощью функции $NLfunction()$) если Lat_EVEN и Lat_ODD находятся в той же широтной зоны. Если нет, то просто выйдем здесь, ждем новых данных, а затем запустим программу снова.

Окончательный широта выбирается в зависимости от временной метки кадров-новейший используется:

$$Lat = \begin{cases} Lat_E & \text{if } (T_0 \geq T_1) \\ Lat_O & \text{else} \end{cases} \quad (3.32)$$

Шаг 4: Вычисление долготы.

Для того чтобы получить долготу, сначала нам нужно вычислить индекс долготы m и ni с функцией $Nfunction()$, которая находится в таблице широты зоны.

$$\begin{aligned} ni &= \begin{cases} N(Lat_E, 0) & \text{if } (T_0 \geq T_1) \\ N(Lat_O, 1) & \text{else} \end{cases} \\ m &= \begin{cases} \text{floor}[Lon_{CPR-E} * (NL(Lat_E) - 1) - Lon_{CPR-O} * NL(Lat_E) + 0.5] & \text{if } (T_0 \geq T_1) \\ \text{floor}[Lon_{CPR-E} * (NL(Lat_O) - 1) - Lon_{CPR-O} * NL(Lat_O) + 0.5] & \text{else} \end{cases} \end{aligned} \quad (3.33)$$

$$\begin{aligned} Lon &= \begin{cases} \frac{360.0}{ni} * (\text{Mod}(m, ni) + Lon_{CPR-E}) & \text{if } (T_0 \geq T_1) \\ \frac{360.0}{ni} * (\text{Mod}(m, ni) + Lon_{CPR-O}) & \text{else} \end{cases} \\ Lon &= Lon - 360 \text{ if } (Lon \geq 180) \end{aligned} \quad (3.34)$$

Шаг 5: Вычисление высоты.

Бит №48 в пакете данных указывает, с каким шагом закодирована в данных высота (25 или 100 футов на дискрету) (0 - 100 футов, 1 - 25 футов).

При $Q = 1$, мы можем вычислить высоту следующим образом:

Во-первых, удалим Q-бит $N = 0001001\ 1(Q\ \text{бит})\ 1011 \Rightarrow 155$ (in decimal).

Высота имеет точность ± 25 футов, когда Q-бит равен 1.

Окончательное значение высоты будет: $\text{Alt} = N \cdot 25 - 1000(\text{ft.})$

Шаг 6: Полное положение воздушного судна.

У нас есть все три компонента (широта / долгота / высота) о местоположении воздушного судна:

Полученные результаты:

Latitude=49.8998°
Longitude=36.0464°
Altitude=3775ft

МАТЛАБ-программа, выполняющая определение координат ВС:

```

    clc;clear all;close all;
    load('even.mat');
    bit_bin=str2num(bit);
    for i=1:4:length(bit_bin)

        bit_dec=bin2dec(num2str(bit_bin(i:(i+
        3))));

        bit_hex_even((i+3)/4)=dec2hex(bit_de
        c);
        end

        DF_even=bit_bin(1:5)           ;%
        Downlink Format
        disp(['DF_even'
        ='num2str(DF_even)]);

        ICAO_even=bit_bin(9:32);% ICAO
        aircraft address 24 bits
        disp(['ICAO_even'
        ='num2str(ICAO_even)]);

        TC_even=bit_bin(33:37);%Type
        code
        disp(['TC_even'
        ='num2str(TC_even)]);

        F_even=bit_bin(54);%CPR
        odd/even frame flag
        disp(['F_even'
        ='num2str(F_even)]);
        bit_even=bit_bin;
        %% Do CRC computation, only
        analyze DF 17 even
        generator = [1 1 1 1 1 1 1 1 1 1 1
        1 0 1 0 0 0 0 0 0 1 0 0 1];
        [divid,
        remainder]=deconv(bit_even,generato
        r);
        remainder=mod(remainder(end-
        23:end),2);
        if isequal(remainder,[0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0]);
            disp(['CRC_even right !
            ['num2str(bit_even(1:88)),']]);
        else
            disp(['CRC_even
            error:['num2str(remainder),']]);
        end
        %%
        load('odd.mat');
        bit_bin=str2num(bit);
        for i=1:4:length(bit_bin)

            bit_dec=bin2dec(num2str(bit_bin(i:(i+
            3))));

```

```

bit_hex_odd((i+3)/4)=dec2hex(bit_dec
);
    end

    DF_odd=bit_bin(1:5);% Downlink
Format
    disp(['DF_odd'
    =',num2str(DF_odd)]);

    ICAO_odd=bit_bin(9:32);% ICAO
aircraft address 24 bits
    disp(['ICAO_odd'
    =',num2str(ICAO_odd)]);

    TC_odd=bit_bin(33:37);%Type
code
    disp(['TC_odd'
    =',num2str(TC_odd)]);

    F_odd=bit_bin(54);%CPR
odd/even frame flag
    disp(['F_odd'
    =',num2str(F_odd)]);
    bit_odd=bit_bin;
    %% Do CRC computation, only
analyze DF 17 even
    generator = [1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 0 0 0 0 1 0 0 1];
    [divid,
remainder]=deconv(bit_odd,generator
);
    remainder=mod(remainder(end-
23:end),2);
    if isequal(remainder,[0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0]);
        disp(['CRC_odd right !
[' ,num2str(bit_odd(1:88)),']']);
    else
        disp(['CRC_odd
error:[' ,num2str(remainder),']']);
    end
    %% Decode the position
information

    %Convert the binary string to
decimal value
    LAT_EVEN=bin2dec(num2str(bit_
even(55:71)));
    lat0=LAT_EVEN/(2^17);
    LON_EVEN=bin2dec(num2str(bit_
even(72:88))) ;lon0=LON_EVEN/
(2^17);
    LAT_ODD=bin2dec(num2str(bit_o
dd(55:71))); lat1=LAT_ODD/(2^17);
    LON_ODD=bin2dec(num2str(bit_
odd(72:88))); lon1=LON_ODD/
(2^17);
    % Compute the Latitude
    AirDlat0 = 360.0 / 60;
    AirDlat1 = 360.0 / 59;
    j = floor(59*lat0 - 60*lat1 + 0.5);
    rlat0 = AirDlat0 * (mod(j,60) +
lat0 );
    rlat1 = AirDlat1 * (mod(j,59) +
lat1);
    if (rlat0 >= 270)
        rlat0 = rlat0 - 360;
    end
    if (rlat1 >= 270)
        rlat1 = rlat1 - 360;
    end
    % Check that both are in the same
latitude zone, or abort.
    if (cprNLFunction(rlat0) ~=
cprNLFunction(rlat1))
        return;
    end
    T0=bit_even(53);T1=bit_odd(53);
    %Calculate longitude,ni and the
longitude index m
    if (T0 >= T1)%depending on the
time stamp of the frames—the newest one
is used:
        % Use even packet
        ni = cprNFunction(rlat0,0);

```

```

        m      = floor(((lon0      *      lon = lat.lon - 360;
(cprNLFunction(rlat0)-1))-(lon1      *      end
cprNLFunction(rlat0)))) + 0.5);      %Calculate altitude
        lat = rlat0;      alt=bit_odd(41:52);
        lon      =      360/ni*      Q=alt(8);
(mod(m,ni)+lon0);      if Q==1
        else      alt=[alt(1:7),alt(9:length(alt))];
        % Use odd packet      N=bin2dec(num2str(alt));
        ni = cprNLFunction(rlat1,1);      Alt=N*25-100;
        m      = floor(((lon0      *      else
(cprNLFunction(rlat1)-1))-(lon1      *      disp('altitude not in multiples of
cprNLFunction(rlat1)))+ 0.5);      25 or 100ft')
        lat = rlat1;      return;
        lon      =      360/ni*      end
(mod(m,ni)+lon1);      disp(['Latitude=',num2str(lat),'°'])
        end      disp(['Longitude=',num2str(lon),'°'
        if (lon > 180)      ])
        disp(['Altitude=',num2str(Alt),'ft'])

```

3.7.4 Определение вектора скорости ЛА

Есть два различных типа сообщений для скоростей, определяемые 3-битным значением подтипа в сообщении. Если значение подтипа равно 1 или 2, то это поверхностная скорость (путевая скорость). А если значение подтипа равно 3 или 4, сообщается о воздушной скорости.

На практике подтип 3 и 4 используется очень редко. В этом дипломном проекте мы будем декодировать сообщения подтипа 1.

Сообщение о скорости воздушного судна имеет DF: 17, TC: 19. и код подтипа представлены в битах от 38 до 40. Теперь мы можем расшифровать эти сообщения [19].

Получено следующее сообщение (в 16-ричном виде) - Таблица 3.9, в двоичном формате – Таблицы 3.10 – 3.12.:

8F4B84EF9910B00B082485E17BDC

Таблица 3.9 – Сообщение в 16-ричном виде

DF	ICAO24	DATA	Parity check(PC)
8F	4B84EF	9910B00B082485	E17BDC

Таблица 3.10 – Сообщение в двоичном виде

DF	CA	ICAO24 ADDRESS	TC(DATA)
10001	111	010010111000010011101111	10011

Таблица 3.11 – Сообщение в двоичном виде

ST	IC	RESV_A	NAC	S-EW	V-EW	S-NS	V-NS	VrSrc
001	0	0	010	0	0010110000	0	0001011000	0

Таблица 3.12 – Сообщение в двоичном виде

S-Vr	Vr	RESV_B	S-Dif	Dif	CRC
1	000001001	00	1	0000101	111000010111101111011100

Значения компонент сообщений приведены в таблице 3.13.

Таблица 3.13 – Компоненты сообщений

Биты сообщения	N Биты	Аббревиатура	Название
33-37	5	TC	Тип кода
38-40	3	ST	Подтип
41	1	IC	Флаг изменения намерения
42	1	RESV_A	Зарезервировано-А
43-45	3	NAC	неопределенность скорости (NAC)
46	1	S-WE	Восток-Запад скорость знак
47-56	10	V-WE	Скорость Восток-Запад
57	1	S-NS	Скорость знак Север-Юг
58-67	10	V-NS	Скорость Север-Юг
68	1	VrSrc	Источник вертикальной скорости

Продолжение таблицы 3.13

Биты сообщения	N Биты	Аббревиатура	Название
69	1	S-Vr	Знак вертикальной скорости
70-78	9	Vr	Вертикальная скорость
79-80	2	RESV_B	Зарезервировано-В
81	1	S-Dif	Diff от барометрической высоты, знак
82-88	7	Dif	Diff от барометрической высоты

Горизонтальная скорость. Для расчета горизонтальной скорости нам нужны четыре значения, величина скорости Восток-Запад - V-WE, знак скорости Восток-Запад - S-WE, величина скорости Север-Юг - V-NS, знак скорости Север-Юг - S-NS.

S-NS:

1 -> flying North to South (с севера на юг);

0 -> flying South to North (с юга на север).

S-EW:

1 -> flying East to West(с востока на запад);

0 -> flying West to East(с запада на восток).

Скорость (v) и заголовок (h) может быть вычислена следующим образом:

$$\begin{aligned}
 V(we) &= \begin{cases} -1 * [V(ew) - 1] & \text{if } (s(ew) = 1) \\ V(ew) - 1 & \text{if } (s(ew) = 0) \end{cases} \\
 V(sn) &= \begin{cases} -1 * [V(ns) - 1] & \text{if } (s(ns) = 1) \\ V(ns) - 1 & \text{if } (s(ns) = 0) \end{cases} \\
 v &= \sqrt{V_{we}^2 + V_{sn}^2} \\
 h &= \arctan\left(\frac{V_{we}}{V_{sn}}\right) * \frac{360}{2\pi} \quad (\text{deg})
 \end{aligned}
 \tag{3.35}$$

В случае отрицательного значения здесь, мы будем просто добавлять 360 градусов.

$$h = h + 360 (\text{if } h < 0).$$

Вертикальная скорость. Направление вертикального перемещения воздушного судна может быть считано из S поля (Vr), в сообщении (бит - 69):

0 -> Вверх;

1 -> Вниз.

Фактическая вертикальная скорость Vr является двоичным представлением значения в футах / мин (фут / мин). В нашем сообщении:

Vr: 000001001 => 9

S-Vr: 1 => Вверх/на подъеме.

Полученные результаты:

flying West to East (полет с запада на восток);

flying South to North (полет с юга на север);

Horizontal Velocity (горизонтальная скорость) = 100.5393 m/s;

flying Up (подъем);

The heading (угол) = 63.5661deg;

Vertical Rate (вертикальная скорость) = 2.9261m/s.

МАТЛАБ-программа, выполняющая эти расчеты, приведена ниже:

```
%% Decode plane velocity
S_EW=bit_bin(46);%East-West
velocity sign
S_NS=bit_bin(57);%North-South
velocity sign
V_EW=bit_bin(47:56);%East-West
velocity

v_ew=bin2dec(num2str(V_EW));%1knot=1851.9648/3600(m/s)
V_NS=bit_bin(58:67);%North-South
velocity
v_ns=bin2dec(num2str(V_NS));
S_Vr=bit_bin(69);%Vertical rate sign
Vr=bit_bin(70:78);%Vertical rate
vr=bin2dec(num2str(Vr));
if S_EW % 0 means east direction,
1 means west
    disp(['flying East to West']);
    V_ew=-1*(v_ew-1);
else
    disp(['flying West to East']);

V_ew=v_ew-1;
end

if S_NS % 0 means north direction,
1 means south direction
    disp(['flying North to South']);
    V_ns=-1*(v_ns-1);
else
    disp(['flying South to North']);
    V_ns=v_ns-1;
end

v=sqrt(V_ew^2+V_ns^2);%speed
disp(['Horizontal
Velocity=',num2str(v),'knot/h=',num2str(
v*1.852/3.6),'m/s']);
if S_Vr % 0 means up, 1 means
down
    disp(['flying Up']);
else
    disp(['flying Down/Descending']);
end
h=atan(V_ew/V_ns)*360/(2*pi);
```

```

if h<0
    h=h+360;
end
disp(['The

```

```

heading=',num2str(h),'deg']]);
disp(['Vertical
Rate=',num2str(vr),'feet/minute=',num
2str(vr*0.00508*64),'m/s']]);

```

3.8 Оценка помехоустойчивости процедур обнаружения и декодирования ADS-B-данных.

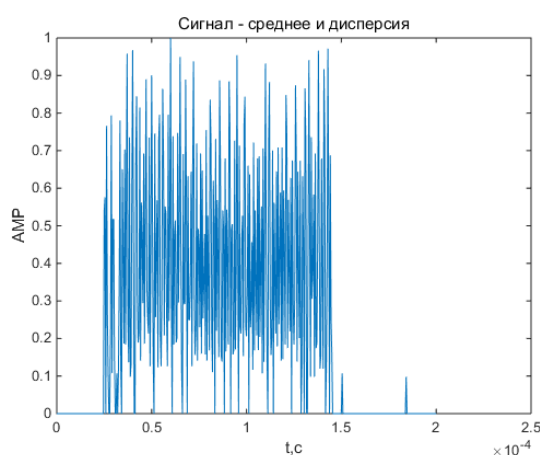
В этом разделе мы будем анализировать зависимость между вероятностью ошибки и отношение сигнала к шуму с помощью различных алгоритмов при обнаружении и декодировании ADS-B сообщения.

Процедура моделирования выполнялась следующим образом. Во-первых, мы приняли реальный сигнал очень высокого качества, и используем этот сигнал в качестве модели полезного сигнала. Затем мы добавим гауссовский белый шум постоянной мощности. Путем изменения уровня сигнала, мы проанализировали результаты обнаружения.

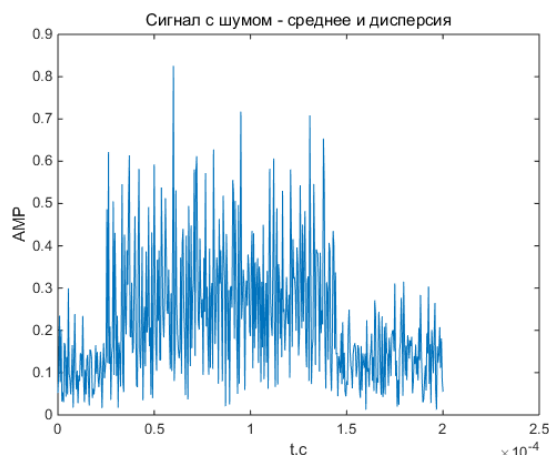
Далее, мы будем исследовать взаимосвязь между $P_{\text{пр.обн}}$ и SNR (Соотношение сигнала и шума) при заданном значении вероятности ложной тревоги $F=F_{\text{зад}}$.

Полученные результаты обнаружения показаны в рисунках 3.19 - 3.20. И на рисунке 3.21 показана зависимость между $P_{\text{пр.обн}}$ и SNR:

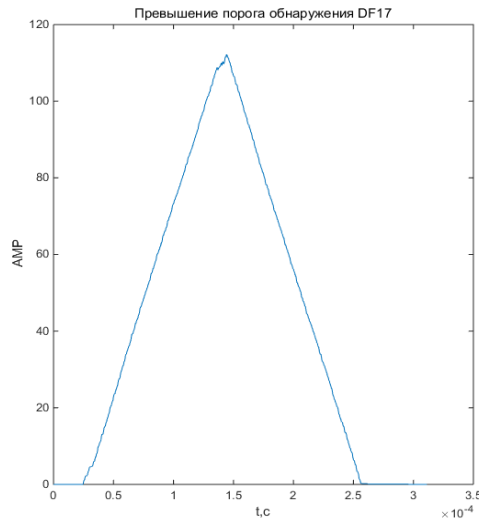
Сигнал не обнаружен (ниже порога) при условии, что сигнал есть:



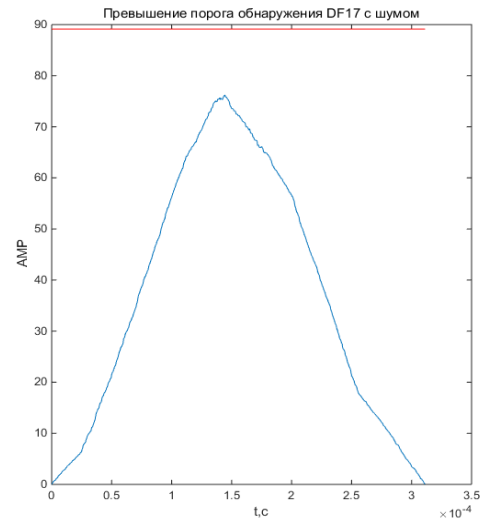
а



б



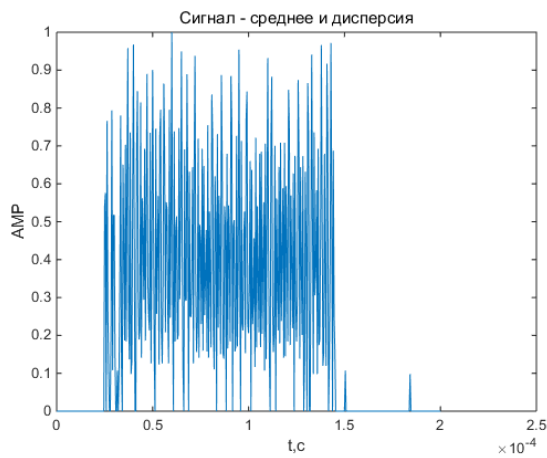
В



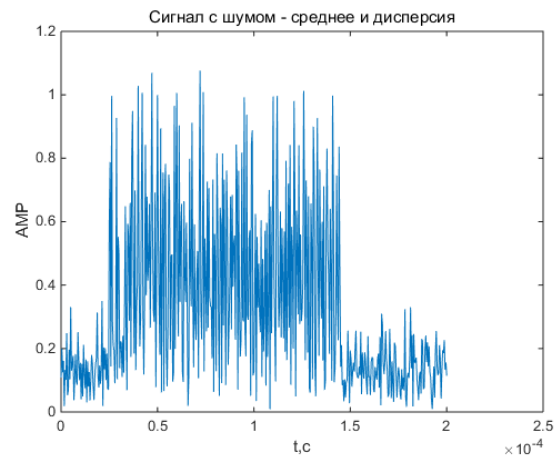
Г

Рисунок 3.19 – Обнаружение сигналов при малом SNR: а – Полезный сигнал без шума; б – Сигнал с шумом ; в – Корреляционная функция сигнала без шума; г – Корреляционная функция сигнала с шумом

Сигнал обнаружен (выше порога) при условии, что сигнал есть:



а



б

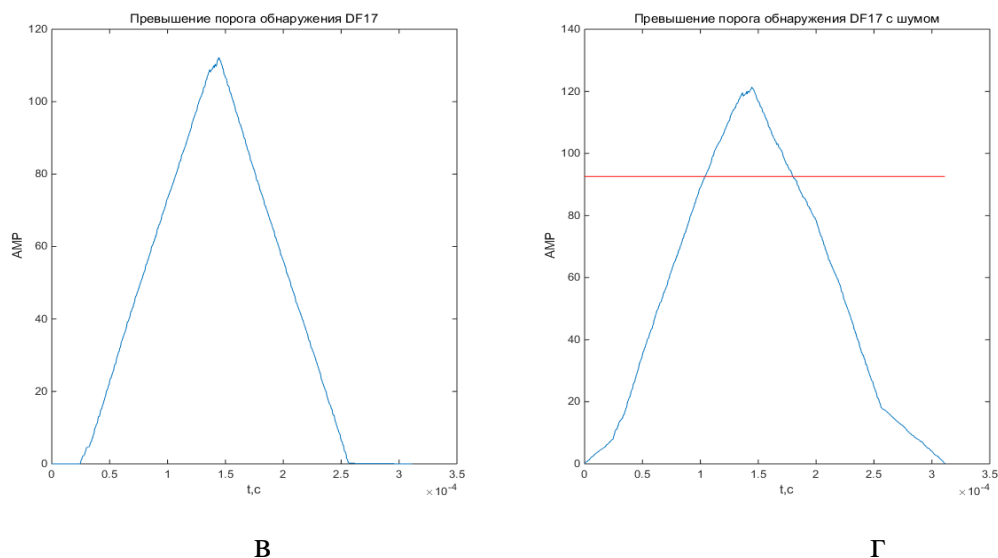


Рисунок 3.20 – Обнаружение сигналов при большом SNR:
а – Полезный сигнал без шума;б – Сигнал с шумом;в – Корреляционная функция сигнала без шума;г – Корреляционная функция сигнала с шумом

Результат оценки зависимости $P_{\text{пр.обн}}$ от SNR приведен на рисунке 3.21.

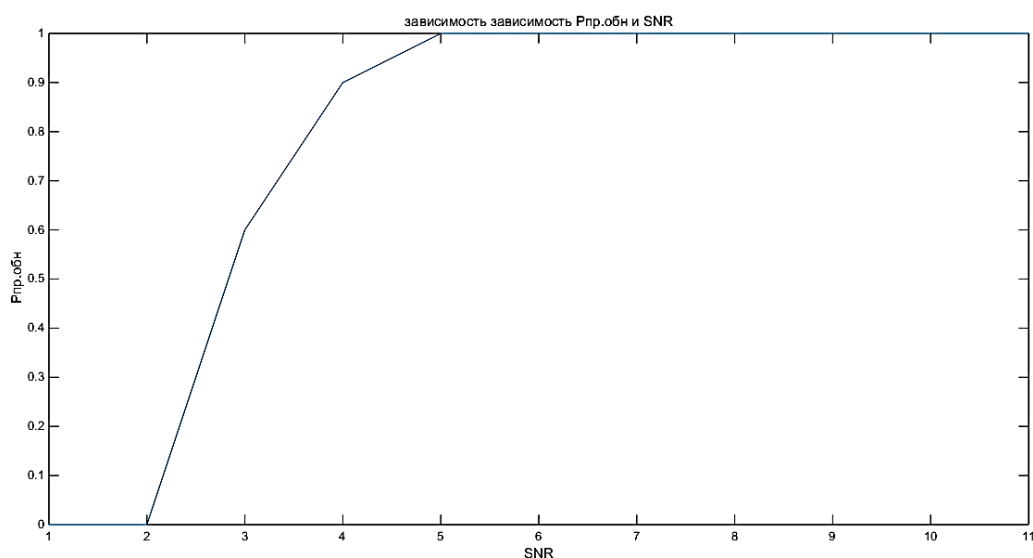


Рисунок 3.21 – Зависимость $P_{\text{пр.обн}}$ от SNR

МАТЛАБ-программа, выполняющая оценку вероятности правильного обнаружения сигнала в зависимости от отношения сигнал/шум в канале связи:

```

clc;
clear all;
close all;
%
a=1;
for m=1:0.05:1.5
    % m=1;
    [y,Fs]
    wavread('SDRSharp_20140312_12270

```

```

1Z_1090000kHz_IQ.wav',[0.2189*240
0000 0.2191*2400000]);
y = y';
y1 = y(1,1:size(y,2));
y2 = y(2,1:size(y,2));
n=0;

Z = zeros (size(y1));

Y = y1+1i*y2;
Ya=abs(Y);
Ya(find(Ya<0.05))=0;
t=0:1/Fs:((length(y)-1)/Fs);
figure
plot(t,Ya);
ylabel('AMP');xlabel('t,c')
title('Сигнал - среднее и дисперсия');
% n=randn(1,2500);
% n=n/std(n);
% n=n-mean(n);
% n=0;
% n=sqrt(1);
% n=a+b*n;

mu=0;sigma=0.08;
% for i=1:
n1=normrnd(mu,sigma,[1,10*size(y,2)
]);
n2=normrnd(mu,sigma,[1,10*size(y,2)
]);
% n1=randn(1);
% n2=randn(1);
y1=repmat(y1,1,10);y2=repmat(y2,1,1
0); %эргодичекой процесс
y1=m*y1+n1;y2=m*y2+n2;
% end
Yn = y1+1i*y2;Yn=abs(Yn);
% Yn(find(Yn<0.05))=0;
figure
t=0:1/Fs:((10*length(y)-1)/Fs);
plot(t,Yn);
ylabel('AMP');xlabel('t,c')

```

```

title('Сигнал с шумом - среднее и
дисперсия'); % signal with noise
len=size(Yn,2);

DF_17(1:268)=1;
Y_a=conv(Ya,transpose(DF_17));
Y_n=conv(Yn,transpose(DF_17));
fprintf('Выполняется сверка.... : \n');
t_t1=0:1/Fs:((length(Y_a)-1)/Fs);
t_t2=0:1/Fs:((length(Y_n)-1)/Fs);
figure
subplot(1,2,1);
plot(t_t1,Y_a);
ylabel('AMP');xlabel('t,c')
title('Превышение порога
обнаружения DF17');

F=1e-9;%вероятность ложной
тревоги
l0=mean(sqrt(2*log(1/F))*(conv(abs(n
ormrnd(mu,sigma,[1,size(y,2)])),transp
ose(DF_17))));
subplot(1,2,2);
plot(t_t2,Y_n);
hold on
line(t_t2,l0*ones(1,length(Y_n)),'Color'
,'r')
ylabel('AMP');xlabel('t,c')
title('Превышение порога
обнаружения DF17 с шумом');
k=1;
while k<length(Y_n)
    if Y_n(k)>l0
        n=n+1;
        k=k+258*1.5;
    % while Y_n(k)>l0 ;
    % k=k+1;
    % end
    end
    k=k+1;
end
N(a)=n;
a=a+1;
end

```

```
close all;
figure
p=N/10;
plot(p);
```

```
title('зависимость      зависимость
Pnp.обн и SNR');
ylabel('Pnp.обн');xlabel('SNR')
```

При исследовании декодирования кода Манчестера с использованием трех методов, так же как и при обнаружении сигнала, мы выбираем в качестве модели сигнала реальный ADS-B сигнал, принятый практически без помех, и используем его в качестве модели полезного сигнала, а затем анализируем взаимосвязь между вероятностью ошибки и мощностью шума. Вероятность ошибки получается по формуле

$$P_{\text{ошдек}} = \frac{n_{\text{ош}}}{N} \times 100\% \quad (3.36)$$

где $P_{\text{ошдек}}$ - вероятность ошибки декодирования; $n_{\text{ош}}$ - количество ошибок; N - общее количество бит.

Полученный результат показан на рисунке 3.22:

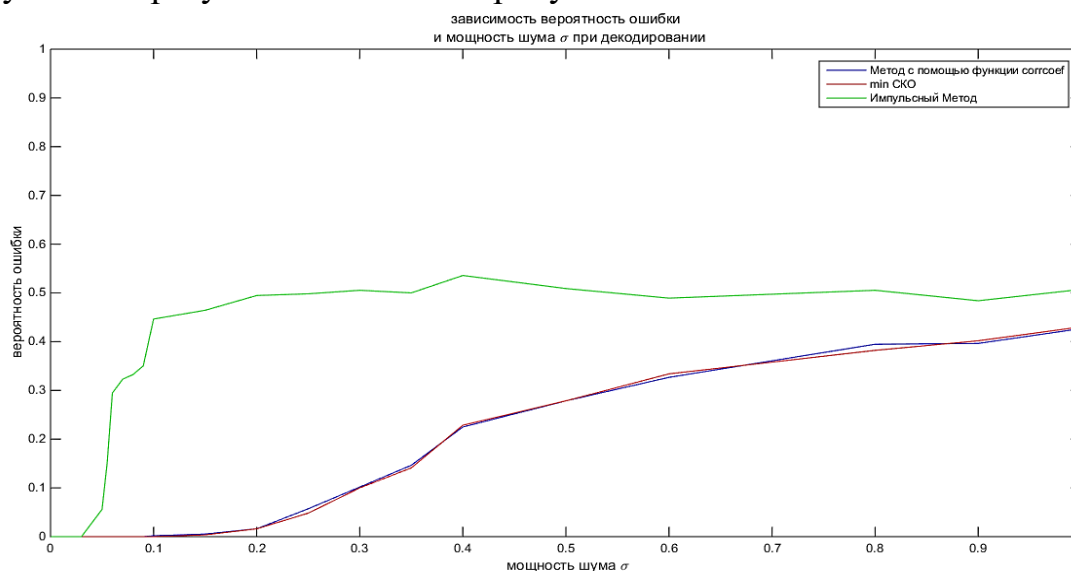


Рисунок 3.22 – Зависимость $P_{\text{ош.дек}}$ от мощности шума δ

Из рисунка очевидно, помехоустойчивости методов по максимуму корреляции по минимуму SKO гораздо лучше, чем импульсного метода. Таким образом, при декодировании зашумленных сигналов лучше не использовать импульсный метод.

4 Сравнение качества приема и декодирования данных ADS-B при использовании приемников RTL-SDR и AIRSPY

В разделе 2.2.3 мы сравнили AIRSPY приемник и RTL-SDR приемник, приёмник по теории. AIRSPY явно выигрывает практически по всем параметрам кроме цены. Он имеет более широкий динамический диапазон, большую частоту

дискретизации и разрядность АЦП. В этом мы будем сравнивать характеристики приемники по эксперименту.

Эксперимент выполняем так же, как и в предыдущем разделе. В качестве модели полезного сигнала используем отрезок принятого практически без шумов реального сигнала, шумы моделируем в МАТЛАБ. Далее, выполнив декодирование кода Манчестера, мы рассчитаем вероятность ошибки по формуле 3.36. В конечном счете, мы будем сравнивать характеристик приемники AIRSPY и RTL-SDR с помощью зависимости между вероятностью ошибки $P_{\text{ош.дек}}$ и мощности шума δ .

Зависимости между вероятностью ошибки $P_{\text{ош.дек}}$ и мощности шума δ показана в рисунке 3.23.

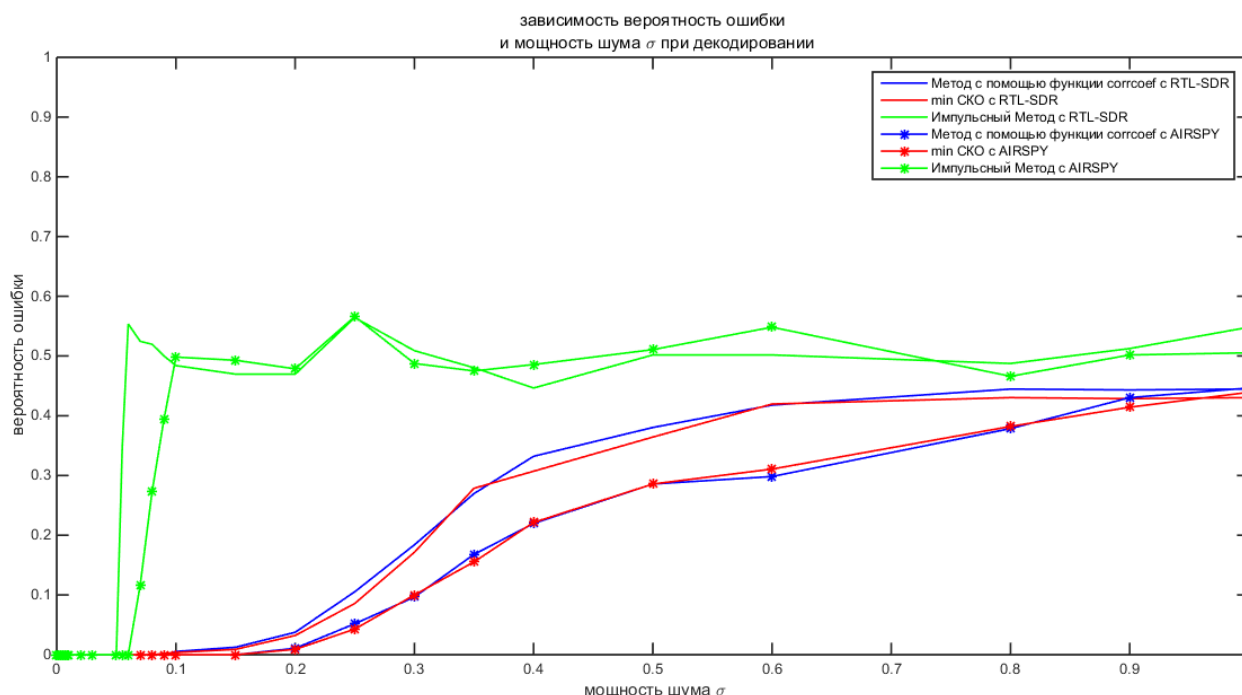


Рисунок 4.1 – Зависимость $P_{\text{ош.дек}}$ и мощности шума δ с использованием AIRSPY и RTL-SDR приёмники

По зависимости между вероятностью ошибки $P_{\text{ош.дек}}$ и мощности шума, мы получаем, что качества сигнала принятого приемником AIRSPY лучше, чем качество сигнала принимаемого приемником RTL-SDR с экспериментальной точки зрения.

5 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

5.1 Обзорная часть

Любая производственная деятельность может быть целесообразной только в том случае, если она дает продукт, способный удовлетворять ту или иную потребность людей. В рыночной экономики этот продукт выступает в условиях форме товара, т.е. его нужно продать, чтобы получить денежный доход. Производитель товара, принимая решение о том, следует ли изготавливать данный товар, прежде всего, должен выяснить, возможна ли в принципе его продажа на рынке. Для этого нужно оценить конкурентоспособность изделия. Такая оценка предполагает изучение возможного рынка сбыта, спроса потенциальных покупателей, получение информации о качестве конкурирующих товаров, издержках конкурентов. Обязательное условие решения о выпуске товара - его достаточно высокая конкурентоспособность. В противном случае изготовление товара нецелесообразно.

Затем необходимо определить эффективность производства, которая в значительной степени влияет на конкурентоспособность товаров. В современных условиях рыночной экономики ставится задача повышения эффективности производства на каждом предприятии. Работа различных их видов требует непрерывного совершенствования производства, повышения его эффективности за счет рационального использования всех видов ресурсов, постоянного улучшения качества продукции и внедрение передовых технологий. Успешное решение этих задач в значительной степени обеспечивает стабильность работы предприятия и его конкурентоспособность. Главной целью этой части дипломного проекта является расчет, обоснование, анализ экономических параметров, себестоимости разработки программного обеспечения для обнаружения воздушных судов по технологии ADS-B на базе приемники RTL-SDR и AIRSPY.

Конечный продукт не предназначен для массового применения, заинтересованы в его приобретении могут быть только некоторые университеты, институты или специализированные НИИ. В связи с этим, нет необходимости рассчитывать емкость рынка и рентабельность.

5.2 Расчёт себестоимости программного продукта

В данной части представлены оценка себестоимости, расчет сметы затрат на проектирование программного продукта

Смета затрат на научно исследовательскую работу (НИР) включает затраты, идущие как непосредственно для выполнения НИР, так и прибыль (накопления) для развития исследований.

Смета затрат на НИР включает в себя:

- затраты на оплату труда;
- материальные затраты;
- амортизация оборудования;
- отчисления на социальные мероприятия;
- другие затраты.

Статьи затрат на НИР относятся к внепроизводственным расходам и рассчитываются по укрупненной, ниже представленной методике.

К затратам на оплату труда относятся основная и дополнительная заработная плата персонала, занятого непосредственно при выполнении конкретной темы: научные работники, научно–технический, научно–вспомогательный персонал и производственные рабочие.

В данном разделе необходимо составить перечень этапов работ по разработке программного обеспечения и рассчитать трудоемкость и продолжительность работ над проектом.

Продолжительность каждого этапа определяется по формуле:

$$T = \frac{t}{n} \quad (5.1)$$

где Т – продолжительность этапа в рабочих днях;

t– трудоемкость этапа, человеко-дни;

n – принятое количество исполнителей, одновременно занятых на определенном этапе работы.

Для разработки того или иного приложения, потребуется наличие сотрудников, которые, собственно, и будут выполнять разработку программного обеспечения.

Данный программный продукт требует участия двух исполнителей: профессор (руководитель) и младший научный сотрудник.

Результаты расчета трудоемкости представлены в табл. 5.1. Согласно этим подсчетам, на разработку программного обеспечения потребуется 40 дней. Рабочий месяц положим равным 21 день.

Таблица 5.1 – Перечень работ по разработке SDR—приемников

№ п/п	Этапы работ	Трудоемкость, чел./дн		Продолжительность дн
		Руководитель профессор	Младший научный сотрудник	
1	Разработка технического задания	2	2	2
2	Постановка задачи, выяснение ключевых целей проекта	2	2	2
3	Разработка алгоритма работы программы	3	5	5
4	Разработка программного кода	5	15	15
5	Тестирование программы, проведение экспериментов	2	7	7
6	Доработка ПО	2	3	3
7	Окончательное тестирование и проведение экспериментов	2	2	2
8	Разработка документации	2	4	4
Итого:		20	40	40

Для проектирования и разработки программного продукта, понадобится использовать труд профессора (20 чел/дн), а также труд младшего научного сотрудника (40 чел/дн).

В конечном итоге, мы получили, что на разработку программного обеспечения нам потребуется команда, состоящая из двух человек: профессор и младший научный сотрудник, срок выполнения – 40 дней.

Стоимость необходимого оборудования, инструментов и материалов, а также их назначение и необходимое количество представлены в табл. 5.2 и табл. 5.3.

Таблица 5.2 – Стоимость оборудования

№ п/п	Наименование материала	Назначение	Количество, шт.	Цена за единицу, грн.	Общая стоимость
1	Ноутбук Lenovo IdeaPad Y480	Работа с программной средой	1	19615,00	19615,00
2	Компьютерная беспроводная мышь Logitech Wireless Mouse M175 Black	Навигация в среде разработки	1	329,00	329,00
3	Принтер HP LaserJet P2035 (CE461A)	Распечатка документации	1	5 984,00	5 984,00
4	Приёмник RTL - SDR	Приём сигнала	1	300	300
5	Приёмник AIRSPY	Приём сигнала	1	6000	6000
6	Антенна	Приём сигнала	2	300	600
Итого:					32828,00

Таблица 5.3. – Расходы на материалы и инструменты

№	Наименование материала	Цена за единицу, Грн.	Количество, шт.	Сумма, грн.
1	Папка для проекта	11,00	2	22,00
2	Бумага(пачка)	100,00	1	100,00
3	Ручка	4,00	4	16,00
4	Карандаш	3,75	2	7,50
5	Маркер	5,50	3	16,50
6	Флеш память USB 32 Гб	249,00	1	249,00
7	Печать на ч/б принтере	0,30	120 листов	36,00
Итого				447,00

Таким образом, расходы на материалы и инструменты для данного проекта составляют $M = 447,00$.

Заработная плата — вознаграждение за труд в зависимости от квалификации работника, сложности, количества, качества и условий выполняемой работы, а также компенсационные выплаты и стимулирующие выплаты.

Заработная плата профессора и младшего научного сотрудника состоит из основной и дополнительной заработной платы.

В разработке программного модуля принимали участия профессор и младший научный сотрудник. Разработка длилась 40 рабочих дня. Рабочий день — восьмичасовой.

Для данного проекта рассчитывается заработная плата профессора, оклад которого составляет 5200 грн/мес (21 рабочих дня в месяце) и младшего научного сотрудника, оклад которого составляет 3200 грн/мес (табл. 5.4).

Таблица 5.4 – Средняя месячная и дневная заработные платы

№ п/п	Должность	Заработная плата/ месяц (21 день), грн.	Заработная плата/ день, грн.
1	Профессор(руководитель)	5200	247,61
2	Младший научный сотрудник	3200	152,38

Основная заработная плата ($ЗП_{осн}$) состоит из суммы средней дневной заработной платы профессора и младшего научного сотрудника, умноженной на их трудоемкость и рассчитывается следующим образом:

$$ЗП_{осн} = ЗП_{днпро} \cdot T_{про} + ЗП_{днмлсот} \cdot T_{млсот},$$

где $ЗП_{днпро}$, $ЗП_{днмлсот}$ — дневная заработная плата профессора и младшего научного сотрудника соответственно (таблица 5.4); $T_{про}$, $T_{млсот}$ — трудоемкость доцента и младшего научного сотрудника (таблица 5.1).

Таким образом, основная заработная плата равна :

$$ЗП_{осн} = 247,61 \cdot 40 + 152,38 \cdot 40 = 9904,4 + 6095,2 = 15999,6 \text{ (грн)}.$$

Следовательно, основная заработная плата составляет 15999,6 грн.

Далее производится расчет дополнительной заработной платы ($ЗП_{доп}$), которая составляет 20% от основной заработной платы младшего научного сотрудника и 70% от основной заработной платы профессора, рассчитывается

$$ЗП_{доп} = ЗП_{осн} \cdot N, \quad (5.2)$$

где N — коэффициент дополнительной заработной платы.

Следовательно, по формуле (5.2) дополнительная заработная плата младшего научного сотрудника равна:

$$ЗП_{\text{доп.млад}} = 6095,2 \cdot 0,2 = 1219,04 \text{ (грн).}$$

Дополнительная заработная плата профессора равна:

$$ЗП_{\text{доп.проф}} = 9904,4 \cdot 0,7 = 6933,08 \text{ (грн).}$$

Общая дополнительная заработная плата равна:

$$ЗП_{\text{доп}} = ЗП_{\text{доп.проф}} + ЗП_{\text{доп.млад}} = 6933,08 + 1219,04 = 8152,12 \text{ (грн).}$$

Итого, общий фонд заработной платы составляет:

$$ЗП = ЗП_{\text{осн}} + ЗП_{\text{доп}} = 15999,6 + 8152,12 = 24151,72 \text{ (грн).}$$

Начисления на заработную плату (единый социальный фонд – $H_{\text{соц}}$) составляют 22% и рассчитывается по формуле:

$$ЗП_{\text{соц}} = (ЗП_{\text{осн}} + ЗП_{\text{доп}}) \cdot \frac{H_{\text{соц}}}{100}, \quad (5.3)$$

Следовательно, по формуле (4.4) начисления на заработную плату составляют:

$$ЗП_{\text{соц}} = 24151,72 \cdot 0,22 = 5313,38 \text{ (грн).}$$

Затем производится расчет амортизации (АМ) оборудования, в нашем случае для ноутбука Ноутбук Lenovo IdeaPad Y480 и принтера HP LaserJet P2035 (CE461A).

Амортизацию оборудования как элемент себестоимости данной продукции предлагается рассчитывать по такой методике:

1) С учетом первоначальной стоимости оборудования и годовой нормы амортизации рассчитывается годовая сумма амортизационных отчислений по данному виду оборудования по формуле:

$$A_{\text{год}} = \frac{C_{\text{Пj}} \times H_{\text{ам}}}{100\%}, \quad (5.4)$$

где $A_{\text{год}}$ - годовые амортизационные отчисления по данному виду оборудования, грн;

$C_{\text{Пj}}$ - первоначальная стоимость оборудования данного вида, грн.;

$H_{\text{ам}}$ - годовая норма амортизации, в процентах.

2) Определяется величина амортизационных отчислений в расчете на один час работы оборудования данного вида:

$$A_{\text{час}} = \frac{A_{\text{год}}}{T_j}, \quad (5.5)$$

где $A_{j\text{час}}$ - величина амортизации оборудования данного вида в течение одного часа его использования, грн.;

T_j - годовой фонд работы данного оборудования в часах.

3) В зависимости от времени использования оборудования в процессе изготовления продукта рассчитывается величина амортизационных отчислений, связанных с производством единицы данного вида:

$$A_{j\Pi} = A_{j\text{час}} \times t_{j\Pi}, \quad (5.6)$$

где $A_{j\Pi}$ - амортизация оборудования данного вида при изготовлении продукта, грн.;

$t_{j\Pi}$ - время использования оборудования данного вида при изготовлении продукта, часы.

Если использовать различное оборудование, то величина амортизации как элемент себестоимости продукта должна быть определена путем суммирования амортизационных отчислений по отдельным видам оборудования:

$$A_{\Pi} = \sum_{j=1}^m A_j, \quad (5.7)$$

где A_{Π} - сумма амортизации по продукту, грн.;

m – количество видов оборудования.

Следовательно, по формуле (5.4) годовая сумма амортизационных отчислений для ноутбука Lenovo IdeaPad Y480:

$$A_{\text{ноутгод}} = 19615,00 \cdot 0,25 = 4903,75 \text{ (грн.)}.$$

Тогда, по формуле (5.5) величина амортизационных отчислений для ноутбука Lenovo IdeaPad Y480:

$$A_{j\text{час}} = 4903,75 / 1976 = 2,48 \text{ (грн.)}.$$

По формуле (5.6) величина амортизационных отчислений в зависимости от времени использования ноутбука Lenovo IdeaPad Y480:

$$A_{\text{ноут}\Pi} = 2,48 \cdot 320 = 793,6 \text{ (грн.)}.$$

Следовательно, по формуле (5.4) годовая сумма амортизационных отчислений для принтера HP LaserJet P2035 (CE461A):

$$A_{\text{прингод}} = 5\,984,00 \cdot 0,25 = 1496 \text{ (грн.)}.$$

Тогда, по формуле (5.5) величина амортизационных отчислений для принтера HP LaserJet P2035 (CE461A):

$$A_{\text{принчас}} = 1496 / 1976 = 0,76 \text{ (грн.)}.$$

По формуле (5.6.) величина амортизационных отчислений в зависимости от времени использования принтера HP LaserJet P2035 (CE461A):

$$A_{\text{принт}} = 0,76 \cdot 320 = 243,20 \text{ (грн)}.$$

По формуле (5.7) сумма амортизации по продукту:

$$A_{\text{п}} = 793,6 + 243,20 = 1036,80 \text{ (грн)}.$$

Стоимость технологической электроэнергии вычисляется по формуле

$$S_{\text{э}} = T_{\text{ар}} \cdot T \cdot W, \quad (5.8)$$

где $T_{\text{ар}}$ – тариф электроэнергии за один кВт, равный 1,6 грн.;

T – количество часов работы;

W – потребляемая мощность.

Следовательно, по формуле (5.9) стоимость энергии ноутбука, при мощности 0,032 кВт:

$$S_{\text{Эноут}} = 1,6 \cdot 320 \cdot 0,032 = 16,38 \text{ (грн)}.$$

Стоимость энергии принтера, при мощности 0,11 кВт:

$$S_{\text{Эприн}} = 1,6 \cdot 320 \cdot 0,11 = 56,32 \text{ (грн)}.$$

Следовательно, стоимость технологической электроэнергии

$$S_{\text{э}} = 16,38 + 56,32 = 72,70 \text{ (грн)}.$$

Стоимость осветительной электроэнергии рассчитывается по формуле (5.8) при мощности, потребляемая осветительным прибором (электролампой) в час $W = 0,10$ кВт:

$$S = 1,6 \cdot 0,1 \cdot 320 = 51,2 \text{ грн.}$$

$T_{\text{ар}}$ – тарифная стоимость 1 кВт электроэнергии, $T_{\text{ар}} = 1,6$ грн.

Стоимость осветительной электроэнергии равна $S = 51,2$ грн.

Себестоимость — это стоимостная оценка используемых в процессе производства продукции (работ, услуг) природных ресурсов, сырья, материалов, топлива, энергии, основных фондов, трудовых ресурсов и других затрат на ее производство и реализацию. Себестоимость равна сумме всех расходов на разработку проекта и рассчитывается по формуле:

$$C = 3\Pi_{\text{осн}} + 3\Pi_{\text{дон}} + 3\Pi_{\text{соц}} + AM + M + S_{\text{э}} + S, \quad (5.9)$$

Следовательно, из формулы (5.11) себестоимость составляет: С
 $=15999,6 + 8152,12 + 5313,38 + 1036,80 + 447,00 + 72,70 + 51,2 = 31072,8$ (грн).

Калькуляционные статьи на разработку программного обеспечения представлены в таблице 5.5

Таблица 5.5 – Статьи калькуляции на разработку программного продукта

№	Статья калькуляции	Расходы,
1	Материалы и инструменты	447,00
2	Основная заработная плата	15999,6
3	Дополнительная заработная плата	8152,12
4	Общая заработная плата	24151,72
5	Начисления на заработную плату	5313,38
6	Амортизация	1036,80
7	Стоимость технологической	72,70
8	Стоимость осветительной	51,2
9	Общая себестоимость разработки	31072,8

Для оптимального решения данной задачи, рассмотрим второй вариант. Выполнения работы, в котором для разработки программного продукта будут задействованы профессор (руководитель) и молодой доцент (исполнитель).

Результаты расчета трудоемкости для этого варианта, представлены в табл. 5.6. Согласно этим подсчетам, на разработку программного обеспечения потребуется 28 дня. Количество рабочих дней в месяце – 21.

Таблица 5.6 – Перечень работ

№ п/п	Этапы работ	Трудоемкость, чел./дн		Продолжительность дн
		Руководитель профессор	Исполнитель доцент	
1	Разработка технического задания	2	2	2
2	Постановка задачи, выяснение ключевых целей проекта	2	2	2
3	Разработка алгоритма работы программы	3	4	4

Продолжение таблицы 5.6

№ п/п	Этапы работ	Трудоемкость, чел./дн		Продолжительность дн
		Руководитель профессор	Исполнитель доцент	
4	Разработка программного кода	4	8	8
5	Тестирование программы, проведение экспериментов	2	4	4
6	Доработка ПО	2	3	3
7	Окончательное тестирование и проведение экспериментов	2	2	2
8	Разработка документации	2	3	3
Итого:		19	28	28

Для проектирования и разработки программного продукта, понадобится использовать труд профессора руководителя (19 чел/дн), а также труд доцента исполнителя (28 чел/дн).

В конечном итоге, мы получили, что на разработку программного обеспечения нам потребуется команда, состоящая из двух человек: профессор (руководитель) и доцент, срок выполнения – 28 дней.

Стоимость необходимого оборудования, инструментов и материалов, а также их назначение и необходимое количество представлены в табл. 5.2 и табл. 5.3

Расходы на материалы и интрусенты для данного проекта составляют $M = 447,00$.

Заработная плата профессора (руководителя) и доцента состоит из основной и дополнительной заработной платы.

В разработке программного модуля принимали участия доцент и младший научный сотрудник. Разработка длилась 28 рабочих дня. Рабочий день – восьмичасовой.

Для данного проекта рассчитывается заработная плата профессора, оклад которого составляет 5200 грн/мес (21 рабочих дня в месяце) и доцента, оклад которого составляет 4950 грн/мес (табл. 5.7.).

Таблица 5.7 – Средняя месячная и дневная заработные платы

№ п/п	Должность	Заработная плата/ месяц (21 день), грн.	Заработная плата/ день, грн.
1	Профессор (руководитель)	5200	247,61
2	Доцент	4950	235,72

Основная заработная плата ($ЗП_{осн}$) состоит из суммы средней дневной заработной платы профессора и доцента, умноженной на их трудоемкость и рассчитывается следующим образом:

$$ЗП_{осн} = ЗП_{днпро} \cdot T_{про} + ЗП_{дндоч} \cdot T_{доц},$$

где $ЗП_{днпро}$, $ЗП_{дндоч}$ – дневная заработная плата профессора и доцента соответственно (таблица 5.7); $T_{про}$, $T_{доц}$ – трудоемкость доцента и младшего научного сотрудника (таблица 5.6).

Таким образом, основная заработная плата за месяц равна:

$$ЗП_{осн} = 247,61 \cdot 28 + 235,72 \cdot 28 = 6933,08 + 6600,16 = 13533,24 \text{ (грн)}.$$

Следовательно, основная заработная плата составляет 13533,24 грн.

Следовательно, по формуле (5.2) дополнительная заработная плата доцента равна:

$$ЗП_{доп доц} = 6600,16 \cdot 0,4 = 2640,06 \text{ (грн)}.$$

Дополнительная заработная плата профессора равна:

$$ЗП_{доп проф} = 6933,08 \cdot 0,7 = 4853,16 \text{ (грн)}.$$

Общая дополнительная заработная плата равна:

$$ЗП_{доп} = ЗП_{доп проф} + ЗП_{доп доц} = 4853,16 + 2640,06 = 7493,22 \text{ (грн)}.$$

Итого, общий фонд заработной платы составляет:

$$ЗП = ЗП_{осн} + ЗП_{доп} = 13533,24 + 7493,22 = 21026,46 \text{ (грн)}.$$

По формуле (5.4) начисления на заработную плату составляют:

$$ЗП_{соц} = 21026,46 \cdot 0,22 = 4625,82 \text{ (грн)}.$$

По формуле (5.5) годовая сумма амортизационных отчислений для ноутбука Lenovo IdeaPad Y480:

$$A_{ноутгод} = 19615,00 \cdot 0,25 = 4903,75 \text{ (грн)}.$$

Тогда, по формуле (5.6) величина амортизационных отчислений для ноутбука Lenovo IdeaPad Y480:

$$A_{\text{нбас}} = 4903,75 / 1976 = 2,48 \text{ (грн)}.$$

По формуле (5.7) величина амортизационных отчислений в зависимости от времени использования ноутбука Lenovo IdeaPad E10:

$$A_{\text{ноутП}} = 2,48 \cdot 224 = 555,52 \text{ (грн)}.$$

Следовательно, по формуле (5.5) годовая сумма амортизационных отчислений для принтера HP LaserJet P2035 (CE461A):

$$A_{\text{прингод}} = 5\,984,00 \cdot 0,25 = 1496 \text{ (грн)}.$$

Тогда, по формуле (5.6) величина амортизационных отчислений для принтера HP LaserJet P2035 (CE461A):

$$A_{\text{принчас}} = 1496 / 1976 = 0,76 \text{ (грн)}.$$

По формуле (5.7) величина амортизационных отчислений в зависимости от времени использования принтера HP LaserJet P2035 (CE461A):

$$A_{\text{принП}} = 0,76 \cdot 224 = 170,24 \text{ (грн)}.$$

По формуле (5.8) сумма амортизации по продукту:

$$A_{\text{П}} = 555,52 + 170,24 = 725,76 \text{ (грн)}.$$

Следовательно, по формуле (5.8) стоимость энергии ноутбука, при мощности 0,032 кВт:

$$S_{\text{Эноут}} = 1,6 \cdot 224 \cdot 0,032 = 11,47 \text{ (грн)}.$$

Стоимость энергии принтера, при мощности 0,11 кВт:

$$S_{\text{Эприн}} = 1,6 \cdot 224 \cdot 0,11 = 39,42 \text{ (грн)}.$$

Следовательно, стоимость технологической электроэнергии

$$S_{\text{Э}} = 11,47 + 39,42 = 50,89 \text{ (грн)}.$$

Стоимость осветительной электроэнергии равна $S = 1,6 \cdot 224 \cdot 0,1 = 35,84$ грн.

Из формулы (5.9) себестоимость составляет:

$$C = 13533,24 + 7493,22 + 4625,82 + 725,76 + 447,00 + 50,89 + 35,84 = 26911,77 \text{ (грн)}.$$

Калькуляционные статьи на разработку программного обеспечения представлены в таблице 5.8.

Таблица 5.8 – Статьи калькуляции на разработку программного продукта

№	Статья калькуляции	Расходы,
1	Материалы и инструменты	447,00
2	Основная заработная плата	13533,24
3	Дополнительная заработная плата	7493,22
4	Общая заработная плата	21026,46
5	Начисления на заработную плату	4625,82
6	Амортизация	725,76
7	Стоимость технологической	50,89
8	Стоимость осветительной электроэнергии	35,84
9	Общая себестоимость разработки	26911,77

Таким образом, себестоимость разработки при участии профессора и младшего научного сотрудника составляет 31072,8 грн, а при участии профессора и доцента себестоимость равна 26911,77 грн.

Отсюда следует, что с экономической точки зрения целесообразно использовать 2-й вариант, т.к. в этом случае издержки, связанные с выполнением работы, меньше.

Рассчитанные показатели дают полное представление о затратах, необходимых на разработку программы.

ЗАКЛЮЧЕНИЕ

В данной работе было осуществлены прием и обработки ADS-B сигнала с использованием RTL-SDR и AIRSPY приемники от самолетных ответчиков. Для приема использовалось оборудование кафедры 501, которое включает коллинеарную антенну, установленную на крыше радиокорпуса ХАИ, малошумящий усилитель, настроенный на частоту примерно 1090 МГц, недорогой SDR приемник RTL 2832 и хороший SDR приемник AIRSPY, подключенный к компьютеру. Для приема и обработки сигнала использовалась программа SDR#, а также среда математического моделирования MatLab осуществляющий обработки ADS-B сигнала.

В среде MatLab, решаются задачи обработки–детектирование, идентификация и декодирование принимаемого ADS-B сигнала, полученного при помощи приемники RTL-SDR и AIRSPY. Кроме того, проведен оценка помехоустойчивости процедур обнаружения и декодирования ADS-B-данных.

В ходе работы над детектировании, был предложен энергетический обнаружитель, который имеет гораздо большую энергию (длительность - 100 – 200 отсчетов), и вероятность ее правильного обнаружения может быть гораздо выше, по сравнению с обнаружением по преамбуле.

Суть идентификации ADS-B сигнала заключается в различии между различными длинами сквиттера. Из-за различных длин информации привела к энергии различной.

При декодирования ADS-B сообщении, Используя три различных метода были точно такие же результаты. В то время как использование алгоритма CRC для проверки. Это является еще одним доказательством правильности алгоритма. И для декодирования различных типах сигнала используются различные алгоритмы : Идентификация летательного аппарата, Определение координат и высоты ЛА, Определение вектора скорости ЛА и т.д.

Результаты моделирования в ходе оценка помехоустойчивости процедур обнаружения и декодирования ADS-B-данных характеризуют различные свойства шумозащитных алгоритма.

Таким образом, основные результаты работы над данным проектом могут быть использованы в учебном процессе при исследовании стандарта ADS-B.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по авиационному наблюдению:Дос 9924 AN/474 ,AN10-4 ИКАО -Международная организация гражданской авиации ,2012. -320 с.
2. 张蔚, 徐炎祥/二次雷达原理/ 37 стр.
3. Flightradar24 / [Электронный ресурс] – Режим доступа: <https://www.flightradar24.com/48.94,32.37/5> – 24.03.2016 г.
4. RADARSPORTTERS TEAM/ADSBRADAR [Электронный ресурс] – Режим доступа: <http://adsbradar.ru/adsb-technology>
5. GitHub/Code ADS-B Decoding Guide [Электронный ресурс] –Режим доступа: <http://adsb-decode-guide.readthedocs.io/en/latest/introduction.html#ads-b>
6. Руководство по специальным услугам режима S :Дос 9688 AN/952. - Международная организация гражданской авиации,2012.-27с
- 7 Руководство по вторичным обзорным радиолокационным :Дос 9684 AN/951 ИКАО. /245стр (ВОРЛ) системам.-Международная организация гражданской авиации, 2004. -256 с.
8. E4000 Datasheet 4v0 [Электронный ресурс] / –Режим доступа: www.elonics.com
- 9.Компания RTL-SDR/ RTL [Электронный ресурс] –Режим доступа:<http://www.rtl-sdr.com/about-rtl-sdr/>
10. Компания Airspy/ AIRSPY [Электронный ресурс] –Режим доступа: <http://airspy.com/>
11. Электроника Радиотехника Радиолобительство/ MyHamRadio[Электронный ресурс] –Режим доступа:<http://myhamradio.ru/sdr/sravnenie-sdr-priyomnikov-airspy-sdrplay-i-hackrf-chast-1.html>
12. Limor "Ladyada" Fried MIT engineer/ adafruit [Электронный ресурс]–Режим доступа: <https://learn.adafruit.com/getting-started-with-rtl-sdr-and-sdrsharp/download-and-install-software>
13. Левин Б. Р. Теоретические основы статистической радиотехники: Уч.пособ.перераб. и доп. —М.: Радио и связь, 1989. - 656 с.
14. Борисов В.И. Помехозащищенность систем радиосвязи с расширением спектра сигналов методом псевдослучайной перестройки рабочей частоты: Уч.пособ.- М.: Радио и связь, 2000. - 384 с.
- 15.Википедия[Электронный ресурс]–Режим доступа: https://ru.wikipedia.org/wiki/Среднеквадратическое_отклонение
- 16.Компания MATLAB/ mathworks [Электронный ресурс] –Режим доступа:<http://cn.mathworks.com/help/matlab/ref/corrcoef.html>
17. E. Chnag, R. Hu, D. Lai, R. Li, Q. Scott, T. Tyan (2000)/The Story of Mode S: An Air Traffic Control Data-Link Technology, Massachusetts Institute of Technology (MIT), Cambridge, MA.
- 18.Шульгин В.И. ОСНОВЫ ТЕОРИИ ЦИФРОВОЙ СВЯЗИ:Уч.пособ.-Харьков: “ХАИ”,2008 . -166с.
19. RADARSPORTTERS TEAM/ADSBRADAR [Электронный ресурс]/ – Режим доступа: <http://adsb-decode-guide.readthedocs.io/en/latest/introduction.html>

ПРИЛОЖЕНИЕ А

Код программы приема сигналов

```

clc;
clear all;
close all;

N1=11000000;

[y,Fs] =
wavread('SDRSharp_20160506_11235
1Z_1090000kHz_IQ.wav',N1);
Ta = 1;
dt = Ta/Fs;

Ns = 1;
Len = 10000000;

Nf = Ns+Len;
T = Len*dt;
y = y';

y1 = y(1,Ns:Nf);
y2 = y(2,Ns:Nf);

Y = y1+1i*y2;
Y=abs(Y);

t=0:1/Fs:((length(Y)-1)/Fs);

figure(1);
subplot(3,1,1);
plot(t,y1);
ylabel('AMP');xlabel('t,c')
title('Реальная часть - Re');

subplot(3,1,2);
plot(t,y2);
ylabel('AMP');xlabel('t,c')
title('Мнимая часть- Im');

subplot(3,1,3);
plot(t,Y);
ylabel('AMP');xlabel('t,c')
title('Модель (Y)');

```


ПРИЛОЖЕНИЕ Б

Код программы декодирования сигналов (в двоичном формате)

```

clc;
clear all;
close all;
N1=2800000;
% load('RTL.mat','y');
[y,Fs] =
wavread('SDRSharp_20160427_08523
0Z_1090000kHz_IQ.wav',[1 1*N1]);
y=resample(y,10000000,Fs);
Fs=10000000;

% save('RTL3.mat','y')
Ta = 1;
dt = Ta/Fs;

Ns = 1;
Len = 10e6;

Nf = Ns+Len-1;
T = Len*dt;
df = 1/(2*T);
y = y';

y1 = y(1,Ns:Nf);
y2 = y(2,Ns:Nf);
Z = zeros (size(y1));

Y = y1+1i*y2;
Y=abs(Y);

[b,a]=butter(2,500/1000,'low');
Y=filtfilt(b,a,Y);

Z=mean(Y);
Y=Y-Z;
DEV=std(Y);
Y=Y-DEV/3;
%Y(Y<0)=0;

%Задание временных отсчетов
t=0:1/Fs:((length(Y)-1)/Fs);
figure(1);
subplot(6,1,1);
plot(t,Y);
ylabel('AMP');xlabel('t,c')
title('Сигнал - среднее и дисперсия');

t_d=0:1/Fs:((length(Y)-1)/Fs);

DF_11(560:1120)=1;
DF_11(1,560)=0;
DF_17(1:1120)=1;

Y_1=conv(Y,transpose(DF_11));
Y_2=conv(Y,transpose(DF_17));
fprintf('Выполняется сверка.... : \n');

t_t1=0:1/Fs:((length(Y_1)-1)/Fs);
t_t2=0:1/Fs:((length(Y_2)-1)/Fs);

subplot(6,1,2);
plot(t_t1,Y_1);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на сквиттер
DF11');
subplot(6,1,3);
plot(t_t2,Y_2);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на сквиттер
DF17');

h_t=[1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1 1];
h_t=h_t-mean(h_t);

Z=conv(Y,transpose(h_t));

```

```

t_kor=0:1/Fs:((length(Z)-1)/Fs);

subplot(6,1,4);
plot(t_kor,Z);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на преамбулу');
DF17=Y_2;
DF11=Y_1;

subplot(6,1,5);
plot(t_t2,DF11);
ylabel('AMP');xlabel('t,c')
title('Превышение порога
обнаружения DF11');

subplot(6,1,6);
plot(t_t2,DF17);
ylabel('AMP');xlabel('t,c')
title('Превышение порога
обнаружения DF17');

k=1;kk=1;
while k<length(Y)
    k=k+1;
    if DF11(k)>10
        DF11C=DF11((k-
0.0002*Fs):(k+0.0002*Fs));
        DF17C=DF17((k-
0.0002*Fs):(k+0.0002*Fs));
        s1=find(DF11C==max(DF11C));
        s2=find(DF17C==max(DF17C));
        detaDF=max(DF17C)-
1.5*max(DF11C);
        if mean(mean(detaDF)>0)
            display('суммар DF17 ');
            k=min(s2+k-0.0002*Fs);k;
%         display(k/Fs);
        else display('суммар DF11');
            k=min(s1+k-0.0002*Fs);
        end
        num(kk)=k;
        kk=kk+1;
        k=k+0.0002*Fs;
    end
end
display(num);
display(num/Fs);

for j=1:length(num)
    preamble=Z(num(j)-1200:num(j)-
1120);

    start(j)=find((preamble==max(preamb
le)))+num(j)-1200+30;
end
display(start);
%%
% for q=1:numel(start)
q=1;
error=20; % возможное отклонение
YY=Y(start(q)-80:num(q)+error);
figure
plot(YY);
ylabel('AMP');xlabel('t,c')
title('Полезный сигнал - среднее и
дисперсия');

YY((YY-mean(YY))>0)=1;
YY((YY-mean(YY))<0)=0;
figure
plot(YY);
ylabel('AMP');xlabel('t,c')
title('Полезный сигнал - среднее и
дисперсия');
ylim([0 1.5])
yy=YY(80:length(YY));
figure
plot(yy)
ylim([0 1.5])

%%
for en=1:length(yy)-1
    detayy=yy(en+1)-yy(en);
    if abs(detayy)>0.5
        ed=en+5;
    end
end

```

```

end
bit0=1;
if yy(1)==1
    bit0=1;
else
    for b=1:10
        detayy=yy(b+1)-yy(b);
        if detayy>0.5
%    display(b)
            if b<4
                bit0=b;
            elseif b>5
                bit0=b-5;
            end
        end
    end
end
%%%
% метод1:метод с помощью
% функции

%
% clear bit;bit0=bit0+1;
% S1=[1 1 1 1 1 0 0 0 0 0];S0=[0 0 0
0 0 1 1 1 1 1];
% for i=bit0:10:ed-9
%
sum11=0;sum01=0;sum12=0;sum02=
0;sum13=0;sum03=0;
% for j=i:i+9
%    sum01=power((yy(j)-S0(j-
i+1)),2)+sum01; sum02=power((yy(j-
1)-S0(j-i+1)),2)+sum02;
sum03=power((yy(j+1)-S0(j-
i+1)),2)+sum03;
%    sum11=power((yy(j)-S1(j-
i+1)),2)+sum11;sum12=power((yy(j-
1)-S1(j-
i+1)),2)+sum12;sum13=power((yy(j+
1)-S1(j-i+1)),2)+sum13;
% end

%
aa=[sum01,sum02,sum03];bb=[sum11
,sum12,sum13];
%    aaa=min(aa);bbb=min(bb);
%    if aaa<bbb
%        bit((i+10-bit0)/10)=0;
%    else
%        bit((i+10-bit0)/10)=1;
%    end
% end
% bit=num2str(bit);
% display(bit)
%%

% метод2:Метод с помощью
% функции corrcoef
%
% S1=[1 1 1 1 1 0 0 0 0 0];S0=[0 0 0
0 0 1 1 1 1 1];
% clear bit;
% for i=bit0:10:ed-9
%    if
sum(sum(corrcoef([yy(i:i+9)',S1'])))>s
um(sum(corrcoef([yy(i:i+9)',S0'])))
%        bit((i+10-bit0)/10)=1;
%    else
%        bit((i+10-bit0)/10)=0;
%    end
% end
% bit=num2str(bit);
% display(bit)
% save([num2str(q),'.mat'],'bit');
% end
%%

% метод3:Импульсный Метод
%
clear bit;bit0=bit0+1;
for i=bit0:10
    detayy=yy(i)-yy(i-1);
    if detayy>0.5
        if i-bit0<2
            bit(1)=1;
        else bit(1)=0;
    end
end

```

```

        end
    end
end
%
T=5;a=2;
for i=bit0:10:(ed-9)
    for j=i+7:i+19
        detayy=yy(j+1)-yy(j);
        if abs(detayy)>0.5
            if bit(a-1)==1
                if ((j-i-5)/T)<1.5
%           0.7<((j-i-5)/T) & ((j-i-5)/T)<1.3
                    bit(a)=1;
                else
% 0.7<((j-i-5)/T/2) & ((j-i-5)/T/2)<1.3
                    bit(a)=0;
                end
            elseif bit(a-1)==0
                if ((j-i-5)/T)<1.5

```

```

%           0.7<((j-i-5)/T) & ((j-i-5)/T)<1.3
                    bit(a)=0;
                else
%           0.7<((j-i-5)/T/2) & ((j-i-5)/T/2)<1.3
                    bit(a)=1;
                end
            break;
        end
    break;
    end
end
a=a+1;
end
bit=num2str(bit);
display(bit);
% %

```

ПРИЛОЖЕНИЕ В

Оценка помехоустойчивости процедур декодирования ADS-B-данных

```

clc;
clear all;
close all;
%
[y,Fs] =
wavread('SDRSharp_20140312_12270
1Z_1090000kHz_IQ.wav',[0.2189*240
0000 0.2191*2400000]);
% y = awgn(x,SNR) %Добавляет
белый гауссов шум к вектору сигнала
x.
%Скаляр snr задает отношение
сигнал/шум в децибелах.
%Если значения x являются
комплексными, функция awgn
добавляет комплексный шум.

y=resample(y,10000000,Fs);
Fs=10000000;

Ta = 1;
dt = Ta/Fs;

Ns = 1;
Len =0.002e6;

Nf = Ns+Len;
T = Len*dt;
df = 1/(2*T);
y = y';

y1 = y(1,Ns:Nf);
y2 = y(2,Ns:Nf);
Z = zeros (size(y1));

Y = y1+1i*y2;
Y=abs(Y);

%Задание временных отсчетов
t=0:1/Fs:((length(Y)-1)/Fs);

figure;
plot(t,Y);
ylabel('AMP');xlabel('t,c')
title('Сигнал - среднее и дисперсия');

t_d=0:1/Fs:((length(Y)-1)/Fs);

DF_17(1:1120)=1;

Y_1=conv(Y,transpose(DF_17));
fprintf('Выполняется сверка.... : \n');

t_t1=0:1/Fs:((length(Y_1)-1)/Fs);

figure
subplot(3,1,1);
plot(t_t1,Y_1);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на сквиттер DF17');

h_t=[1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1 1];
h_t=h_t-mean(h_t);

Z=conv(Y,transpose(h_t));

t_kor=0:1/Fs:((length(Z)-1)/Fs);

subplot(3,1,2);
plot(t_kor,Z);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на преамбулу');

DF17=Y_1;

```

```

subplot(3,1,3);
plot(t_t1,DF17);
ylabel('AMP');xlabel('t,c')
title('Превышение порога
обнаружения DF17');

k=1;kk=1;
while k<length(Y)
    k=k+1;
    if DF17(k)>20
        DF17C=DF17(k:(k+0.0002*Fs));
        s1=find(DF17C==max(DF17C));
        k=min(s1+k);k;
        break
    end
end
num(kk)=k;
display(num);
display(num/Fs);

for j=1:length(num)
    preamble=Z(num(j)-1200:num(j)-
1120);

    start(j)=find((preamble==max(preamb
le)))+num(j)-1200+30;
end
display(start);
%%

q=1;
YY=Y(start(q)-80:num(q)+20);

YYmax=max(YY);
figure
plot(YY(81:length(YY)));
ylabel('AMP');xlabel('t,c')
title('Полезный сигнал - среднее и
дисперсия');
YY((YY-mean(YY))>0)=1;
YY((YY-mean(YY))<0)=0;
figure

```

```

plot(YY);
ylabel('AMP');xlabel('t,c')
title('Полезный сигнал - среднее и
дисперсия');
ylim([0 1.5])
yy=YY(80:length(YY));
figure
plot(yy)
ylim([0 1.5])

%%
for en=1:length(yy)-1
    detayy=yy(en+1)-yy(en);
    if abs(detayy)>0.5
        ed=en+5;
    end
end

bit0=1;
if yy(1)==1
    bit0=1;
else
    for b=1:10
        detayy=yy(b+1)-yy(b);
        if detayy>0.5
            % display(b)
            if b<4
                bit0=b;
            elseif b>5
                bit0=b-5;
            end
        end
    end
end

end

%%
% %min CKO

%
clear bit;bit0=bit0+1;
S1=[1 1 1 1 1 0 0 0 0 0 ];S0=[ 0 0 0 0 0
1 1 1 1 1 ];
for i=bit0:10:ed-9

```

```

sum11=0;sum01=0;sum12=0;sum02=
0;sum13=0;sum03=0;
    for j=i:i+9
        sum01=power((yy(j)-S0(j-
i+1)),2)+sum01; sum02=power((yy(j-
1)-S0(j-i+1)),2)+sum02;
sum03=power((yy(j+1)-S0(j-
i+1)),2)+sum03;
        sum11=power((yy(j)-S1(j-
i+1)),2)+sum11;sum12=power((yy(j-
1)-S1(j-
i+1)),2)+sum12;sum13=power((yy(j+
1)-S1(j-i+1)),2)+sum13;
    end

aa=[sum01,sum02,sum03];bb=[sum11
,sum12,sum13];
    aaa=min(aa);bbb=min(bb);
    if aaa<bbb
        bit((i+10-bit0)/10)=0;
    else
        bit((i+10-bit0)/10)=1;
    end
end
bit=num2str(bit);
display(bit)
bit_bin=str2num(bit);
%% Do CRC computation, only analyze
DF 17, since their CRC are
generator = [1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 0 0 0 0 0 0 1 0 0 1];
[divid,
remainder]=deconv(bit_bin,generator);
remainder=mod(remainder(end-
23:end),2);
if isequal(remainder,[0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0]);
    disp(['CRC                right      !
[' ,num2str(bit_bin(1:88)),']']);
else
    disp(['CRC
error: [' ,num2str(remainder),']']);
end

```

```

BIT_right=[bit_bin bit_bin bit_bin
bit_bin bit_bin];
%% Add Noise
y1=(y1(start(1):start(1)+ed-2))/YYmax;
y2=y2(start(1):start(1)+ed-2)/YYmax;
y1=repmat(y1,1,5);y2=repmat(y2,1,5);
%эпзодичеокй процесс
Sigma=[0.001 0.003 0.005 0.007
0.009 0.01 0.02 0.03 0.05 0.055 0.06
0.07 0.08 0.09 0.1 0.15 0.2 0.25 0.3 0.35
0.4 0.5 0.6 0.8 0.9 1.0]
for m=1:26
    mu=0;sigma=Sigma(m);

n1=normrnd(mu,sigma,[1,size(y1,2)]);
n2=normrnd(mu,sigma,[1,size(y2,2)]);

[b,a]=butter(2,1/3,'low');
n1=filtfilt(b,a,n1);
n2=filtfilt(b,a,n2);

mean_n1=mean(abs(n1));
mean_n2=mean(abs(n2));
mean_n=mean_n1+mean_n1;

y1=y1+n1;y2=y2+n2;

%y1=filtfilt(b,a,y1);
%y2=filtfilt(b,a,y2);

Yn = y1+1i*y2;Yn=abs(Yn) -
0.75*mean_n;

% Yn(find(Yn<0.05))=0;
t=0:1/Fs:((length(Yn)/5-1)/Fs);
figure
plot(t,Yn(1:length(Yn)/5));
ylabel('AMP');xlabel('t,c')
title('Сигнал с шумом - среднее и
дисперсия'); % signal with noise
len=size(Yn,2);
Umax=max(Y);
l0=1/8*Umax;
Yn=Yn-l0;

```



```

end

% bit=num2str(bit);
% display(bit)
BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error))
disp(['The number of errors'
=' ,num2str(error)]);
p2(m)=error/length(BIT_right);
%%
% метод3:Импульсный Метод

clear bit BIT_error;

Yn((Yn-mean(Yn))>0)=1;
Yn((Yn-mean(Yn))<0)=0;

yy=Yn;

figure
plot(Yn);

yy3=[yy,zeros(1,50)];
bit=zeros(1,560);
bit(1)=1;
T=5;a=2;tn=5;
for i=1:10:(length(yy)-9)
    for j=i+7:i+19
        detayy=yy3(j+1)-yy3(j);
        if abs(detayy)>0.5
            if bit(a-1)==1
                if ((j-tn)/T)<1.5
%           0.8<((j-i-5)/T) & ((j-i-5)/T)<1.2
                    bit(a)=1; tn=j+5;
                end
            end
        end
    end
end
BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error))
disp(['The number of errors'
=' ,num2str(error)]);
p3(m)=error/length(BIT_right);
%%
end
close all;
figure
plot(Sigma,p1,'b')
hold on
plot(Sigma,p2,'r')
hold on

```

```

plot(Sigma,p3,'g')
ylim([0 1])
title({'зависимость      вероятность
ошибки';' и мощность шума \sigma
при декодировании'});

```

```

ylabel('вероятность
ошибки');xlabel(' мощность шума
\sigma')
legend('Метод с помощью функции
corrcoef','min      СКО','Импульсный
Метод')

```

ПРИЛОЖЕНИЕ Г

Сравнение качества декодирования данных ADS-B при использовании приемников RTL-SDR и ARSPY

```

clc;
clear all;
close all;
%
[y,Fs] =
wavread('SDRSharp_20160506_11235
8Z_1090379kHz_IQ.wav',[0.4366*100
00000 0.4370*10000000]);

Ta = 1;
dt = Ta/Fs;

Ns = 1;
Len = 0.0002e7;

Nf = Ns+Len-1;
T = Len*dt;
df = 1/(2*T);
y = y';

y1 = y(1,Ns:Nf);
y2 = y(2,Ns:Nf);
Z = zeros (size(y1));

Y = y1+1i*y2;
Y=abs(Y);

Y(find(Y<0.005))=0;
%Задание временных отсчетов

t=0:1/Fs:((length(Y)-1)/Fs);

figure;
plot(t,Y);
ylabel('AMP');xlabel('t,c')
title('Сигнал - среднее и дисперсия');

t_d=0:1/Fs:((length(Y)-1)/Fs);

DF_17(1:1120)=1;

Y_1=conv(Y,transpose(DF_17));
fprintf('Выполняется сверка.... : \n');

h_t=[1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1 1];
h_t=h_t-mean(h_t);

Z=conv(Y,transpose(h_t));

DF17=Y_1;
k=1;kk=1;
while k<length(Y)
    k=k+1;
    if DF17(k)>10
        DF17C=DF17(k:(k+0.0001*Fs));
        s1=find(DF17C==max(DF17C));
        k=min(s1+k);k;
        break
    end
end
num(kk)=k;
display(num);
display(num/Fs);

for j=1:length(num)
    preamble=Z(num(j)-1200:num(j)-
1120);

start(j)=find((preamble==max(preamb
le)))+num(j)-1200+30;
end
display(start);
%%
q=1;
YY=Y(start(q)-80:num(q)+20);

```

```

YYmax=max(YY);
figure
plot(YY(81:length(YY)));
ylabel('AMP');xlabel('t,c')
title('Полезный сигнал - среднее и
дисперсия');
YY((YY-mean(YY))>0)=1;
YY((YY-mean(YY))<0)=0;
figure
plot(YY);
ylabel('AMP');xlabel('t,c')
title('Полезный сигнал - среднее и
дисперсия');
ylim([0 1.5])
yy=YY(80:length(YY));
figure
plot(yy)
ylim([0 1.5])

%%
for en=1:length(yy)-1
    detayy=yy(en+1)-yy(en);
    if abs(detayy)>0.5
        ed=en+5;
    end
end

bit0=1;
if yy(1)==1
    bit0=1;
else
    for b=1:10
        detayy=yy(b+1)-yy(b);
        if detayy>0.5
%    display(b)
            if b<4
                bit0=b;
            elseif b>5
                bit0=b-5;
            end
        end
    end
end

%%
% %min CKO
%
clear bit;bit0=bit0+1;
S1=[1 1 1 1 1 0 0 0 0 0 ];S0=[ 0 0 0 0
0 1 1 1 1 1 ];
for i=bit0:10:ed-9

    sum11=0;sum01=0;sum12=0;sum02=
0;sum13=0;sum03=0;
    for j=i:i+9
        sum01=power((yy(j)-S0(j-
i+1)),2)+sum01; sum02=power((yy(j-
1)-S0(j-i+1)),2)+sum02;
        sum03=power((yy(j+1)-S0(j-
i+1)),2)+sum03;
        sum11=power((yy(j)-S1(j-
i+1)),2)+sum11;sum12=power((yy(j-
1)-S1(j-
i+1)),2)+sum12;sum13=power((yy(j+
1)-S1(j-i+1)),2)+sum13;
    end

    aa=[sum01,sum02,sum03];bb=[sum11
,sum12,sum13];
    aaa=min(aa);bbb=min(bb);
    if aaa<bbb
        bit((i+10-bit0)/10)=0;
    else
        bit((i+10-bit0)/10)=1;
    end
end
bit=num2str(bit);
display(bit)
bit_bin=str2num(bit);
%% Do CRC computation, only
analyze DF 17, since their CRC are
generator = [1 1 1 1 1 1 1 1 1 1 1 1
0 1 0 0 0 0 0 0 1 0 0 1];
[divid,
remainder]=deconv(bit_bin,generator)
;

```

```

remainder=mod(remainder(end-
23:end),2);
if isequal(remainder,[0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0]);
    disp(['CRC right !
',num2str(bit_bin(1:88)),']');
else disp(['CRC
error:',num2str(remainder),']');
end
BIT_right=[bit_bin bit_bin bit_bin
bit_bin bit_bin];
%% Add Noise
y1=(y1(start(1):start(1)+ed-
2))/YYmax;
y2=y2(start(1):start(1)+ed-2)/YYmax;
y1=repmat(y1,1,5);y2=repmat(y2,1,5);
%эргодической процесс
Sigma=[0 0.001 0.003 0.005 0.007
0.009 0.01 0.02 0.03 0.05 0.055 0.06
0.07 0.08 0.09 0.1 0.15 0.2 0.25 0.3
0.35 0.4 0.5 0.6 0.8 0.9 1.0];
for m=1:27
mu=0;sigma=Sigma(m);

n1=normrnd(mu,sigma,[1,size(y1,2)]);
n2=normrnd(mu,sigma,[1,size(y2,2)]);

[b,a]=butter(2,1/2,'low');
n1=filtfilt(b,a,n1);
n2=filtfilt(b,a,n2);

mean_n1=mean(abs(n1));
mean_n2=mean(abs(n2));
mean_n=mean_n1+mean_n1;

y1=y1+n1;y2=y2+n2;

y1=filtfilt(b,a,y1);
y2=filtfilt(b,a,y2);

Yn = y1+1i*y2;Yn=abs(Yn) -
0.75*mean_n;

% Yn(find(Yn<0.05))=0;

```

```

t=0:1/Fs:((length(Yn)/5-1)/Fs);
len=size(Yn,2);
Umax=max(Y);
l0=1/8*Umax;
Yn=Yn-l0;
%Задание временных отсчетов

t_d=0:1/Fs:((length(Yn)/5-1)/Fs);

DF_17(1:1120)=1;

Y_1=conv(Yn(1:length(Yn)/5),transpose(DF_17));

Z=conv(Yn,transpose(h_t));

t_kor=0:1/Fs:((length(Z)-1)/Fs);

%Yn((Yn-mean(Yn))>0)=1;
%Yn((Yn-mean(Yn))<0)=0;

yy=Yn;
%%
% Метод с помощью функции
corrcoef
S1=[1 1 1 1 1 0 0 0 0 0 ];S0=[ 0 0 0 0
0 1 1 1 1 1 ];
clear bit;
for i=1:10:length(yy)-9
    if
        sum(sum(corrcoef([yy(i:i+9)',S1'])))>sum(sum(corrcoef([yy(i:i+9)',S0'])))
        bit((i+10-1)/10)=1;
    else
        bit((i+10-1)/10)=0;
    end
end
% bit=num2str(bit);
% display(bit)

```

```

BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error));
p1(m)=error/length(BIT_right);
%%
%min CKO
clear bit BIT_error;
yy2=[0,yy,0];
for i=2:10:length(yy2)-10

sum11=0;sum01=0;sum12=0;sum02=
0;sum13=0;sum03=0;
    for j=i:i+9
        sum01=power((yy2(j)-S0(j-
i+1)),2)+sum01; sum02=power((yy2(j-
1)-S0(j-i+1)),2)+sum02;
        sum03=power((yy2(j+1)-S0(j-
i+1)),2)+sum03;
        sum11=power((yy2(j)-S1(j-
i+1)),2)+sum11;sum12=power((yy2(j-
1)-S1(j-
i+1)),2)+sum12;sum13=power((yy2(j
+1)-S1(j-i+1)),2)+sum13;
    end

aa=[sum01,sum02,sum03];bb=[sum11
,sum12,sum13];
    aaa=min(aa);bbb=min(bb);
    if aaa<bbb
        bit((i+10-2)/10)=0;
    else
        bit((i+10-2)/10)=1;
    end
end

```

```

% bit=num2str(bit);
% display(bit)
BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error));
p2(m)=error/length(BIT_right);
%%
% метод3:Импульсный Метод

clear bit BIT_error;

Yn((Yn-mean(Yn))>0)=1;
Yn((Yn-mean(Yn))<0)=0;

yy=Yn;

figure
plot(Yn);

yy3=[yy,zeros(1,50)];
bit=zeros(1,560);
bit(1)=1;
T=5;a=2;tn=5;
for i=1:10:(length(yy)-9)
    for j=i+7:i+19
        detayy=yy3(j+1)-yy3(j);
        if abs(detayy)>0.5
            if bit(a-1)==1
                if ((j-tn)/T)<1.5
                    % 0.8<((j-i-5)/T) & ((j-i-5)/T)<1.2
                    bit(a)=1; tn=j+5;
                else
                    % 0.8<((j-i-5)/T/2) & ((j-i-5)/T/2)<1.2
                    bit(a)=0;tn=j;
                end
            end
        end
    end
end

```

```

elseif bit(a-1)==0
    if ((j-tn)/T)<1.5
%       0.8<((j-i-5)/T) & ((j-i-
5)/T)<1.2
        bit(a)=0;tn=j+5;
    else
%       0.8<((j-i-5)/T/2) & ((j-i-
5)/T/2)<1.2
        bit(a)=1;tn=j;
    end
    break;
end
break;
end
end
a=a+1;
end
BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error));
p3(m)=error/length(BIT_right);
%%
end
save airspyerror.mat p1 p2 p3
clear all;close all;
[y,Fs] =
wavread('SDRSharp_20160506_11235
1Z_1090000kHz_IQ.wav',[0.4495*280
0000 0.4498*2800000]);
y=resample(y,10000000,Fs);
Fs=10000000;
Ta = 1;
dt = Ta/Fs;
Ns = 1;
Len =0.0003*10e6;

```

```

Nf = Ns+Len-1;
T = Len*dt;
df = 1/(2*T);
y = y';

y1 = y(1,Ns:Nf);
y2 = y(2,Ns:Nf);
Z = zeros (size(y1));

Y = y1+1i*y2;
Y=abs(Y);

Y(find(Y<0.001))=0;
%Задание временных отсчетов

t=0:1/Fs:((length(Y)-1)/Fs);

figure;
plot(t,Y);
ylabel('AMP');xlabel('t,c')
title('Сигнал - среднее и дисперсия');

t_d=0:1/Fs:((length(Y)-1)/Fs);

DF_17(1:1120)=1;

Y_1=conv(Y,transpose(DF_17));

h_t=[1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 0 0 0 0 0 1 1 1 1 1];
h_t=h_t-mean(h_t);

Z=conv(Y,transpose(h_t));
DF17=Y_1;

k=1;kk=1;
while k<length(Y)
    k=k+1;
    if DF17(k)>10
        DF17C=DF17(k:(k+0.0001*Fs));
        s1=find(DF17C==max(DF17C));

```

```

        k=min(s1+k);k;
        break
    end
    end
    num(kk)=k;
    display(num);
    display(num/Fs);

    for j=1:length(num)
        preamble=Z(num(j)-1200:num(j)-
            1120);

        start(j)=find((preamble==max(preamb
            le)))+num(j)-1200+30;
        end
        display(start);
        %%

        q=1;
        YY=Y(start(q)-80:num(q)+20);

        YYmax=max(YY);
        YY((YY-mean(YY))>0)=1;
        YY((YY-mean(YY))<0)=0;
        yy=YY(80:length(YY));

        %%
        for en=1:length(yy)-1
            detayy=yy(en+1)-yy(en);
            if abs(detayy)>0.5
                ed=en+5;
            end
        end

        bit0=1;
        if yy(1)==1
            bit0=1;
        else
            for b=1:10
                detayy=yy(b+1)-yy(b);
                if detayy>0.5
                    %    display(b)

                    if b<4
                        bit0=b;
                    elseif b>5
                        bit0=b-5;
                    end
                end
            end
        end
    end

    %%
    % %%min CKO

    %
    S1=[1 1 1 1 1 0 0 0 0 0 ];S0=[ 0 0 0 0
    0 1 1 1 1 1 ];
    clear bit;
    for i=1:10:length(yy)-9
        if
            sum(sum(corrcoef([yy(i:i+9)',S1'])))>s
            um(sum(corrcoef([yy(i:i+9)',S0'])))
                bit((i+10-1)/10)=1;
            else
                bit((i+10-1)/10)=0;
            end
        end
        bit=num2str(bit);
        display(bit)
        bit_bin=str2num(bit);
        %% Do CRC computation, only
        analyze DF 17, since their CRC are
        generator = [1 1 1 1 1 1 1 1 1 1 1 1
        0 1 0 0 0 0 0 0 1 0 0 1];
        [divid,
        remainder]=deconv(bit_bin,generator)
        ;
        remainder=mod(remainder(end-
        23:end),2);
        if isequal(remainder,[0 0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0]);
            disp(['CRC right !
            [' ,num2str(bit_bin(1:88)),']']);
        else disp(['CRC
        error:[' ,num2str(remainder),']']);
        end
    end

```



```

BIT_right=[bit_bin bit_bin bit_bin
bit_bin bit_bin];
%% Add Noise
y1=(y1(start(1):start(1)+ed-
1))/YYmax;
y2=y2(start(1):start(1)+ed-1)/YYmax;
y1=repmat(y1,1,5);y2=repmat(y2,1,5);
%эргодичекой процесс
Sigma=[0 0.001 0.003 0.005 0.007
0.009 0.01 0.02 0.03 0.05 0.055 0.06
0.07 0.08 0.09 0.1 0.15 0.2 0.25 0.3
0.35 0.4 0.5 0.6 0.8 0.9 1.0];
for m=1:27
mu=0;sigma=Sigma(m);
% mu=0;
n1=normrnd(mu,sigma,[1,size(y1,2)]);
n2=normrnd(mu,sigma,[1,size(y2,2)]);
%
% [b,a]=butter(2,1/3,'low');
% n1=filtfilt(b,a,n1);
% n2=filtfilt(b,a,n2);

mean_n1=mean(abs(n1));
mean_n2=mean(abs(n2));
mean_n=mean_n1+mean_n1;

y1=y1+n1;y2=y2+n2;

Yn = y1+1i*y2;Yn=abs(Yn) -
0.75*mean_n;

t=0:1/Fs:((length(Yn)/5-1)/Fs);
figure
plot(t,Yn(1:length(Yn)/5));
ylabel('AMP');xlabel('t,c')
title('Сигнал с шумом - среднее и
дисперсия'); % signal with noise
len=size(Yn,2);
l0=1/8*YYmax;
Yn=Yn-l0;
%Задание временных отсчетов
t_d=0:1/Fs:((length(Yn)/5-1)/Fs);

```

```

DF_17(1:1120)=1;

Y_1=conv(Yn(1:length(Yn)/5),transpos
e(DF_17));

t_t1=0:1/Fs:((length(Y_1)-1)/Fs);

figure
plot(t_t1,Y_1);
ylabel('AMP');xlabel('t,c')
title('Отклик СФ на сквиптер
DF17');
Z=conv(Yn,transpose(h_t));
%
% Yn((Yn-mean(Yn))>0)=1;
% Yn((Yn-mean(Yn))<0)=0;
yy=Yn;

%%
% Метод с помощью функции
corrcoef
S1=[1 1 1 1 1 0 0 0 0 0 ];S0=[ 0 0 0 0
0 1 1 1 1 1 ];
clear bit;
for i=1:10:length(yy)-9
if
sum(sum(corrcoef([yy(i:i+9)',S1'])))>s
um(sum(corrcoef([yy(i:i+9)',S0'])))
bit((i+10-1)/10)=1;
else
bit((i+10-1)/10)=0;
end
end
% bit=num2str(bit);
% display(bit)
BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
d= BIT_right(i)-BIT_error(i);

```

```

    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error));
p11(m)=error/length(BIT_right);
%%
%min CKO
clear bit BIT_error;
yy2=[0,yy,0];
for i=2:10:length(yy2)-10

    sum11=0;sum01=0;sum12=0;sum02=
0;sum13=0;sum03=0;
    for j=i:i+9
        sum01=power((yy2(j)-S0(j-
i+1)),2)+sum01; sum02=power((yy2(j-
1)-S0(j-i+1)),2)+sum02;
        sum03=power((yy2(j+1)-S0(j-
i+1)),2)+sum03;
        sum11=power((yy2(j)-S1(j-
i+1)),2)+sum11;sum12=power((yy2(j-
1)-S1(j-
i+1)),2)+sum12;sum13=power((yy2(j
+1)-S1(j-i+1)),2)+sum13;
    end

    aa=[sum01,sum02,sum03];bb=[sum11
,sum12,sum13];
    aaa=min(aa);bbb=min(bb);
    if aaa<bbb
        bit((i+10-2)/10)=0;
    else
        bit((i+10-2)/10)=1;
    end
end

% bit=num2str(bit);
% display(bit)
BIT_error=bit;
error=0;

```

```

for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error));
p22(m)=error/length(BIT_right);
%%
% метод3:Импульсный Метод

clear bit BIT_error;

Yn((Yn-mean(Yn))>0)=1;
Yn((Yn-mean(Yn))<0)=0;

yy=Yn;

figure
plot(Yn);

yy3=[yy,zeros(1,50)];
bit=zeros(1,560);
bit(1)=1;
T=5;a=2;tn=5;
for i=1:10:(length(yy)-9)
    for j=i+7:i+19
        detayy=yy3(j+1)-yy3(j);
        if abs(detayy)>0.5
            if bit(a-1)==1
                if ((j-tn)/T)<1.5
                    % 0.8<((j-i-5)/T) & ((j-i-5)/T)<1.2
                    bit(a)=1; tn=j+5;
                else
                    % 0.8<((j-i-5)/T/2) & ((j-i-5)/T/2)<1.2
                    bit(a)=0;tn=j;
                end
            elseif bit(a-1)==0
                if ((j-tn)/T)<1.5
                    % 0.8<((j-i-5)/T) & ((j-i-
5)/T)<1.2

```

```

        bit(a)=0;tn=j+5;
    else
%           0.8<((j-i-5)/T/2) & ((j-i-
5)/T/2)<1.2
        bit(a)=1;tn=j;
    end
    break;
end
break;
end
end
a=a+1;
end
BIT_error=bit;
error=0;
for
i=1:min(length(BIT_right),length(BIT_
error))
    d= BIT_right(i)-BIT_error(i);
    if d
        error=error+1;
    end
end
error=error+abs(length(BIT_right)-
length(BIT_error));
p33(m)=error/length(BIT_right);
%%
end
close all;
figure
plot(Sigma,p11,'b')
hold on
plot(Sigma,p22,'r')
hold on
plot(Sigma,p33,'g')
load airspyerror.mat
hold on
plot(Sigma,p1,'b*-' )
hold on
plot(Sigma,p2,'r*-' )
hold on
plot(Sigma,p3,'g*-' )
ylim([0 1])

```

```

title({'зависимость вероятность
ошибки';' и мощность шума \sigma
при декодировании'});
ylabel('вероятность
ошибки');xlabel(' мощность шума
\sigma')
legend('Метод с помощью функции
corrcoef с RTL-SDR','min CKO с RTL-
SDR','Импульсный Метод с RTL-
SDR','Метод с помощью функции
corrcoef с AIRSPY','min CKO с
AIRSPY','Импульсный Метод с
AIRSPY')

```