**Список введенных обозначений**

- Задержка на 1 такт; - амплитуда в i-канале трафика; - элемент матрица Адамара ; - порождающий многочлен циклического кода; - генераторы сверточного кода; - элемент скремблирующей ПСП; - длительность чипа ПСП; - оценка веса луча с амплитудой и фазой ; - задержка -го луча; - комплексная огибающая на входе приемника; - поток “мягких” значений с выхода Rake-приемника длиной 2N символов, полученная на выходе декодера Витерби с -м состоянием в конце; –любой коэффициент нормировки;

**1.Передатчик**

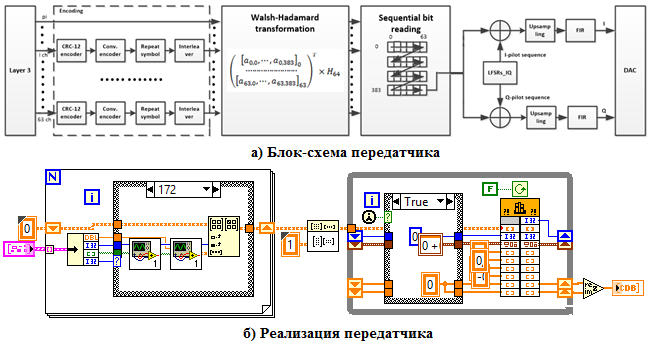


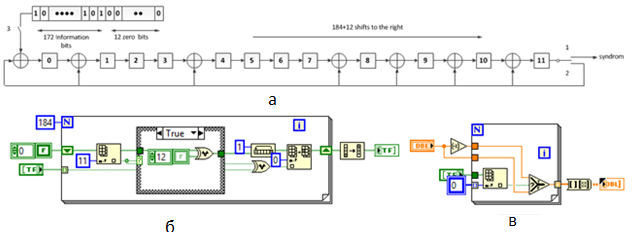
Рисунок 1 – Блок-схема передатчика IS-95 и реализация в LabVIEW.

Пусть (l+1)-й кадр сетевого уровня в i-м канале трафика задан многочленом

Непрерывный поток битов в i-м канале зададим многочленом

Остаток задается выражением

Формула (2) описывает процесс добавления циклического префикса к кадру сетевого уровня с целью контроля правильности декодирования на приемной стороне. Данная операция осуществляется блоком CRC-12(8) (рисунок 1 а). Блок-схема и реализация в LabVIEW представлены на рисунке 2 а, б.

Рисунок 2 Блок-схема и реализация циклического кодера и перемежителя.

Сверточный кодер образует поток битов заданный многочленом

Правая часть выражения (4) осуществляется блоком Convolution encoder (рисунок 1 а) , описание и реализация представлены на рисунке 3.

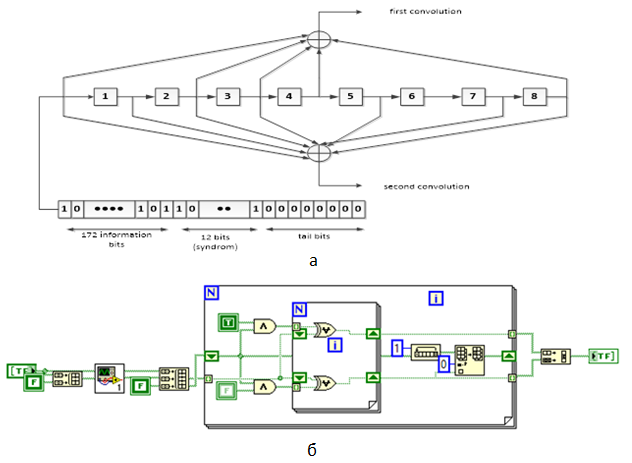


Рисунок 3 –Блок-схема и реализация сверточного кодера в LabVIEW.

На рисунке 3 а показана структура кадра после добавления циклического префикса , что отражено в формуле (2).В формуле (4) задает перестановку блочного перемежителя

Формула (5) показывает что блочный перемежитель задается перестановкой, что показано на рисунке 4, реализация в LabVIEW показана на рисунке 2 в.

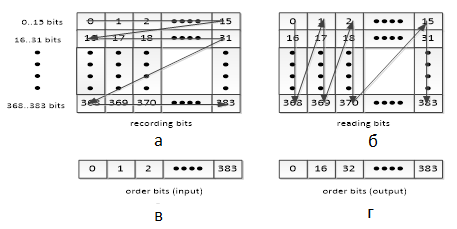


Рисунок 4 – Иллюстрация работы перемежителя.

Огибающая на выходе передатчика

Формула (6) отражает процесс формирования комплексной огибающей на выходе передатчика (рисунок 5, эпюра з, указано что огибающая представляет собой ряд выборок и показана только синфазная компонента). Эпюры а,б,в показывают сопоставление битов векторам Уолша. Эпюра г показывает результат выполнения преобразования Уолша – Адамара (рисунок 1 а, блок Walsh-Hadamard transformation).



Рисунок 5 – Эпюры работы передатчика.

Далее производится скремблирование сигнала (эпюра г, рисунок 1 а перемножители и блок LFSRs).

В формуле (7) импульсный отклик формирующего фильтра задается выражением (сигнал типа root raised cosine)

Поскольку LabVIEW оперирует лишь дискретными величинами далее подразумевается , что все указанные формулы реализуются в дискретном виде.

Импульс вида (7) определяет полосовой сигнал без межсимвольной интерференции c быстрой асимптотикой затухания пропорциональной . Ширина полосы составляет и для ее корректной дискретизации достаточно *двух выборок* на символ, при этом коэффициент передискретизации составит , что удовлетворяет условиям теоремы Котельникова. Поведение импульса показано на рисунке 6. Заметим, что указанный импульс обладает свойством *нормировки*: =.

Однако при восстановлении сигнала по выборкам могут возникнуть ошибки интерполяции и желательно использовать четыре выборки на битовый интервал.

Коэффициент задается выражением (9600 бит/с, рисунок 5 г)

Для скорости 4800 бит/с элемент повторяется дважды подряд(блок repetition, рисунок 1 а).

Дискретная свертка (6) выполняется с помощью блоков Upsampling и FIR (рисунок 1 а, рисунок 5 ж з), их реализация выполнена в подключенной DLL библиотеки с кодом ,указанным в листинге 1.

include “stdafx.h”

#pragma pack(1)

extern “C” \_\_declspec(dllexport) int dsp( double \*arr, int phase, complex \*samples, double \*even, double \*odd, double \*I, double \*Q, double \*res, double \*ims);

int dsp( double \*arr, int phase, complex \*samples,

double \*even,double \*odd,double \*I,double \*Q, double \*res, double \*ims) {

int I,j,k,cnt=phase;

for(i=0;i<24576;i++) {

for(j=9;j>0;j--) samples[j]=samples[j-1];//сдвиг выборок в КИХ-фильтре

samples[0].re=I[cnt]\*arr[i]; // обновляем первую выборку в буфере

samples[0].im=Q[cnt]\*arr[i];

k=i\*2;

res[k]=0;

ims[k]=0;

for(j=0;j<10;j++) {

res[k]+=even[j]\*(samples[j].re);//производим свертку с ИХ в синфазном

ims[k]+=even[j]\*(samples[j].im); //тоже в квадратурном канале}

k++;

res[k]=0;

ims[k]=0;

for(j=0;j<10;j++) {

res[k]+=odd[j]\*(samples[j].re);

ims[k]+=odd[j]\*(samples[j].im); }

cnt=cnt++%32768;}

return cnt;

} # pragma pack(1)

Листинг 1- Реализация формирующих фильтров Найквиста.

Из рисунка 5, эпюра ж видно, что при выполнении дискретной свертки некоторые выборки будут заведомо нулевыми , что позволяет уменьшить число выполняемых операция и реализовано в приведенном коде. Данный участок является наиболее требовательным к вычислительным ресурсам. Описанная реализация передатчика способна работать в режиме реального времени.

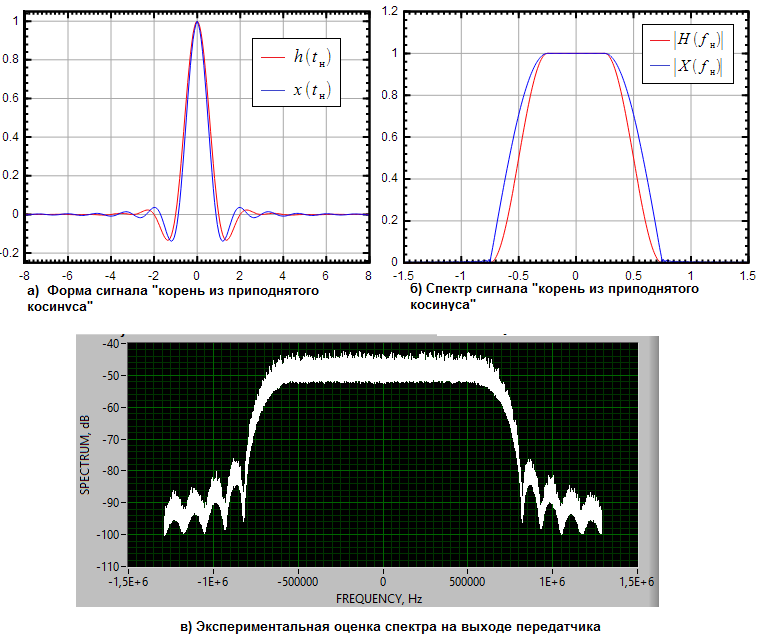


Рисунок 6 – Спектр мощности сигнала на выходе передатчика.

**2.Приемник**



Рисунок 7 – Блок-схема приемника.

Импульсная характеристика канала связи может быть задана выражением

Формула (9) показывает наличие радиоэха на приемной стороне. Для нейтрализации его вредного влияния используют Rake-приемник (рисунок 7 а), который состоит из синхронного детектора огибающей (envelope detector),”гребенки” корреляторов и комбинационного устройства, на выходе имеем т. н. “мягкое ” значение

Эпюры а,б,в,г на рисунке 8 иллюстрируют процесс формирования в соответствии с формулой (6), но в данном случае следует положить и выполнить комплексное сопряжение.

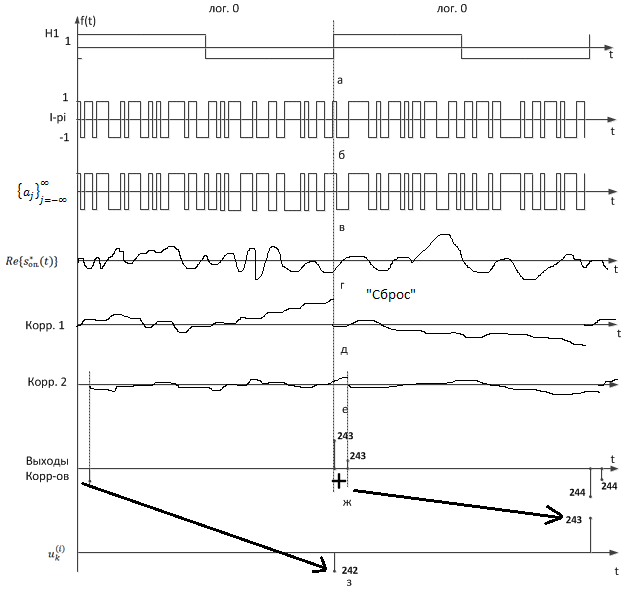
****

Рисунок 8 – Эпюры работы Rake-приемника.

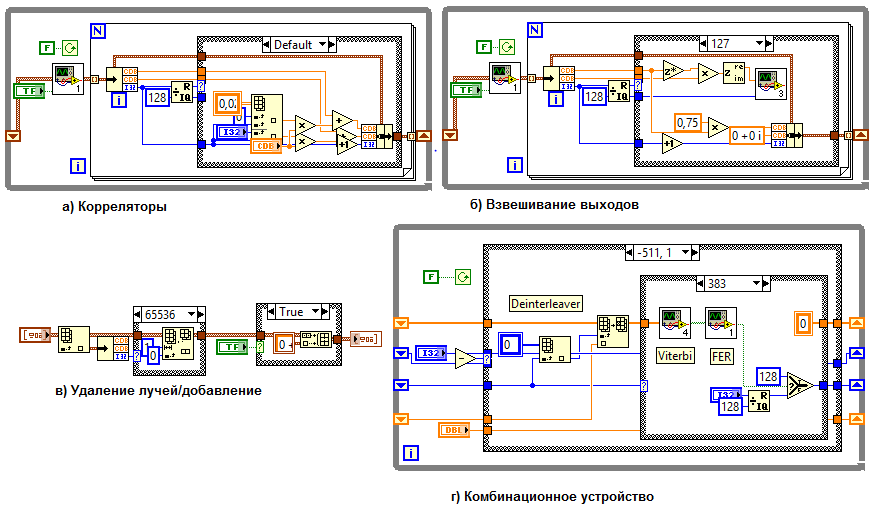


Рисунок 9 – Реализация Rake-приемника в LabVIEW.

Для повышения быстродействия при дискретизации огибающей на выходе USRP-приемника используются также две выборки на битовый интервал , что приводит к смазыванию эффекта ортогональности. Максимальная временная неопределенность может составить .

Поскольку является *полосовым* сигналом, а корреляция осуществляется на конечном интервале, то это приводит к появлению дополнительного корреляционного шума на выходе коррелятора, который можно ослабить введением дополнительных усложнений в реализацию. Однако здесь рассматривается простейший вариант.

Из формулы 10 видно, что необходимо оценить вес луча. Устройство оценки веса луча в простейшем виде представляет собой перемножитель , “снимающий” модуляцию с пилот-сигнала и фильтра скользящего среднего, выделяющего постоянную комплексную константу из шума. Заметим, что полученная величина *фактически* не является весом луча, который представляет комплексное безразмерное число с модулем строго меньше единицы (в канале энергия сигнала рассеивается) . Однако для оптимального приема в соответствии с формулой (10) необходимо лишь чтобы модуль полученной оценки был пропорционален модулю веса луча, а их фазы примерно совпадали (с поправкой на присутствие шума).

Недостатками данного метода оценки являются наличие операции сдвига в фильтре скользящего среднего , когда на одну входную выборку приходится большое число операций копирования в соседнюю ячейку, и наличие массива, занимающего память.

Первый недостаток устраняется введением *циклической очереди*. Второй не критичен на современном этапе развития аппаратных средств.

Можно предложить в качестве фильтра скользящего среднего простейший *рециркулятор*, близкий к нему по свойствам , но использующий только одну ячейку памяти (блок задержки на один такт) и одно сложение. Данная схема используется в описываемой реализации, поскольку программный код получается значительно менее громоздким.

Оценка веса луча (формула (9)) дается блоком оценки луча (channel estimator, рисунок 7 б) c точностью до размерного множителя K , который представляет коррелятор , настроенный на пилот-сигнал, а вместо сброса в ноль накопленного значения производится его умножение на безразмерный коэффициент , в модели задано .

Частотная расстройка между несущей передатчика и опорным сигналом приемника отражается в комплексной экспоненте умноженной на огибающую сигнала на входе Rake-приемника (рисунок 7 а, после детектора огибающей).

Учитывая линейность блока оценки луча и использование при взвешивания выходов корреляторов операции комплексного сопряжения(формула 10, рисунок 7 а), можем заметить , что операция комплексного взвешивания и блок оценки веса луча действуют как *разомкнутая система ФАПЧ*. Однако появляется сдвиг фазы и дополнительное искажение амплитуды.

Искажения оценки веса луча даны на рисунке 10 (произведено моделирование в LabVIEW, здесь обозначает коэффициент рециркулятора). Поскольку в качестве приемопередатчиков используются USRP-2920 с расстройкой порядка , и учитывая малую поправку на фазовый сдвиг (в идеале выходы корреляторов лежат на действительной оси и фазовый сдвиг вносит квадратичную поправку ) ,а также тем что амплитудные искажения одинаковы для всех лучей ( условие пропорциональности сохраняется , потери в ОСШ на выходе коррелятора при малых также квадратичны), вызванными искажениями можно пренебречь.

Эпюры д е на рисунке 8 отражают работу корреляторов (указана вещественная часть накопленного значения в интеграторе ).

Эпюры ж з показывают работу комбинационного устройства , которое складывает выходы разных корреляторов. Каждое переданное значение с корреляторов циклически последовательно нумеруется в диапазоне 0..511. При этом началу пилот-сигнала соответствует символьный интервал с индексом ноль. (На эпюре а указано два символьных интервала).

Если разность между последовательными по времени лучами не превышает 64 чиповых интервалов (равняется символьному интервалу, эпюра а, такая задержка весьма велика , но как правило, она составляет не более 20-30 чиповых интервалов), то корреляторы последовательно выдают значения принадлежащие одному биту ,для чего и необходима нумерация. Комбинационное устройство складывает выходы корреляторов с одинаковым индексом , а приход значения с о следующим индексом означает необходимость передать накопленное значение на деинтерливер и начать суммирование заново (эпюра ж).

При ложной тревоге с выхода блока поиска лучей (Path searcher, рисунок 7 в) подается псевдолуч с неправильной нумерацией, которая ведет к полному нарушению работы комбинационного устройства. Реализация комбинационного устройства (рисунок 9 г) отслеживает подобные ситуации (при разумной вероятности ложной тревоги).

Реализация блока поиска лучей упрощена и не учитывает возможность наличия сигналов от нескольких базовых станций , а все обнаруженные лучи подаются на Rake-приемник, хотя некоторые лучи должны быть отброшены ввиду их малой энергии.

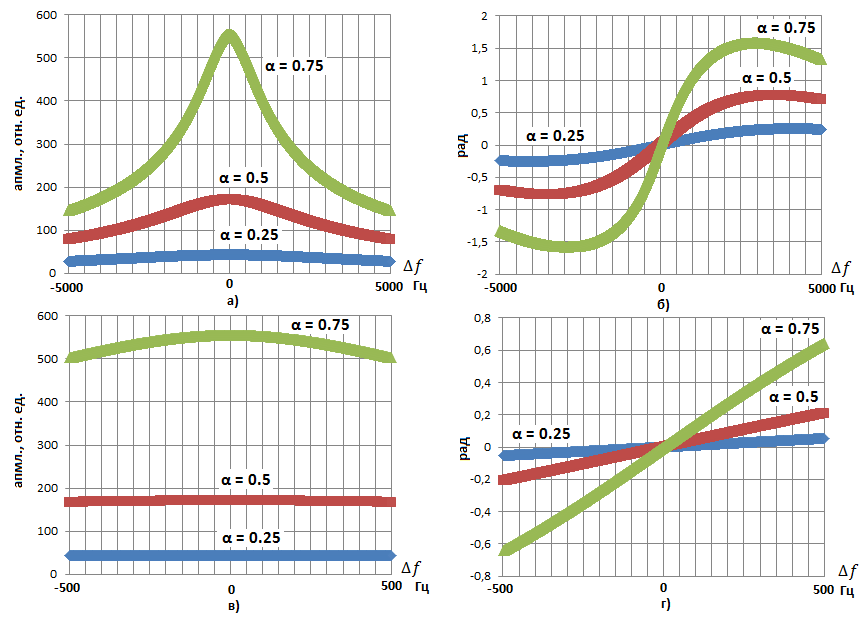


Рисунок 10 –Линейные искажения оценки веса луча в зависимости от частотной расстройки передатчика и приемника.

Полученная последовательность “мягких” решений получается на выходе блока Deinterleaving (рисунок 8 а, формула (10) , левая часть) путем перестановки элементов в потоке мягких решений с Rake-приемника, что приводит к рызрывам пакетов ошибок и повышению помехоустойчивости работы декодера Витерби.

Реализация деперемежителя в LabVIEW представлена на рисунке 9 г (также здесь видим блок декодера Витерби (Viterbi), идущий после деперемежителя , рисунок 7 а).

Деперемежитель совместно с блоком FER и комбинационным устройством обеспечивают достижение и контроль состояния *кадрового синхронизма.*

В N такте работы декодер Витерби формирует вектор

Функционал правдоподобия задается выражением

Компоненты вектора (11) монотонно зависят от компонент вектора

*Критерий максимума правдоподобия* задает условия максимизации компонент (11,13)

.

Значения являются *метриками ребер* (рисунок 12 а, г).



Рисунок 11–Кривая помехоустойчивости декодера Витерби.

Из условия (14) следует , что метрики путей непрерывно растут. При низкоуровневой реализации декодера возможно переполнение ячеек , хранящих значение метрик, поскольку ,как правило, они имеют небольшое число разрядов (это следует из того, что помехоустойчивость при квантовании мягкого значение на входе на восемь уровней (три разряда) не уступает теоретически возможному, отсюда возможность использования ячеек с небольшим числом разрядов ). Использование ячеек с небольшим числом разрядов упрощает реализацию декодера.

Для предотвращения переполнения используются приемы *нормализации* метрик.

Поскольку LabVIEW является высокоуровневым языком с типами большой разрядности, то указанная проблема не стоит остро. Однако нормализацию нетрудно осуществить, вычитая на каждом шаге треллиса значение одной из метрик из всех других. Тогда метрики будут случайными числами с гауссовским распределением и ничтожно малой вероятностью переполнения разрядов, что и требуется.

На рисунке 11 показана зависимость частоты FER (Frame Error Ratio) появления искаженных информационных кадров на выходе декодера Витерби от отношения сигнал/ шум (SNR) на входе декодера (получена моделированием в LabVIEW).

Требуемая частота составляет примерно 0.02 (из 50 принятых кадров в секунду допускается принятие одного искаженного). Данное значение частоты не оказывает существенного влияния на качество декодирования речи. Большая частота нежелательна, а меньшая указывает на то, что SNR на входе взято со значительным запасом, и его можно уменьшить.

В данной системе абоненты создают интерференционный шум , поэтому амплитуду сигнала в каждом канале снижают для достижения FER порядка ~0.01.

Как видно из рисунка 11 доля искаженных кадров быстро снижается от неприемлемого уровня (~0.1) до пренебрежимо малого ((~0.001) на небольшом интервале 2-3.9 дБ . Быстрый спад кривых помехоустойчивости является типичным для помехоустойчивых кодов.

Для данных параметров сверточного кода (длина кодового ограничения и скорости ) генераторы кода и обеспечивают максимальную помехоустойчивость.

Видим что оптимальное значение SNR составляет примерно 3,3 дБ.

Указанное выше свойство вектора (11) дает оценку на выходе декодера Витерби *общую* для всех путей (при рабочих уровнях SNR) , что упрощает его программную реализацию , однако вносит существенную задержку(рисунок 12 в).

Полученный с выхода кадр пересобирается и проверяется на корректность с помощью вычисления синдрома циклического кода (рисунок 12 б). ). В качестве критерия *срыва* состояния кадрового синхронизма принимается принятие подряд 7 некорректных кадров (упрощение реального процесса).

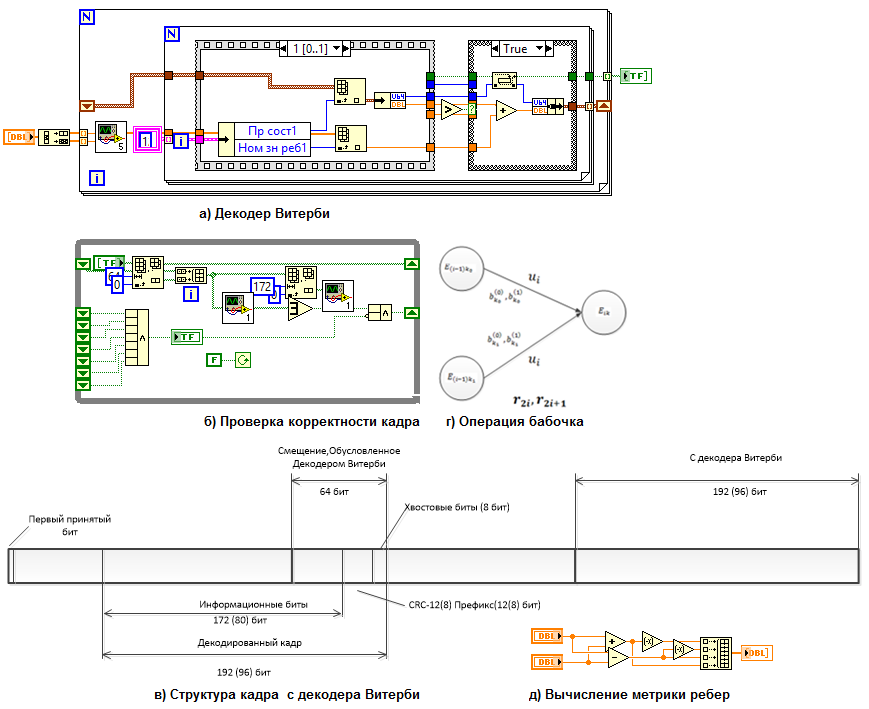


Рисунок 12-Структура кадра на выходе декодера Витерби и реализация блока контроля правильности декодирования в LabVIEW.

Для достижения тактовой синхронизации используется блок поиска лучей (рисунок 7 в). Мгновенная мощность на выходе согласованного фильтра в момент (fast convolution, рисунки 7 и 14) дается выражением

Импульсная характеристика фильтра задается выражением

Свертка (15) выполнена на основе быстрого преобразования Фурье

что приводит к значительному увеличению быстродействия модели.

Полученный корреляционный профиль (формула 15, рисунок 14 в) усредняется с целью снижения вероятностей ложного срабатывания и пропуска сигнала (рисунки 7 в, 14 а б в). Усредненный профиль хранится в *циклической очереди*, что позволяет сэкономить вычислительные операций на сдвиг элементов в массиве .

По усредненному профилю (average profile, рисунок 14) вычисляется порог (рисунок 14 а) , а в качестве блока compute threshold выступает рециркулятор (рисунок 13) .

Условие обнаружение луча в момент задается выражением (рисунок 14 г)

В модели задано (рисунок 13).

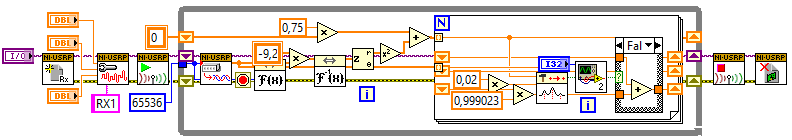


Рисунок 13 – Блок поиска лучей.

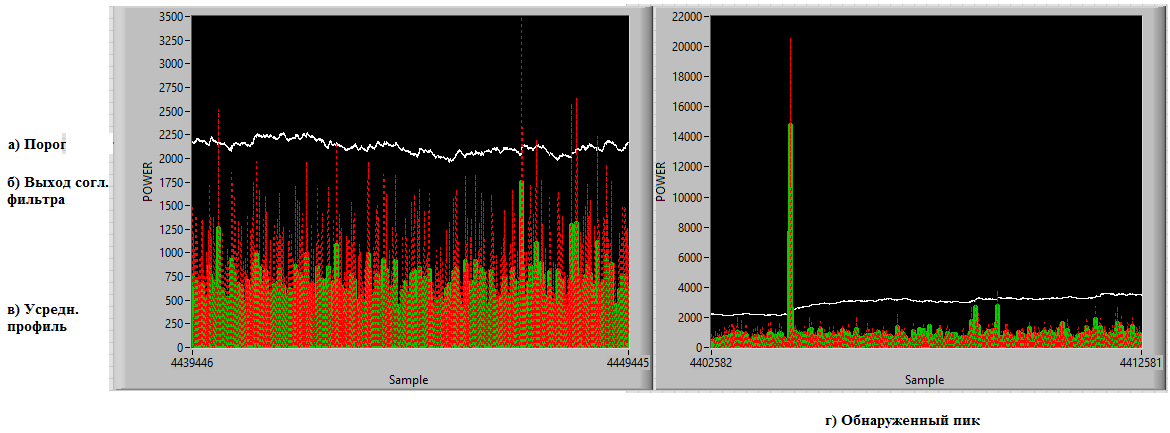


Рисунок 14 – Эпюры работы блока поиска лучей .

Параметр подается (рисунок 7 в) при обнаружение начала нового периода пилот-сигнала в *l-*м луче. Поскольку период пилот-сигнала повторяется 32.5 раза в секунду, и учитывая, что USRP выдерживает заданную тактовую частоту с высокой точностью , обновление лучей происходит с достаточной частотой , обеспечивающей поддержание состояния тактового синхронизма. При этом уже используемые лучи обрабатываются только в течение одного периода (так заложено в реализацию Rake-приемника, рисунок 9 а, б), а затем отбрасываются. Это обеспечивает непрерывное обновление параметров для работы Rake-приемника и позволяет отказаться от системы тактовой синхронизации.

В блоке Path Searcher также компенсируется кратковременное повышение порога, вызванное корреляционным пиком, что приводит к пропуску лучей с малой энергией. В модели USRP-2932 имеется GPS- синхронизатор, позволяющий достичь тактовой синхронизации без использования блока поиска лучей, но для его использования необходимы GPS–антенны, при этом время вхождения в синхронизм может составлять несколько минут.

**Общие замечания и описание экспериментов.**

В рассматриваемой модели используется значительное число массивов-констант: перестановка перемежителя, опорные сигналы корреляторов, коэффициенты формирующих фильтров Найквиста, сформированных с использованием вспомогательных виртуальных приборов, указанных на рисунке 15. Данный прием называется препроцессингом и позволяет сделать код менее громоздкими, значительно повысив быстродействие работы модели. В данных виртуальных приборах используется готовые подприборы из библиотеки Modulation toolkit.

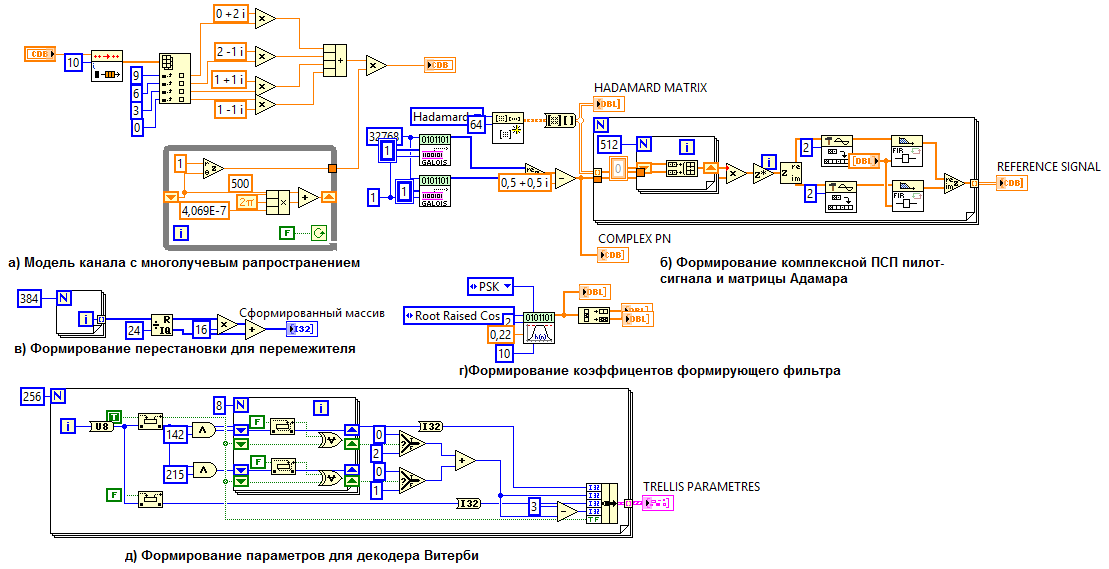


Рисунок 15-Вспомогательные подприборы.

Взаимодействие с аппаратными средствами проиллюстрировано на рисунке 16, указаны основные конфигурационные параметры USRP. Для управления моделью приемника (рисунок 7) был разработан подприбор с программный интерфейсом на рисунке 17 а. Настройка приемника осуществляется с помощью интерфейса подприбора , указанного на рисунке 13 а (рисунок 17 б).



Рисунок 16 – Взаимодействие программной модели и USRP.

Сетевой уровень системы заменен источником случайных битов, формирующих аналог речевых кадров (рисунок 16,”Источник информационных кадров”). Выход приемника отображается на панели подприбора , указанного на рисунке 18 (“Принятый кадр”). При правильной работе модели поток кадров, формируемые на передающей стороне должен корректно восстанавливаться на приемной. Для наглядности работы модели в каждом канале можно передавать повторяющуюся копию соответствующего ему кадра, на приемной стороне значение кадров не меняется (отсутствует мелькание), что при сравнении дает наглядную визуальную картину корректности работы. Также (рисунок 18) индикатор “Ошибка” указывает на значение синдрома декодированного кадра (нулевой или нет). При корректности реализации блока вычисления синдрома (рисунок 2 а, б) это хороший указатель правильности работы системы. Также вычисляется значение частоты появления искаженных кадров. Для оценки быстродействия системы оценивается скорость обработки кадров(“rate”). В дальнейшем вместо случайных битов можно передавать осмысленную информацию: тексты, изображение. Передача звука проблематична, поскольку необходимо реализовать сложный низкоскоростной речевой кодек.

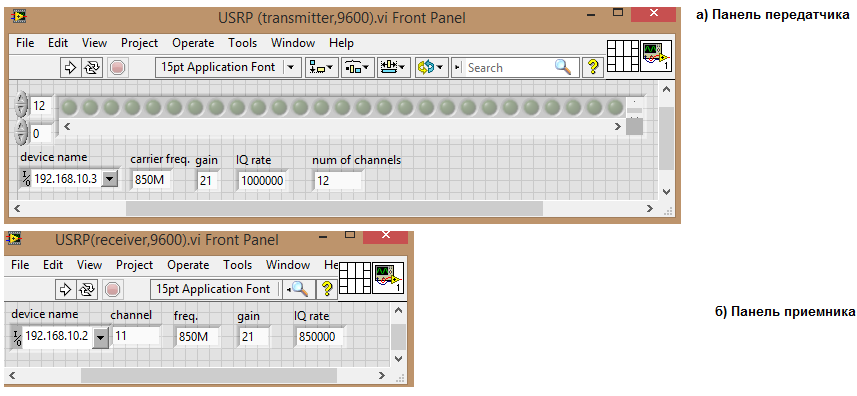


Рисунок 17-Интерфейсы передатчика и приемника.

Конфигурация на рисунке 16 была запущена и указанные контрольные точки проверены. При этом при снижении скорости дискретизации USRP до 850000 выборок/сек удалось выйти на режим реального времени. Как видно (рисунок 18) скорость обработки составляет примерно 17 кадров в секунду. При большей скорости дискретизации происходит переполнение буфера USRP (либо оперативной памяти) и вызывается ошибка переполнения.

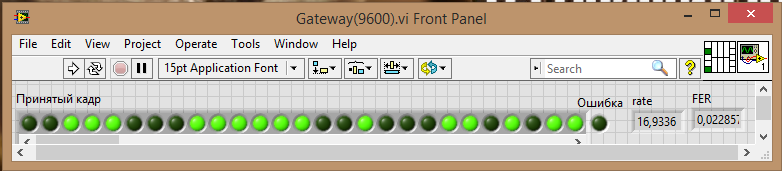


Рисунок 18 –Выход модели приемника.

Произведено несколько опытов на различных расстояниях. Максимальное расстояние соответствует 50 метрам. Модель в LabVIEW с использованием простейшей модели канала связи (рисунок 15 а) дает лучшие результаты чем эксперимент с USRP (рисунок 16) по частоте появления искаженных кадров, что связано с неопределенностью тактовой синхронизации в (максимум) и низкой частотой сэмплирования (две выборки на символ, что удовлетворяет теореме Котельникова, но может быть недостаточно для аппаратного восстановления сигнала). Также обнаружено влияние среды распространения (металлические объекты и т. д.) на помехоустойчивость приема. Выход приемника можно реализовать в виде UDP –клиента для отправки по сети на другие компьютеры для дальнейшей обработки кадров.

**Библиографический список**

1.CDMA: прошлое, настоящее, будущее[Текст] : монография / Л. Е. Варакин [и др.]; под ред. Л. Е. Варакина, Ю. С. Шинакова. –М.:Междунар. акад. cвязи, 2003,-601 c.

2.Петрович, Н. Т. Системы связи с шумоподобными сигналами [Текст]/ Н. Т. Петрович, М. К. Размахнин.-М.: Советское радио, 1969.-232 с.

3.Морелос-Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение [Текст]/ Р. Морелос-Сарагоса-М.: Техносфера, 2005.- 320 с.

4.Вернер , М. Основы кодирования [Текст]/ М. Вернер- М.: Техносфера, 2004.-288 с.

5.Ipatov, V. P. Spread spectrum and CDMA. Principles and Applications [Text]/ V. P. Ipatov –WILEY,2004.-373 c.

6.Yang, Samuel C. CDMA RF System Engineering [Text]/ Samuel C. Yang-Norwood: Artech House, 1998.-289 с.

7. Кехтарнаваз, Н. Цифровая обработка сигналов на системном уровне с использованием LabVIEW [Текст]/ Н. Кехтарнаваз, Н. Ким.-М.: Издательский дом “ДОДЭКА-XXI”, 2007.-304 c.

8. Прокис, Дж. Цифровая связь. Пер. с англ.[Текст]/ Дж. Прокис-М.: Радио и связь, 2000.-800 с.

9.Лезин , Ю. С. Оптимальные фильтры и накопители импульсных сигналов.[Текст]/ Лезин Ю. С.-М.: Госэнергоиздат,1963.-319 c.

10.Tretter , St. A. Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C6713 DSK[Текст]/St. A. Tretter-New York:Springer,2008.-344 p.

11. Суранов, А. Я. LabVIEW 8.20. Справочник по функциям.[Текст]/ Суранов А. Я.-М.:ДМК Пресс, 2007.-536 c.

12.Yang, Y. LabVIEW graphical programming cookbook.[Text]/Y. Yang-Birmingham:Packt, 2014.-235 р.

13.Clark, L. C.LabVIEW Digital Signal Processing and digital comms.[Text]/ L. C. Clark-New York:MacGraw-Hill Companies, 2005.-205 p.

14. Blume, Peter A. The LabVIEW Style Book.[Text]/Peter A. Blume-Crawfordsville:Prentice Hall,2007.-362 p.

15.Варакин, Л. Е. Системы связи с шумоподобными сигналами.[Текст]/Л. Е. Варакин-М.: Радио и связь, 1985. — 384 с., ил.

16. 214. Скляр Б. Цифровая связь. Теоретические основы и практическое при-

менение. Изд. 2-е: Пер. с англ. – М.: Из-дательский дом «Вильямс», 2003. – 1104 с.

17. Ettus Research [Electronic resource]. - <http://www.ettus.com>

18.National Instruments [Electronic resource].-http://www.ni.com/datasheet/pdf/en/ds-355

19.Шумоподобные сигналы в системах передачи информации .[Текст]/ Пестряков В.Б., Афанасьев В.П., Гурвиц В.И.[ и др.]-М.:Советское радио,1973.-424 с.

20.Баскаков, С. И. Радиотехнические цепи и сигналы.[Текст]/ С. И. Баскаков-М.:Высшая школа,2000.-462 c.

21.Керниган Брайан У. Язык программирования С.[Текст]/ Брайан У. Керниган, Деннис М. Ритчи.-2-е издание.:пер. с англ.-М.: дом “Вильямс”, 2009.-304 с. : ил.-Парал. тит. англ.