

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

## Телекоммуникационные технологии

Отчет по лабораторной работе №12  
GNU radio

**Работу выполнил:**  
Махонько Я. В.  
Группа: 3530901/90202  
**Преподаватель:**  
Богач Н.В.

Санкт-Петербург  
2022

# 1 Цели работы

- Посмотреть на результаты искажения сигнала, разобраться с проблемой искажения.
- Разобрать этапы, необходимые для восстановления канала:
  - добавление канала;
  - восстановление синхронизации;
  - многолучевые каналы;
  - фазовая и частотная коррекции;

, такие как восстановление синхронизации, многолучевые примеры, фазовая и частотная коррекции, декодирование символов, упорядочивание битовой информации.

## 2 Ход работы

### 2.1 Передача сигнала

На представленном ниже рисунке продемонстрированы сгенерированные избыточные полосы пропускания с различными значениями  $\alpha$ , где  $\alpha$  - коэффициент спада фильтра:

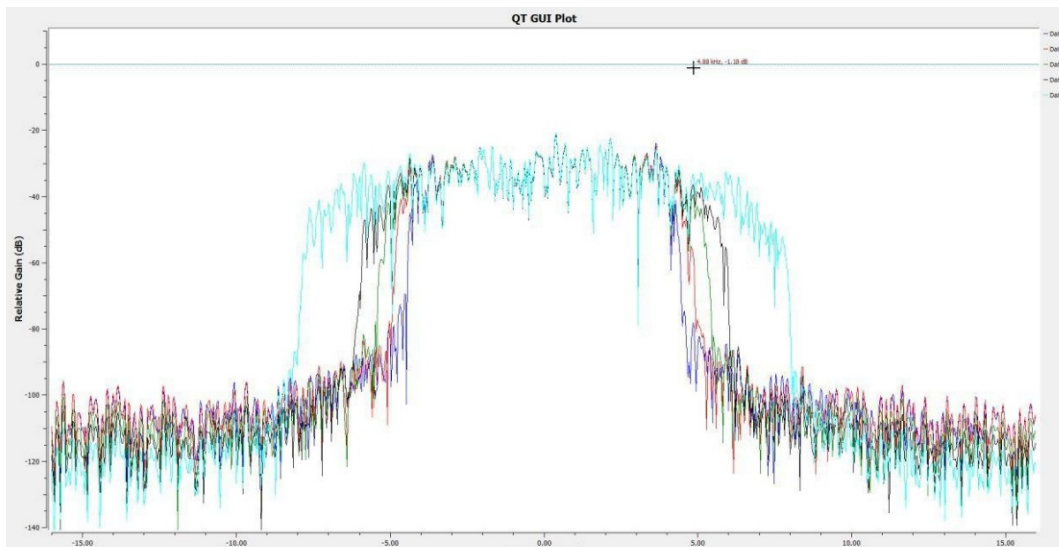


Рис. 1: Избыточные полосы пропускания при различных  $\alpha$

Генератор случайной выборки передаёт значения (0-255) в модулятор сигнальных созвездий, который использует фильтр формирования импульсов корневого косинуса (RRC) с параметром  $\alpha, 4$  :

- Data0 = 0.1 = 100m
- Data1 = 0.22 = 220m
- Data2 = 0.35 = 350m
- Data3 = 0.5 = 500m
- Data4 = 1 = 1000m

Используя данный блок, построим несколько графиков:

- График сигнальных созвездий, на котором можно увидеть эффект увеличения разрешения и сам процесс фильтрации. При этом RRC-фильтр тоже генерирует особый тип помех - межсимвольные (ISI).
- Частотный график демонстрирует форму сигнала, который в конце превращается в шум.
- QPSK-график показывает передаваемый и получаемый (отфильтрованный) QPSK-сигнал.

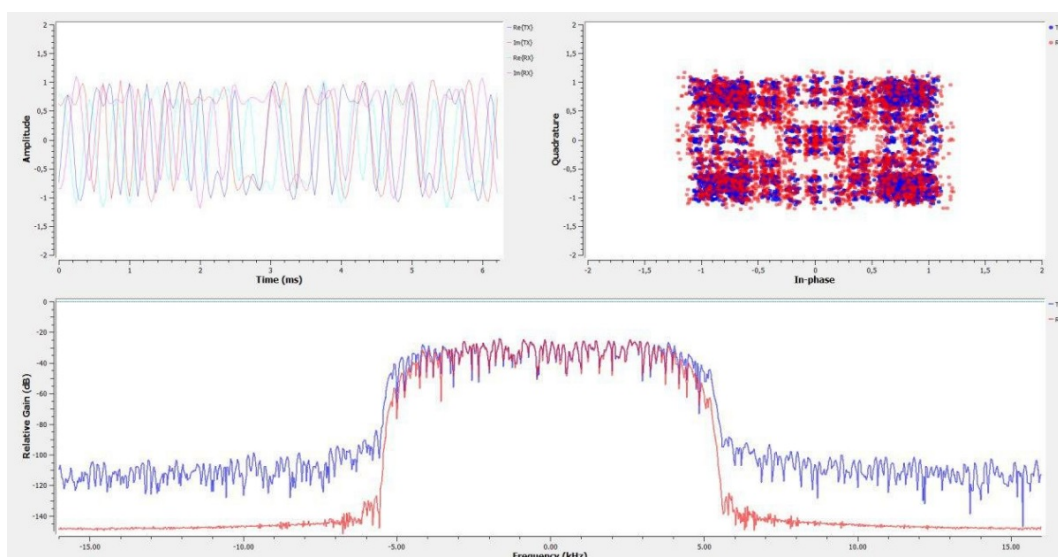


Рис. 2: Набор графиков при RRC-фильтре

На следующем графике можем наблюдать возникновение межсимвольных помех (ISI). Как было сказано ранее, данный тип помех возникает из-за использования RRC-фильтра:

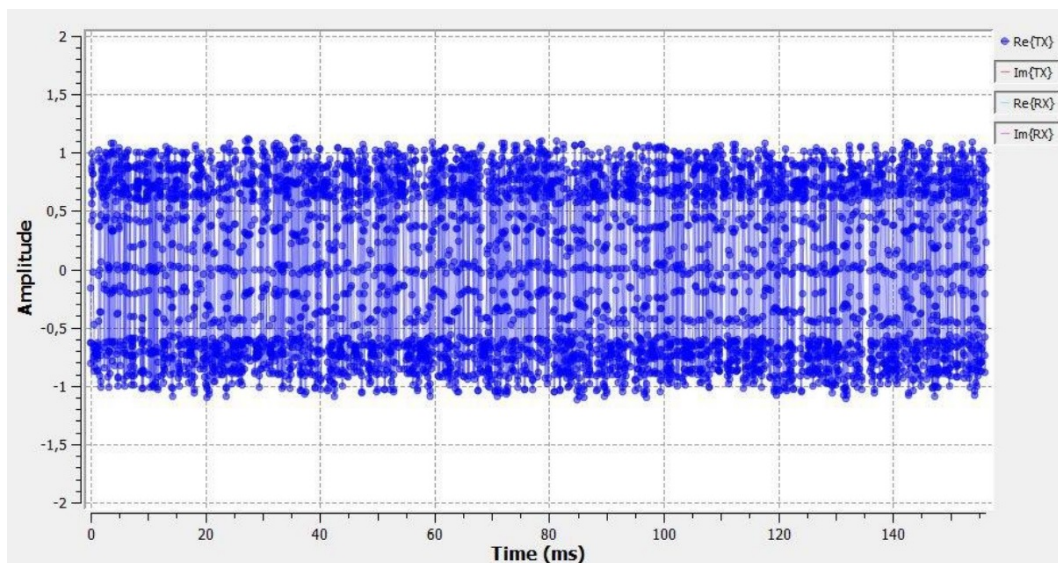


Рис. 3: Значения сигнала при возникающих ISI-помехах

При добавлении второго аналогичного фильтра получаем фильтр с повышенным косинусом (форма фильтра Найквиста). Другое преимущество - отсутствие рассеивающего эффекта из-за использования согласованного фильтра на приёмнике.

Сигнал на приёмнике имеет три отдельных значения: -1, 0 и 1, что заметно на следующем изображении:

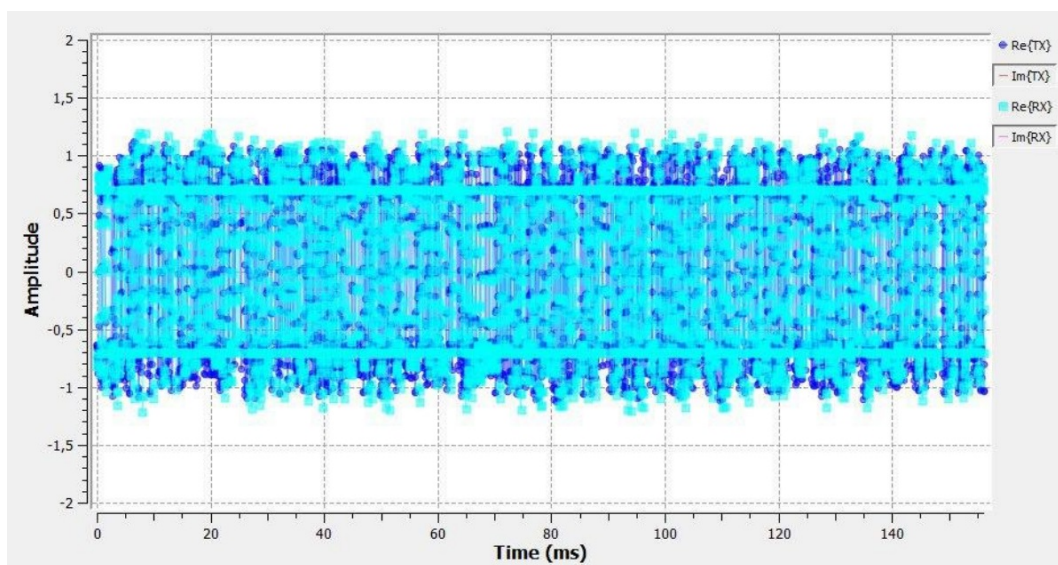


Рис. 4: Значения сигнала на приёмнике

## 2.2 Добавление канала

Добавим блок Channel model. С его помощью можем посмотреть на несколько наиболее типовых проблем:

- влияние шумов;
- отклонения в тактовых сигналах;
- проблема нахождения идеальной выборки.

Выполним симуляцию, получим следующий набор графиков:

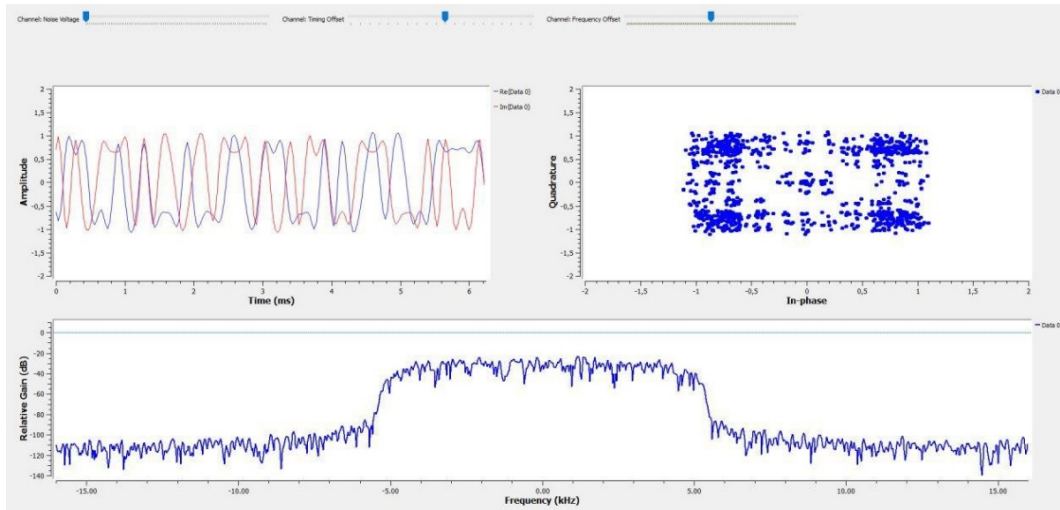


Рис. 5: Графики блока channel model

Ползунки сверху отвечают за шум, частотное и временное смещения.

Сигнальное созвездие выглядит хуже, чем в прошлый раз.

Отменим все эффекты, поставим максимальное значение параметра шума:

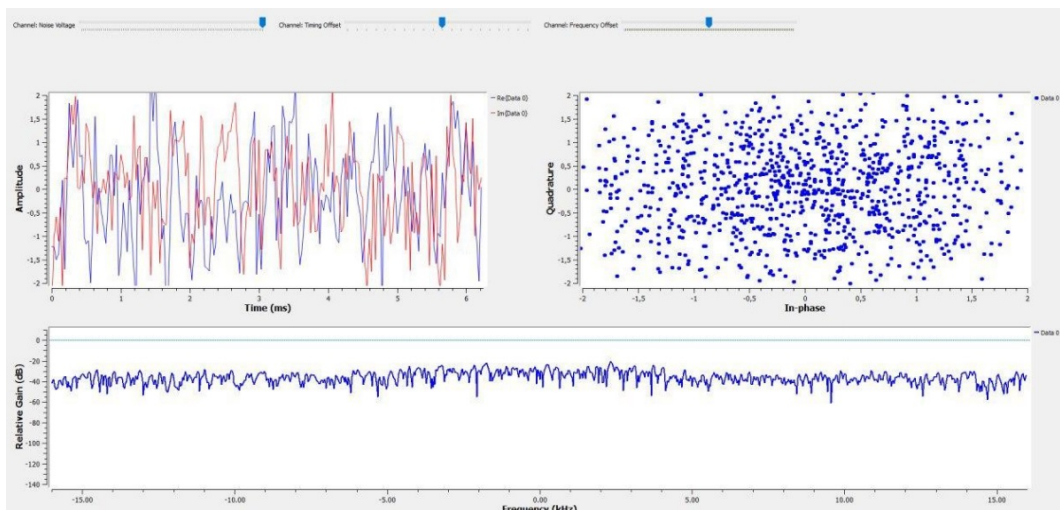


Рис. 6: Графики блока channel model, максимальное значение шума

Сигнальное созвездие стало практически неразличимо.

## 2.3 Восстановление синхронизации

Для демонстрации проблем, связанных с проявлением ISI, создадим 4 символа и отфильтруем их RRC-фильтром:

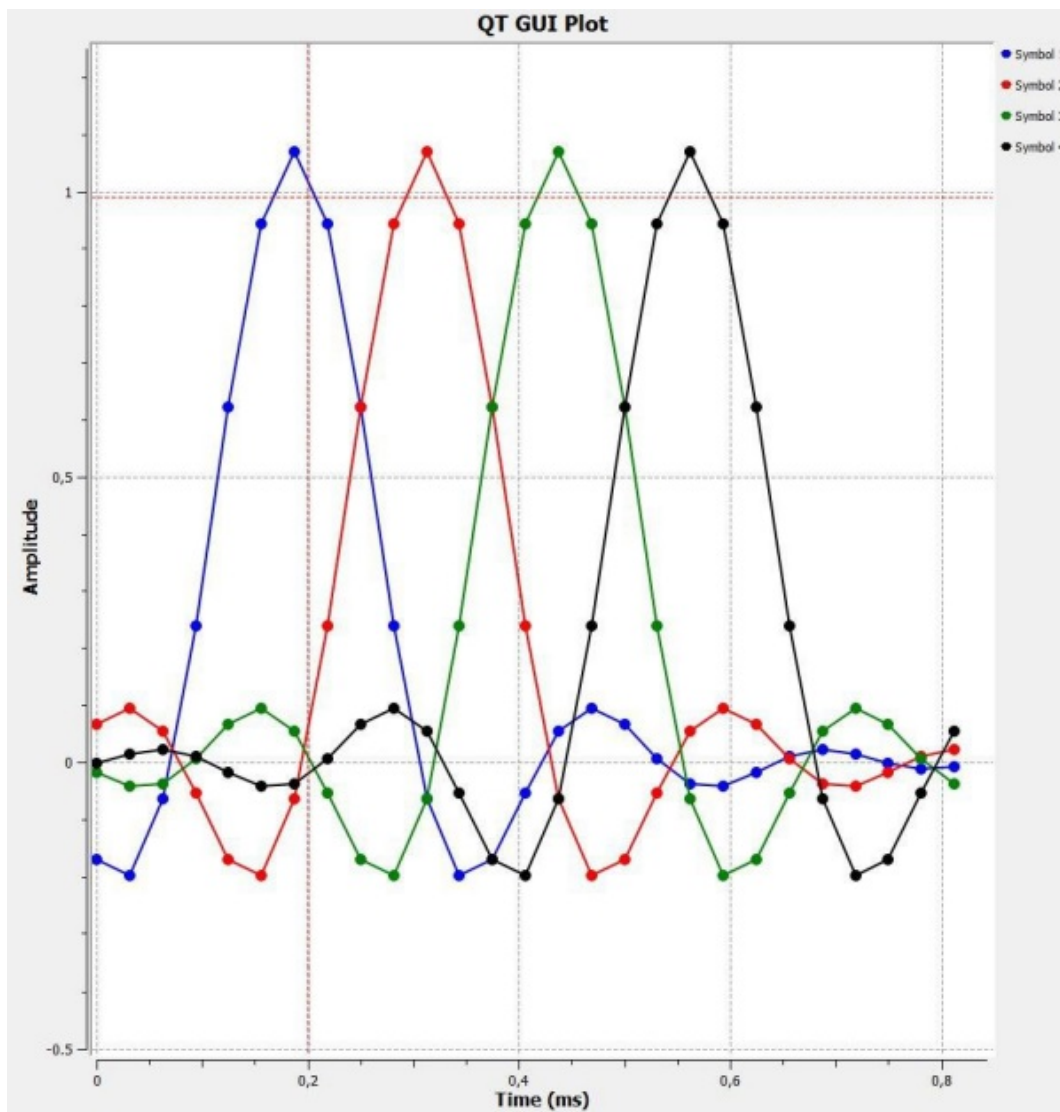


Рис. 7: Графики после фильтрации RRC

Видно, что после первого этапа всё равно сохраняется часть энергии.



Используем фильтр Найквиста с повышенным косинусом, после которого энергия почти полностью теряется:

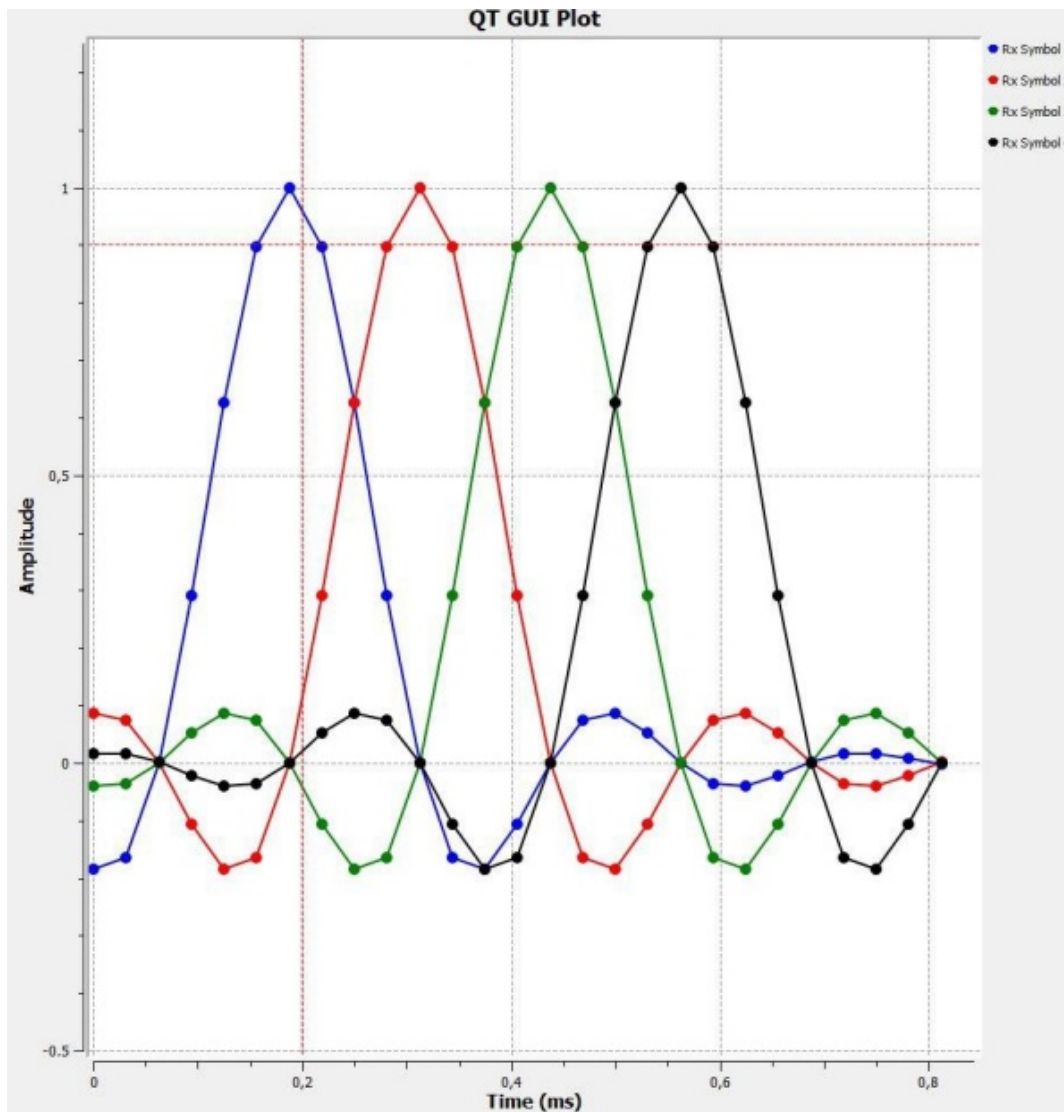


Рис. 8: Графики после фильтрации фильтром Найквиста

При снимаемой в разные моменты времени выборке идеальный период выборки неизвестен и будет страдать по качеству из-за ISI.

Необходимо каким-то образом синхронизировать сигналы приёмника и передатчика, используя только информацию из сигнала.

## 2.4 Детали блока многофазной фильтрации тактовых сигналов

Для восстановления тактового сигнала воспользуемся блоком многофазных фильтров. этот блок выполняет несколько функций:

- Выполняет восстановление тактового сигнала;
- Выполняет функцию фильтра, согласованного с приёмником, для борьбы с ISI;
- Понижает частоту сигнала, производит выборку.

Симуляция работы этого блока:

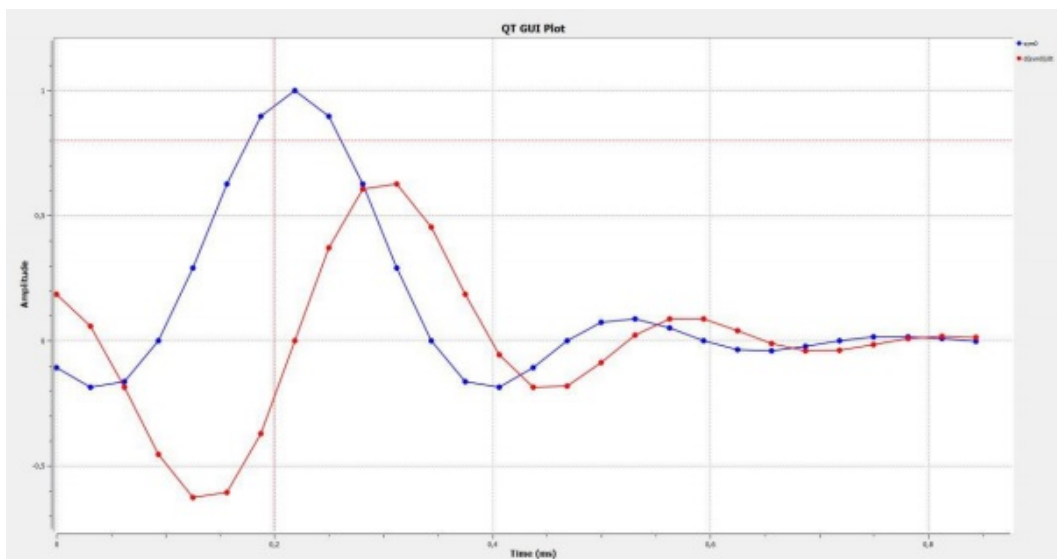


Рис. 9: Графики после блока многофазной фильтрации

При этом значение  $rate = 1$ , а выход фильтра в нужной точке равен 0, что означает, что мы находимся в оптимальной точке выборки.

Несколько сместимся для выхода из "зоны комфорта":

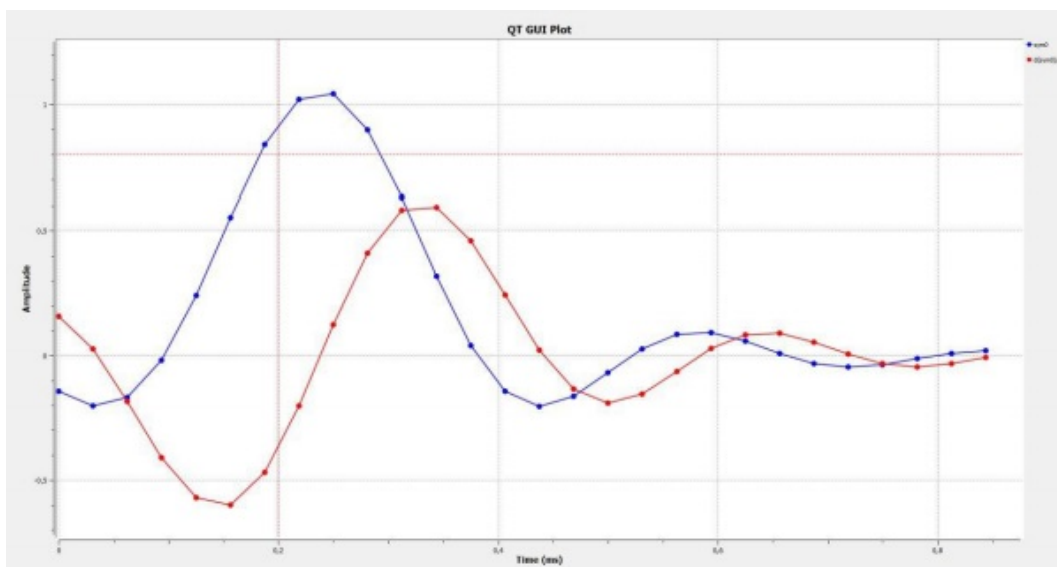


Рис. 10: Графики после блока многофазной фильтрации со сдвигом

Также, вместо использования одного фильтра мы можем использовать несколько фильтров с разными фазами.



Посмотрим на результат симуляции с 5 фильтрами:

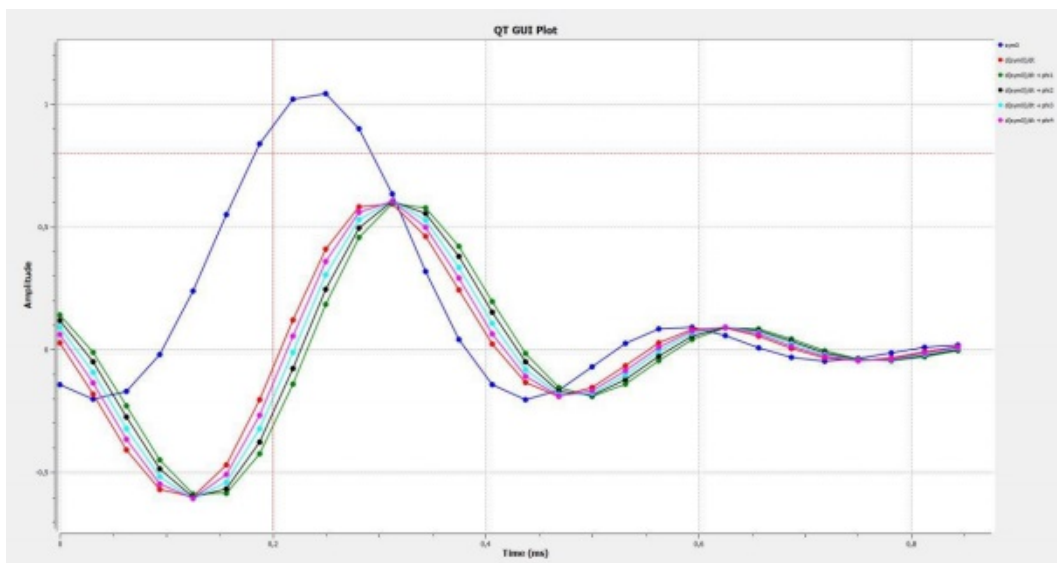


Рис. 11: Графики после блока многофазной фильтрации с пятью фильтрами

Можно заметить, что сигнал  $d(sym0)/dt + \phi3$  имеет точку выборки в 0 - значит, это фазовый сдвиг идеальной фазы.

Таким образом, настроив фазу RRC на  $\phi3$ , можно корректировать рассинхронизацию и выбирать идеальную точку выборки.

При увеличении объема фильтров до 32 можно добиться максимального коэффициента ISI:

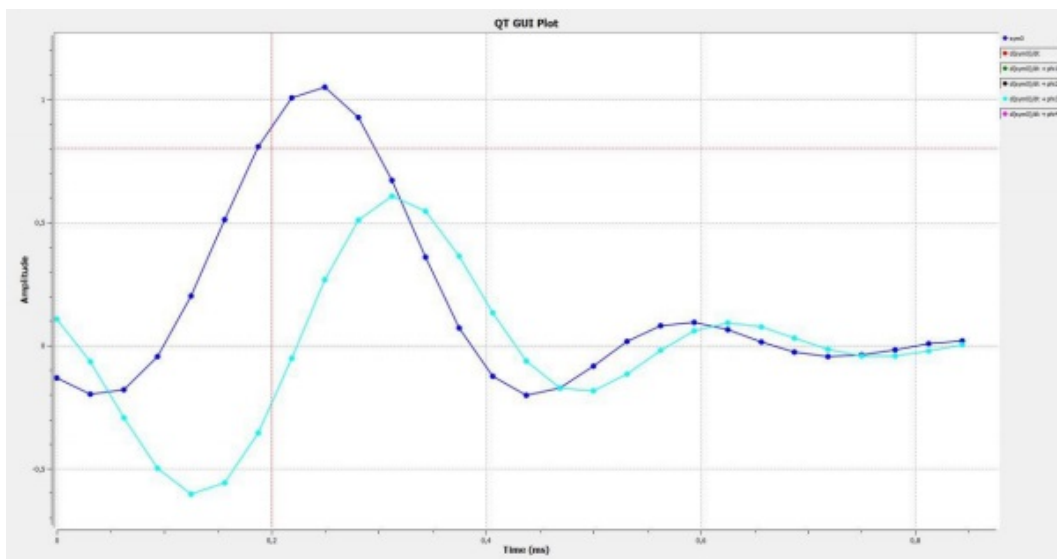


Рис. 12: Графики после блока многофазной фильтрации с 32 фильтрами

Цикл управления переключается между фильтрами в зависимости от ошибки и останавливается на наиболее оптимальном - с наиболее близкой к 0 ошибкой.

## 2.5 Использование блока многофазной синхронизации на приёмнике

Включим данный блок в нашу модель. Блок имеет 32 фильтра и пропускную способность в  $2\pi/100$ . Блок может адаптироваться к значению на основе частоты входного сигнала.

Посмотрим на результаты восстановления сигнального созвездия:

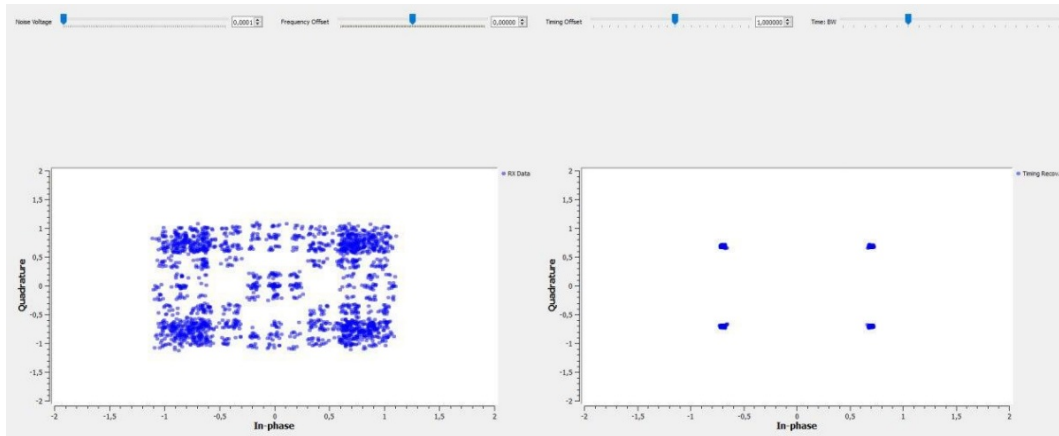


Рис. 13: Принятое и восстановленное сигнальное созвездие

Шумы всё ещё присутствуют как следствие ISI, но при этом почти не заметны. При добавлении смещения частоты созвездие становится круглым:

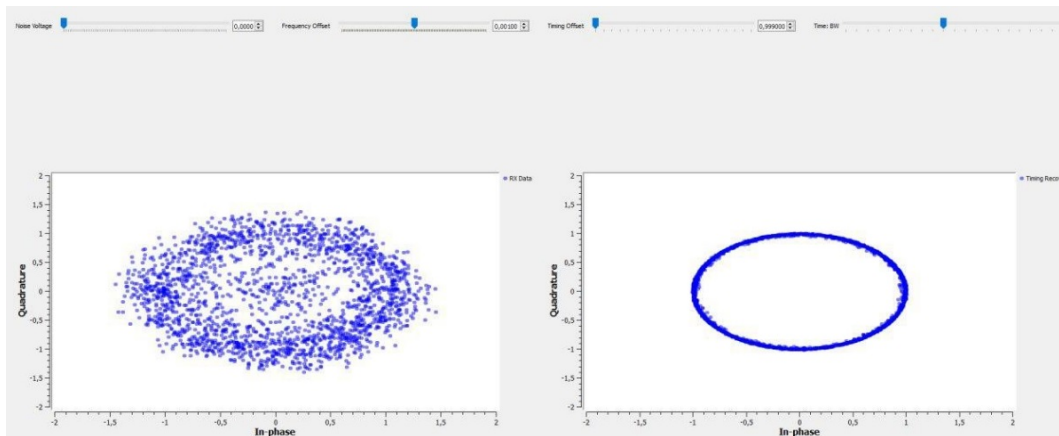


Рис. 14: Принятое и восстановленное сигнальное созвездие с сдвигом

## 2.6 Многолучевое распространение

Феномен многолучевого распространения - результат того, что в большей части коммуникационных сред нет единого пути передачи данных. Например, распространение радиоволн в воздухе - в результате многократного отражения сигнала от зданий, растений, etc. каждый из этих сигналов будет поступать на приёмник в разное время. Суммирование этих сигналов в итоге способно вызывать деструктивные помехи.

ISI может проявляться в том числе из-за описанного выше явления. Такая проблема исправляется эквалайзерами - с их помощью мы можем усилить нужные нам частоты и ослабить вредные.

Следующая модель - модель с пятью ручками эквалайзера, четыре из которых мы способны контролировать. При значении 1 во всех эквалайзерах сигнал проходит беспрепятственно.

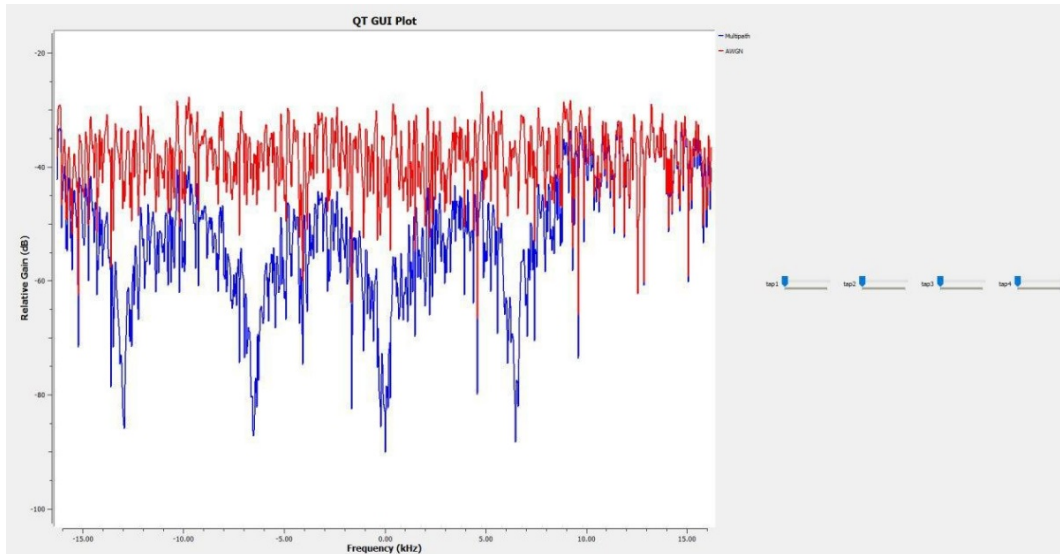


Рис. 15: Модель эквалайзера

В идеале нам надо сделать так, чтобы выход эквалайзера был максимально "плоским" для этого есть специальные алгоритмы.

## 2.7 Эквалайзеры

Вместо ранее использованного "слепого" эквалайзера, который не знает сигнал при принятии решения, LMS-DD эквалайзеру необходимо знание о принятом сигнале для корректной работы. Он использует решения о выборках для исправления taps в эквалайзере.

Этот эквалайзер хорошо работает с модуляциями типа QAM (квадратурной амплитудной модуляцией). Если сигнал слишком плохого качества, эквалайзер сделает только хуже, поэтому распространённая модель - использование "слепого" эквалайзера для получения сигнала достаточного качества для работы LMS-DD.

Продemonстрируем работу LMS-DD эквалайзера:

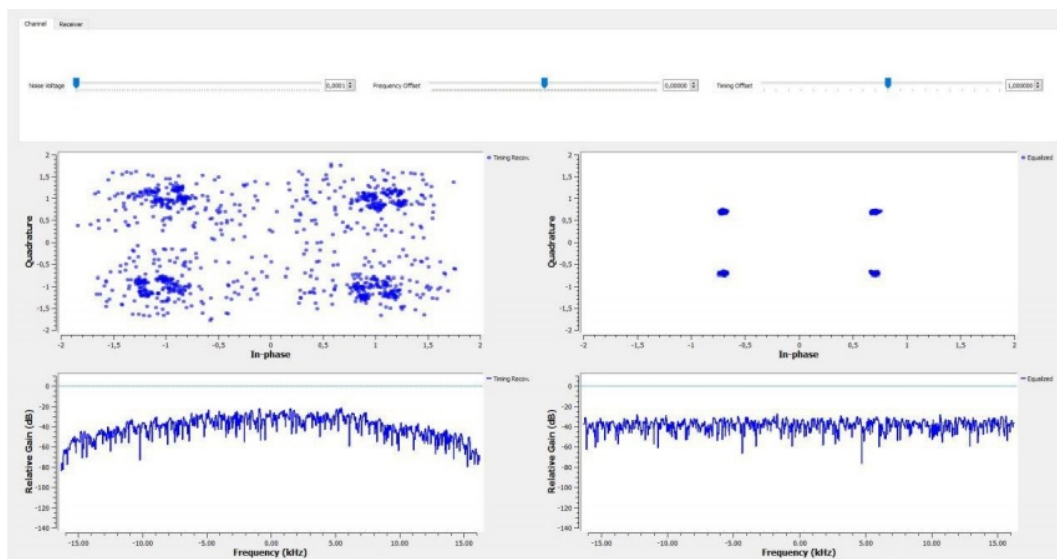


Рис. 16: Модель эквалайзера LMS-DD

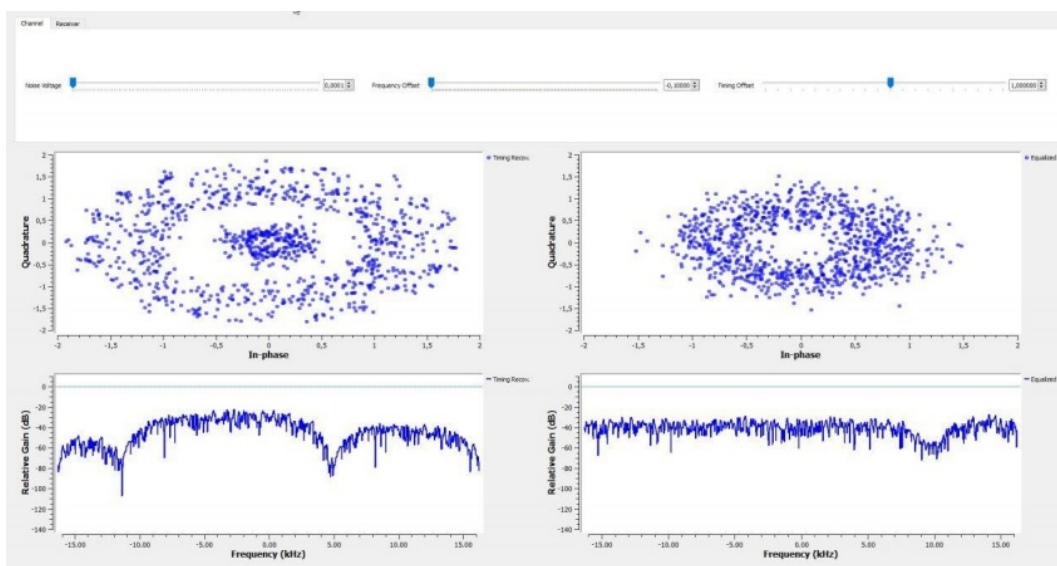


Рис. 17: Модель эквалайзера LMS-DD при смещении частот

## 2.8 Фазовая и точная частотная коррекция

Канал выровнен, но проблема сдвига фазы и частоты при этом осталась.

Для решения этой проблемы воспользуемся модулем Costas Loop, который поддерживает синхронизацию BPSK, QPSK И 8PSK.

Этот блок использует циклы второго порядка, и ему необходимо знать порядок модуляции PSK (2, 4 или 8 соответственно).

Установим небольшой шум, смещение частоты и синхронизации, а также простой многолучевой канал:

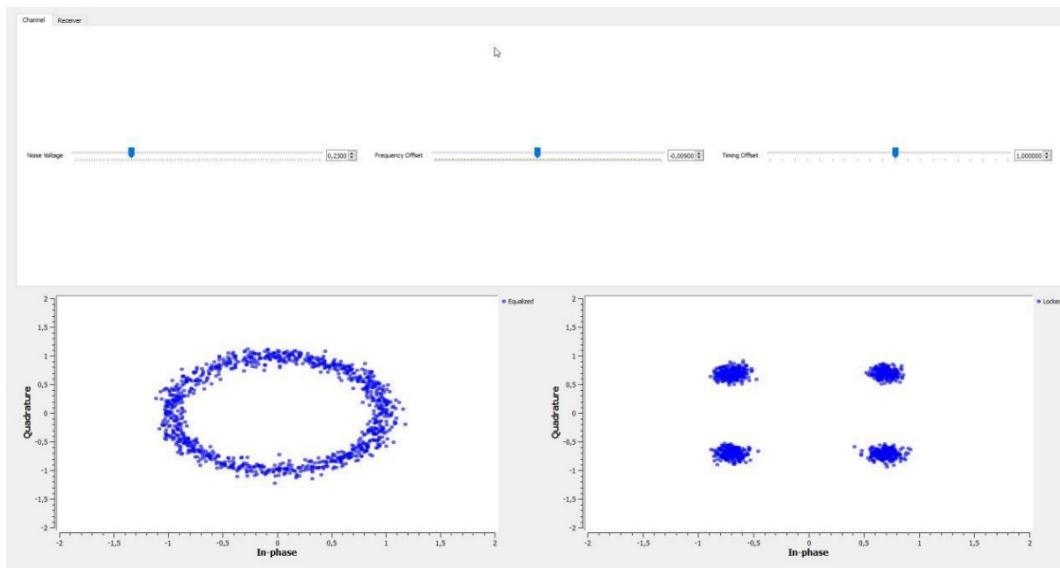


Рис. 18: Симуляция Costas Loop

На выходе блока мы видим, что все символы находятся на единичной окружности созвездия, а так же немного зашумлены, но с этим шумом мы ничего не можем сделать.

## 2.9 Декодирование

Осталась последняя часть передачи данных - их декодирование.

Сейчас мы должны максимально верно интерпретировать некий символ, основываясь на том, что мы получили. Используем блок Map для преобразования символов в исходные. Таким образом, мы получим символы от 0 до 3. Эти два бита данных необходимо распаковать, для этого используем блок распаковки битов. Таким образом, мы получаем оригинальный поток данных. Распакуем значения не в формате 8 бит на байт, а 1 бит на байт, преобразуем эти потоки в значения с плавающей точкой. Принятый сигнал отстаёт на некоторое количество битов относительно передаваемого, поэтому необходимо задержать биты, используя блок Delay:

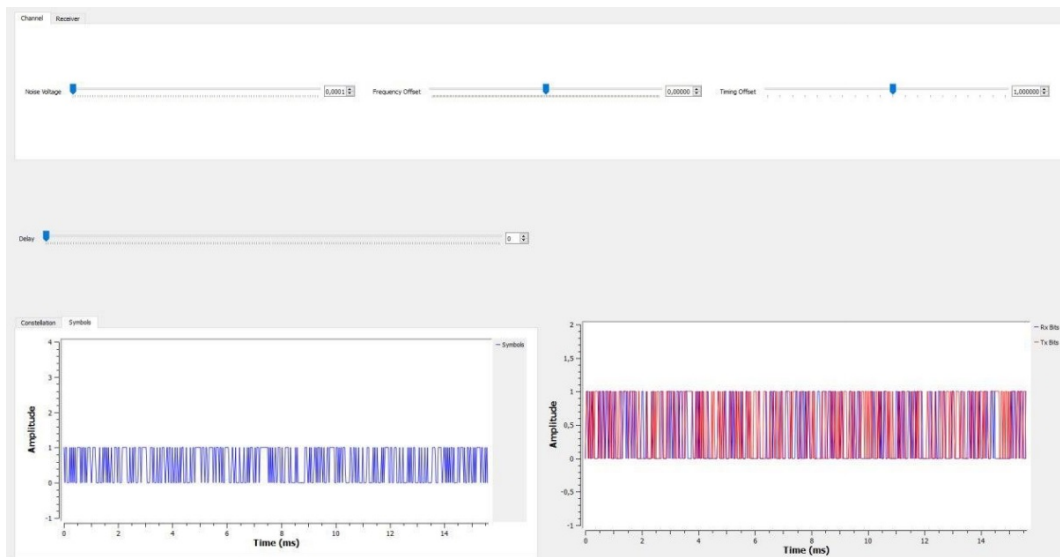


Рис. 19: Результирующий сигнал

### 3 Выводы

В этой лабораторной работе я поближе познакомился с приёмниками сигнала. Они оказались устроены гораздо сложнее, чем передатчики, а также вынуждены решать гораздо более широкий круг проблем: рассинхронизацию, сдвиг частоты, подавление шумов, etc.