

# Лабораторная 12. GNU Radio.

Шерепа Никита

31 мая 2021 г.

## Содержание

<b>1</b>	<b>Передача сигнала</b>	<b>4</b>
<b>2</b>	<b>Добавление канальных искажений</b>	<b>7</b>
<b>3</b>	<b>Временная коррекция</b>	<b>8</b>
3.1	Коррекция тактовых сигналов . . . . .	8
3.2	Информация о блоке синхронизации тактовых сигналов . .	9
3.3	Использование блока многофазной синхронизации в приёмнике . . . . .	12
<b>4</b>	<b>Многолучевое распространение</b>	<b>14</b>
<b>5</b>	<b>Эквалайзеры</b>	<b>16</b>
<b>6</b>	<b>Фазовая и точная частотная коррекция</b>	<b>18</b>
<b>7</b>	<b>Декодирование</b>	<b>19</b>
<b>8</b>	<b>Вывод</b>	<b>21</b>

## Список иллюстраций

1	mpsk_rrc_rolloff граф . . . . .	4
2	Результат . . . . .	5
3	mpsk_stage1 граф . . . . .	5
4	Результат . . . . .	6
5	mpsk_stage2 граф . . . . .	7
6	Результат . . . . .	7
7	symbol_sampling граф . . . . .	8
8	Результат . . . . .	8
9	symbol_sampling_diff граф . . . . .	9
10	Результат . . . . .	9
11	symbol_differential_filter граф . . . . .	10
12	Результат . . . . .	10
13	Добавили тактовое смещение . . . . .	11
14	5 фильтров . . . . .	11
15	Результат . . . . .	12
16	mpsk_stage3 граф . . . . .	12
17	Результат . . . . .	13
18	multipath_sim граф . . . . .	14
19	Результат . . . . .	15
20	mpsk_stage4 граф с CMA . . . . .	16
21	Результат работы CMA . . . . .	16
22	mpsk_stage4 граф с LMS-DD . . . . .	17
23	Результат работы LMS-DD . . . . .	17
24	mpsk_stage5 граф . . . . .	18
25	Результат работы Costas Loop . . . . .	19
26	mpsk_stage6 граф . . . . .	19
27	Результат декодирования . . . . .	20

# 1 Передача сигнала

Первый этап - передача сигнала QPSK. Мы генерируем поток битов и модулируем его на сложное созвездие. Для этого мы используем блок **Constellation Modulator**, который использует **Constellation Rect** (созвездие)

Объект созвездия позволяет нам определить, как кодируются символы. Затем блок модулятора использует эту схему модуляции с или без дифференциального кодирования. Также, для модулятора созвездия, используется генератор случайных битов со значениями от 0 до 255.

Имея дело с количеством выборок на символ, мы хотим, чтобы это значение было как можно меньшим (минимальное значение 2). Здесь происходит моделирование количество отсчетов на символ важно только для того, чтобы обеспечить соответствие скорости передачи по всей потоковой диаграмме, здесь будет 4 что избыточно, но полезно для визуализации

Также установим значение избыточной пропускной способности. Модулятор созвездия использует фильтр формирования импульсов RRC, который позволяет нам изменять один параметр для регулирования roll-off коэффициента фильтра - «альфа».

Граф ниже визуализирует генерацию различных значений избыточной пропускной способности.

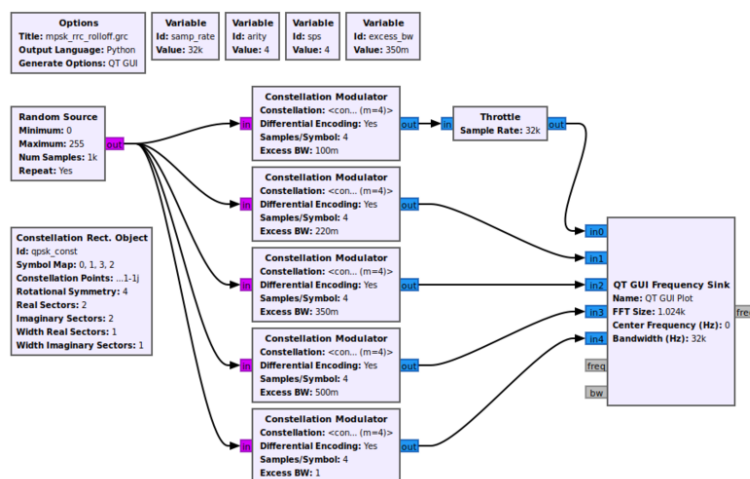


Рис. 1: mpsk\_rrc\_rolloff граф

Получаем следующий результат

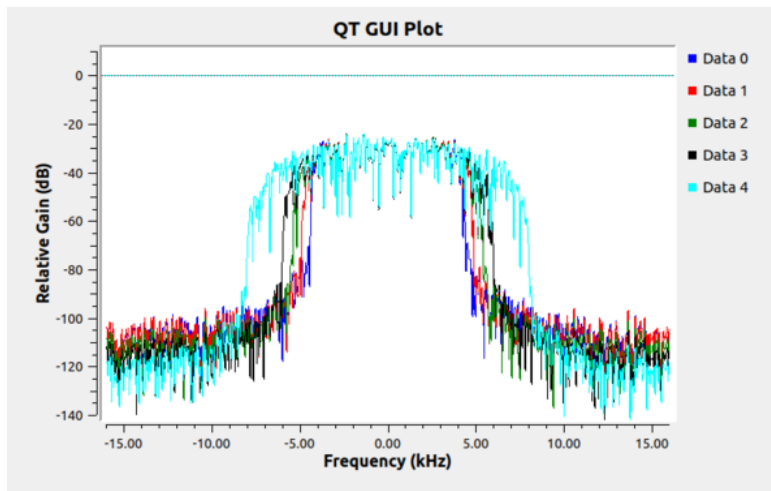


Рис. 2: Результат

Теперь рассмотрим граф, который передает созвездие QPSK. Он отображает передаваемый сигнал и часть цепи приемника во времени, частоте и диаграмме созвездия.

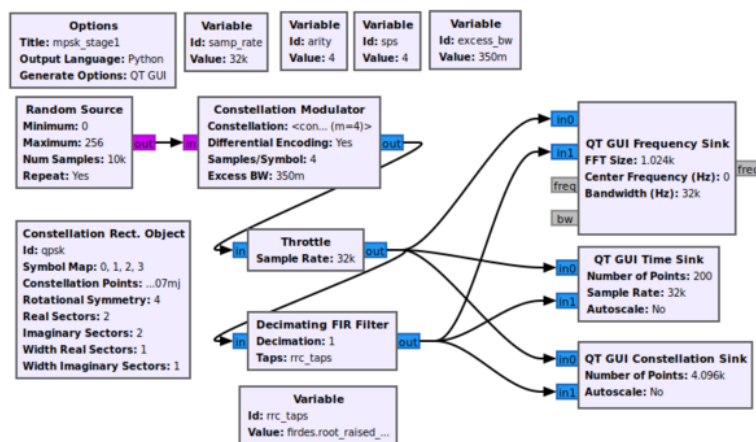


Рис. 3: mpsk\_stage1 граф

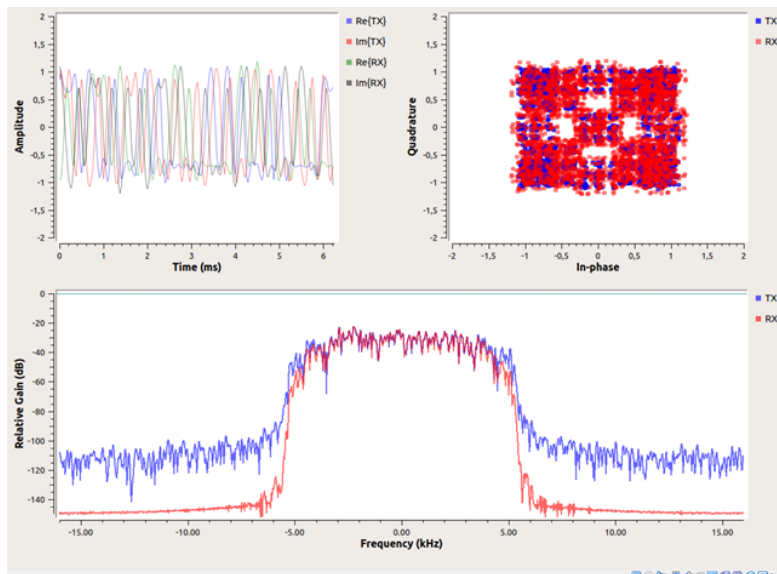


Рис. 4: Результат

Можно увидеть процесс фильтрации и эффект увеличения частоты дискретизации - генерирование 4х выборок на символ.

На стороне приема мы избавляемся от ISI с помощью RRC фильтра. Фильтрация здесь просто свертка, поэтому выходной сигнал RRC-фильтра на приемной стороне представляет собой сигнал в форме приподнятого косинусоидального импульса с минимизированным ISI.

## 2 Добавление канальных искажений

Теперь к передаче сигнала добавим искажений в канал передачи. Воспользуемся блоком **Channel Model**, который позволит смоделировать появление шума, а также различные тактовые сигналы, которые определяют частоту радиомодулей.

Граф ниже позволяет экспериментировать с этими эффектами и наблюдать их влияние на сигнал.

Выставим шум = 0.2, смещение частоты = 0.025, смещение времени = 1.0005

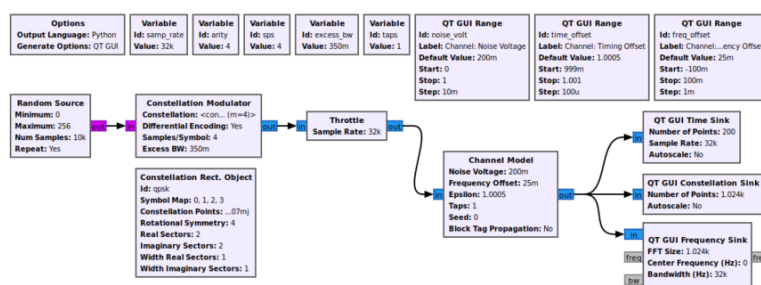


Рис. 5: mpsk\_stage2 граф

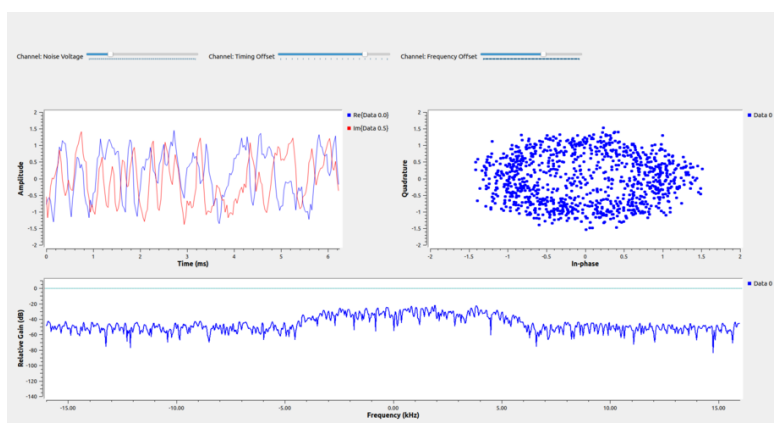


Рис. 6: Результат

Результат получился хуже, чем в прошлом пункте.

## 3 Временная коррекция

### 3.1 Коррекция тактовых сигналов

Теперь попытаемся восстановить сигнал. Начнем с временного восстановления. Попытаемся найти наилучшее время для дискретизации входящих сигналов, что позволит максимизировать SNR (отношение сигнал-шум) каждой выборки, а также уменьшить влияние ISI (межсимвольных помех).

Создадим четыре отдельных символа из единиц и отфильтруем их.

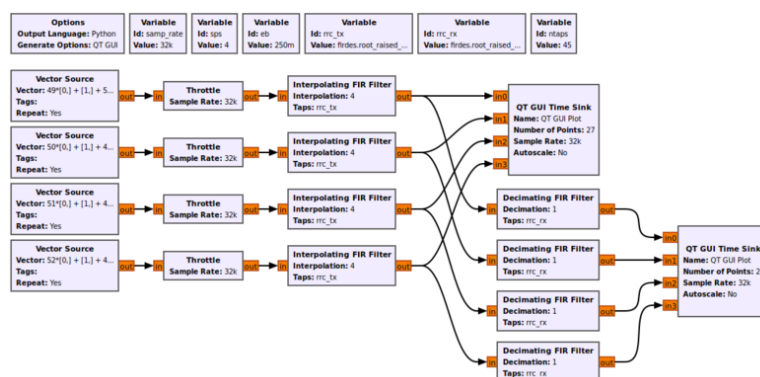


Рис. 7: symbol\_sampling граф

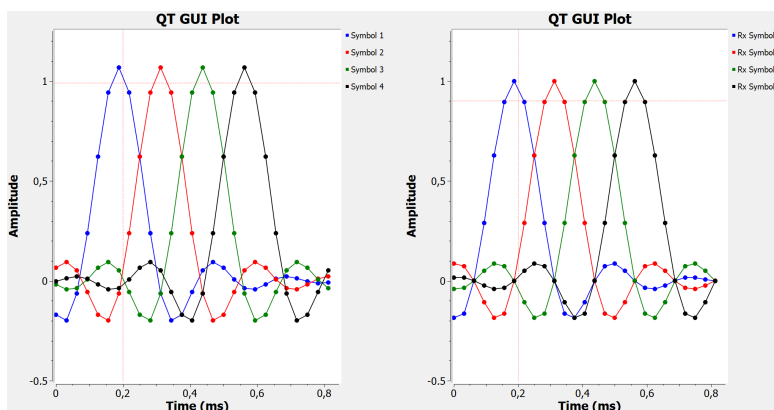


Рис. 8: Результат

Теперь рассмотрим влияние разных тактовых сигналов на точки выборки между передатчиком и приемником. Для модуляции этого добавим Resampler, который регулирует время выборки сигнала между переданным сигналом и приёмником.



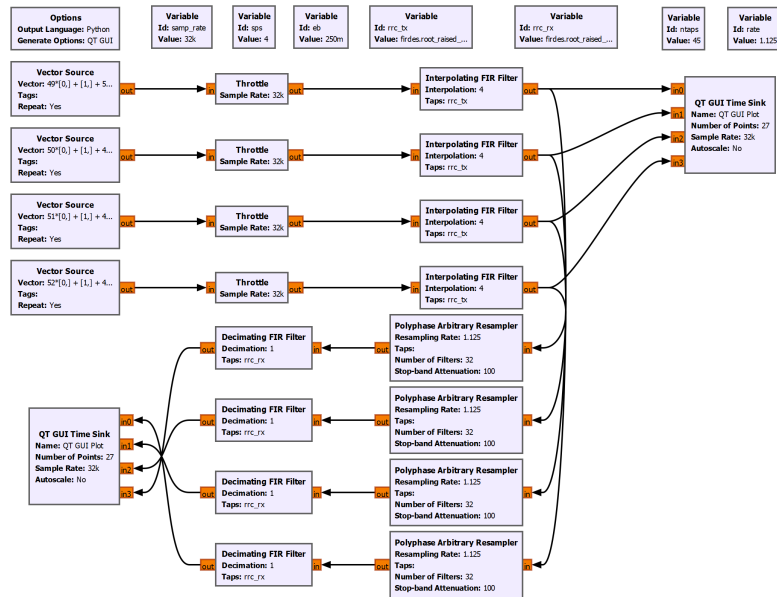


Рис. 9: symbol\_sampling\_diff граф

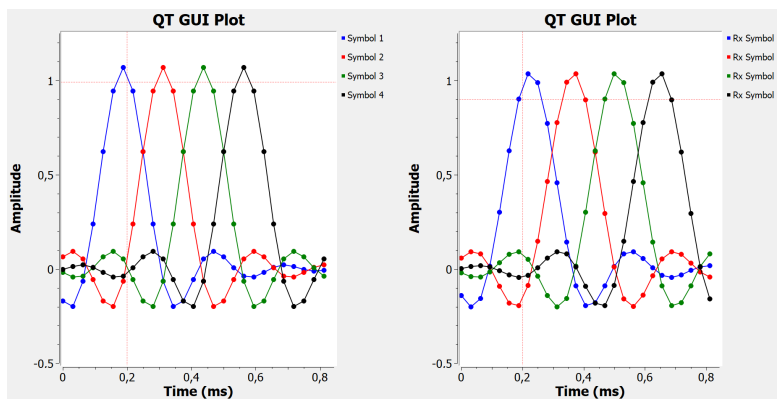


Рис. 10: Результат

### 3.2 Информация о блоке синхронизации тактовых сигналов

Функции, выполняемые блоком синхронизации

1. Восстановление тактовых сигналов
2. Согласованная фильтрация приёмника для устранения ISI

### 3. Уменьшение частоты дискретизации сигнала

Блок синхронизации вычисляет первый дифференциал входящего сигнала, который будет связан с его смещением тактовой частоты.

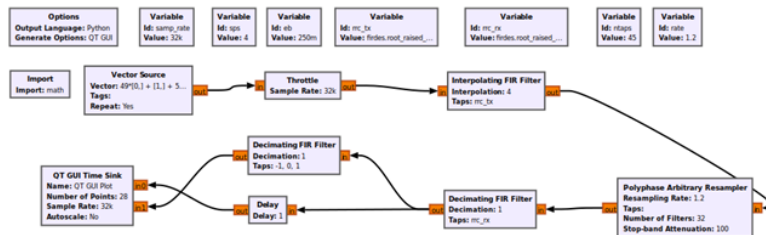


Рис. 11: symbol\_differential\_filter граф

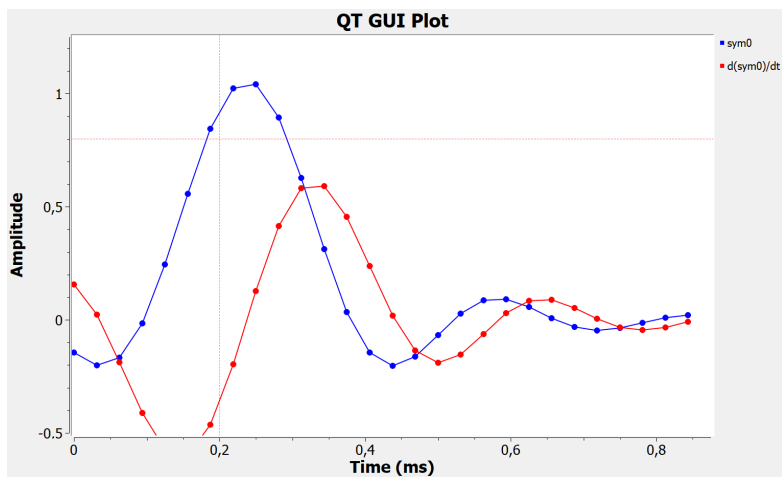
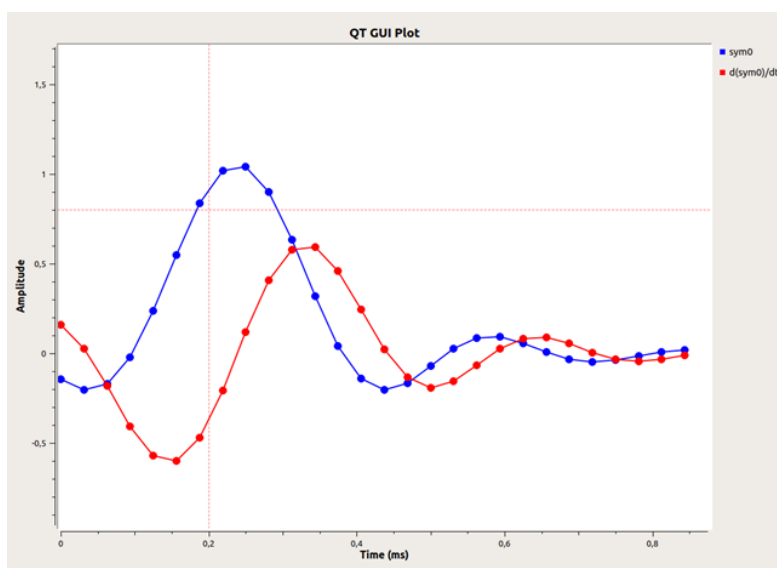


Рис. 12: Результат

Попробуем добавить тактовое смещение



Видим, что в пиковой точке дифференциальный фильтр уже не даёт 0.

Вместо использования одного фильтра можно создать несколько фильтров с разной фазой. Из этих нескольких фильтров можно будет найти фильтр с правильной фазой, которая даст нужное значение синхронизации.

Проведем эксперимент с 5ю фильтрами.

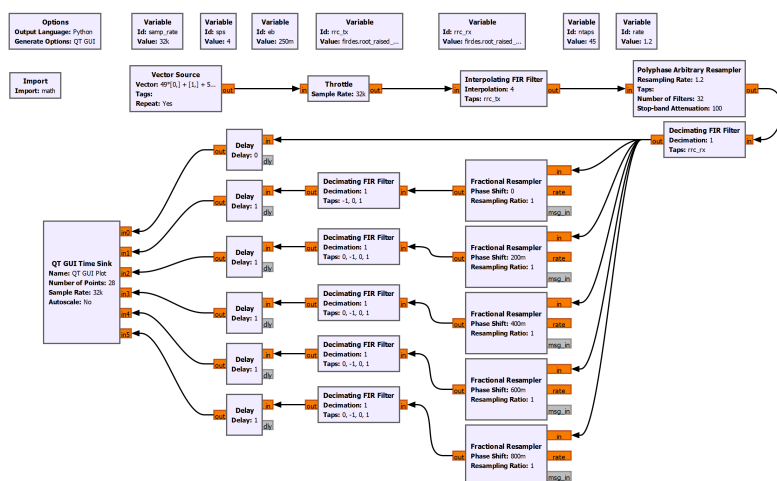


Рис. 14: 5 фильтров

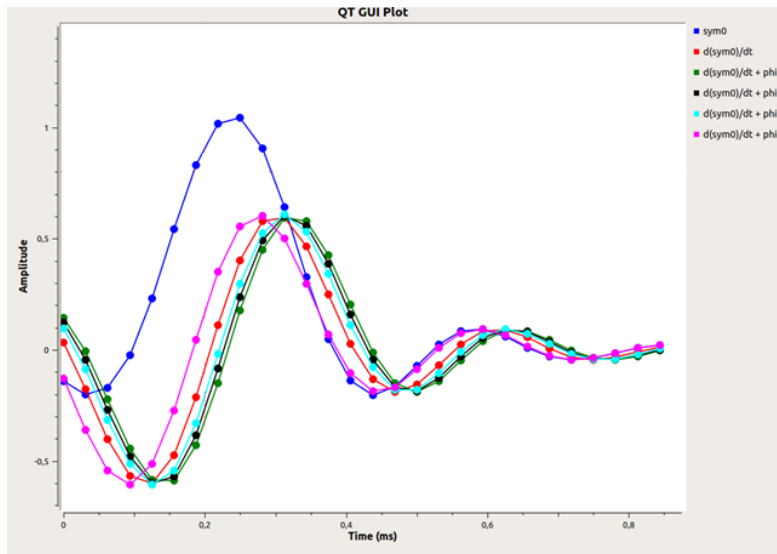


Рис. 15: Результат

Видим, что сигнал  $d(\text{sum0})/dt + \text{ph3}$  в правильной точке выборки даёт 0.

### 3.3 Использование блока многофазной синхронизации в приёмнике

Теперь используем этот блок синхронизации в приёмнике. Настроим его на использование 32 фильтров

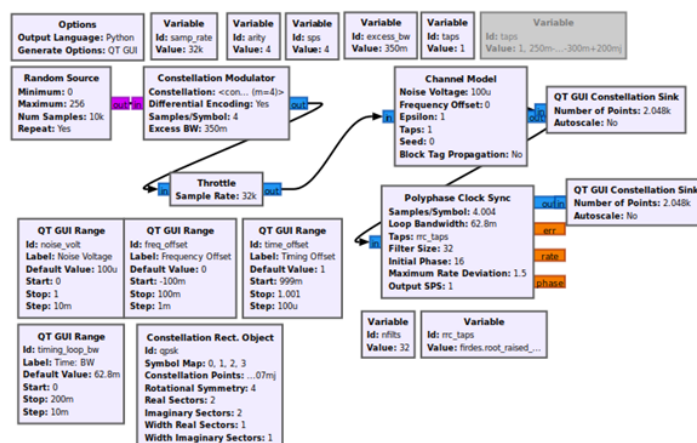


Рис. 16: mpsk\_stage3 граф

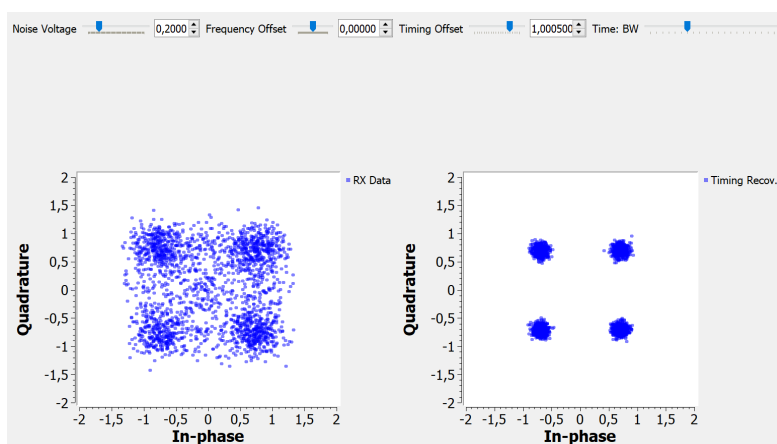


Рис. 17: Результат

Слева - сигнал до восстановления синхронизации

Справа - сигнал после восстановления синхронизации

## 4 Многолучевое распространение

Бывает так, что на пути сигнала от передатчика к приемнику могут появиться какие-нибудь объекты, отражающие сигнал - возникает искажение. В следствии этого, возникает многолучевое распространение - каждый раз сигнал будет проходить по разному пути и в разное время.

Исправить это можно с помощью стерео-эквалайзера: можно усилить определенные частоты, чтобы подавить/усилить сигнал.

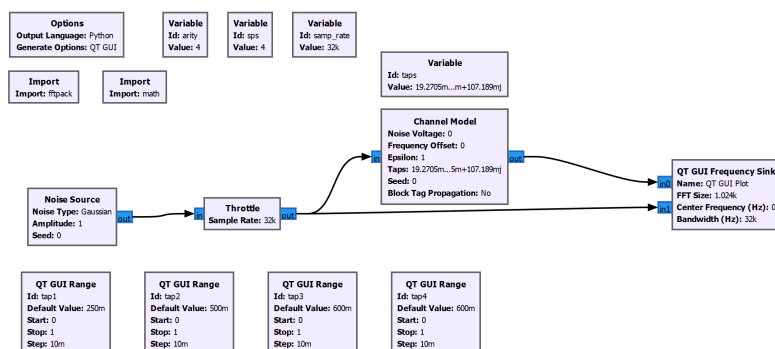


Рис. 18: `multipath_sim` граф

Это моделирование настраивает модель канала, чтобы предоставить каналу пять элементов управления эквалайзером, четыре из которых мы можем изменить. Эти элементы управления настроены одинаково по частоте, и их можно настроить от 0 до 1. При значении 1 элемент управления позволит этим частотам проходить без помех. При значении 0 они будут производить глубокий ноль в спектре, который повлияет на все частоты вокруг него. График частоты установлен на среднее значение.

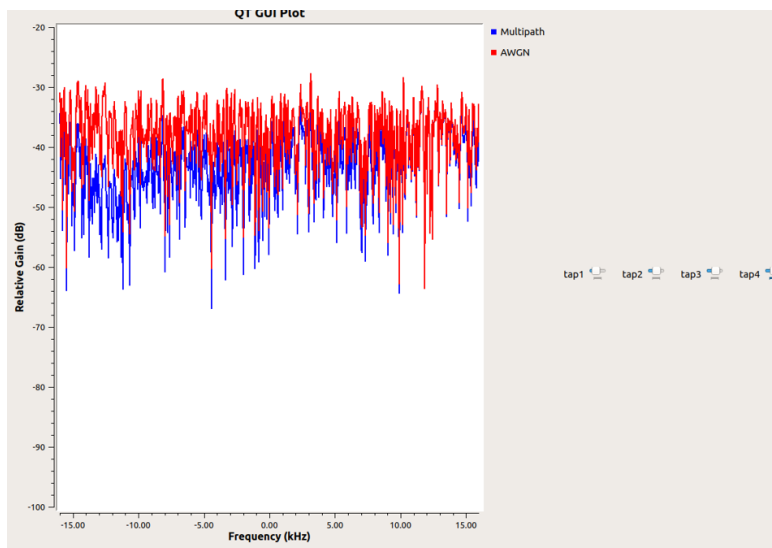


Рис. 19: Результат

Хотя в этом примере мы явно контролируем частотную область, на самом деле мы создаем эквалайзер, который может корректировать или регулировать частотную характеристику принятого сигнала. В итоге, многолучевой канал создает некоторые искажения в сигнале, как показано в частотной области. Задача эквалайзера - инвертировать этот канал.

## 5 Эквалайзеры

В GNU Radio используется два эквалайзера

1. CMA Equalizer - "слепой" эквалайзер, который работает только с сигналами с постоянной амплитудой или модулем
2. LMS-DD Equalizer - должен знать точки созвездия, чтобы понимать, как обновлять параметры эквалайзера. Более сложен в вычислениях. Требуется большей производительности.

Рассмотрим работу CMA Equalizer.

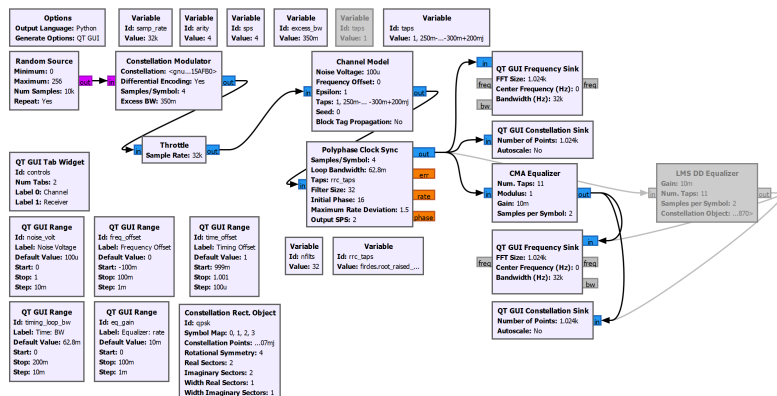


Рис. 20: mpsk\_stage4 граф с CMA

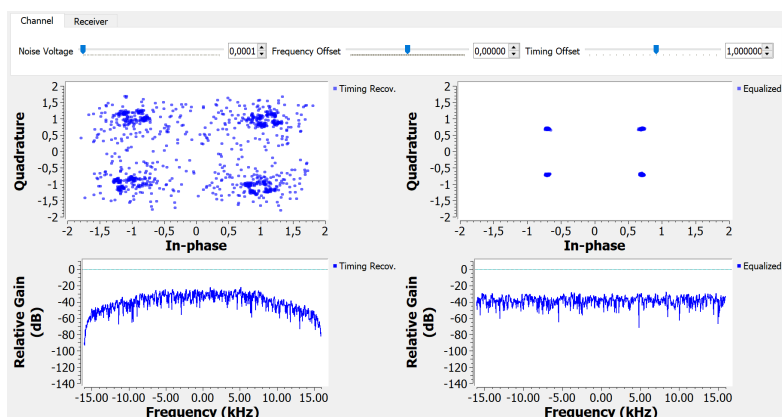


Рис. 21: Результат работы CMA

Видим эффект синхронизированного по времени многолучевого сигнала до и после эквалайзера.



Перед эквалайзером - некрасивый сигнал без шумов  
Эквалайзер понимает что нужно инвертировать, чтобы получился чистый сигнал.

В конце концов видим, что канал выравнился после работы эквалайзера.

Теперь рассмотрим работу LMS-DD Equalizer

Этот эквалайзер хорошо подходит для сигналов, которые не подходят под требование постоянной амплитуды, поэтому его можно использовать и для QAM-модуляции.

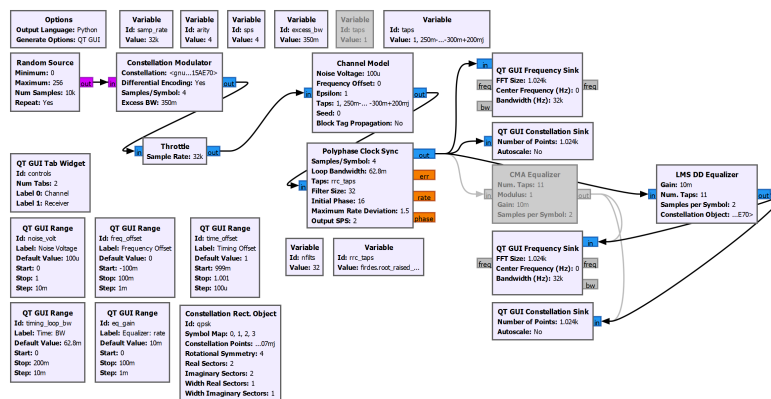


Рис. 22: mpsk\_stage4 граф с LMS-DD

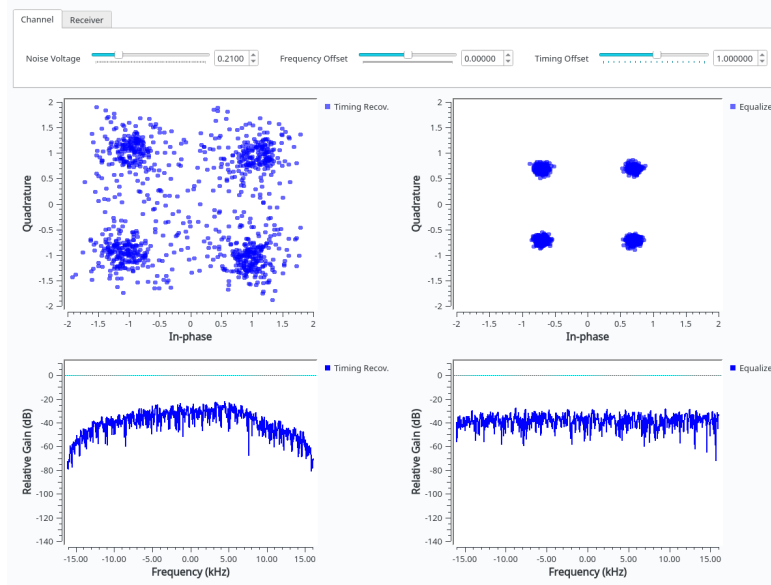


Рис. 23: Результат работы LMS-DD

Видим, что канал успешно выравнялся.

## 6 Фазовая и точная частотная коррекция

Учитывая, что мы выровняли канал, у нас все еще есть проблема смещения фазы и частоты. Эквалайзеры, как правило, не адаптируются быстро, поэтому смещение частоты может быть легко за пределами возможностей эквалайзера. Кроме того, если мы просто запускаем эквалайзер СМА, все, о чем он заботится, - это схождение к единичной окружности. Он ничего не знает о созвездии, поэтому, когда он блокируется, он блокируется на любой заданной фазе. Теперь нам нужно исправить любой сдвиг фазы, а также любой сдвиг частоты.

Для этой задачи мы собираемся использовать цикл Костаса. Блок Costas Loop может синхронизировать BPSK, QPSK и 8PSK. Приемник созвездия будет привязан к любому заданному объекту созвездия, хотя в зависимости от созвездия функция принятия решения может быть более или менее сложной.

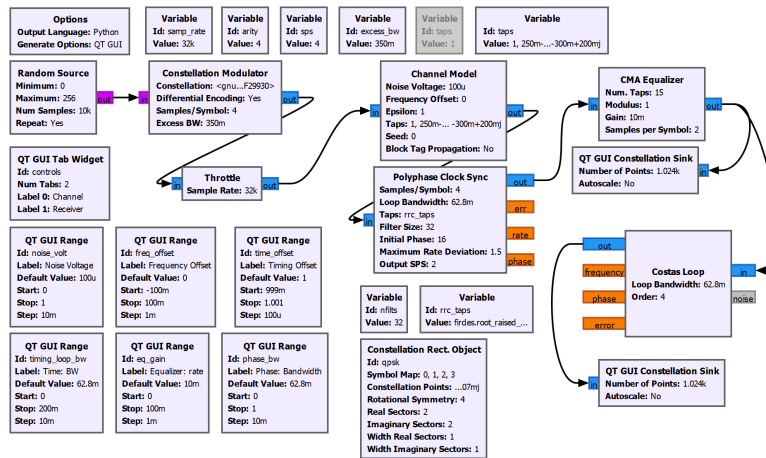


Рис. 24: mpsk\_stage5 граф

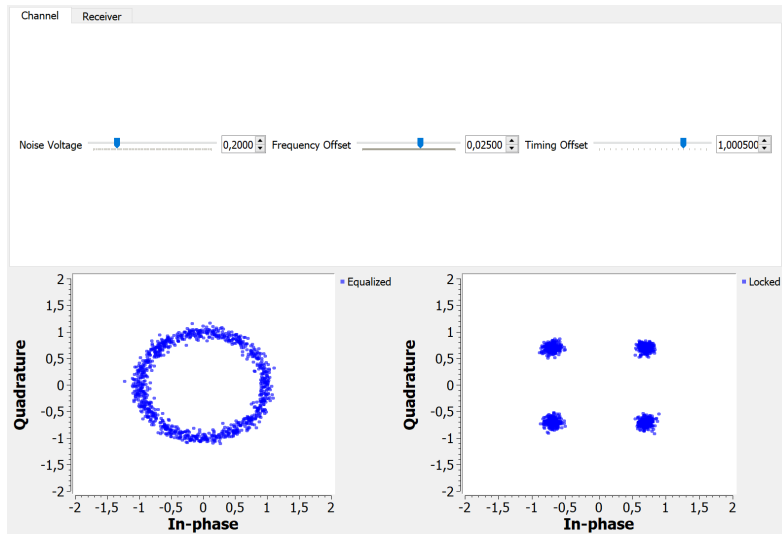


Рис. 25: Результат работы Costas Loop

Видим, что в результате обработки сигнал избавился от искажений, хотя небольшой шум остался.

## 7 Декодирование

Теперь попробуем декодировать сигнал. Вставим Constellation Decoder после Costas Loop и изменим параметр в Differential на false в блоке Constellation Modulator, тем самым выключив дифференциальное кодирование

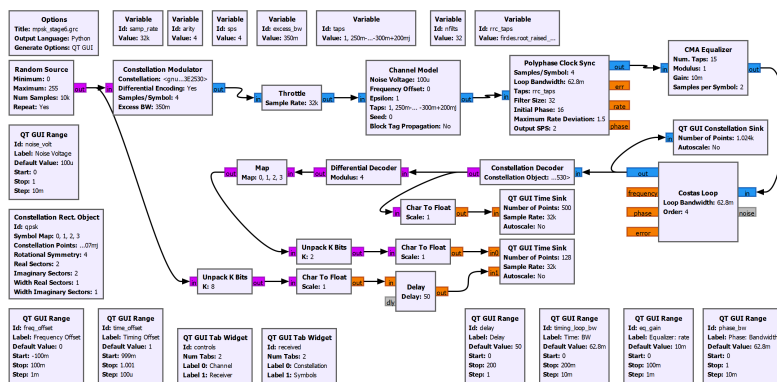


Рис. 26: mpsk\_stage6 граф

Также граф использует блок Map для преобразования символов из

дифференциального декодера в исходные символы и блок **Unpack Bits** для распаковки полученных битов. Блок **Delay** используется для компенсации узлов, которые задерживают данные.

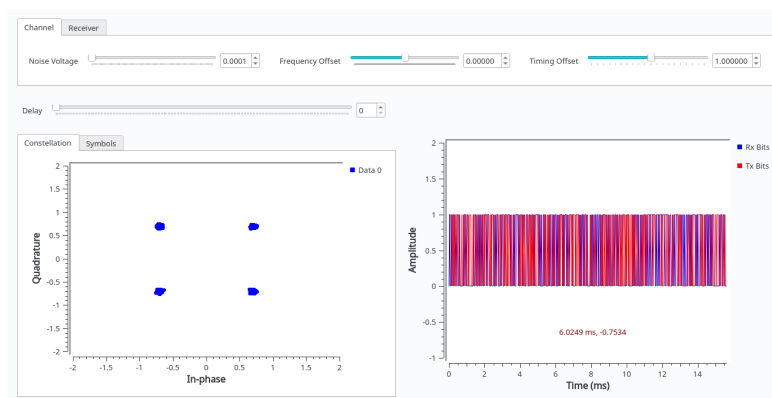


Рис. 27: Результат декодирования

Получили исходный поток данных.

## 8 Вывод

В результате выполнения работы получены навыки работы с QPSK-сигналами: мы моделировали условия передачи данных, выявляли проблемы которые могут возникнуть при передаче данных (например, канальные искажения) и рассматривали способы решения этих проблем (например, восстанавливали сигнал с помощью эквалайзеров).