

Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

GNU Radio

Работу выполнил студент
3-го курса, группа 3530901/80201
Солянкин Илья Андреевич

Преподаватель:
Богач Наталья Владимировна

Санкт-Петербург 2021

Содержание

1	Постановка задачи	5
2	Часть №1: Передача сигнала	6
3	Часть №2: Добавление канала передачи	8
4	Часть №3: Восстановление времени	11
5	Часть №4: Восстановление синхронизации	16
6	Часть №5: Многолучевое распространение	18
7	Часть №6: Эквалайзеры	20
8	Часть №7: Фазовая и частотная коррекция	23
9	Часть №8: Декодирование	25
10	Выводы	27

Список иллюстраций

1	Передача созвездия QPSK	6
2	График QPSK	7
3	Добавление канала передачи	9
4	Первый график результата	9
5	Второй график результата	10
6	Схема без рассинхронизации	11
7	Передача четырех символов без рассинхронизации	12
8	Схема со рассинхронизацией	12
9	Передача четырех символов с рассинхронизацией	13
10	Схема	13
11	Синхронизация	14
12	Схема со синхронизацией	15
13	Расширенный экран синхронизации	15
14	Схема 3 шага	16
15	Первая схема синхронизации	17
16	Вторая схема синхронизации	17
17	Схема	19
18	График многолучевого распространения	19
19	Эквализация СМА	20
20	Результат для СМА	21
21	Эквализация LMS	22
22	Результат для LMS	22

23	Фазовая и высокочастотная коррекция	23
24	Первый результат фазовой и высокочастотной коррекции	24
25	Результат фазовой и высокочастотной коррекции	24
26	Передача созвездия QPSK	25
27	Вид результата	26

1 Постановка задачи

В данной лабораторной работе нам необходимо для начала обнаружить причины искажения сигнала и проблемы эффектов канала, после чего распознать этапы, которые будут необходимы для восстановления сигналов, а именно:

- Сроки восстановления
- Многолучевые каналы
- Фазовая и частотная корреляция
- Декодирования символов и упорядочение блоков

Помимо перечисленного необходимо разработать передатчик и приемник сигнала с использованием квадратурной фазовой манипуляции средствами GNU Radio, и рассмотреть этапы с использованием алгоритмов уже доступных в GNU Radio для приема и демодуляции сигнала с реализацией фазовой манипуляции.

2 Часть №1: Передача сигнала

В первой части нам необходимо создать квадратурную фазовую манипуляцию и передачу модифицированного сигнала. Генерируется сигнал (случайный поток битов), после чего он моделируется в сложное созвездие. Используется блок Constellation Modulator, который принимает объект Constellation и применяет различные настройки для управления передаваемым сигналом. Объект созвездия позволяет определить, каким образом кодируются символы. Модулятор Constellation использует фильтр формирования импульсов RRC (root raised cosine), который предоставляет единственный параметр для настройки коэффициента спада фильтра - «альфа».

Полученная диаграмма передает созвездие QPSK. Он строит как переданный сигнал, так и часть цепи приемника во времени, частоте и графике созвездия. ИТакже стоит отметить, что имеются 3 ограничения: сигнал обязан укладываться в выделенную ему полосу пропускания. Передаваемые символы должны быть различимы. И, наконец, даже, если канал передачи не наложит значительных искажений, от шума все-равно не избавиться, поэтому передача сигнала должна обеспечивать хорошее соотношение сигнал/шум.

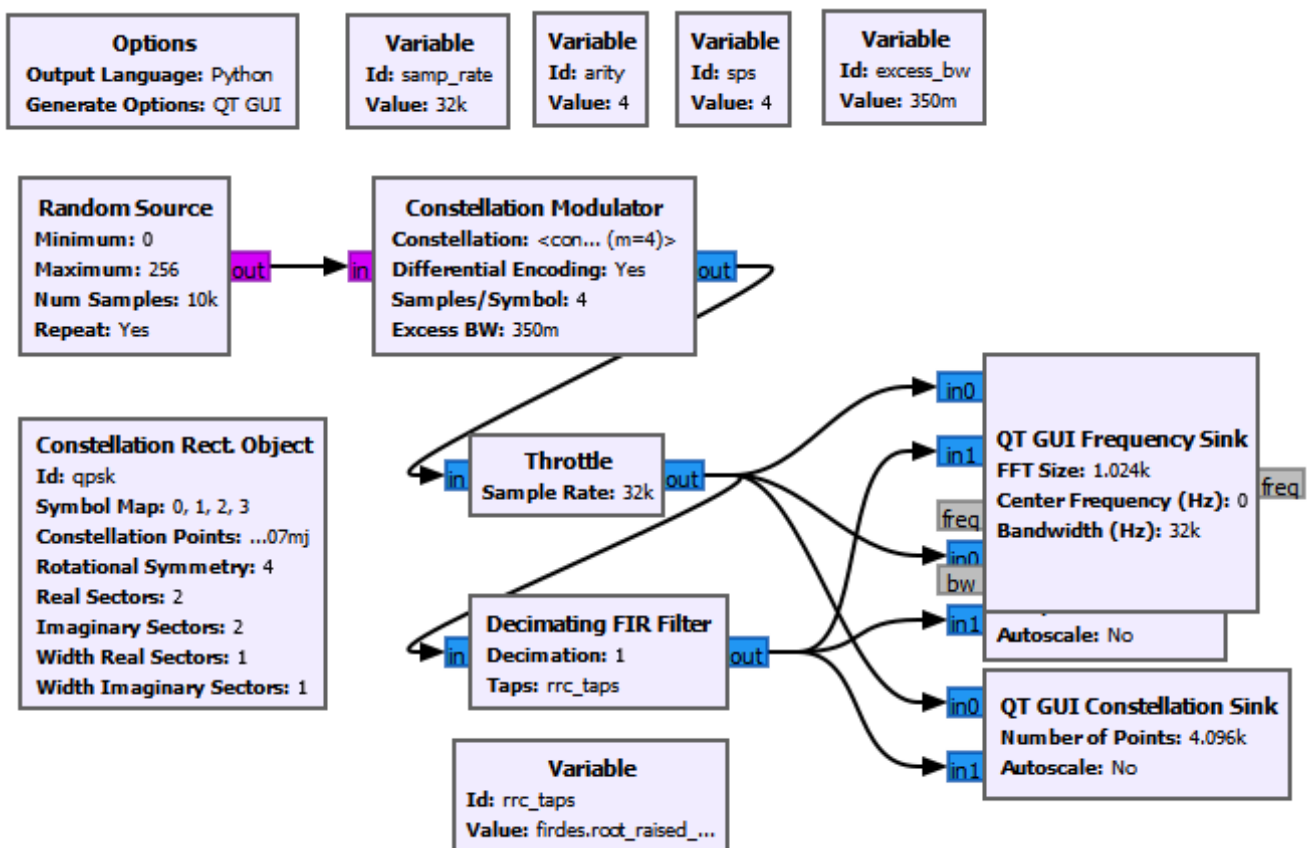


Рис. 1: Передача созвездия QPSK

Ниже представлен график, демонстрирующий характеристики сигнала в “канале передачи” и на приемнике.

На графике Constellation виден эффект повышения частоты дискретизации и фильтрации. В этом случае фильтр RRC добавляет собственные помехи, известные как межсимвольные помехи (ISI). ISI не является подходящим для принятого сигнала, потому что он размывает символы. Если посмотреть на передаваемые сигналы с данного графика, можно видеть, что частотный график показывает сигнал с хорошей формой, который скатывается в шум. Если бы формирующий фильтр не был установлен на сигнал, передавались бы прямоугольные волны, которые производят много энергии в соседних каналах. Благодаря уменьшению внеполосных излучений сигнал остается в пределах полосы пропускания канала.

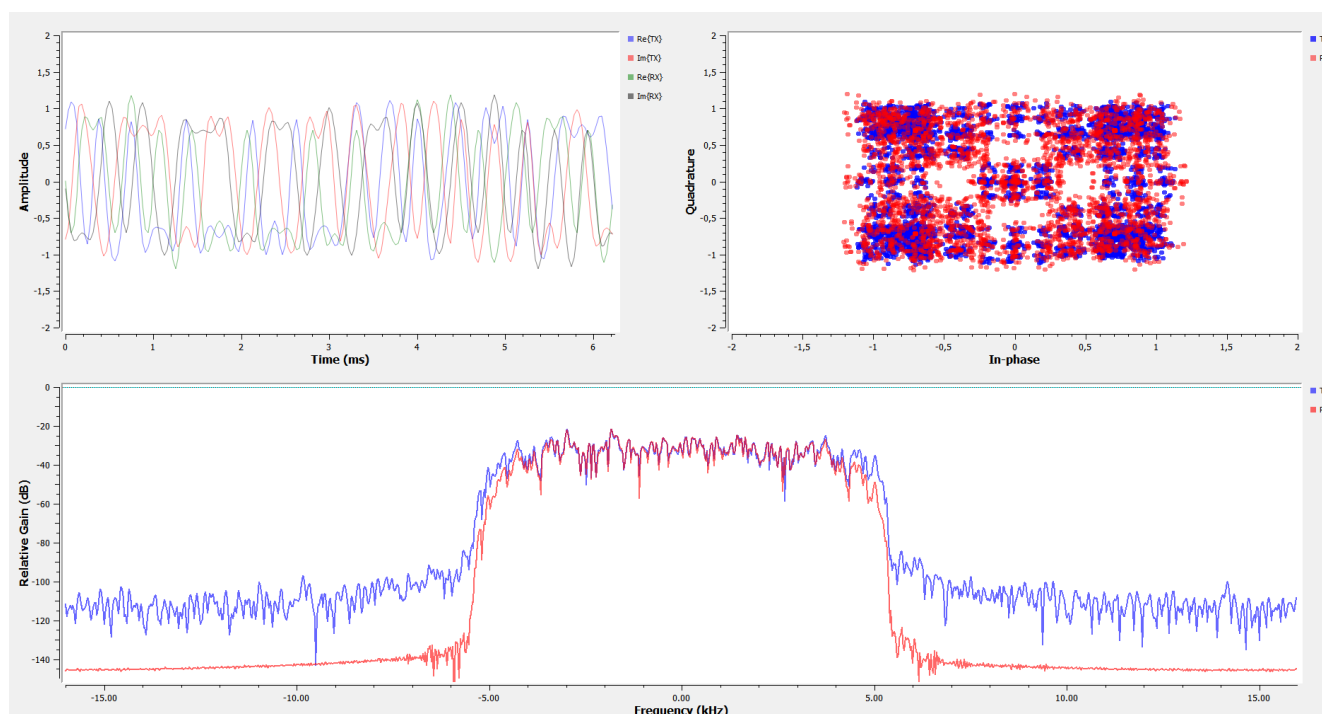


Рис. 2: График QPSK

3 Часть №2: Добавление канала передачи

Во второй части лабораторной работы будут рассмотрены эффекты канала, то, как сигнал искажается между передачей и видимостью в приемнике. Для начала будет использоваться самый базовый блок Channel Model в GNU Radio. Этот блок позволяет смоделировать несколько основных проблем, с которыми далее приходится иметь дело. Первая проблема с приемниками - возникающий аддитивным белый гауссовый шум (AWGN). Устанавливается мощность шума, регулируя значение шумового напряжения модели канала. Здесь указывается напряжение вместо мощности, потому что необходимо узнать ширину полосы сигнала, чтобы правильно рассчитать мощность. Другой существенной проблемой является нахождение идеальной точки выборки. Была увеличена частота дискретизации сигнала в передатчике, и он был сформирован, но при его получении необходимо произвести выборку сигнала в исходной точке выборки, чтобы максимизировать мощность сигнала и минимизировать межсимвольные помехи. Как и в имитации этапа 1 после добавления второго фильтра RRC, можно видеть, что среди четырех выборок на символ одна из них находится в идеальной точке выборки +1, -1 или 0. Но опять же, две радиостанции работают на разных скоростях, поэтому идеальная точка отбора проб неизвестна.

Второй этап моделирования позволяет провести эксперименты над эффектами аддитивного шума, смещения частоты и смещения синхронизации. Было добавлено немного шума (0.2), некоторое смещение частоты (0.025) и некоторое временное смещение (1.0005), чтобы увидеть результирующий сигнал с начального его положения.

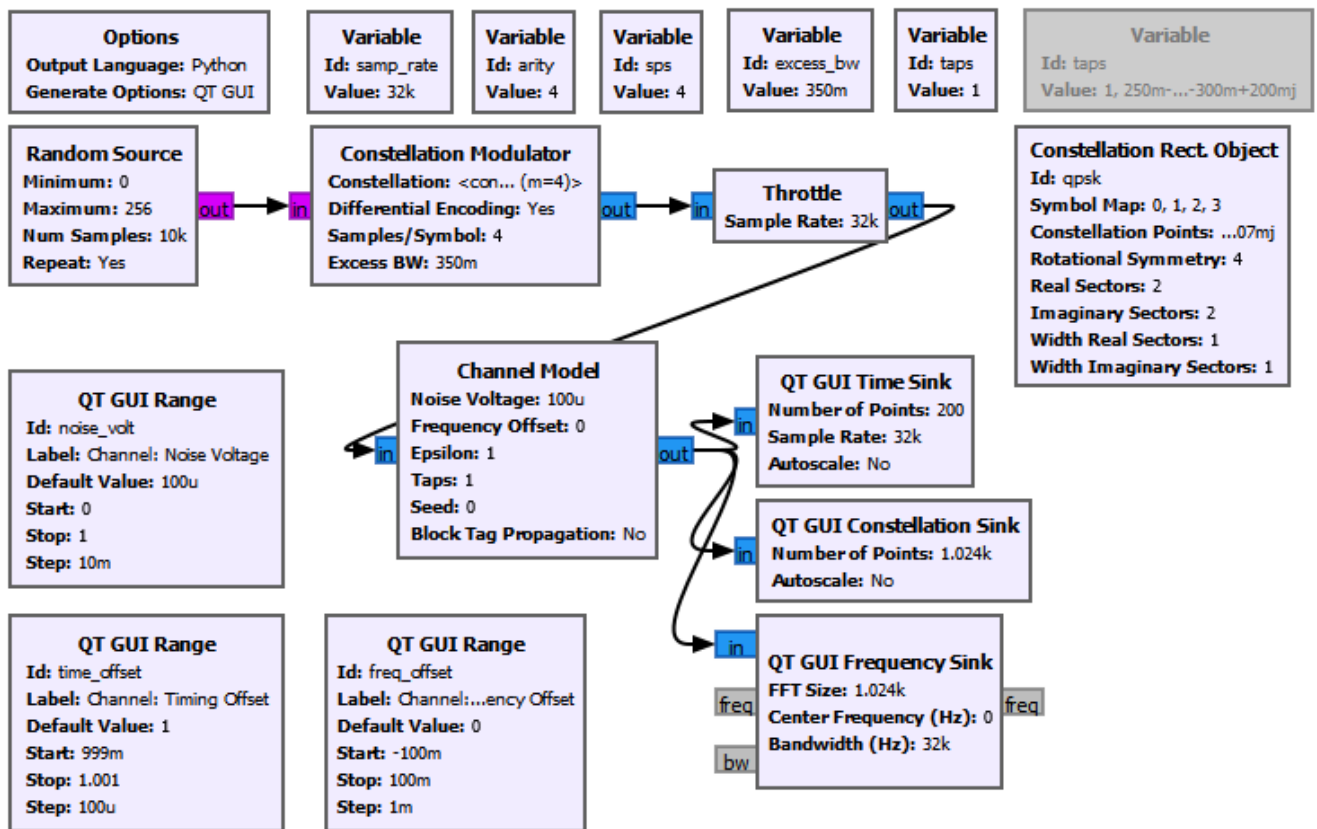


Рис. 3: Добавление канала передачи

На графиках результата второго этапа изображенв исходный сигнал с наложенными на него искажениями канала. Из исходного созвездия мы получили облако точек, из которого нужно будет в дальнейшем выделить наш сигнал.

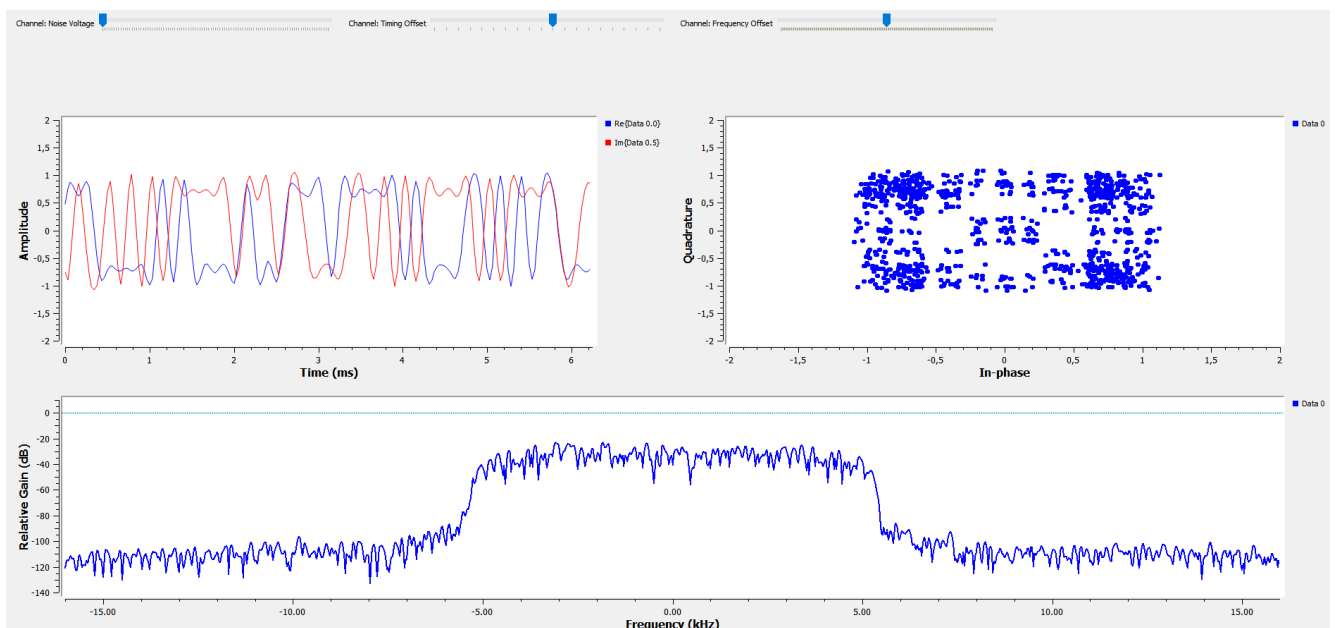


Рис. 4: Первый график результата

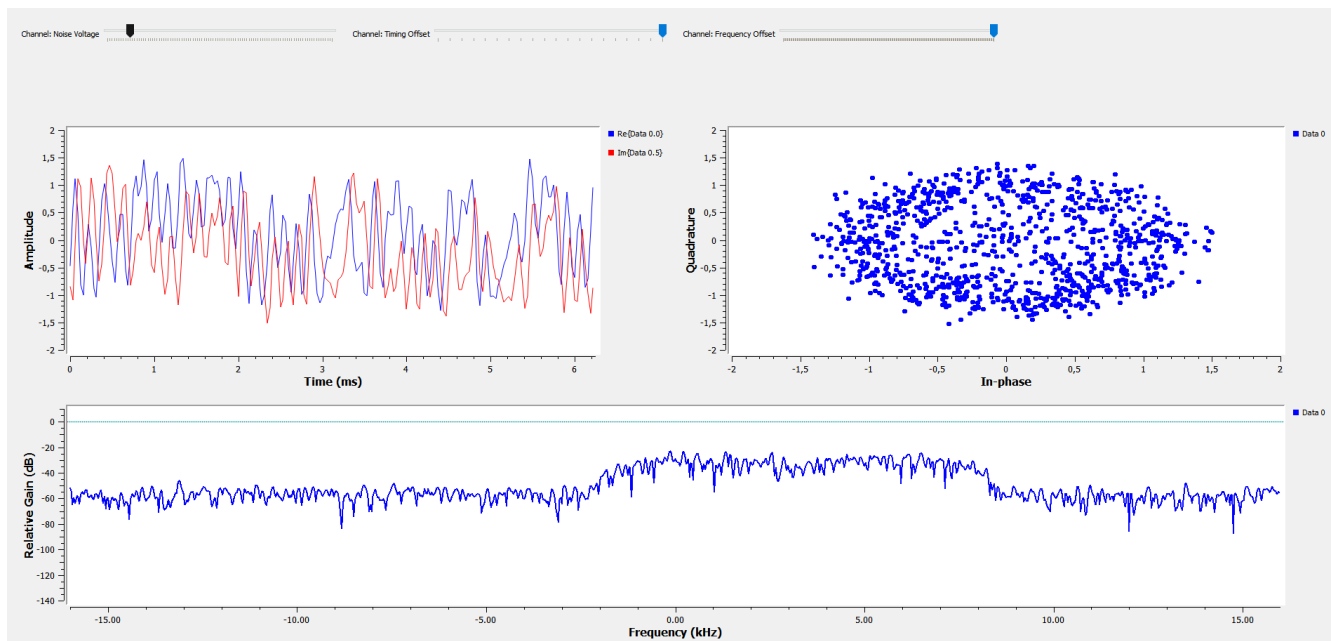


Рис. 5: Второй график результата

4 Часть №3: Восстановление времени

В третьей части будет пройден процесс восстановления. Будет использоваться алгоритм *polyphase clock recovery*. Первым шагом будет восстановление времени. Задачей является найти лучшее время для выборки входящих сигналов, что позволит максимизировать отношение сигнал / шум (SNR) каждого образца, а также уменьшить влияние межсимвольных помех (ISI).

Можно проиллюстрировать проблему ISI, используя пример flowgraph *symbolsampling*. Первый этап фильтрации изменяет частоту выборки до значения 'sps' на элемент и использует ISI root raised cosine фильтр. За этим следует другой root raised cosine фильтр, который не изменяет скорость. Вторым фильтром RRC здесь преобразует сигналы от использования не-Nyquist RRC-фильтра в Nyquist raised cosine (RC) фильтр. Можно наблюдать как восстановление синхронизации применяет согласованный фильтр для удовлетворения критерию Nyquist ISI.

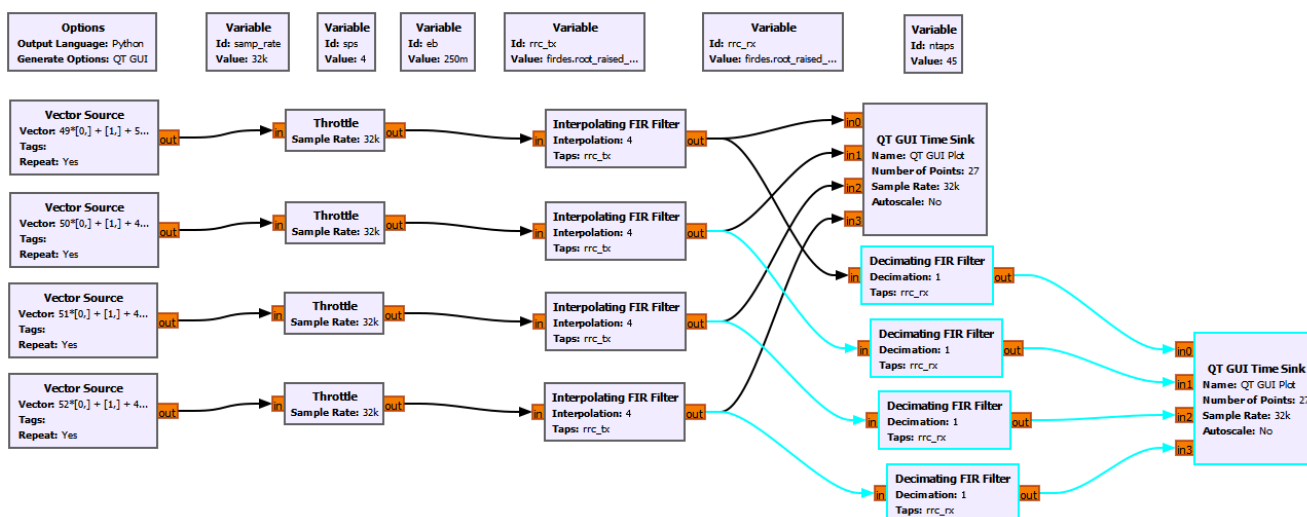


Рис. 6: Схема без рассинхронизации

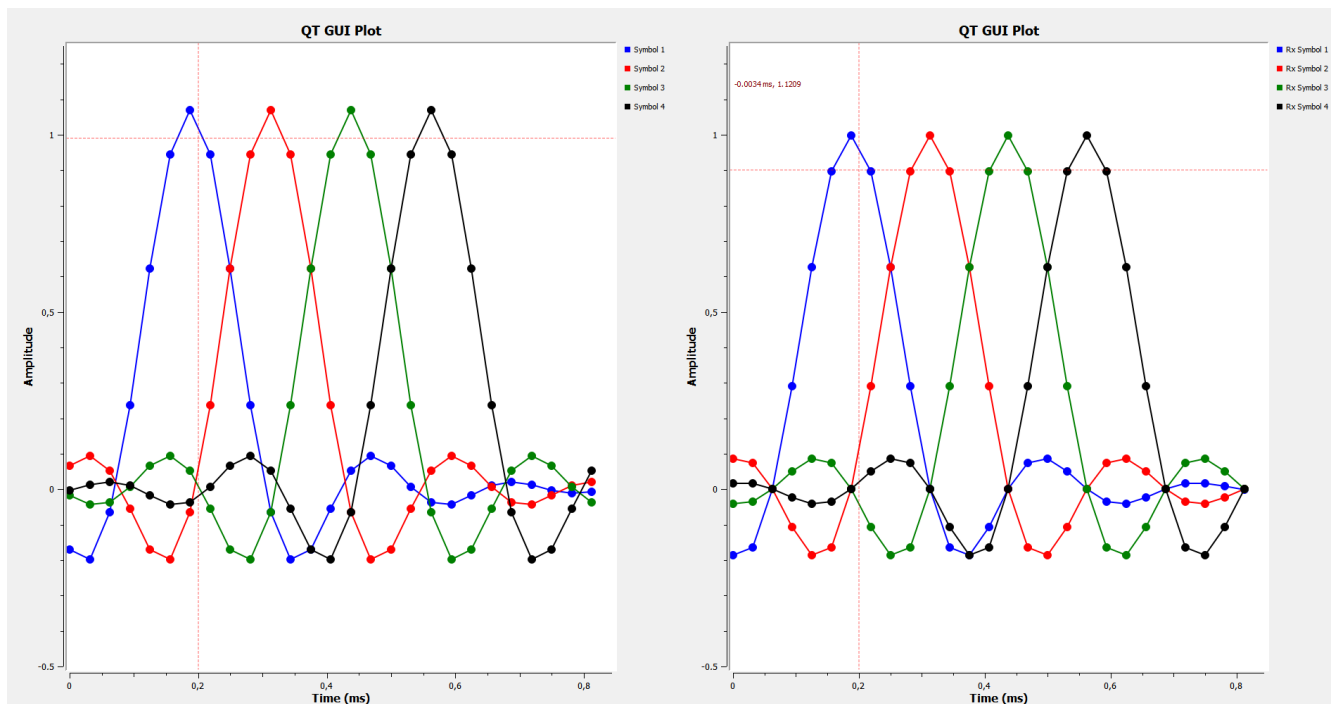


Рис. 7: Передача четырех символов без рассинхронизации

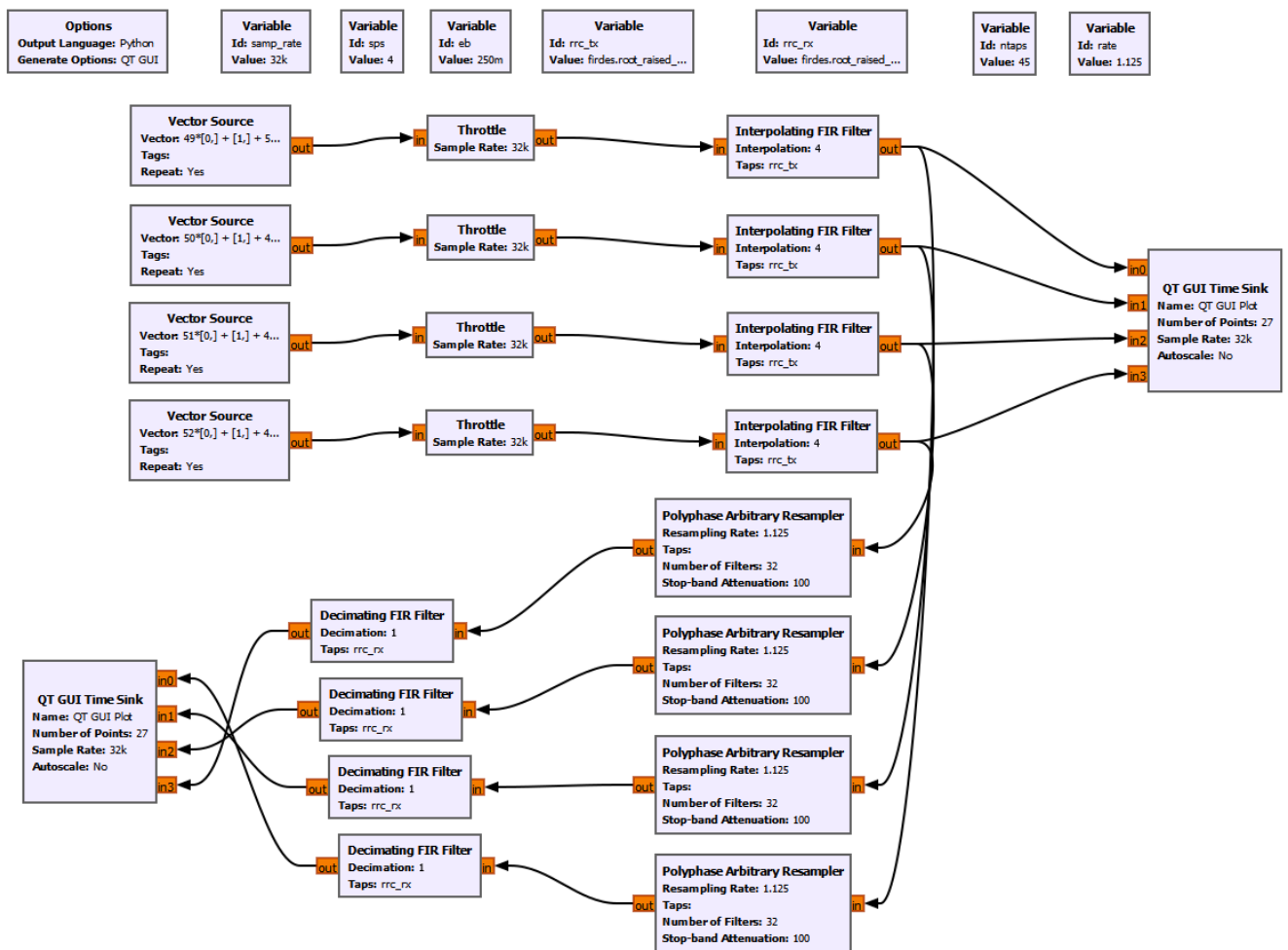


Рис. 8: Схема со рассинхронизацией

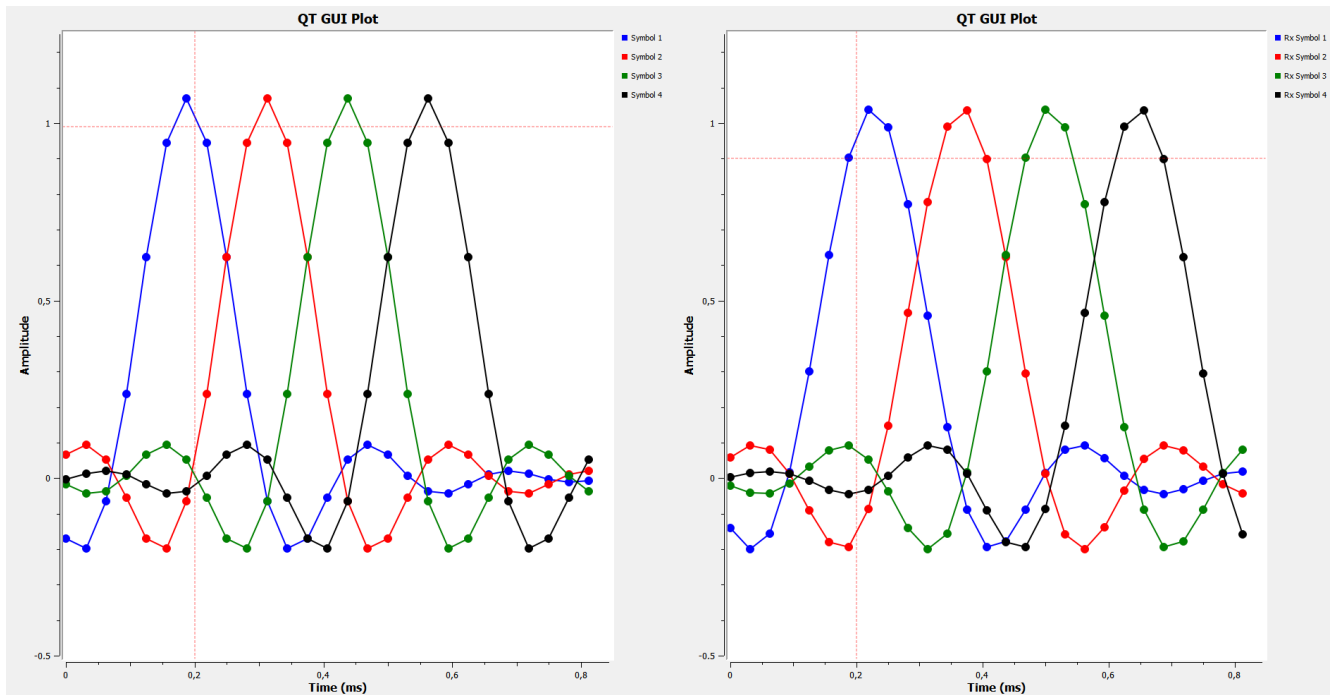


Рис. 9: Передача четырех символов с рассинхронизацией

Для восстановления времени на приемнике будет использоваться метод восстановления тактового сигнала многофазного фильтра. Этот блок выполняет восстановление времени и избавляет от проблемы ISI. Он работает путем вычисления первого дифференциала входящего сигнала, который будет связан с его тактовым смещением. Выборка, которую хочется видеть, находится в 0.25 мс. Разностный фильтр $[-1, 0, 1]$ генерирует дифференциал символа, и, как показано на следующем рисунке, выход этого фильтра в правильной точке выборки равен 0. Оптимальная точка выборки считается найденной тогда, когда выход дифференциального фильтра равен 0.

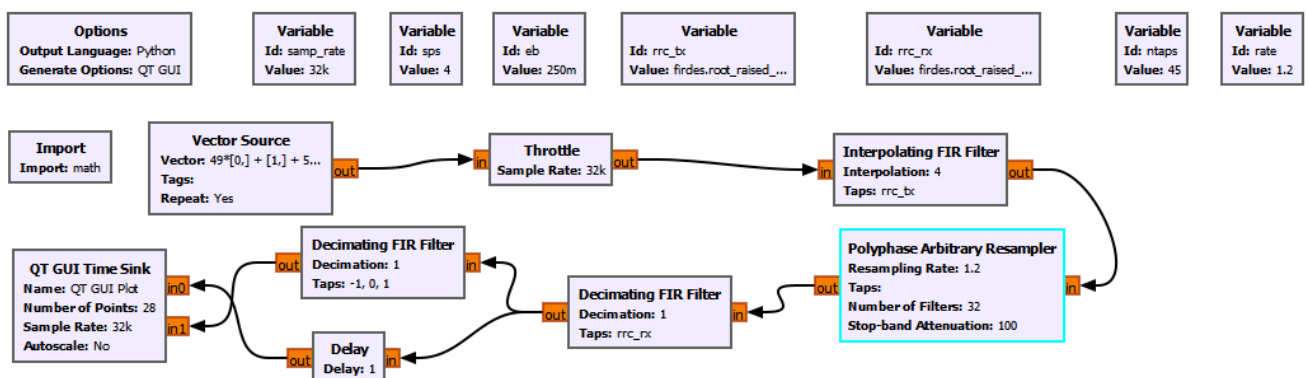


Рис. 10: Схема

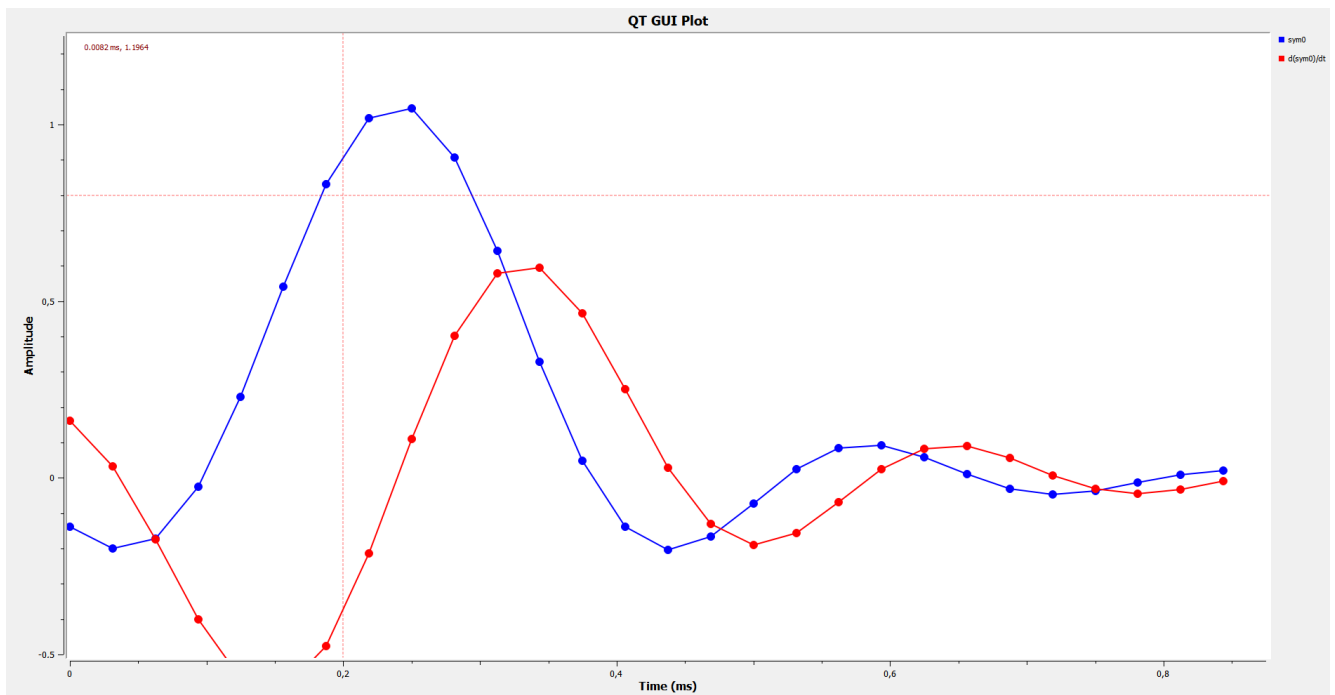


Рис. 11: Синхронизация

Было рассмотрено моделирование, которое использует 5 фильтров (т.е. 5 различных фаз). В данном случае используется `fractional resampler`, так как он позволяет легко произвести сдвиг фазы (между 0 и 1), и он также изменяет задержку фильтра сигналов, что необходимо скорректировать с помощью блоков задержки.

Можно видеть, что сигнал $d(\text{sym0})/dt + \phi_3$ имеет точку отсчета ровно в 0. Это говорит о том, что идеальная точка отсчета происходит при этом смещении фазы. Поэтому, если взять фильтр RRC этого приемника и отрегулировать его фазу по ϕ_3 , то мы можем исправить несоответствие времени и выбрать идеальную точку выборки.

Была компенсирована частота дискретизации некоторым известным коэффициентом 1.2 и оказалось, что можно использовать один из пяти фильтров в качестве идеальной точки выборки. Любое смещение выборки между этими фазами все еще будет производить несвоевременную выборку с добавлением ISI как исследовали ранее. Поэтому вместо этого используется более 5 фильтров в данном алгоритме восстановления.

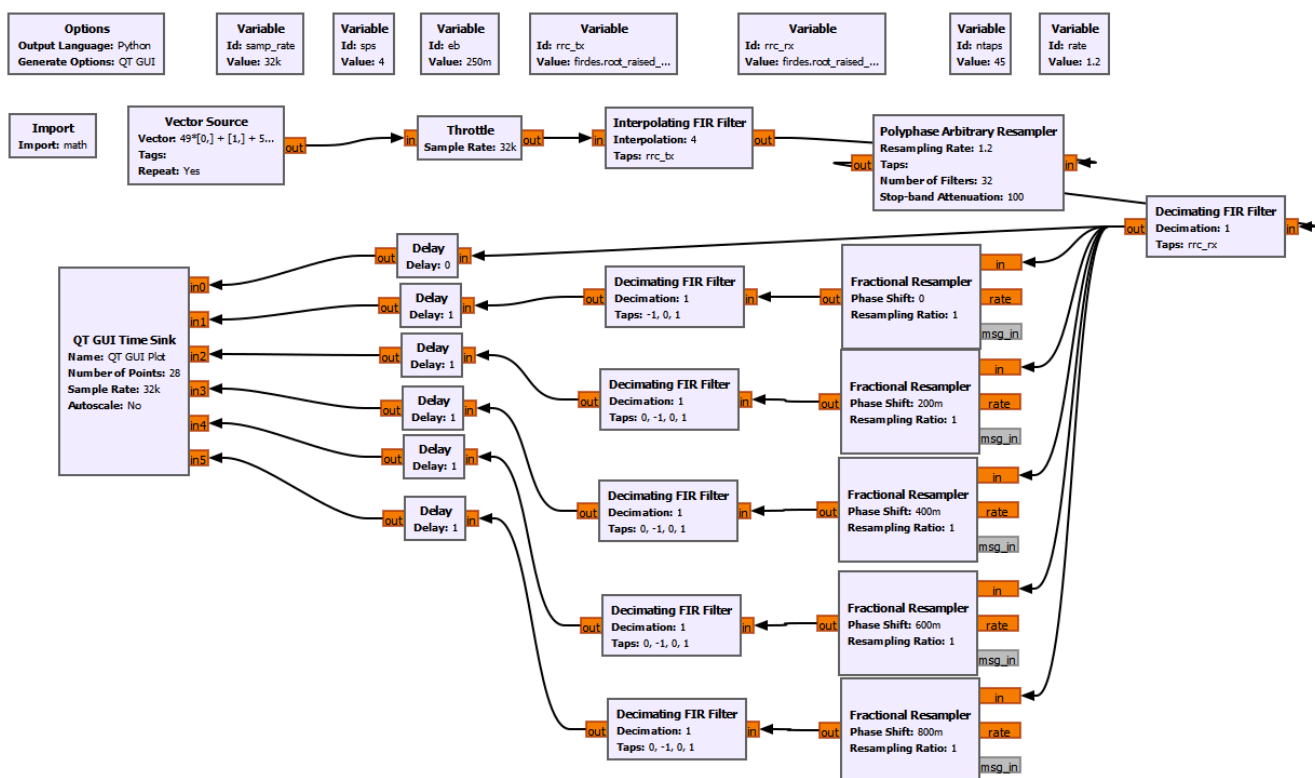


Рис. 12: Схема со синхронизацией

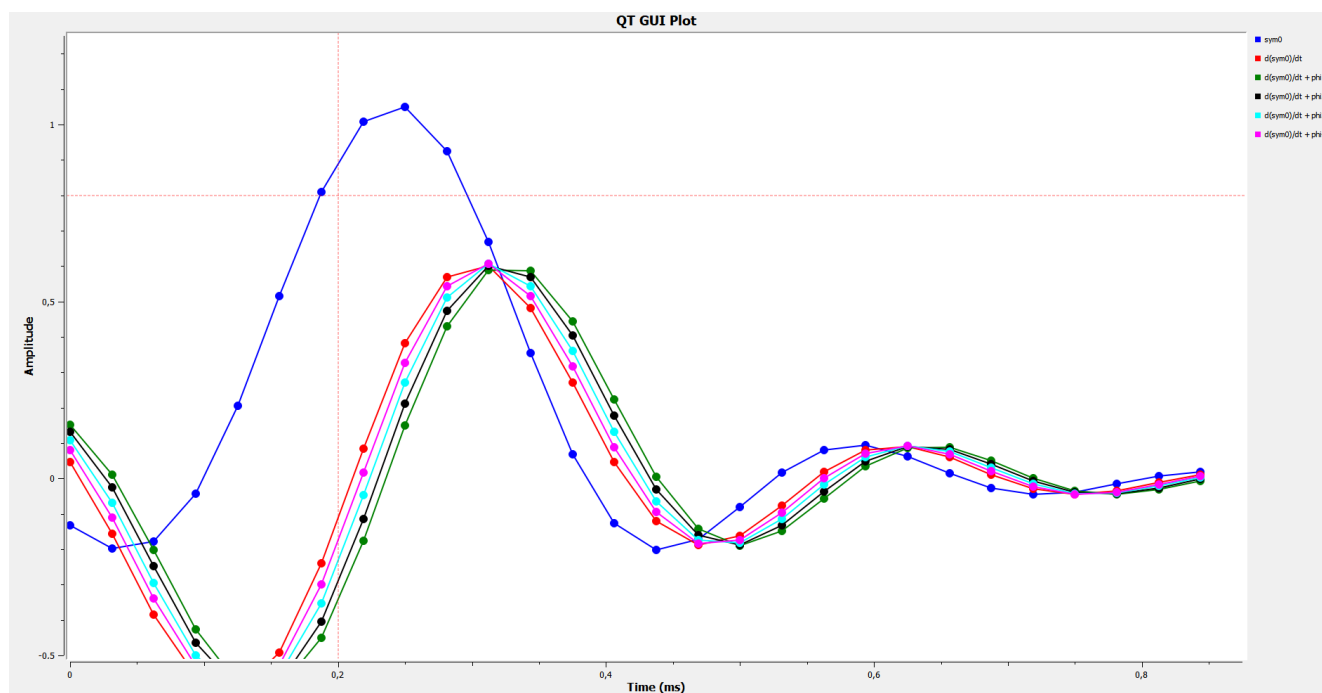


Рис. 13: Расширенный экран синхронизации

5 Часть №4: Восстановление синхронизации

В четвертом пункте будет использоваться результат упомянутых выше рассуждений.

Данный граф `mpskstage3.gcr` принимает выходной сигнал модели канала и пропускает его через наш многофазный блок синхронизации часов. Этот блок настроен с 32 фильтрами и пропускной способностью петли $2\pi/100$, принимает значение для ожидаемых выборок на символ. Внутренне, блок будет адаптироваться вокруг этого значения на основе скорости входящего сигнала. Можно наблюдать сближение созвездия.

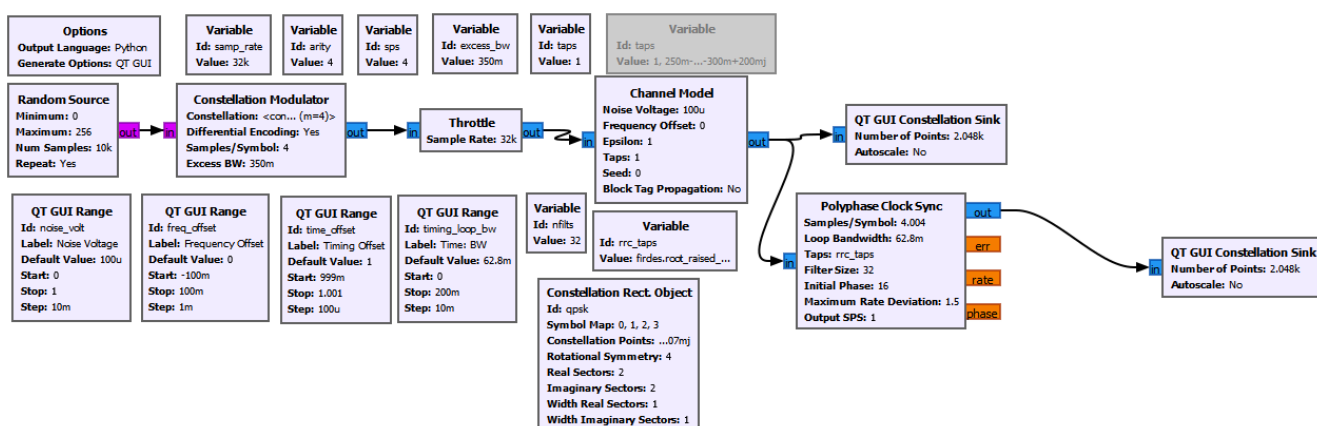


Рис. 14: Схема 3 шага

При запуске этого сценария отображается созвездие: слева как полученный сигнал до восстановления синхронизации и справа после восстановления синхронизации. Изображение все еще немного зашумлено в результате ISI после 32 фильтров.

Когда происходит смещение частоты, можно видеть, что созвездие становится кругом. Однако блок все еще не позволяет исправить смещение частоты. Аналогичным образом, можно изменить среду многолучевого моделирования, изменив переменную `taps`. Добавление `multipath` покажет, что блок восстановления времени устойчив к `multipath`, но не будет исправлять его, поэтому снова нужно прибегнуть к дополнительным методам решения проблемы.

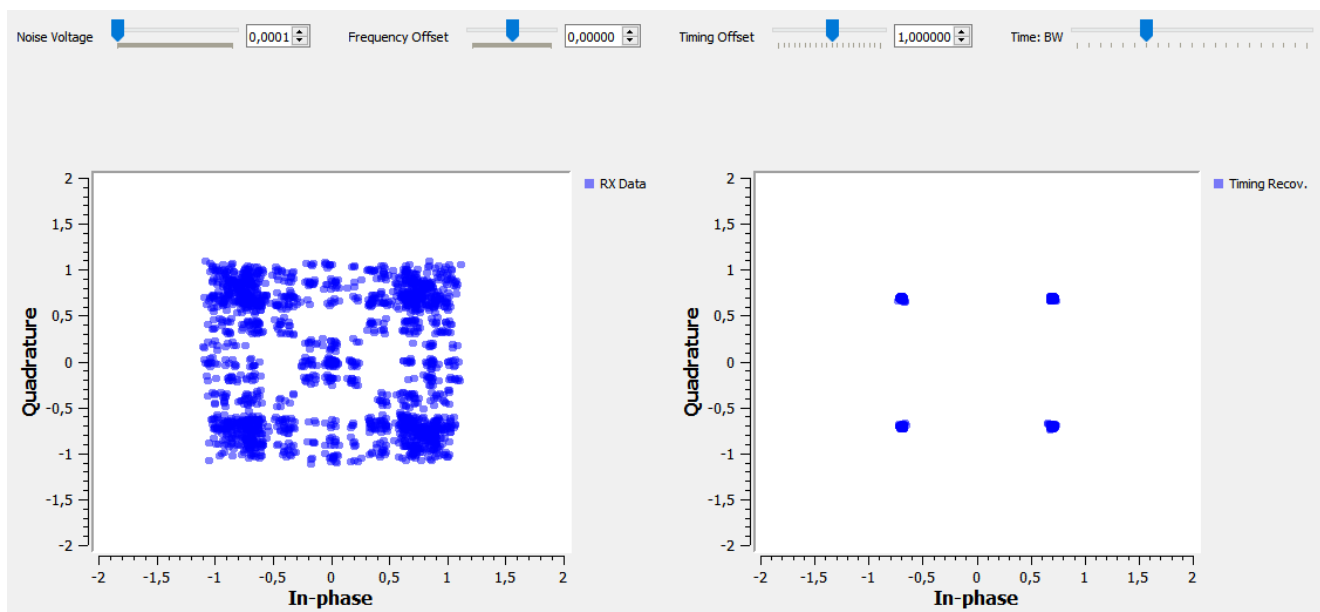


Рис. 15: Первая схема синхронизации

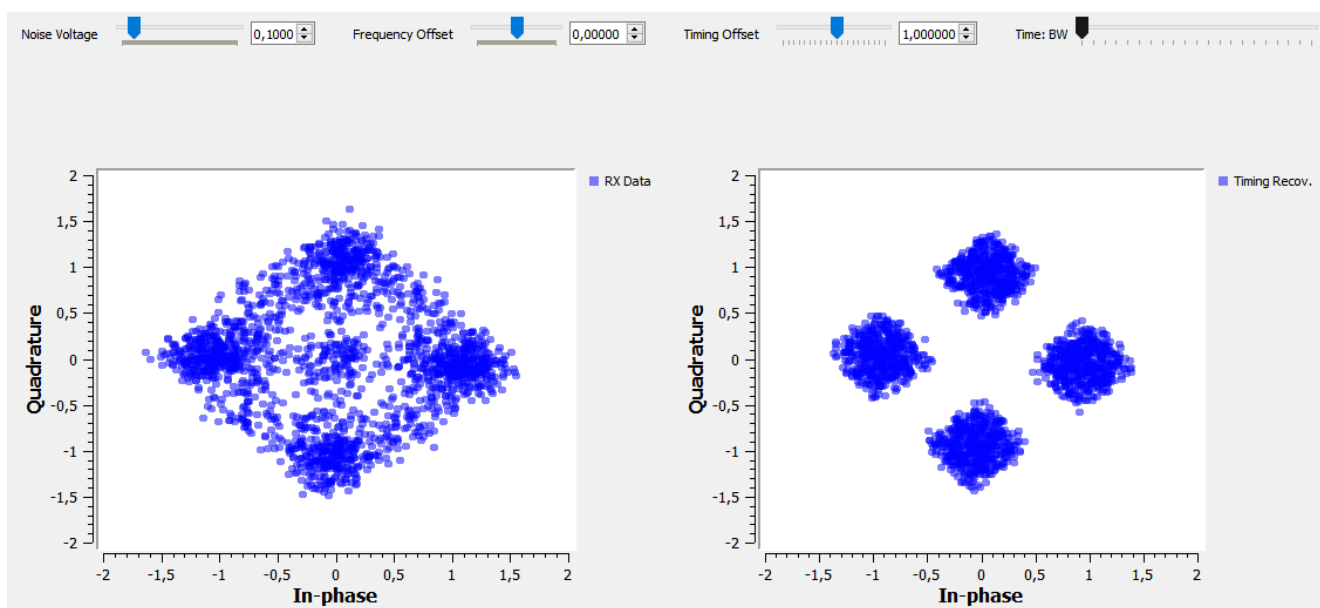


Рис. 16: Вторая схема синхронизации

6 Часть №5: Многолучевое распространение

В пятом пункте будем рассматривать многолучевое распространение.

Multipath является результатом того факта, что в большинстве коммуникационных сред нет единого пути для передачи сигнала от передатчика к приемнику. Каждый раз, когда есть объект, отражающий сигнал, между двумя узлами может быть установлен новый путь. Каждый из этих отражающих путей будет отображаться на приемнике в разное время в зависимости от длины пути. Суммирование их вместе у приемника вызывает искажения. Если разница во времени между отражениями достаточно мала относительно ширины символа, то искажение может быть внутри символа – внутрисимвольная интерференция. Когда отражения будут длиннее времени символа, отражение от одного символа повлияет на следующие сигналы - еще одна причина для интерференции между символами. Нужно исправить это поведение с помощью механизма, очень похожего на стереоэквалайзер (уравнитель). С помощью стереоэквалайзера можно изменить усиление определенных частот, чтобы либо подавить, либо усилить эти сигналы-бас и высокие частоты являются общими.

Multipathsim.grc просто устанавливает модель канала, чтобы обеспечить канал с пятью кнопками эквалайзера, четыре из которых можем изменить. На рисунке ниже многолучевой канал создает некоторое искажение в сигнале. Задача эквалайзера - инвертировать этот канал. Необходимым является использование подходящего алгоритма эквалайзера и правильная настройка параметров. Одним из важных из них в данном эксперименте является количество нажатий в эквалайзере. Как можно видеть при моделировании, пять отводов дают довольно грубый контроль над частотной характеристикой. Также, чем больше отводов, тем больше времени требуется как для вычисления отводов, так и для запуска эквалайзера против сигнала.

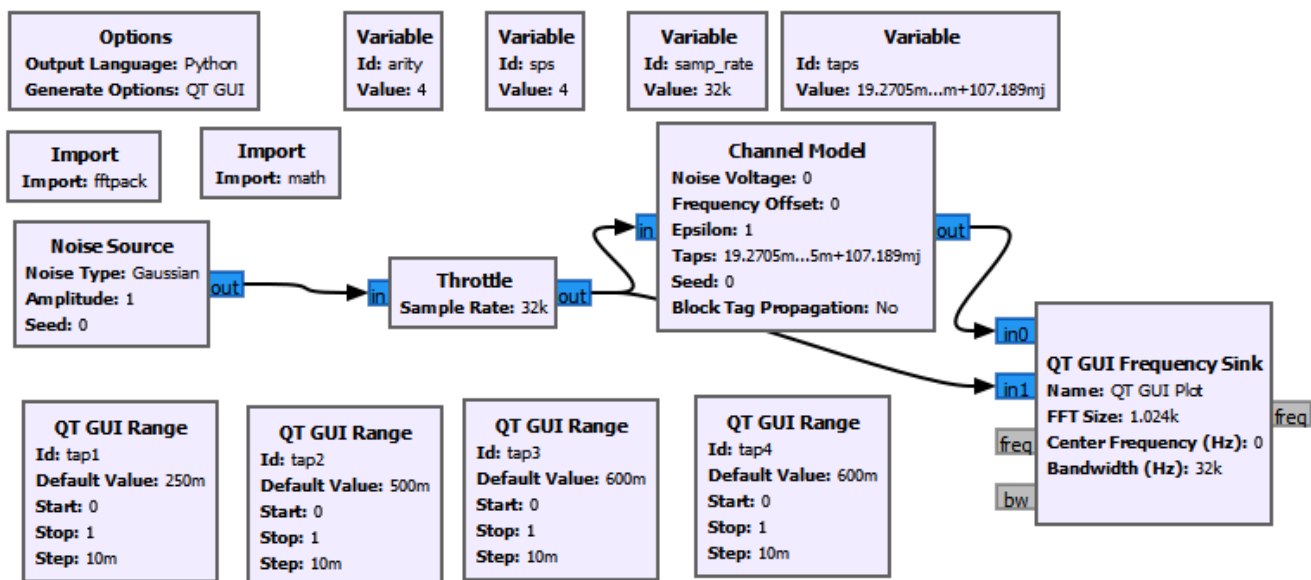


Рис. 17: Схема

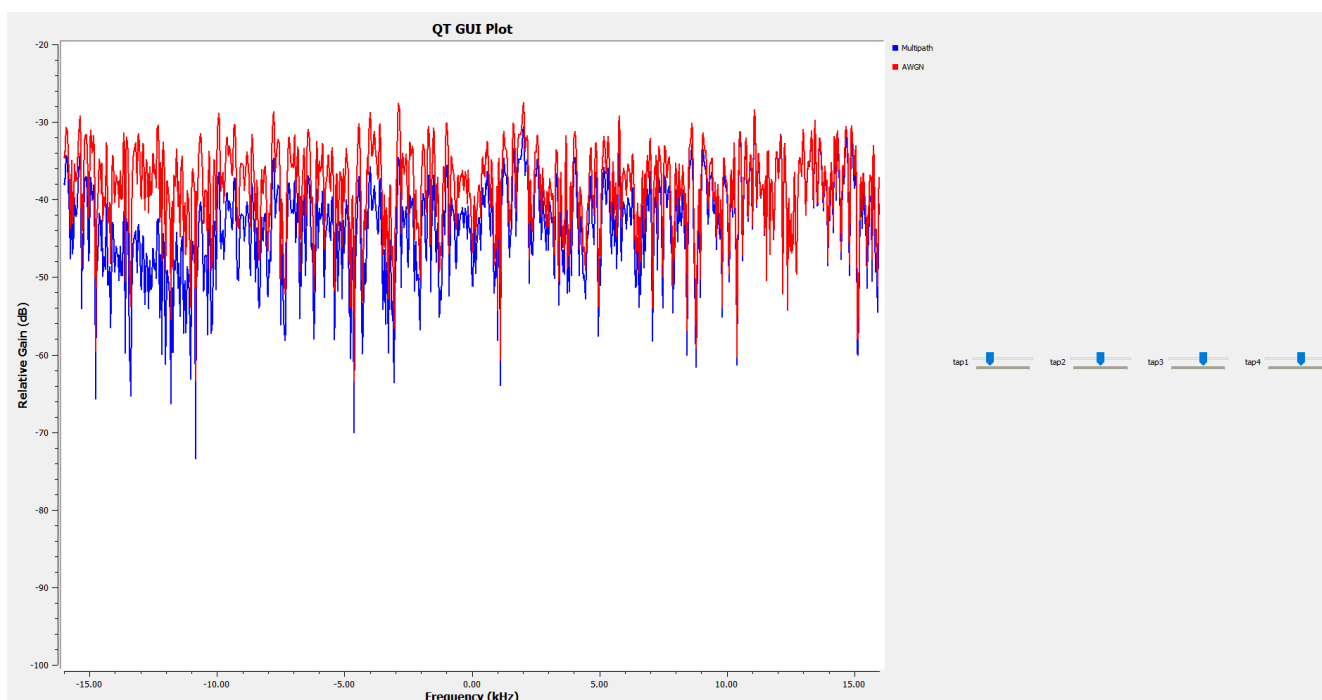


Рис. 18: График многолучевого распространения

7 Часть №6: Эквалайзеры

В шестом пункте мы будем рассматривать эквалайзеры, поставляемые GNU Radio: эквалайзер CMA и эквалайзер LMS DD.

СМА

СМА, или алгоритм постоянного модуля, является слепым эквалайзером. Он работает только на сигналах, которые имеют постоянную амплитуду или модуль. Алгоритм СМА принимает количество нажатий для использования в эквалайзере, которое будет основано на некоторой комбинации. Хочется сохранить это число небольшим, чтобы уменьшить накладные расходы алгоритма, убедившись, что есть достаточно степеней свободы, чтобы исправить для канала.

В `mpskstage4.grc`, используется алгоритм СМА с 11 отводами. Было продемонстрировано, как он влияет на производительность с вычислительной и с сигнальной точки зрения.

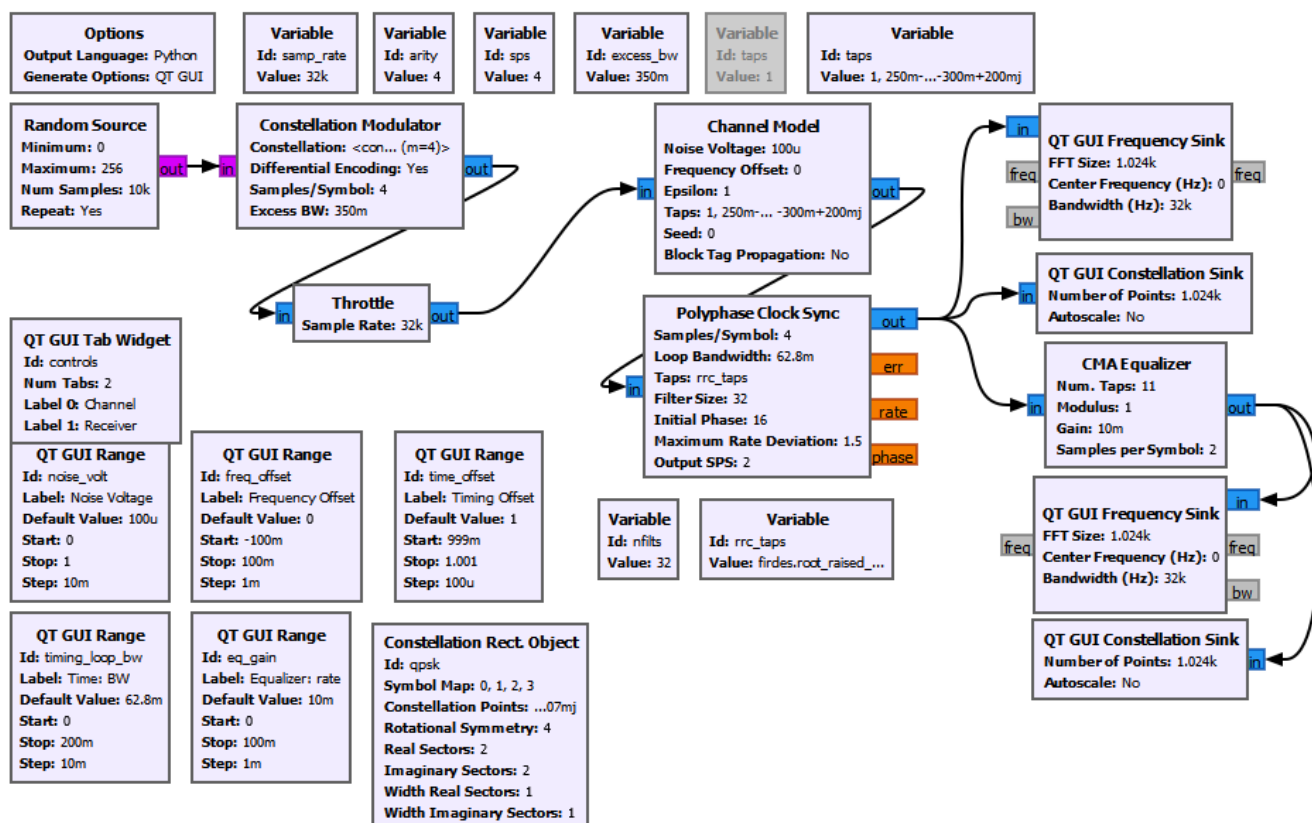


Рис. 19: Эквализация СМА

Можно видеть, как сходится алгоритм СМА. Имеется блок синхронизации по времени, так и блок эквалайзера, они сходятся независимо, но одна стадия влияет на следующую стадию. Таким образом, здесь происходит некоторое вза-

имодействие, когда оба фиксируются на сигнале. В конце концов, однако, можно видеть эффект запертого во времени многолучевого сигнала до и после эквалайзера. Перед эквалайзером имеется зашумленный, неровный сигнал. Применение эквалайзера дает возможность вновь увидеть чистый сигнал. Также можно видеть сам канал и то, как он хорошо выравнивается после эквалайзера.

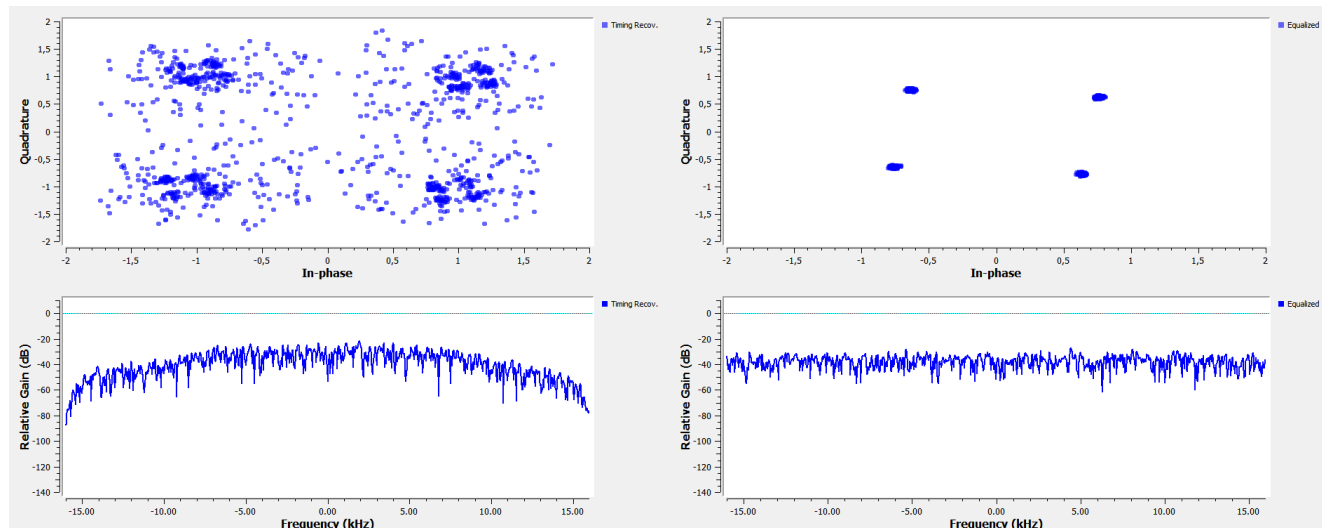


Рис. 20: Результат для СМА

LMS-DD

Задача теперь заключается в использовании блока эквалайзера с наименьшим средним квадратом, направленным на принятие решений (LMS-DD). В отличие слепого эквалайзера, такого как СМА, этот эквалайзер должен знать точки созвездия.

Эквалайзер с наименьшим средним квадратом отлично подходит для сигналов, которые не соответствуют требованиям к постоянному модулю алгоритма СМА, поэтому он может работать с модуляцией типа QAM. С другой стороны, если SNR достаточно плох, принимаемые решения неверны, что может негативно отразиться на производительности приемника. Блок также является более сложным в вычислительном отношении. Данный эквалайзер может производить сигналы лучшего качества, потому что он имеет прямое знание сигнала. Идея состоит в том, чтобы использовать слепой эквалайзер для первоначального сбора, чтобы получить достаточно хороший сигнал, который эквалайзер с наименьшим средним квадратом возьмет на себя.

Был взят `mpskstage4.grc` - файл, в котором эквалайзер СМА был заменен на эквалайзер LMS-DD. Этот блок использует объект созвездия, поэтому сложность здесь заключается в создании правильного объекта созвездия для сигнала QPSK.

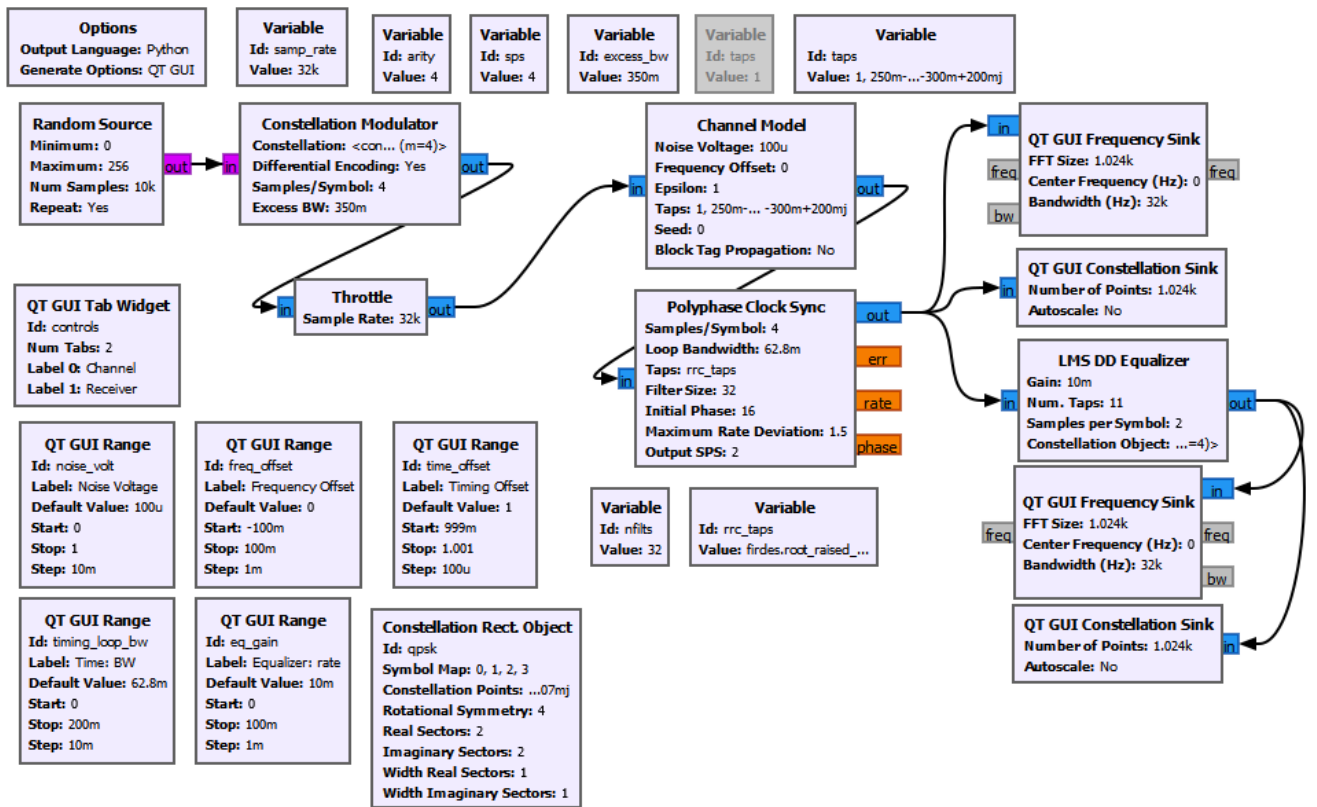


Рис. 21: Эквиализация LMS

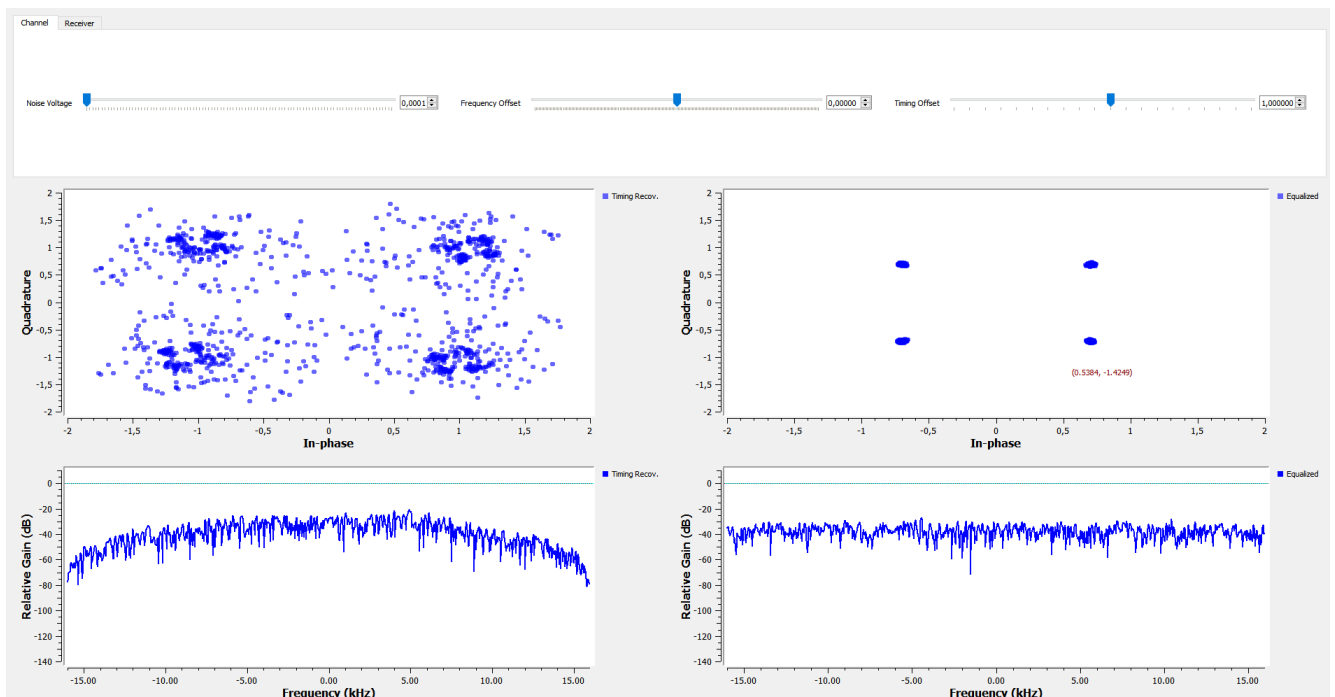


Рис. 22: Результат для LMS

В результате можно сказать, что оба эквалайзера справляются с задачей коррекции искажений, вызванных многолучевым распространением, но LMS эквалайзер к тому же оказывается устойчив к фазовому сдвигу.

8 Часть №7: Фазовая и частотная коррекция

В седьмом пункте посмотрим на фазную и частотную корреляцию. Учитывая, что мы выровняли канал, у нас все еще есть проблема смещения фазы и частоты. Эквалайзеры, как правило, не адаптируются быстро, и поэтому могут не успевать за смещением частоты. Кроме того, если просто запустить эквалайзер СМА, достаточным будет являться его схождение к единичному кругу. Он не имеет никакого представления о созвездии, поэтому, когда он блокирует, он будет блокировать в любой заданной фазе. Теперь нужно настроить ситуацию для любого смещения фазы, а также для любого смещения частоты.

Для решения этой проблемы предлагается использовать петлю Костаса. Как и методы, используемые ранее, этот метод использует обратную связь для вычисления ошибки и коррекции фильтра (`mpskstage5.grc`). Блок петли Costas может синхронизировать BPSK, QPSK и 8PSK. Приемник созвездий будет фиксироваться на любом заданном объекте созвездия.

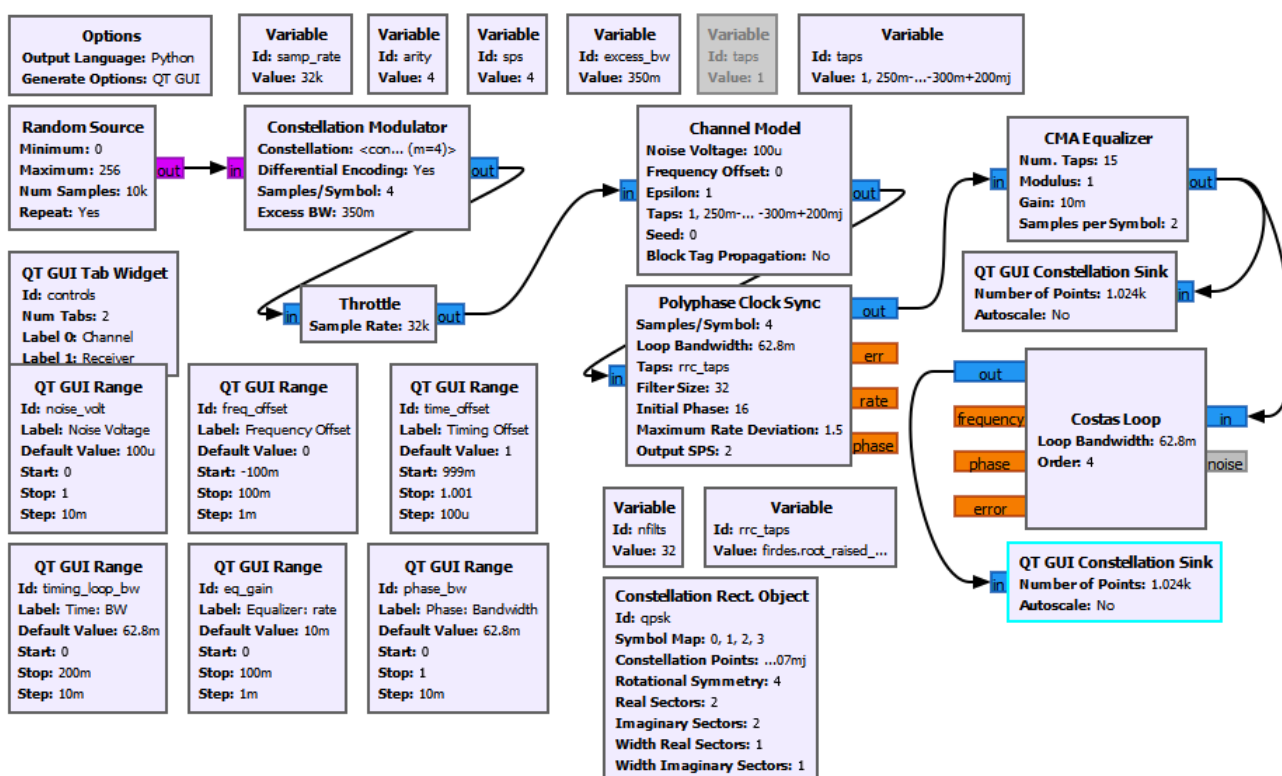


Рис. 23: Фазовая и высокочастотная коррекция

Этот блок, как и все другие, использует цикл второго порядка и поэтому определяется параметром пропускной способности цикла. Другая вещь, которую он должен знать, - это порядок модуляции PSK, поэтому 2 для BPSK, 4 для QPSK и 8 для 8PSK. На следующем изображении мы установили шум, смещение времени, простой многолучевой канал и смещение частоты. После эквалайзера мы видим,

что все символы находятся на единичном круге, но вращаются из-за смещения частоты, которое еще ничего не исправляет. На выходе блока Costas loop мы можем видеть заблокированное созвездие, как мы начали с него, плюс дополнительный шум, с которым мы ничего не можем поделать.

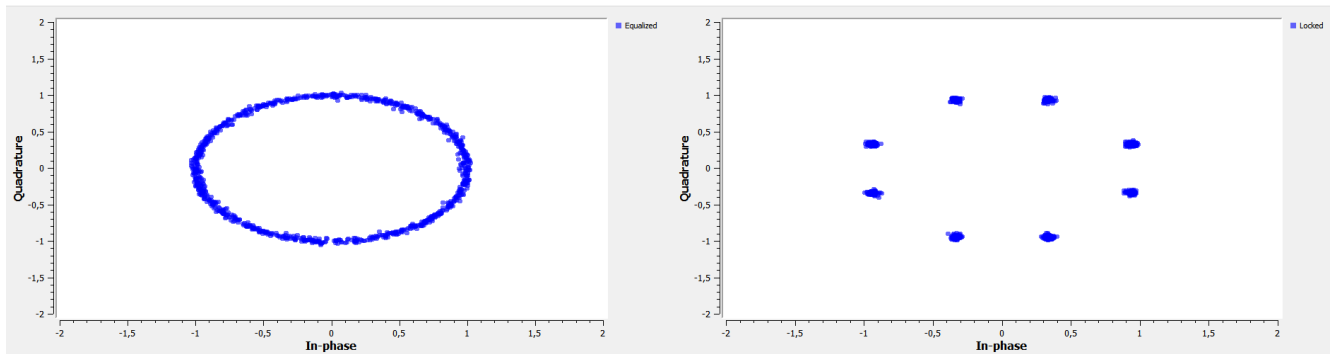


Рис. 24: Первый результат фазовой и высокочастотной коррекции

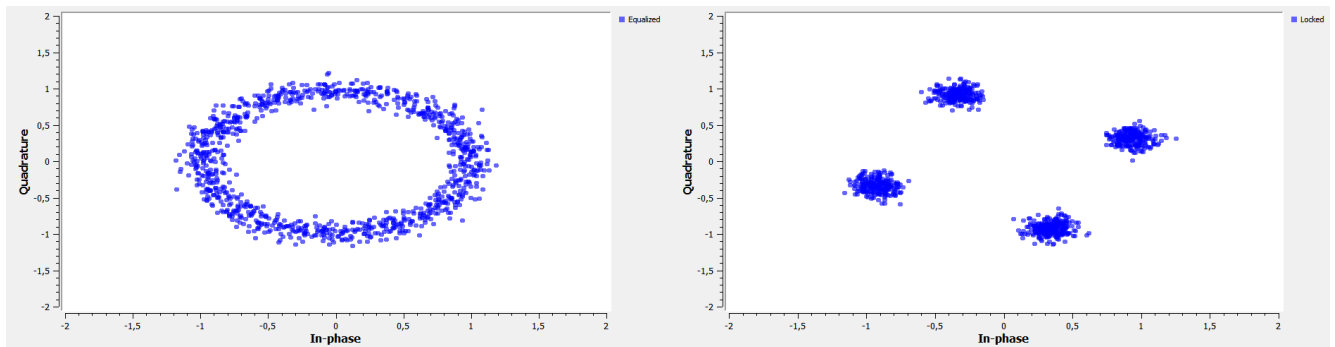


Рис. 25: Результат фазовой и высокочастотной коррекции

9 Часть №8: Декодирование

В заключительном, восьмом пункте рассмотрим декодирования сигнала. Используя `mpskstage6.grc` квадратурный демодулятор был установлен после `Costas loop` вместо `mpqpskdemodulator`. В этот момент получены символы от 0 до 3, т. к. это размер алфавита в схеме QPSK. Само созвездие не было фактически передано, вместо этого передана разница между его символами, установив дифференциальную настройку в блоке модулятора созвездия как `True`.

Чтобы определить, что это исходный байтовый поток, было проведено его сравнение с входным потоком, так как у нас имеется доступ к передаваемым данным при моделировании.

Использовался блок `unpack bit` для распаковки от 8-бит на байт до 1-бит на байт. После эти потоки были преобразованы в значения с плавающей точкой 0.0 и 1.0, т. к. временные приемники принимают только плавающие и комплексные значения. Также нам необходимо задержать передаваемые биты, используя блок `[Delay|delay]`.

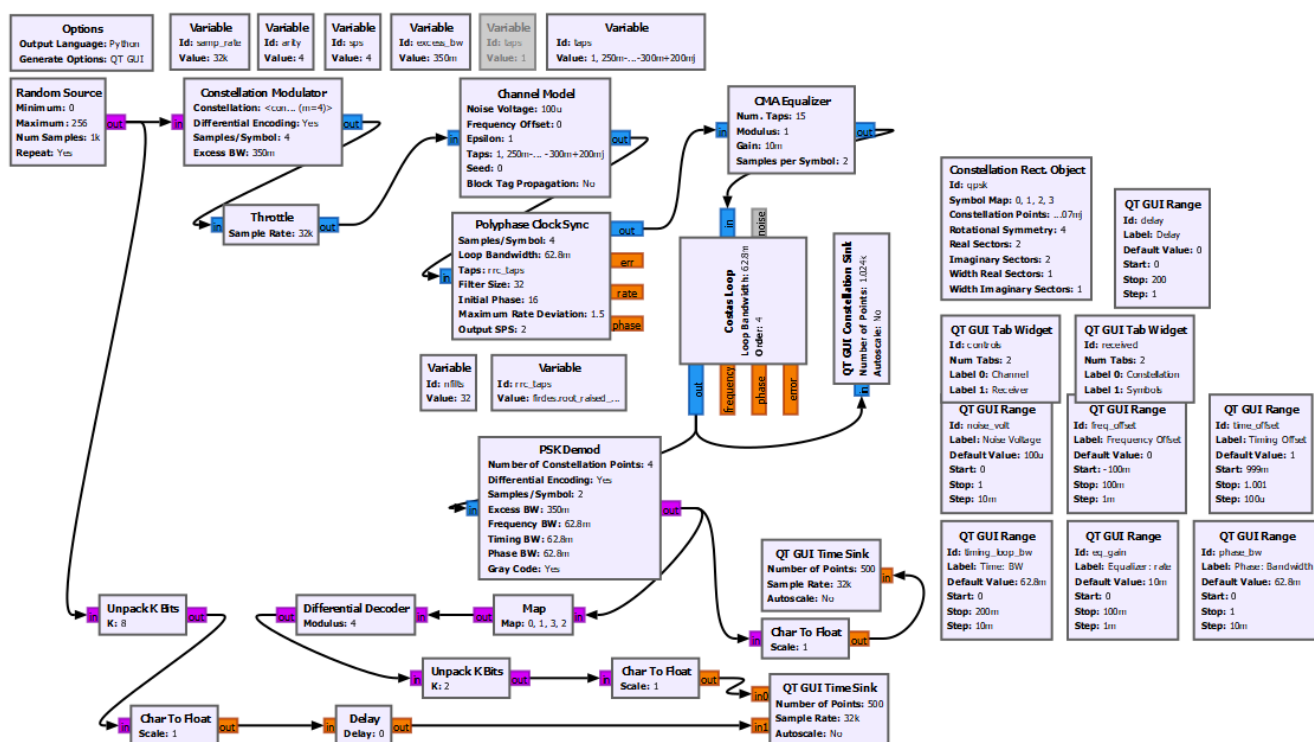


Рис. 26: Передача созвездия QPSK

Чтобы увидеть, когда сигналы синхронизируются, мы можем вычесть один из другого (поскольку выход будет равен 0). Добавление шума и других влияний

канала может быть легко замечено как битовые ошибки (когда этот сигнал не равен 0).

В качестве заключительного эксперимента используется генератор случайных чисел конечной длины, поэтому возможно увидеть шаблон в полученном сигнале. Используя приемник раstra времени графического интерфейса QT, он был настроен так, чтобы этот шаблон можно было увидеть.

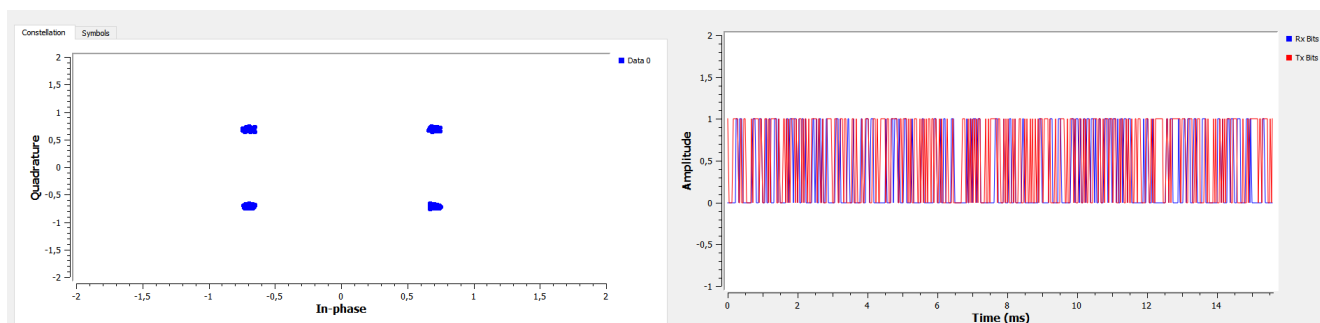


Рис. 27: Вид результата

10 Выводы

В ходе выполнения данной лабораторной работы нами были рассмотрены все стадии процесса приема-передачи процесса. Чтобы было легче его понять, весь процесс был разбит на промежуточные шаги, что позволило лучше разобраться в его работе, учесть все факторы, влияющие на передачу, прием и транслирование сигнала.