

Санкт-Петербургский государственный политехнический
университет Петра Великого.

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

ЛАБОРАТОРНАЯ РАБОТА

GNU Radio

Работу выполнила студентка:

_____ А. И. Луцкевич
« ____ » _____ 2021 г.

Преподаватель лабораторных
работ:

_____ Н. В. Богач
« ____ » _____ 2021 г.

Санкт-Петербург, 2021 г.

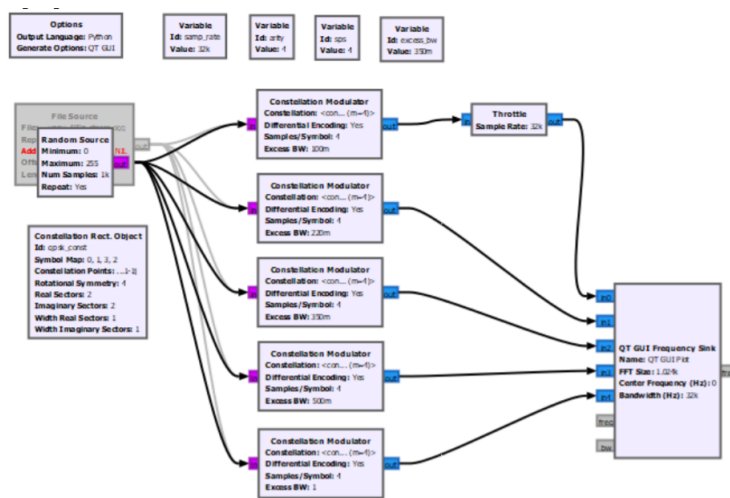
1. Передача сигнала

Сначала необходимо создать квадратурную фазовую манипуляцию и передачу модифицированного сигнала.

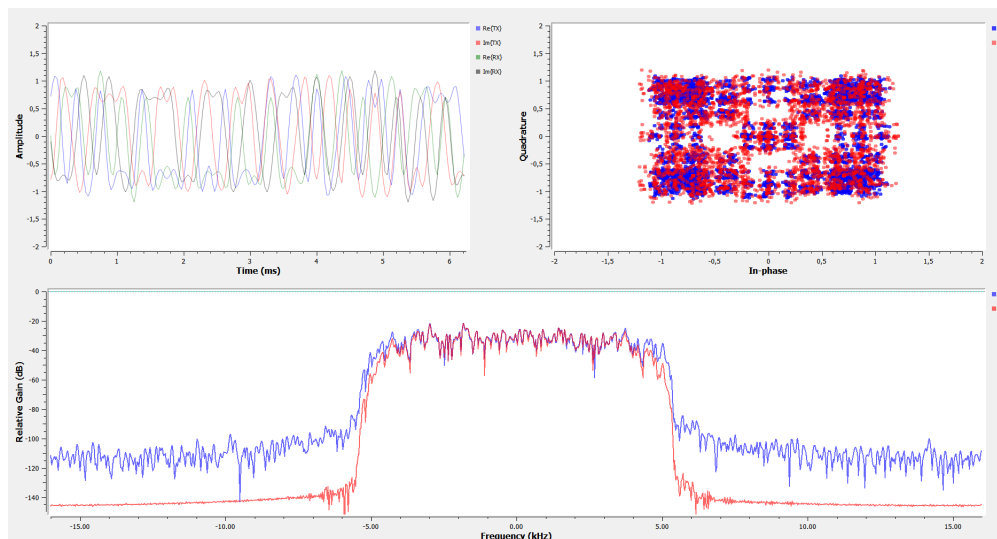
Будет сгенерирован сигнал (случайный поток битов), а далее он будет смоделирован в сложное созвездие. Для этого используем блок Constellation Modulator, который принимает объект Constellation и применяет различные настройки для управления передаваемым сигналом. Объект созвездия позволяет определить, 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 схема передает созвездие QPSK. Он строит как переданный сигнал, так и часть цепи приемника во времени, частоте и графике созвездия. И стоит отметить, что имеются 3 ограничения:

1. сигнал обязан укладываться в выделенную ему полосу пропускания;
2. передаваемые символы должны быть различимы;
3. передача сигнала должна обеспечивать хорошее соотношение сигнал/шум.



График, который находится ниже, демонстрирует характеристики сигнала в “канале передачи” и на приемнике. На графике виден эффект повышения частоты дискретизации и фильтрации. В этом случае фильтр RRC добавляет собственные помехи, известные как межсимвольные помехи (ISI).



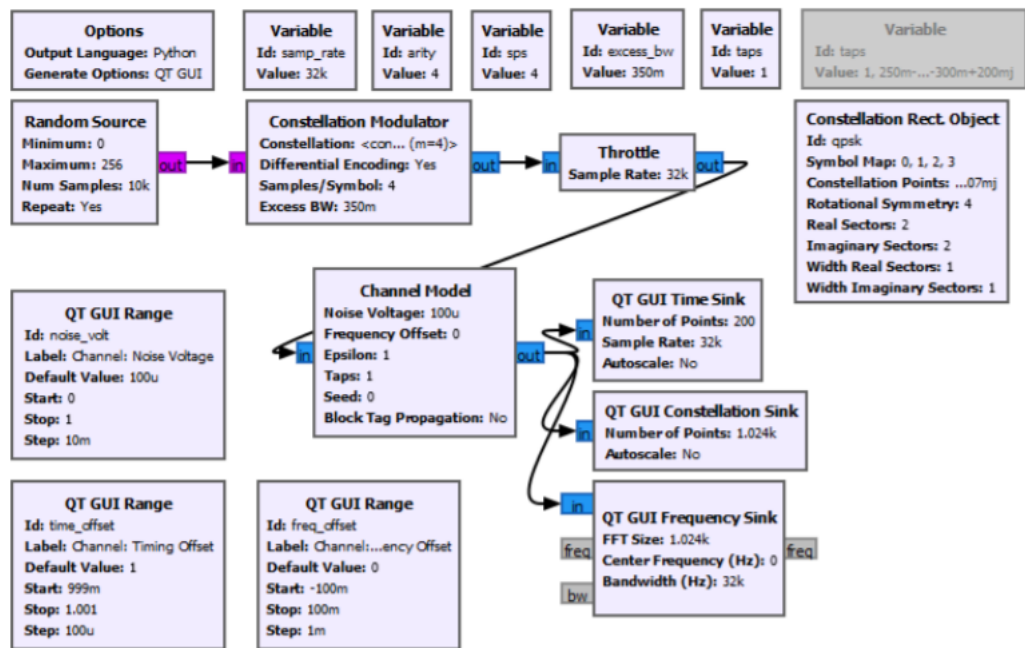
2. Добавление канала передачи

Теперь мы рассмотрим влияние канала и то, как сигнал искажается между тем, когда он был передан, и тем, когда мы видим сигнал в приемнике.

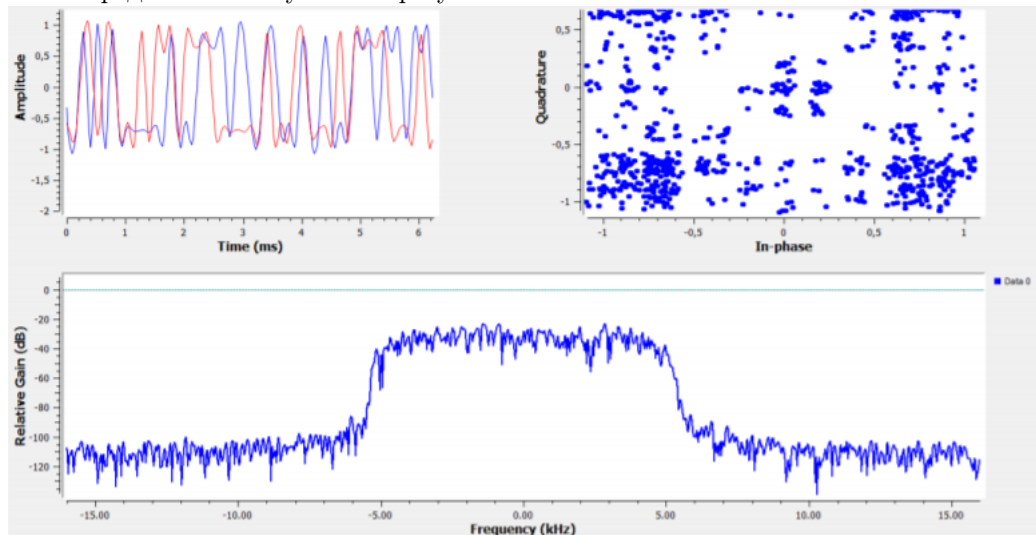
Для начала мы будем использовать самый простой блок модели канала GNU Radio - базовый блок Channel Model. Этот блок позволяет нам смоделировать несколько основных проблем, с которыми нам приходится иметь дело.

Первая проблема с приемниками - это шум. Тепловой шум в нашем приемнике вызывает шум, который мы знаем как аддитивный белый гауссовский шум (AWGN). Устанавливаем мощность шума, регулируя значение напряжения шума модели канала, а также указываем здесь напряжение вместо мощности, потому что нам нужно знать полосу пропускания сигнала, чтобы правильно рассчитать мощность.

Другой 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.



Ниже представлен полученный результат.



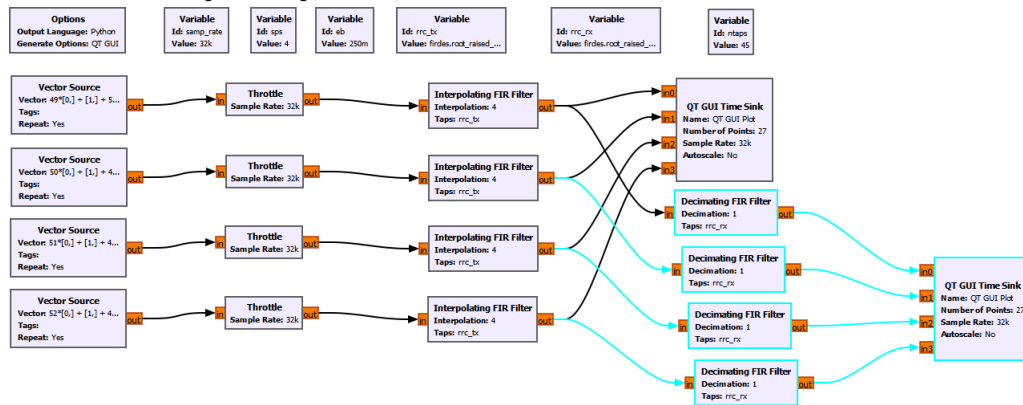
3. Восстановление времени

Существует множество алгоритмов, которые мы могли бы использовать для восстановления на каждом этапе. Нами будет использоваться алгоритм

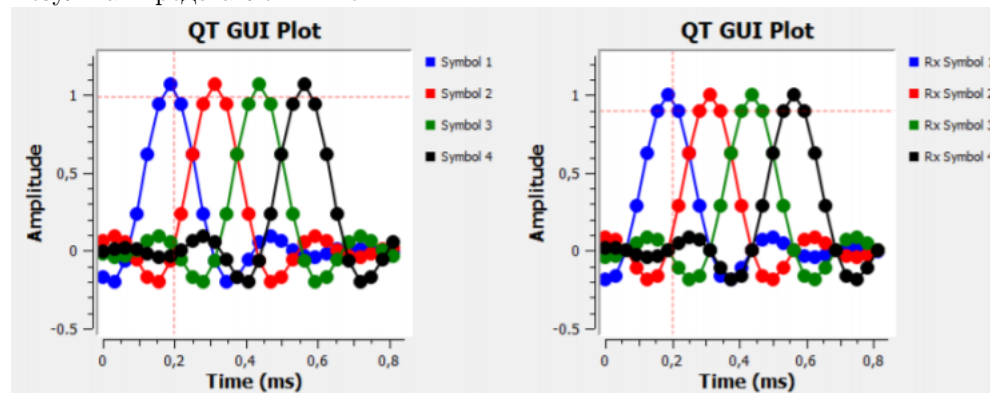
polyphase clock recovery.

Мы пытаемся найти наилучшее время для дискретизации входящих сигналов, что позволит максимизировать отношение сигнал / шум (SNR) каждой выборки, а также уменьшить влияние межсимвольных помех (ISI). Мы можем проиллюстрировать проблему ISI на примере flowgraph symbolsampling, где мы просто создаем четыре отдельных символа из единиц в строке, а затем фильтруем их. 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. Второй фильтр 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.

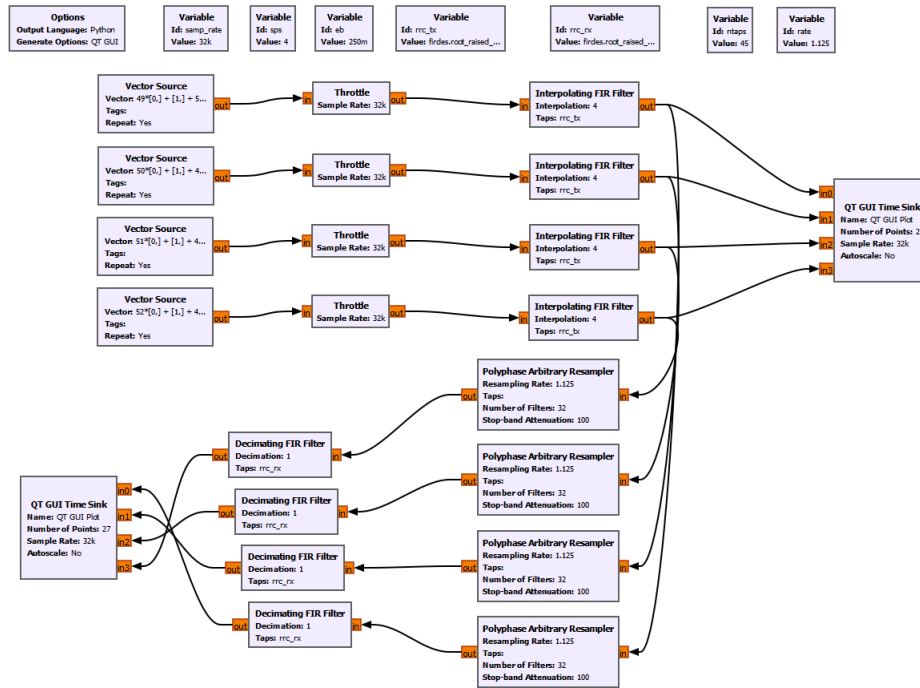
Ниже схема без рассинхронизации.



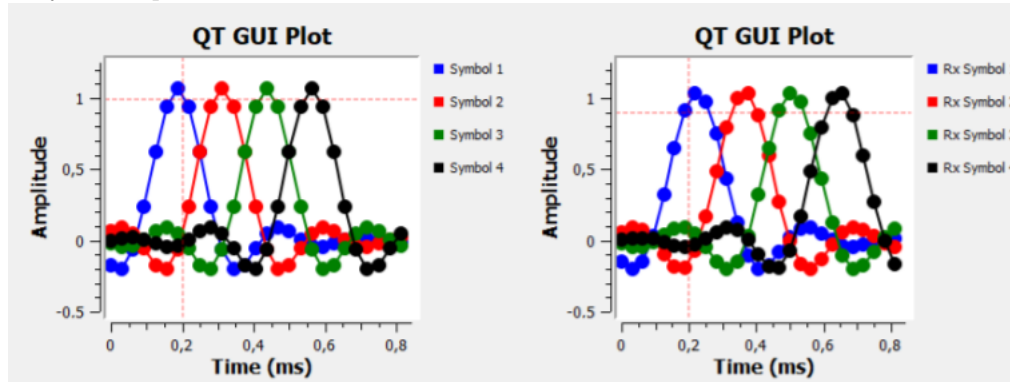
Результат представлен ниже.



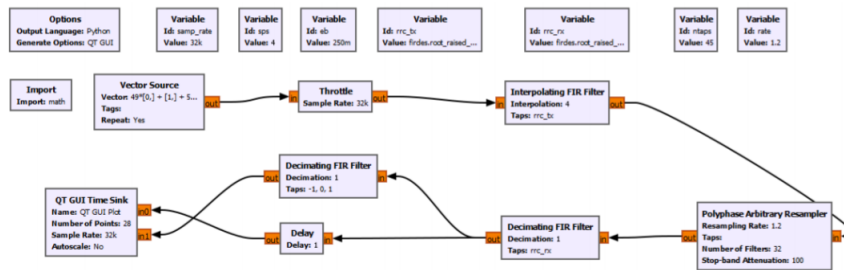
Далее схема с рассинхронизацией.



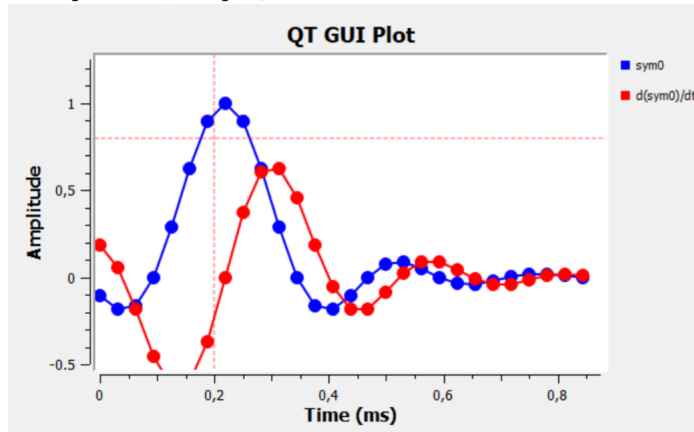
Результат представлен ниже.



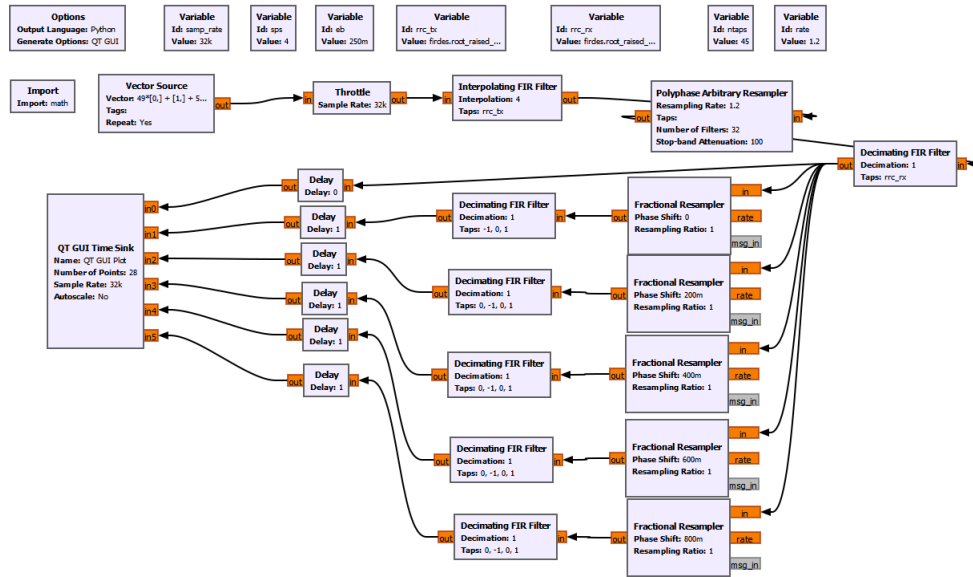
Для восстановления времени на приемнике будет использоваться метод восстановления тактового сигнала многофазного фильтра. Этот блок выполняет восстановление времени и избавляет от проблемы ISI. Он работает путем вычисления первого дифференциала входящего сигнала, который будет связан с его тактовым смещением. 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.



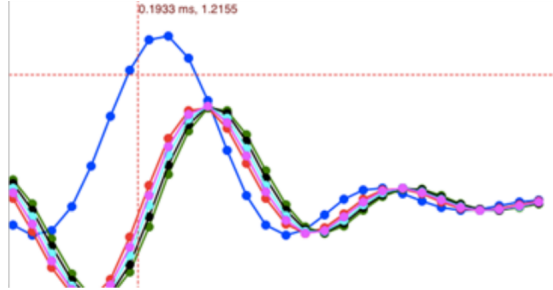
Синхронизация представлена ниже:



Вместо использования одного фильтра можно создать серию фильтров, каждый из которых будет с разной фазой. Если достаточно фильтров на разных фазах, то один из них - правильная фаза фильтра, которая даст нам желаемое значение синхронизации. Посмотрим на симуляцию, которая строит 5 фильтров.



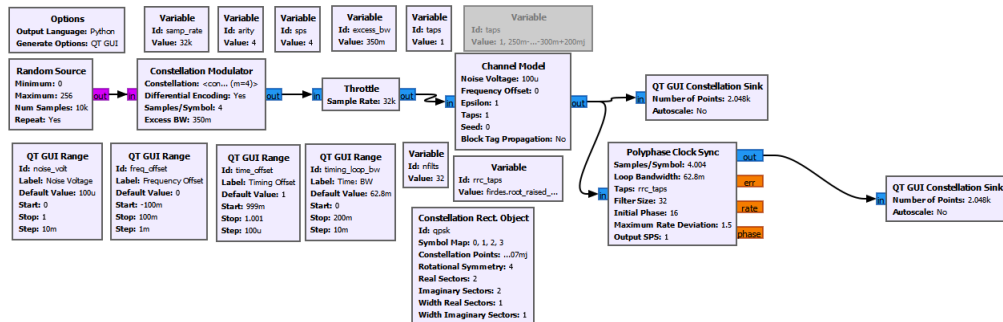
Расширенный экран синхронизации:



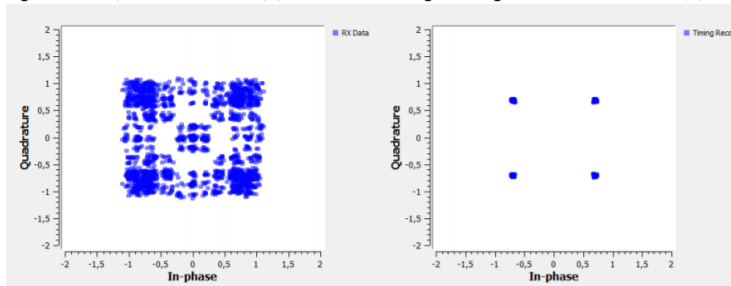
Видно, что сигнал, помеченный как $d(\text{sym0})/dt$ плюс ϕ_3 , имеет точку отсчета в 0. Это говорит о том, что идеальная точка дискретизации происходит при этом сдвиге фазы. Следовательно, если вызовем RRC-фильтр приемника и отрегулируем его фазу на ϕ_3 (что составляет $3 * 2\pi / 5$), то мы сможем исправить рассогласование по времени и выбрать идеальную точку дискретизации в это время дискретизации.

4. Восстановление синхронизации

Граф, который представлен ниже, 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. Внутри блок будет адаптироваться вокруг этого значения на основе скорости входящего сигнала. Можно наблюдать сближение созвездия.



При запуске этого скрипта мы видим созвездие слева как полученный сигнал до восстановления синхронизации и справа после восстановления синхронизации. На выходе - более отфильтрованное созвездие.



5. Многолучевое распространение

Рассмотрим 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. 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 стереоэквалайзера (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 подходя-

щего 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.

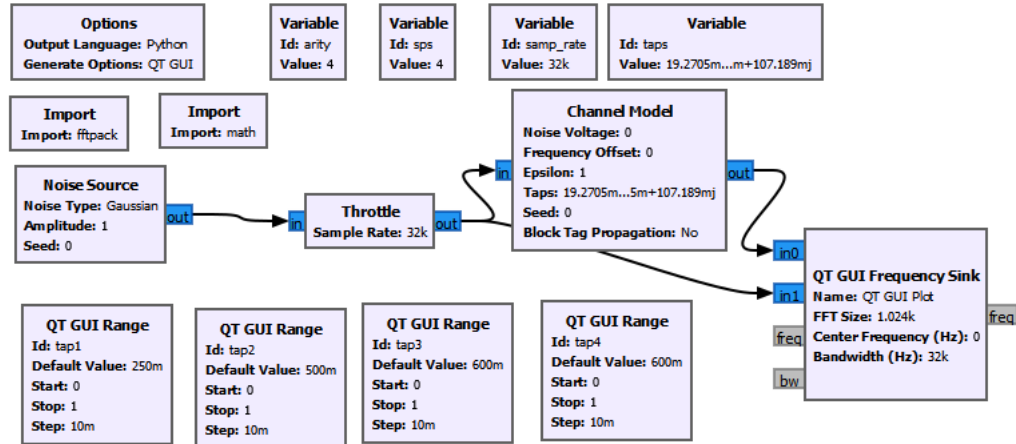
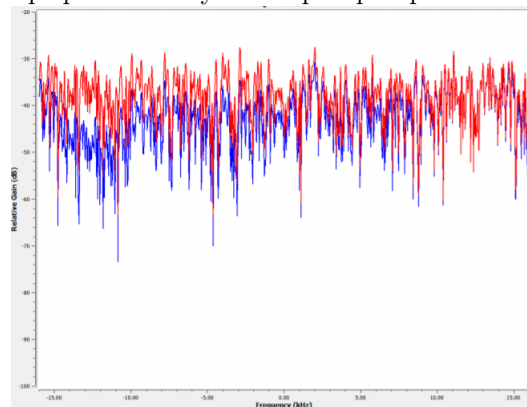


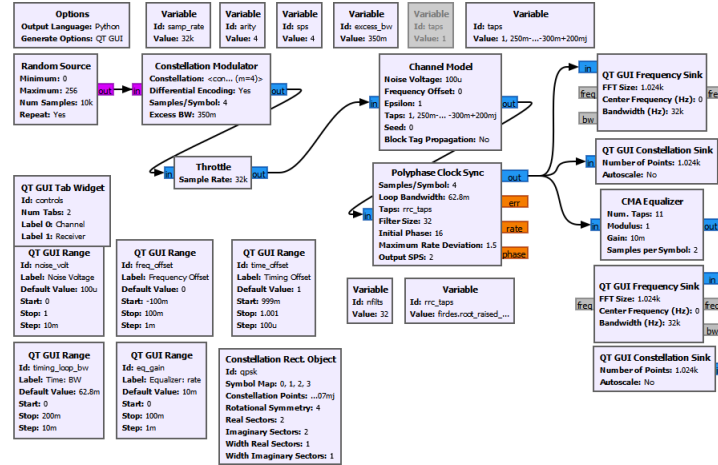
График многолучевого распространения представлен ниже:



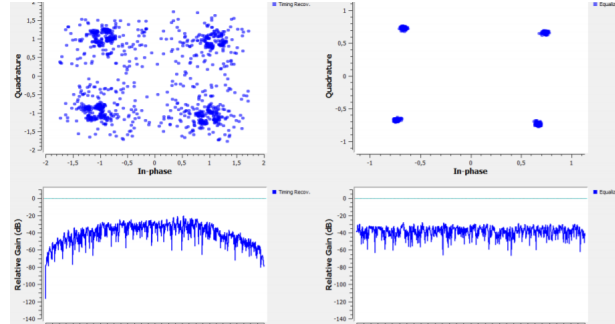
5. Эквалайзеры

GNU Radio поставляется с двумя легко используемыми эквалайзерами. Эквалайзер СМА и эквалайзер LMS DD. СМА или алгоритм постоянного модуля - это слепой эквалайзер, но он работает только с сигналами с постоянной амплитудой или модулем. Алгоритм СМА принимает количество нажатий для использования в эквалайзере, которое будет основано на некоторой комбинации обоснованного предположения, известных передовых практик и, возможно, некоторых фактических знаний о самом канале. В примере

мы используем алгоритм СМА с 11 нажатиями.

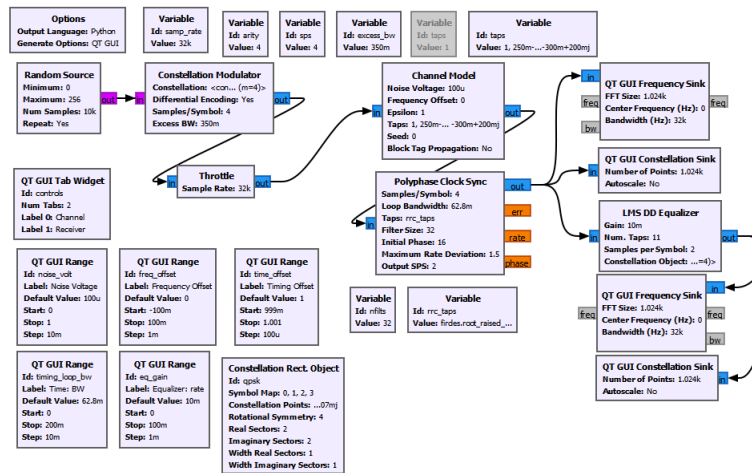


Результат моделирования:

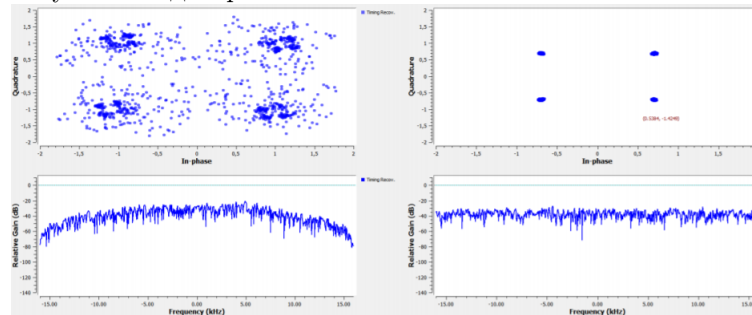


6. Эквалайзер LMS-DD

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. 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 файл `mpskstage4.grc`, 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.



Результат моделирования:



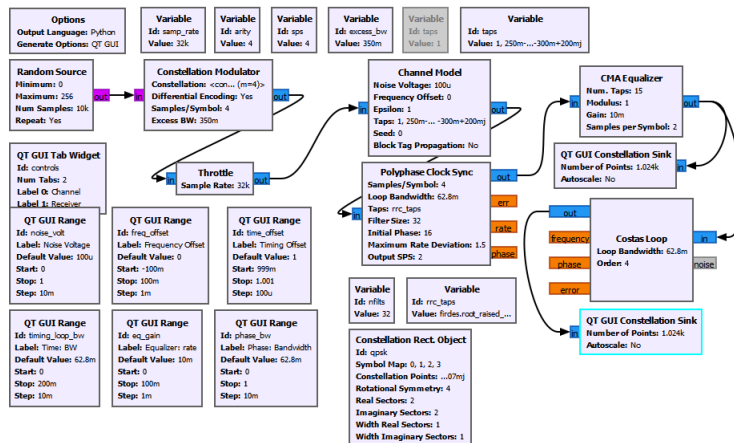
В результате можно сказать, что оба эквалайзера справляются с задачей коррекции искажений, вызванных многолучевым распространением.

7. Фазовая и частотная коррекция

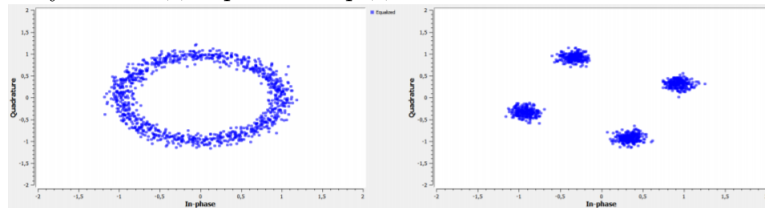
У нас все еще имеется проблема смещения фазы и частоты. Эквалайзеры, как правило, не адаптируются быстро, поэтому могут не успевать за смещением частоты.

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.

Есть способы сделать грубую коррекцию частоты. Для этой задачи мы собираемся использовать петлю Костаса. Этот метод использует обратную связь для вычисления ошибки и коррекции фильтра.



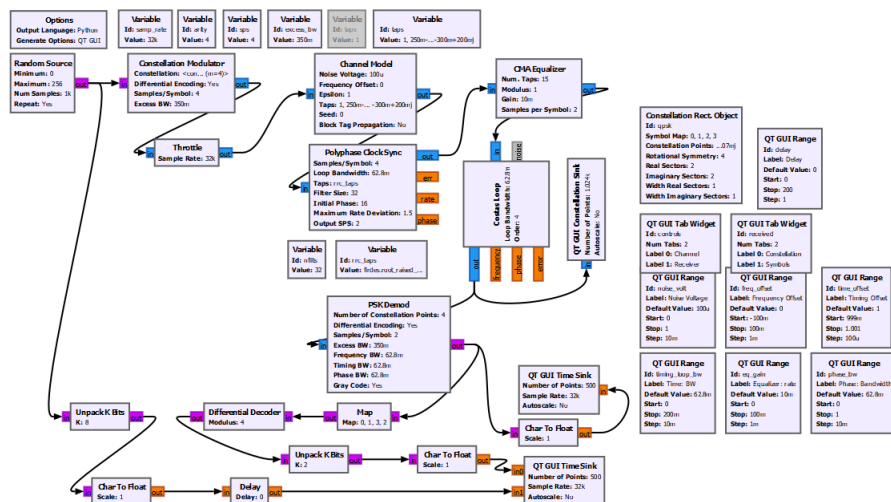
Результат моделирования представлен ниже:



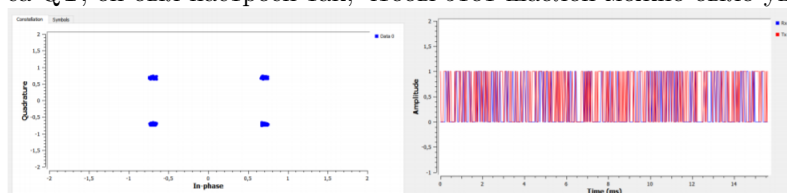
7. Декодирование

Теперь рассмотрим декодирования сигнала. В данном файле квадратурный демодулятор был установлен после Costas loop вместо myqpskdemodulator. 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 uprsk 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.



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.



8. Заключение

В данной лабораторной работе мною были рассмотрены все стадии процесса приема-передачи процесса, узнала что происходит при приеме и передаче с реальным оборудованием по шагам. Были использованы алгоритмы, доступные в GNU Radio для приема и демодуляции сигнала PSK.