

ВВЕДЕНИЕ В СИСТЕМЫ РАДИОСВЯЗИ

Лабораторные работы с NI USRP™ и LabVIEW Communications

Брюс А. Блэк

Отпечатано в Венгрии



326348A-01 Декабрь 2014

Введение в
системы радиосвязи

Лабораторные работы с NI USRP
и LabVIEW Communications

Руководство для студента

Брюс А. Блэк

Техническая поддержка по всему миру и информация о выпускаемой продукции

ni.com

Штаб-квартира корпорации National Instruments

11500 N Морас Expwy Austin, Texas 78759-3504 USA Tel: 512 683 0100

Офисы по всему миру

Посетите страницу ni.com/global для получения самой последней контактной информации.

Andean and Caribbean +58 212 503-5310, Argentina 0800 666 0037, Australia 1800 300 800, Austria 43 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada 800 433 3488, Chile 800 532 951, China 86 21 5050 9800, Czech Republic/Slovakia 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 89 741 31 30, Hungary 36 23 501 580, India 1 800 425 7070, Ireland 353 0 1867 4374, Israel 972 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Lebanon 961 0 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 495 783 68 51, Singapore 1800 226 5886, Slovenia/Croatia, Bosnia/Herzegovina, Serbia/Montenegro, Macedonia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, U.K. 44 0 1635 523545, Uruguay 0004 055 114

© 2014 National Instruments. All rights reserved.

Важная информация

Гарантии

Для носителя информации, на котором вы получили программное обеспечение National Instruments, гарантируется выполнение программных инструкций при отсутствии дефектов в материале и качестве изготовления носителя в течение 90 дней со дня продажи, который указан в гарантийном талоне или другом документе. National Instruments обязуется в течение гарантийного срока, по своему усмотрению, восстановить или заменить носитель информации с программным обеспечением, на котором обнаружены подобные дефекты, если получит сообщение о браке в течение срока действия гарантии. National Instruments не гарантирует стабильность и безошибочность работы программного обеспечения. Чтобы оборудование было принято на гарантийное обслуживание, на внешней поверхности его упаковки должен быть нанесен номер возврата товара, полученный на заводе. National Instruments оплатит почтовые расходы при возврате комплектующих, находящихся на гарантии. National Instruments ручается за достоверность информации в настоящем документе. Тщательно проверена техническая грамотность документа. При обнаружении технических или орфографических ошибок National Instruments оставляет за собой право внесения изменений в последующих изданиях этого документа без предварительного уведомления обладателей настоящего издания. В случае подозрений о наличии ошибок читателю следует поставить об этом в известность National Instruments. Ни при каких обстоятельствах National Instruments не несет ответственность за неисправности, связанные с использованием информации, содержащейся в настоящем документе.

КРОМЕ УКАЗАННЫХ ЗДЕСЬ СЛУЧАЕВ, NATIONAL INSTRUMENTS НЕ ДАЕТ НИКАКИХ ГАРАНТИЙ, ЯВНО ВЫРАЖЕННЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, НА КОММЕРЧЕСКУЮ ПРИГОДНОСТЬ ПРОДУКТА ДЛЯ КОНКРЕТНЫХ ЦЕЛЕЙ. ПРАВА КЛИЕНТОВ НА УСТРАНЕНИЕ ДЕФЕКТОВ, ВЫЗВАННЫХ НЕБРЕЖНЫМ ОБРАЩЕНИЕМ С ПРОДУКЦИЕЙ NATIONAL INSTRUMENTS, ДОЛЖНО ПРОИЗВОДИТЬСЯ ЗА СЧЕТ ПОКУПАТЕЛЯ. NATIONAL INSTRUMENTS НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА УБЫТКИ, ОБУСЛОВЛЕННЫЕ ПОТЕРЕЙ ДАННЫХ, ПРИБЫЛИ, ИСПОЛЬЗОВАНИЯ ПРОДУКЦИИ, А ТАКЖЕ ПОБОЧНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ДАЖЕ ЕСЛИ О НИХ ИЗВЕСТНО.

Ответственность National Instruments ограничена независимо от того, было ли изделие повреждено случайно или умышленно. Любые иски против National Instruments должны быть рассмотрены в течение одного года после возникновения оснований. National Instruments не несет ответственности за любую задержку гарантийного обслуживания в случае возникновения обстоятельств непреодолимой силы. Гарантийному ремонту не подлежат неисправности, дефекты, отказы, вызванные несоблюдением инструкций по установке, работе и обслуживанию продукции; самостоятельной модификацией; небрежным и неправильным обращением; сбоями и импульсными помехами в питающей сети, пожарами, наводнениями, авариями, действиями третьих лиц или обстоятельствами непреодолимой силы.

Авторское право

Согласно законам об авторском праве, это руководство нельзя переиздавать или распространять как в электронной, так и в печатной форме путем ксерокопирования, перезаписи, хранения в информационно-поисковых системах. Также нельзя осуществлять полный или частичный перевод без предварительного письменного согласия корпорации National Instruments. National Instruments относится с уважением к интеллектуальной собственности и призывает к этому же своих клиентов. Программное обеспечение NI защищено законами об охране авторских прав и прав на интеллектуальную собственность. Вы имеете право передавать программное обеспечение и прочие материалы, разработанные с помощью программного обеспечения National Instruments, третьим лицам в соответствии с условиями приобретенной Вами лицензии и другими законодательными ограничениями.

Торговые марки

LabVIEW, National Instruments, NI, ni.com и USRP являются торговыми марками National Instruments. За более подробной информацией о торговых марках national Instruments обратитесь к разделу Terms of Use на странице ni.com/legal. Названия других упомянутых в данном руководстве изделий и производителей также являются торговыми марками, у которых есть правообладатели.

Патенты

Для получения информации о патентах, которыми защищена продукция National Instruments, посетите страницу ni.com/patents.

Некоторые части этого продукта защищены патентом Соединенных Штатов #6,560,572.

ТЕХНИКА БЕЗОПАСНОСТИ ПРИ РАБОТЕ С ИЗДЕЛИЯМИ NATIONAL INSTRUMENTS

(1) ИЗДЕЛИЯ NATIONAL INSTRUMENTS НЕ СОДЕРЖАТ КОМПОНЕНТОВ, ПОЗВОЛЯЮЩИХ ИХ ИСПОЛЬЗОВАТЬ СОВМЕСТНО С ХИРУРГИЧЕСКИМИ ИМПЛАНТАМИ, А ТАКЖЕ В ЛЮБЫХ СИСТЕМАХ ЖИЗНЕОБЕСПЕЧЕНИЯ, ОТКАЗ КОТОРЫХ МОЖЕТ НАНЕСТИ СУЩЕСТВЕННЫЙ ВРЕД ЧЕЛОВЕКУ. ПРОВЕРКА НАДЕЖНОСТИ ФУНКЦИОНИРОВАНИЯ ИЗДЕЛИЙ В ТАКИХ СИСТЕМАХ НЕ ПРОВОДИЛАСЬ.

(2) В ЛЮБЫХ ПРИЛОЖЕНИЯХ, ВКЛЮЧАЯ ВЫШЕУПОМЯНУТЫЕ, НАДЕЖНОСТЬ ФУНКЦИОНИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ МОЖЕТ БЫТЬ СНИЖЕНА ЗА СЧЕТ ФЛЮКТУАЦИЙ ПИТАЮЩЕГО НАПРЯЖЕНИЯ, НЕИСПРАВНОСТЕЙ КОМПЬЮТЕРА, НАСТРОЕК ОПЕРАЦИОННОЙ СИСТЕМЫ, НАСТРОЕК КОМПИЛЯТОРОВ И СРЕДЫ ПРОГРАММИРОВАНИЯ, В КОТОРОЙ РАЗРАБАТЫВАЕТСЯ ПРИЛОЖЕНИЕ, ОШИБОК ПРИ УСТАНОВКЕ, ПРОБЛЕМ СОВМЕСТИМОСТИ АППАРАТНОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, НЕИСПРАВНОСТЕЙ И СБОЕВ ЭЛЕКТРОННЫХ УСТРОЙСТВ МОНИТОРИНГА ИЛИ УПРАВЛЕНИЯ, ПЕРЕМЕЖАЮЩИХСЯ ОТКАЗОВ (АППАРАТНОГО И/ИЛИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ), НЕПРАВИЛЬНОГО ОБРАЩЕНИЯ, ОШИБОК ПОЛЬЗОВАТЕЛЯ ИЛИ РАЗРАБОТЧИКА ПРИЛОЖЕНИЯ (ВСЕ ЭТИ ФАКТОРЫ НАЗЫВАЮТСЯ "СИСТЕМНЫМИ СБОЯМИ").

ЛЮБОЕ ПРИЛОЖЕНИЕ, ГДЕ СИСТЕМНЫЙ СБОЙ МОЖЕТ ПРИВЕСТИ К ПОВРЕЖДЕНИЮ ИМУЩЕСТВА, А ТАКЖЕ ТРАВМИРОВАНИЯ ИЛИ СМЕРТИ ЛЮДЕЙ, НЕ МОЖЕТ ВЫЗЫВАТЬ ДОВЕРИЕ ИЗ_ЗА РИСКА СИСТЕМНОГО СБОЯ.

ВО ИЗБЕЖАНИЕ ТАКИХ СЛУЧАЕВ РАЗРАБОТЧИК ИЛИ ПОЛЬЗОВАТЕЛЬ ПРИЛОЖЕНИЯ ДОЛЖНЫ ПРЕДУСМОТРЕТЬ МЕРЫ ЗАЩИТЫ ОТ СИСТЕМНЫХ СБОЕВ, ВКЛЮЧАЯ МЕХАНИЗМЫ РЕЗЕРВИРОВАНИЯ И ОТКЛЮЧЕНИЯ, НО НЕ ОГРАНИЧИВАЯСЬ ИМИ.

ПОСКОЛЬКУ, ВО-ПЕРВЫХ, КАЖДАЯ ГОТОВАЯ СИСТЕМА ПРИСПОСОБЛИВАЕТСЯ ДЛЯ КОНКРЕТНОГО ПОЛЬЗОВАТЕЛЯ И ОТЛИЧАЕТСЯ ОТ ИСПЫТАТЕЛЬНЫХ ПЛАТФОРМ NATIONAL INSTRUMENTS, ВО-ВТОРЫХ, ПОЛЬЗОВАТЕЛЬ ИЛИ РАЗРАБОТЧИК МОЖЕТ КОМБИНИРОВАТЬ ИЗДЕЛИЯ NATIONAL INSTRUMENTS С ДРУГИМИ ИЗДЕЛИЯМИ СПОСОБАМИ, КОТОРЫЕ НЕ ПРОВЕРЯЛИСЬ СПЕЦИАЛИСТАМИ NATIONAL INSTRUMENTS, ПОЛЬЗОВАТЕЛЬ ИЛИ РАЗРАБОТЧИК ПОЛНОСТЬЮ ОТВЕЧАЕТ ЗА ПРОВЕРКУ И ОБЕСПЕЧЕНИЕ ПРАВИЛЬНОСТИ СБОРКИ, ФУНКЦИОНИРОВАНИЯ И БЕЗОПАСНОСТИ СИСТЕМ ИЛИ ПРИЛОЖЕНИЙ, В СОСТАВ КОТОРЫХ ВХОДЯТ ИЗДЕЛИЯ NATIONAL INSTRUMENTS, ВКЛЮЧАЯ, БЕЗ ОГРАНИЧЕНИЙ, НАДЛЕЖАЩИЕ ПРОЕКТИРОВАНИЕ, ОБРАБОТКУ И УРОВНИ БЕЗОПАСНОСТИ ТАКИХ СИСТЕМ ИЛИ ПРИЛОЖЕНИЙ.

Содержание

Предисловие.....	7
Введение в USRP	9
Амплитудная модуляция.....	17
Частотное разделение каналов	30
Подавление помех от зеркального канала	38
Модуляция с двумя боковыми полосами и подавлением несущей.....	49
Частотная модуляция.....	60
Амплитудная манипуляция	70
Частотная манипуляция	86
Двоичная фазовая манипуляция (BPSK).....	100
Глазковая диаграмма.....	117
Эквалайзинг.....	128
Квадратурная фазовая манипуляция	143

Предисловие

Двенадцать лабораторных работ, представленных в данном курсе, предназначены для дополнения вводного курса по системам коммуникаций, изучаемого на младших или старших курсах программ подготовки по электротехнике или вычислительной технике. В лабораторных работах используется программно-определяемая радиосистема NI USRP; никакого дополнительного оборудования не требуется, за исключением компьютера для запуска LabVIEW Communications и обмена данными с USRP. Трансиверы USRP работают в режиме кольцевой проверки, приемник соединяется с передатчиком при помощи тестового коаксиального кабеля и аттенюатора.

Первая лабораторная работа представляет собой введение в USRP. Это упражнение короткое, чтобы оставить время преподавателю для ознакомления с LabVIEW Communications тех студентов, которые еще не знакомы с этой средой программирования. В лабораторных работах со второй по шестую рассматриваются различные аспекты аналоговой связи, включая амплитудную модуляцию (AM), частотное разделение каналов, подавление помех от зеркального канала, двухполосную модуляцию с подавленной несущей частотой и частотную модуляцию (FM). В лабораторных работах с шестой по двенадцатую рассматриваются различные методы передачи цифровых сигналов, в том числе амплитудная манипуляция, частотная манипуляция, двоичная фазовая манипуляция, глазковая диаграмма, эквалайзинг и квадратурная фазовая манипуляция. Ознакомление с аналоговой модуляцией служит в качестве введения в цифровые методы, такой подход используется в самых современных учебниках по системам связи.

Эти лабораторные работы предназначены для студентов, имеющих базовые знания о сигналах и системах и начинающих изучать системы связи. Некоторый предварительный опыт работы в LabVIEW будет полезен, однако преподаватель может провести краткий инструктаж в начале курса. Этот курс лабораторных работ не предназначен для самостоятельного изучения, и должен следовать за традиционным курсом по системам связи. Описание каждого лабораторного проекта начинается с обсуждения основ теории связи, однако оно сведено к минимуму, поскольку предполагается, что более подробная информация и контекст лабораторных проектов предоставлены в сопутствующем курсе.

Каждый из лабораторных проектов после вводной работы состоит из задания для подготовки к лабораторной работе, упражнения, выполняемого в ходе лабораторного занятия, и отчета. Как правило, каждое задание для подготовки к лабораторной работе включает создание «виртуальных приборов» (VI) в LabVIEW для реализации передатчика и приемника с использованием заданного метода модуляции. Чтобы помочь студентам структурировать свои программы и упростить взаимодействие с USRP предоставляются готовые шаблоны. Во время лабораторных занятий студенты будут тестировать свои

виртуальные приборы на USRP и исправлять ошибки. В некоторых лабораторных проектах от студентов требуется изучить влияние изменения параметров модуляции, другие проекты включают создание дополнительных VI для изучения альтернативных методов, например, дифференциальной фазовой манипуляции. Предполагается, что после выполнения каждой лабораторной работы студенты должны сдать работающие программы и функции с соответствующей документацией и результатами измерений.

Двенадцати лабораторных работ должно быть достаточно для односеместрового курса. Для курса продолжительностью десять недель можно исключить лабораторные 3 и 4, не нарушая целостности курса.

USRP является мощной и гибкой платформой для изучения систем связи, которая предоставляет благоприятные условия для выполнения продвинутых экспериментов, находящихся за рамками рассматриваемых в данном курсе лабораторных работ.

Для выполнения работ с 2 по 12 вам потребуются скачать соответствующие файлы программ LabVIEW. Скачайте файлы со страницы <http://www.ni.com/white-paper/52344/en/> и разархивируйте их в удобное для вас место. Вы будете обращаться к ним в соответствии с инструкцией к каждой лабораторной работе.

Преподаватели, пожалуйста, посетите страницу www.ntspress.com/publications/usrp-labs/ для получения дополнительной информации.

Введение в USRP

1.1 Цель

Цель этой вводной лабораторной работы - убедиться, что на компьютерах студентов установлена рабочая версия LabVIEW Communications, и что студенты знают, как подключаться к программируемой радиосистеме USRP.

1.2 Вводная информация

На форуме Wireless Innovation Forum программно-определяемая радиосистема (Software Defined Radio, SDR) определяется как:

"Радиосистема, в которой некоторые или все функции физического уровня определяются программно"¹.

SDR относится к технологии, где для реализации функций радиосвязи используются программные модули, выполняемые на аппаратной платформе общего назначения. Объединяя оборудование NI USRP с программным обеспечением LabVIEW, можно создать гибкую и функциональную платформу SDR для быстрого прототипирования беспроводных систем связи, включая разработку физического уровня, запись и воспроизведение сигналов, радиотехническую разведку, проверку правильности алгоритмов и т.п.

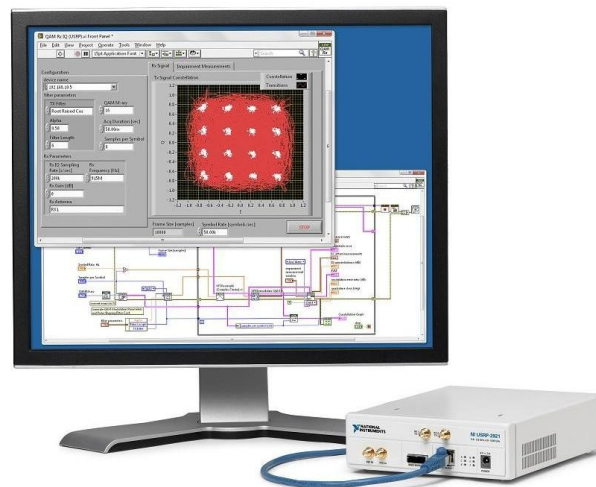


Рисунок 1. Установка оборудования в лаборатории беспроводных коммуникаций

Оборудование NI USRP

Подключение NI USRP к хост-компьютеру создает программно-определяемую радиосистему. Поступающие на разъем SMA сигналы микшируются приемником непосредственного преобразования для получения узкополосных I/Q компонент, которые дискретизируются аналого-цифровым преобразователем (ADC). Оцифрованные I/Q данные следуют затем

¹ http://www.wirelessinnovation.org/what_is_sdr

параллельными путями в тракт цифрового преобразования с понижением частоты (DDC), при этом осуществляется микширование, фильтрация и децимация входного сигнала до заданной пользователем частоты. Прореженные отсчеты передаются на хост-компьютер.

Отсчеты I/Q сигналов синтезируются хост-компьютером и передаются в USRP с заданной частотой дискретизации через Ethernet, USB или PCI Express. USRP аппаратно интерполирует входные сигналы для более высокой частоты дискретизации, используя цифровое преобразование с повышением частоты (DUC), после чего преобразует сигнал в аналоговый при помощи цифро-аналогового преобразователя (DAC). Далее результирующий аналоговый сигнал микшируется с заданной несущей частотой.

Дополнительную информацию об оборудовании NI SDR можно найти в соответствующих руководствах по началу работы² (Getting Started Guide), доступных в стартовом меню.

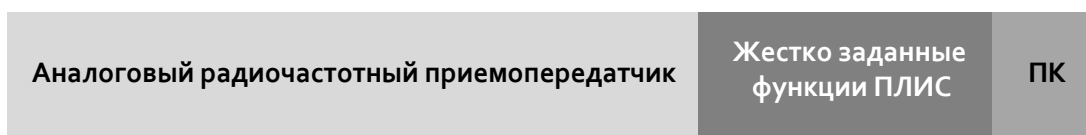
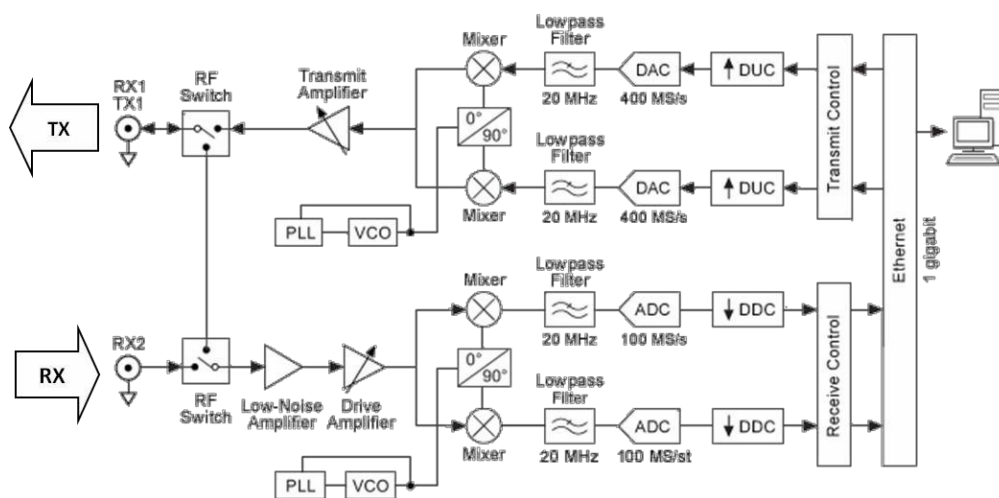


Рисунок 2. Типичная блок-схема NI USRP

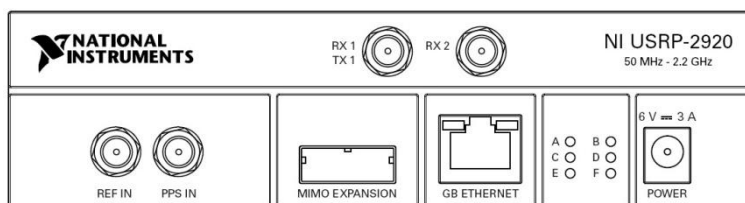


Рисунок 3. Программно-определяемая радиосистема NI USRP-2920 - вид спереди

² Доступна из меню Пуск→Все программы→National Instruments→NI-USRP→Documentation

NI LabVIEW Communications – программное обеспечение для разработки систем радиосвязи

LabVIEW – язык графического программирования, разработанный National Instruments. Основным функциональным блоком LabVIEW является виртуальный измерительный прибор (VI). Концептуально VI – это аналог процедуры или функции в традиционных языках программирования. Каждый VI состоит из *блок-диаграммы* и *лицевой панели*. Блок-диаграмма описывает функциональность VI, а лицевая панель является интерфейсом верхнего уровня VI. Конструкция VI предоставляет два важных преимущества LabVIEW: повторное использование кода и модульность. Графическая природа LabVIEW обеспечивает еще одно преимущество: она позволяет разработчикам легко визуализировать поток данных в своих проектах. NI называет это графическим проектированием систем. Кроме того, LabVIEW – полноценный язык программирования на основе потока данных, он поддерживается большим количеством документации, комплектов инструментальных средств и примеров, которые могут быть использованы при разработке.

В этом курсе вы будете использовать программно-определяемые радиосистемы (SDR) National Instruments. LabVIEW предоставляет простой интерфейс для конфигурирования и работы с внешним вводом-выводом, включая используемое в лабораторных работах оборудование NI SDR. Это главная причина, почему вы будете использовать LabVIEW в качестве языка программирования для построения SDR в этом курсе. Вы должны понимать, что рассмотренные здесь алгоритмы могут быть также запрограммированы на оптимизированном C/C++, ассемблере или VHDL и реализованы на цифровом сигнальном процессоре, микроконтроллере или ПЛИС (FPGA). Программное и аппаратное обеспечение для данного курса выбрано в основном из-за удобства.

В последующих лабораторных работах вы должны быть знакомы с LabVIEW и доступной вам документацией/справочными материалами. Это единственная лабораторная в этом курсе, которая даст вам возможность изучить и попрактиковаться в программировании в LabVIEW, поэтому важно, чтобы вы воспользовались возможностью задать преподавателю любые возникающие у вас вопросы по программированию в LabVIEW. Следующие учебные пособия и справочные материалы помогут вам изучить LabVIEW:

- онлайн-руководство LabVIEW Communications System Design Suite 1.0³
- учебные пособия LabVIEW Communications Guided Help
- контекстная справка (Context help)⁴

Окно контекстной справки отображает основную информацию об объектах LabVIEW при

³ <http://www.ni.com/documentation/en/labview-comms/1.0/manual/labview-comms-manual/>

⁴ Контекстная справка доступна в LabVIEW Communications после открытия программы из меню View→Context Help

наведении курсора на каждый объект. Для того, чтобы открыть окно контекстной справки, выберите в меню View>Show Context Help, или нажмите на клавиши <Ctrl-H>.

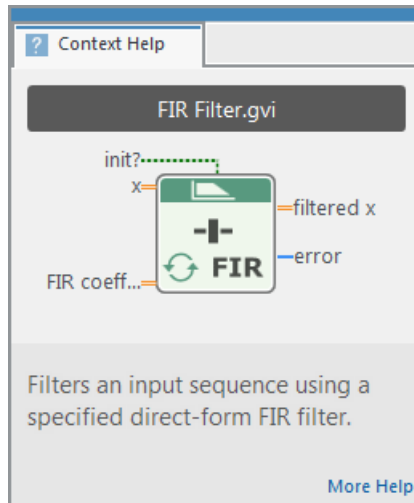


Рисунок 3. Окно Context Help (контекстная справка)

Онлайн-справка LabVIEW – наилучший источник подробной информации о конкретных свойствах и функциях LabVIEW. Статьи онлайн-справки разбивают темы на раздел общих понятий (concepts) с детальными описаниями и раздел, содержащий практические пошаговые инструкции (how-to) по использованию функций LabVIEW.

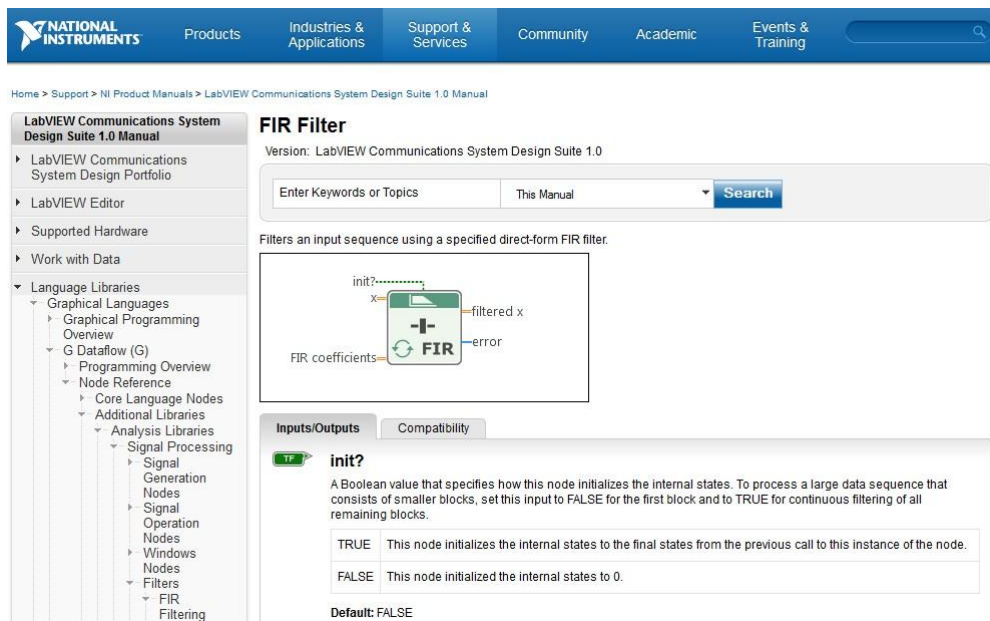


Рисунок 4. Скриншот онлайн-справки LabVIEW

1.3 Подготовка к лабораторной работе

Для выполнения работ с 2 по 12 вам потребуются скачать соответствующие файлы программ LabVIEW. Скачайте эти файлы со страницы <http://www.ni.com/white-paper/52344/en/> и разархивируйте их в удобное вам место. Вы будете обращаться к ним в соответствии с инструкцией к каждой лабораторной работе.

Преподаватели, пожалуйста, посетите страницу www.ntspress.com/publications/usrp-labs/ для получения дополнительной информации.

1. Убедитесь, что система LabVIEW Communications установлена
2. Откройте LabVIEW Communications
3. Из главного окна, называемого также вестибюлем (lobby), откройте интерактивное учебное пособие из меню Learn→Getting Started→Introduction to the LabVIEW Editor и ознакомьтесь с ним
4. Для возврата в главное окно выберите команду File→Close All
5. Откройте пять других интерактивных учебных пособий, чтобы узнать больше о LabVIEW Communications. Из главного окна выберите раздел Learn→Programming Basics, включающий в себя следующие пункты:
 - Разработка интерфейса пользователя (Designing a User Interface)
 - Отладка VI (Debugging your VI)
 - Основные типы данных (Basic Data Types)
 - Массивы (Arrays)
 - Циклы While (While Loops)

С любыми вопросами или затруднениями по поводу LabVIEW или этих учебных пособий обращайтесь к преподавателю. За оставшееся время до конца лабораторной работы вы должны ознакомиться с основами программирования в LabVIEW и узнать, куда обратиться за помощью.

1.4 Порядок выполнения лабораторной работы

1. Подключите выход TX1 к разъему SMA RX2 при помощи предоставленных тестового кабеля и аттенюатора на 30 дБ.
2. Подключите программно-определяемую радиосистему USRP к компьютеру, как описано в руководстве Getting Started Guide⁵ по вашему приемопередатчику NI USRP.
3. Запустите утилиту конфигурирования (NI USRP Configuration Utility)⁶, чтобы определить имя (*Device Name*) вашего устройства NI USRP.
4. Из главного окна LabVIEW Communications откройте следующий пример NI USRP:

Examples→Hardware Input and Output→NI USRP Host→Single Device→Single Channel→Continuous→RX Continuous Async

5. Задайте примеру имя проекта и щелкните Create.
6. Из примера откройте другой пример NI USRP:

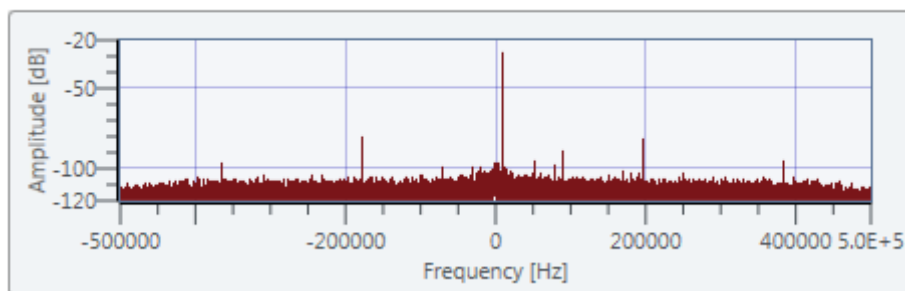
File→Examples→Hardware Input and Output→NI USRP Host→Single Device→Single Channel→Continuous→TX Continuous Async

7. Задайте примеру имя проекта и щелкните Create.
8. В примере Tx Continuous Async (будем называть его программой передатчика), введите имя устройства, найденное вами при помощи утилиты конфигурирования USRP, и обратите внимание на элемент управления *tone frequency* (*частота тона*). Эта программа генерирует один тон в основной полосе частот и отправляет его в USRP.
9. Запустите программу передатчика.
10. В окне примера Rx Continuous Async (будем называть его программой приемника), введите то же имя устройства, что и имя передатчика, и измените параметр *Active Antenna* (*Активная антенна*) на RX2.
11. Запустите программу приемника и проанализируйте график Baseband Power Spectrum (спектр мощности в основной полосе частот). Вы должны увидеть пик около центра графика. Это частота тонового сигнала, сгенерированного передатчиком.

⁵ Доступна из меню Пуск→Все программы→National Instruments→NI-USRP→Documentation

⁶ Доступна из меню Пуск→Все программы→National Instruments→NI-USRP

Baseband Power Spectrum



12. Не изменяя значения элемента управления *Carrier Frequency* (несущая частота) приемника или передатчика, "переместите" частоту сигнала на графике в положение 150 кГц.

Примечание: Вносимые в программу изменения не применяются во время выполнения программы. Вы должны остановить программу и снова запустить ее, чтобы изменения вступили в силу.

Вопросы

1. На диаграмме программы приемника запишите наименование каждого узла (называемого также функцией или блоком) и его назначение на блок-диаграмме. (Например, `niUSRP Open Rx Session` создает идентификатор сессии с устройством для других функций).
2. На диаграмме программы передатчика запишите наименование каждого узла и его назначение на блок-диаграмме. (Например, `niUSRP Open Tx Session` создает идентификатор сессии с устройством для других функций).
3. Опишите, как вы изменили пик на графике спектра мощности на 150 кГц.

1.5 Отчет

Предоставьте ответы на все вопросы из раздела *Вопросы*.

Амплитудная МОДУЛЯЦИЯ

Предварительные требования: Лабораторная работа 1 – Введение в USRP

2.1 Цель

Эта работа преследует две цели. Первая – приобрести практический опыт программирования USRP на работу в качестве передатчика и приемника. Вторая – изучить классическую аналоговую амплитудную модуляцию и детектор огибающей.

2.2 Вводная информация

Амплитудная модуляция

Амплитудная модуляция (AM) - один из старейших методов модуляции. Сегодня он все еще используется в различных системах, включая, разумеется, ширококвотательное радио с амплитудной модуляцией. В цифровой форме это наиболее распространенный метод передачи данных по оптоволокну.

Если $m(t)$ является узкополосным сигналом сообщения с пиковым значением m_p и

$A \cos(2\pi f_c t)$ является сигналом несущей с частотой f_c , тогда можно записать амплитудно-модулированный сигнал $g(t)$ как

$$g(t) = A \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_c t), \quad (1)$$

где параметр μ называется коэффициентом или индексом модуляции (modulation index) и принимает значения в диапазоне $0 \leq \mu \leq 1$ (от 0 до 100%) при нормальной работе. В частном случае, при $m(t) = m_p \cos(2\pi f_m t)$, можно записать

$$\begin{aligned} g(t) &= A \left[1 + \mu \cos(2\pi f_m t) \right] \cos(2\pi f_c t) \\ &= A \cos(2\pi f_c t) + \frac{A}{2} \mu \cos[2\pi (f_c - f_m)t] + \frac{A}{2} \mu \cos[2\pi (f_c + f_m)t]. \end{aligned} \quad (2)$$

В приведенном выше выражении первый терм – несущая, а второй и третий – нижняя и верхняя боковая частота, соответственно. На рисунке 1 Рисунок 1 показан график несущей частотой 20 кГц, модулированной синусоидой частотой 1 кГц с коэффициентом модуляции 50% и 100%.

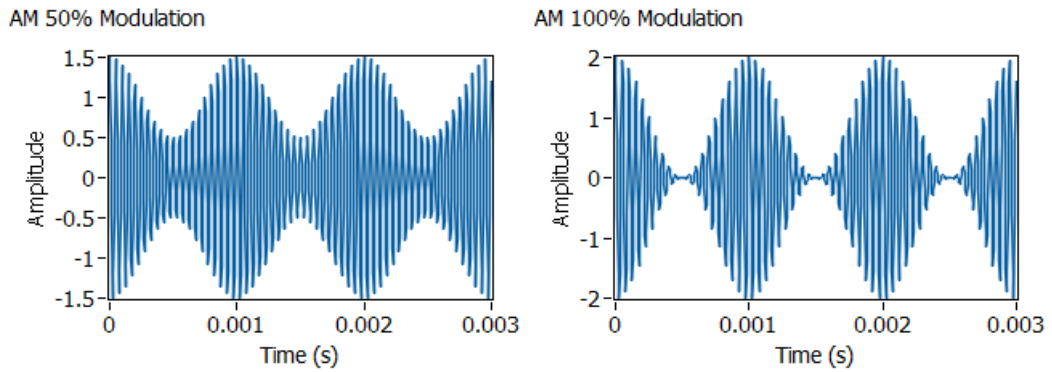


Рисунок 1. Сигналы с амплитудной модуляцией

Когда АМ-сигнал поступает на приемник, он имеет вид

$$r(t) = D \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_c t + \theta), \quad (3)$$

где амплитуда несущей D , как правило, значительно меньше амплитуды A переданной несущей, а угол θ представляет собой разность фаз между сигналами несущей генераторов передатчика и приемника. Мы последуем распространенной практике и сместим частоту генератора приемника f_0 относительно несущей частоты передатчика f_c . При этом получим сигнал

$$r_1(t) = D \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_{IF} t + \theta), \quad (4)$$

где так называемая "промежуточная" частота $f_{IF} = f_c - f_0$. Сигнал $r_1(t)$ может быть пропущен через полосовой фильтр для удаления помех от нежелательных сигналов на частотах вблизи частоты f_c . Как правило, сигнал $r_1(t)$ также подвергается усилению.

Демодуляция сигнала $r_1(t)$ наиболее эффективно выполняется детектором огибающей (envelope detector). Детектор огибающей может быть реализован с помощью выпрямителя, за которым установлен фильтр нижних частот (ФНЧ). Огибающая $A(t)$ сигнала $r_1(t)$ имеет вид

$$A(t) = D \left[1 + \mu \frac{m(t)}{m_p} \right] = D + \frac{D\mu}{m_p} m(t). \quad (5)$$

Настройка USRP

Передатчик

LabVIEW взаимодействует с передатчиком USRP при помощи четырех функций, находящихся в палитре блок-диаграммы в разделе Hardware Interfaces→NI-USRP→Tx. На рисунке 2 Рисунок 2 показана базовая структура передатчика. Эта структура является исходной для всех лабораторных работ этой серии.

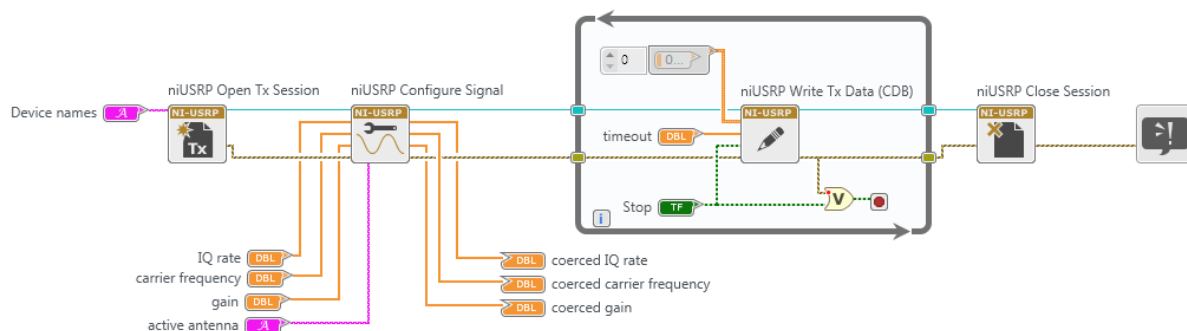


Рисунок 2. Шаблон передатчика



Open Tx Session инициализирует сессию передатчика и генерирует идентификатор сессии и кластер ошибок, которые передаются через все четыре функции. При использовании этой функции вы должны добавить элемент управления "device names" (имена устройств), посредством которого вы сообщите LabVIEW IP-адрес или имя ресурса USRP.



Configure Signal используется для задания значений параметров USRP. Подключите четыре элемента управления и три индикатора к этой функции, как показано на рисунке. Для начала установите частоту IQ равной 200 КОтсчетов/с (наименьшее возможное значение), несущую частоту равной 915.1 МГц, коэффициент усиления равным 0 дБ и выберите в качестве активной антенны TX1. Когда функция выполняется, USRP вернет реальные значения этих параметров. Эти значения отобразятся на подключенных вами индикаторах. Как правило, реальные значения параметров будут совпадать с заданными, однако если значения одного или нескольких заданных параметров лежат за пределами возможностей USRP, вместо сообщения об ошибке будет выбрано ближайшее допустимое значение параметра.



Write Tx Data записывает модулирующий (baseband) сигнал в USRP для передачи. Помещение этой функции в цикл *while* позволяет циклически посылать блок отсчетов модулирующего сигнала, пока не будет нажата кнопка "stop". Обратите внимание, что цикл *while* запрограммирован на останов при обнаружении ошибки. Отсчеты модулирующего сигнала

могут передаваться в *Write Tx Data* как массив комплексных чисел или как комплексный сигнал типа *waveform*. Лента *Configure* в верхней части окна *LabVIEW Communications* позволяет выбирать тип данных. Если модулирующий сигнал записать в виде

$$\tilde{g}(nT_x) = g_I(nT_x) + jg_Q(nT_x), \quad (6)$$

тогда сигнал, передаваемый USRP:

$$g(t) = Ag_I(t) \cos(2\pi f_c t) - Ag_Q(t) \sin(2\pi f_c t). \quad (7)$$

В этом выражении константа A задает параметр "gain" (коэффициент усиления), а f_c – несущая частота. Интервал дискретизации T_x является обратной величиной "IQ rate" (частоты IQ). Обратите внимание, что сигнал $g(t)$, выдаваемый USRP, является непрерывным во времени. Цифро-аналоговое преобразование выполняется внутри USRP.

Следует отметить, что в действительности модулирующий сигнал $\tilde{g}(nT_x)$ это два модулирующих сигнала. По давней традиции, реальная часть $g_I(nT_x)$ называется синфазной (in-phase) компонентой, а мнимая часть $g_Q(nT_x)$ – квадратурной (quadrature) компонентой модулирующего сигнала. AM-сигнал, который мы сгенерируем в данной работе, использует только синфазную компоненту

$$g_I(t) = A \left[1 + \mu \frac{m(t)}{m_p} \right], \quad (8)$$

и

$$g_Q(t) = 0. \quad (9)$$

Мы рассмотрим другие методы модуляции в последующих лабораторных работах, где будут использованы обе компоненты.



Close Session завершает работу передатчика, как только завершается выполнение цикла *while*. Обратите внимание, что выполнение функции должно быть остановлено кнопкой *STOP*, а не кнопкой "Abort Execution" на панели инструментов. Это необходимо для того, чтобы функция *Close Session* гарантировано выполнилась и корректно закрыла все используемые структуры данных.

Приемник

LabVIEW взаимодействует с приемником USRP при помощи шести функций, находящихся в палитре блок-диаграммы в разделе Hardware Interfaces→NI-USRP→Rx. На рисунке 3 показана базовая структура приемника. Эта структура является исходной для всех последующих лабораторных работ этой серии.

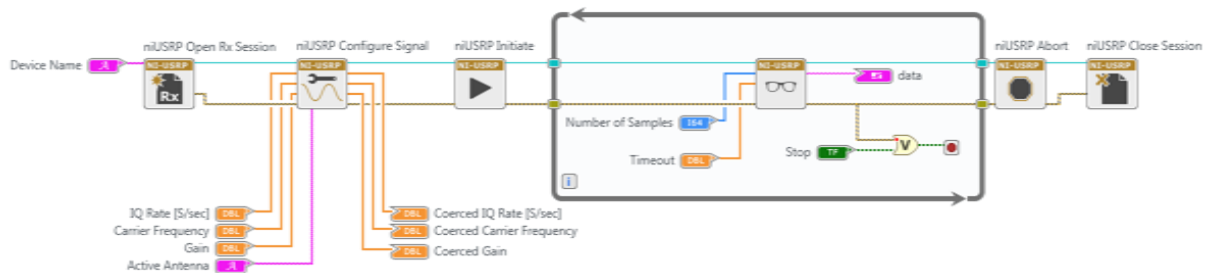


Рисунок 3. Шаблон приемника



Open Rx Session инициализирует сессию приемника и генерирует идентификатор сессии и кластер ошибок, которые передаются через все шесть функций. Вы должны добавить элемент управления "device names", посредством которого вы сообщите LabVIEW IP-адрес или имя ресурса USRP.



Configure Signal выполняет ту же функцию, что соответствующий блок в передатчике. Подключите четыре элемента управления и три индикатора к этой функции, как показано на рисунке. В этот раз установите частоту IQ равной 1 МОтчетов/с, несущую частоту равной 915.0 МГц, коэффициент усиления равным 0 дБ и выберите в качестве активной антенны RX2. Когда функция запустится, USRP вернет реальные значения этих параметров.



Initiate посылает выбранные вами значения параметров в приемник и запускает его.



Fetch Rx Data извлекает отсчеты сообщения, полученные USRP. Помещение этой функции в цикл *while* позволяет извлекать отсчеты сигнала блок за блоком, пока не будет нажата кнопка "stop". Обратите внимание, что цикл *while* запрограммирован на останов при обнаружении ошибки. "number of samples" (Количество отсчетов) позволяет задать количество отсчетов, извлекаемых в каждой итерации цикла *while*. В последующих лабораторных работах этой серии мы не будем использовать цикл *while*, и будет извлекать из приемника лишь один блок данных.

Fetch Rx Data может выдавать отсчеты сообщения пользователю массивом комплексных чисел или комплексным сигналом типа waveform. Вы можете выбрать тип данных отсчетов сообщений из выпадающего меню.



Abort останавливает сбор данных по завершении цикла *While*.



Close Session завершает работу приемника. Как отмечено выше, используйте для останова программы кнопку STOP, чтобы функция *Close Session* гарантированно исполнилась.

Иерархия файлов

Для доступа к файлам и функциям, используемых в данном курсе, щелкните дважды по проекту *Communication Systems.lvproject*. После загрузки проекта вы увидите файловую панель (Files Pane) в верхней левой части экрана, прямо под лентой. Для доступа к файлам и функциям разверните папки и дважды щелкните по нужному файлу. Вы можете также перетаскивать файлы из этой панели на блок-диаграмму.

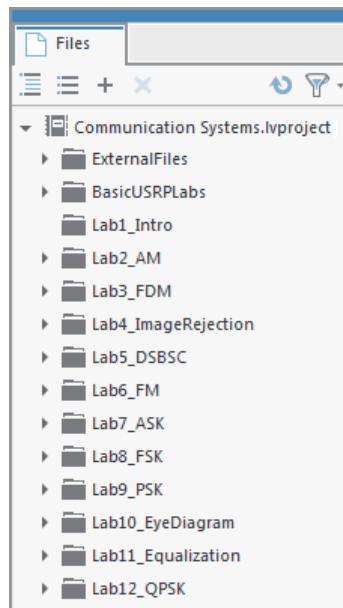


Рисунок 4. Файловая панель в LabVIEW Communications

2.3 Подготовка к лабораторной работе

Передатчик

1. Шаблон для передатчика подготовлен в файле *Lab2TxTemplate.gvi*. Этот шаблон содержит четыре функции интерфейса, описанные выше в разделе "Вводная информация", а также генератор сигналов (Basic Multitone), настроенный на генерацию трехтонального сигнала. Три тона изначально настроены на 1,2 и 3 кГц, но эти частоты можно изменять при помощи элементов управления на лицевой панели. Ваша задача – добавить необходимые блоки для генерации АМ-сигнала, после чего передать АМ-сигнал в цикл *while* в функцию Write Tx Data. Коэффициент модуляции задается пользователем в диапазоне $0 \leq \mu \leq 1$, элемент управления на лицевой панели уже имеется.

Подсказка: АМ-сигнал, сгенерированный вами, будет иметь вид $g_I(nT)$. Для $g_Q(nT)$ задайте массив, содержащий нули, той же длины, что и $g_I(nT)$. Затем объедините их в единый комплексный массив $\tilde{g}(nT) = g_I(nT) + jg_Q(nT)$. Если вы знакомы с синтаксисом .m-файлов, вы можете воспользоваться узлом MathScript для более простой реализации формулы амплитудной модуляции.

Примечания:

- a. Генератор сообщений создает сигнал, являющийся суммой набора синусоид одинаковой амплитуды. Вы можете выбрать количество синусоид, включаемых в набор, их частоты и их общую амплитуду. Однако начальные фазы синусоид выбираются случайным образом и будут изменяться при каждом запуске программы. Тем самым сигнал сообщения будет немного отличаться при каждом запуске программы.
- b. Цифро-аналоговые преобразователи USRP налагают одно практическое ограничение: масштабируйте генерируемые сигналы так, чтобы пиковое значение $|\tilde{g}(nT)|$ не превышало +/-1. (+/-1, как правило, относится к полной шкале цифроаналоговых преобразователей устройства. Любое значение выше приведет к обрезке сигнала.)
- c. Сохраните программу передатчика в файл, в имени которого содержатся буквы "AMTx" и ваши инициалы (например, *AMTx_BAB.gvi*).

Приемник

2. Шаблон для приемника подготовлен в файле *Lab2RxTemplate.gvi*. Шаблон содержит шесть функций интерфейса, рассмотренных выше в разделе "Вводная информация", а также графический индикатор waveform graph, на котором вы будете отображать демодулированный выходной сигнал.

Пропустите массив комплексных чисел с выхода функции *Fetch Rx Data* через полосовой фильтр. Фильтры находятся в папке ExternalFiles на файловой панели (Files Pane). Используйте фильтр Чебышева пятого порядка с верхней частотой среза 105 кГц и нижней частотой среза 95 кГц. Заданная по умолчанию неравномерность в полосе пропускания в 0.1 дБ приемлема. Вход частоты дискретизации (sampling frequency) должен быть получен с выхода функции *Configure Signal* "actual IQ rate".

3. Используя функцию *Complex to Re/Im*, извлеките действительную часть данных с выхода "Filtered X" фильтра Чебышева. Полученная вами действительная часть может быть описана уравнением (4). Для выделения огибающей возьмите абсолютное значение и передайте результат через фильтр нижних частот. Это сработает как двухполупериодный выпрямитель. Затем, чтобы извлечь сообщение, передайте сигнал через фильтр нижних частот. Используйте фильтр Баттерворта второго порядка (DBL) с частотой среза 5 кГц. Как и с полосовым фильтром, на вход частоты дискретизации фильтра нижних частот следует подать сигнал с выхода "actual IQ rate" функции *Configure Signal*. Выход вашего фильтра нижних частот должен быть подключен к графическому индикатору Baseband Output.

Сохраните программу приемника в файл, в имени которого содержатся буквы "AMRx" и ваши инициалы (например, *AMRx_BAB.gvi*).

Вопросы

1. Предположим, сообщение $m(t)$ имеет вид

$$m(t) = \cos(2\pi 1000t) + \cos(2\pi 2000t) + \cos(2\pi 3000t). \quad (10)$$

Найдите спектр мощности сигнала $r_1(t)$ из уравнения (4) и нарисуйте график спектра.

Выразите ваш ответ через D и μ .

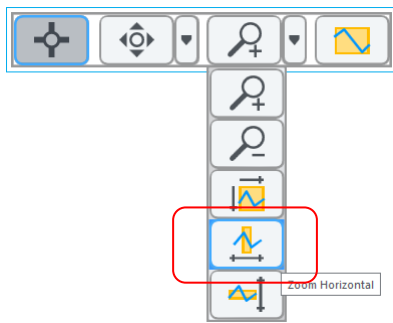
2. Для сообщения $A(t)$ из уравнения (10) найдите спектр мощности и нарисуйте график спектра. Выразите ответ через D и μ .

3. Основываясь на вашем ответе на вопрос 1, определите на качественном уровне, что произойдет с мощностью несущей при изменении индекса модуляции μ ? Что произойдет с мощностью боковых полос?

2.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.
2. Убедитесь, что для передатчика заданы следующие настройки:
Несущая частота (Carrier Frequency): 915.1 МГц
Частота IQ (IQ Rate): 200 кГц
Коэффициент усиления (Gain): 0 дБ
Активная антенна (Active Antenna): TX1
Длина сообщения (Message Length): 200 000 отсчетов - хороший блок данных для работы.
Коэффициент модуляции: Начните с 1.0
Начальная частота (Start Frequency), Дельта частоты (Delta Frequency), Количество тонов (Number of Tones): Представляется, что три тона – хорошо для этого упражнения, но следите, чтобы частота не превышала 5 кГц.
3. Убедитесь, что для приемника заданы следующие настройки:
Несущая частота (Carrier Frequency): 915 МГц
Частота IQ (IQ Rate): 1 МГц
Коэффициент усиления (Gain): Неважно. 0 дБ
Активная антенна (Active Antenna): RX2
Количество отсчетов: Такое же, что и длина передаваемого сообщения.

Запустите сначала передатчик, затем приемник. Через несколько секунд остановите приемник кнопкой STOP, потом остановите передатчик. Используйте инструмент масштабирования по горизонтали из палитры графического индикатора, чтобы растянуть сигнал "message" передатчика и сигнал "baseband output" приемника.



Оба сигнала (переданный и принятый) должны быть одинаковыми, за исключением масштаба и, возможно, смещения постоянной составляющей.

4. Спектр мощности

Перетащите функцию *FFT Power Spectrum for 1 Chan (CDB)* из папки ExternalFiles на блок-диаграмму приемника. Чтобы подать данные на вход функции спектра мощности, используйте функцию Cluster Properties. Подключите вход "Y" функции *Cluster Properties* к выходу полосового фильтра приемника. Подключите вход "dt" функции *Cluster Properties* к "dt", полученному с верхней левой функции *Cluster Properties*. Потом подключите выход "cluster out" функции *Cluster Properties* ко входу "time signal" функции *FFT Power Spectrum*. Осталось подключить только один вход функции вычисления спектра мощности: подключите булевскую константу, равную True, ко входу "dB On". Теперь подключите выход "Power Spectrum/PSD" функции вычисления спектра к графическому индикатору waveform graph. Сделайте видимой палитру "graph palette" этого индикатора, чтобы можно было использовать масштабирование, и измените метку горизонтальной оси на "Frequency".

Настройте передатчик на генерацию сообщения, состоящего из трех тонов, начинающихся с 1 кГц с шагом в 1 кГц. Установите коэффициент модуляции равным $\mu = 1$. Запустите сначала передатчик, затем приемник. Остановите сначала приемник, затем передатчик. Увеличьте график спектра мощности, чтобы можно было разглядеть компоненты в окрестностях 100 кГц. Сделайте скриншот графика спектра мощности.

Сравните спектр мощности с тем, каким предполагали его увидеть в Вопросе 1 раздела "Подготовка к лабораторной работе". На сколько дБ боковые составляющие ниже несущей?

Измените индекс модуляции на $\mu = 0.5$ и снова получите спектр мощности. Сделайте скриншот графика спектра мощности. На сколько дБ боковые составляющие теперь ниже несущей?

5. Константа D , представляющая собой амплитуду принятой несущей, может быть измерена, если пропустить огибающую из уравнения (5) через фильтр нижних частот. Измеренное значение D часто используется в реальных приемниках для подстройки коэффициента

усиления на выходе приемника, что обеспечивает возможность автоматической регулировки усиления. Добавьте ФНЧ к приемнику таким образом, чтобы выходной сигнал фильтра был пропорционален D . Запустите передатчик и приемник и измерьте значение D . Увеличьте коэффициент усиления приемника до 20 дБ и повторите измерение D . Соответствуют ли изменения D изменению коэффициента усиления приемника на 20 дБ?

6. *Элегантный приемник*

Как следует из уравнения (4), сигнал на выходе полосового фильтра имеет действительную часть. Мнимая часть описывается выражением

$$r_2(t) = A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \sin(2\pi f_{IF}t + \theta). \quad (11)$$

Тогда комплексный сигнал на выходе полосового фильтра равен

$$\begin{aligned} \tilde{r}(t) &= r_1(t) + jr_2(t) \\ &= A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_{IF}t + \theta) + jA_r \left[1 + \mu \frac{m(t)}{m_p} \right] \sin(2\pi f_{IF}t + \theta) \\ &= A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \left[\cos(2\pi f_{IF}t + \theta) + j \sin(2\pi f_{IF}t + \theta) \right] \\ &= A_r \left[1 + \mu \frac{m(t)}{m_p} \right] e^{j(2\pi f_{IF}t + \theta)}. \end{aligned} \quad (12)$$

Амплитуда $\tilde{r}(t)$ равна $A_r \left[1 + \mu \frac{m(t)}{m_p} \right]$, и это желаемый демодулированный выходной

сигнал.

Подключите выход полосового фильтра непосредственно к блоку абсолютного значения (минуя функцию *Complex to Re/Im*). Подключите выход абсолютного значения напрямую к графическому индикатору Baseband Output (минуя функцию *Butterworth Filter*). Запустите передатчик и приемник и обратите внимание, что демодулированный сигнал совпадает с полученным в пункте 3. Фильтр нижних частот не нужен! (Обратите внимание, что, поскольку

индекс модуляции μ ограничен сверху единицей, выражение $A_r \left[1 + \mu \frac{m(t)}{m_p} \right]$ всегда

положительное и абсолютное значение на него не влияет).

2.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также документацию для подпрограмм, если вы их создавали. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Ответьте на вопросы из раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также документацию для подпрограмм, если вы их создали. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше. Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы.

Предоставьте графики спектра и ответьте на все вопросы 4 и 5 из раздела "Порядок выполнения лабораторной работы".

Частотное разделение каналов

Предварительные требования: Лабораторная работа 2 – Амплитудная модуляция

3.1 Цель

В этой лабораторной работе вы изучите передачу нескольких сообщений на одной несущей с использованием частотного разделения каналов. Каждое отдельное сообщение модулируется на отдельную поднесущую, и эти промодулированные поднесущие суммируются перед отправкой в USRP. В приемнике отдельные сообщения разделяются при помощи фильтрации, после чего демодулируются. Цель данного упражнения в том, чтобы

- дополнительно попрактиковаться в программировании USRP,
- ознакомиться с концепцией частотного разделения каналов и
- изучить концепцию фильтрации промежуточной частоты в приемнике.

3.2 Вводная информация

Частотное уплотнение каналов широко используется в телеметрии, в спутниковой передаче телевизионных сигналов, и, до широкого распространения оптоволокну, это был стандартный метод передачи телефонных сигналов на большие расстояния. Частотное уплотнение каналов играет также важную роль в технике мультиплексирования с ортогональным частотным разделением каналов (OFDM), используемой в системах цифровой связи (SD-L) и системах сотовой телефонии третьего поколения.

Пусть $m_1(t)$ и $m_2(t)$ - сигналы сообщений, а f_1 и f_2 - соответствующие частоты поднесущих.

Мы можем сформировать модулированные сигналы поднесущих

$$\begin{aligned} g_1(t) &= A_1 \left[1 + \mu_1 \frac{m_1(t)}{m_{1p}} \right] \cos(2\pi f_1 t), \\ g_2(t) &= A_2 \left[1 + \mu_2 \frac{m_2(t)}{m_{2p}} \right] \cos(2\pi f_2 t). \end{aligned} \quad \text{и} \quad (1)$$

Необязательно использовать амплитудную модуляцию для модуляции вспомогательных несущих. В этой лабораторной мы используем ее, поскольку вы знакомы с АМ по Лабораторной работе 2, и потому, что АМ сигнал легко демодулировать. Обратите внимание, что частоты поднесущих f_1 и f_2 должны находиться на достаточном расстоянии друг от друга, чтобы спектры $g_1(t)$ и $g_2(t)$ не перекрывались. Сигналы $g_1(t)$ и $g_2(t)$ вместе создают

$$g_I(t) = g_1(t) + g_2(t), \quad (2)$$

синфазную компоненту сигнала. Как и в лабораторной работе 2, мы приравняем нулю квадратурную компоненту $g_Q(t)$, так что сигнал, посылаемый в USRP, будет равен

$$\tilde{g}(t) = g_I(t) + jg_Q(t) = g_I(t) = g_1(t) + g_2(t). \quad (3)$$

Тогда сигнал, фактически переданный USRP, будет равен

$$\begin{aligned} g(t) &= Ag_I(t) \cos(2\pi f_c t) - Ag_Q(t) \sin(2\pi f_c t) \\ &= A[g_1(t) + g_2(t)] \cos(2\pi f_c t), \end{aligned} \quad (4)$$

где A задается коэффициентом усиления, а f_c является несущей частотой USRP.

На приемной стороне USRP получаем выходной сигнал

$$\tilde{r}(t) = D[g_1(t) + g_2(t)] e^{j\theta}, \quad (5)$$

где θ разность фаз между сигналами генераторов приемника и передатчика, а D - константа, как правило, значительно меньше A . Этот сигнал, состоящий из комплексных компонент, может быть передан на группу из двух полосовых фильтров, настроенных на частоты поднесущих f_1 и f_2 . На выходах фильтров будут отдельные сигналы, описанные в уравнении (1). Теперь их можно демодулировать при помощи детектора огибающей, как в лабораторной работе 2.

3.3 Подготовка к лабораторной работе

Передатчик

Создайте subVI `AMonSubcarrier.gvi` для формирования сигналов $g_1(t)$ или $g_2(t)$, показанных в уравнении (1). Необходимые входы и выходы приведены в

Таблица 1таблице 1.

Таблица 1: AMonSubcarrier

Входы			
	Message waveform in (Вход сигнала сообщения)	$m_i(t)$	сигнал типа waveform (двойной точности)
	Modulation index (Коэффициент модуляции)	μ_i	число двойной точности
	Carrier level (Уровень несущей)	A_i	число двойной точности
	Sampling information for subcarrier (Сведения о дискретизации для поднесущей)		кластер
	Частота поднесущей (Subcarrier frequencies):	f_i	число двойной точности
Выход			
	Modulated signal out (Выход модулированного сигнала)	$g_i(t)$	массив (чисел двойной точности)

Создайте для вашего subVI наглядную иконку.

1. Шаблон для передатчика подготовлен в файле *Lab3TxTemplate.gvi*. Этот шаблон содержит четыре функции для интерфейса с USRP, а также два генератора сообщений, которые сгенерируют сигналы $m_1(t)$ и $m_2(t)$.
 - a. Используйте два экземпляра AMonSubcarrier subVI для создания сигналов $g_1(t)$ и $g_2(t)$.
 - b. Сложите $g_1(t)$ и $g_2(t)$ и нормализуйте пиковое значение суммы относительно единицы.

- c. Создайте еще один массив такой же длины, что и $g_1(t)$ или $g_2(t)$, но содержащий нули, используя функцию *Re/Im to Complex*. Сформируйте комплексный массив $\tilde{g}(t)$, показанный выше в уравнении (3).
 - d. Передайте $\tilde{g}(t)$ в цикл *while* и подключите к функции *Write Tx Data*.
2. Сохраните программу передатчика в файл, в имени которого содержатся буквы "MuxTx" и ваши инициалы (например, *MuxTx_BAB.gvi*).

Приемник

1. Шаблон для приемника подготовлен в файле *Lab3RxTemplate.gvi*. Шаблон содержит шесть функций интерфейса с USRP, а также два графических индикатора *waveform graph*, на которых вы будете отображать демодулированные выходные сигналы.
 - a. Пропустите массив комплексных чисел с выхода функции *Fetch Rx Data* через два полосовых фильтра. Используйте фильтры Чебышева (CDB) из папки *ExternalFiles*. Задайте одному фильтру верхнюю частоту среза 505 кГц и нижнюю частоту среза 495кГц. Задайте другому фильтру верхнюю частоту среза 515 кГц и нижнюю частоту среза 505кГц. Заданная по умолчанию неравномерность в полосе пропускания 0.1 дБ допустима для обоих фильтров. Вход частоты дискретизации (*sampling frequency*) обоих фильтров должен быть равен "actual IQ rate", полученной с функции *Configure Signal* (либо как значение, обратное *dt* из свойств кластера).
 - b. Передайте выходной сигнал каждого фильтра через детектор огибающей. Каждый детектор огибающей состоит из функции, вычисляющей абсолютное значение, и фильтра нижних частот. Настройте фильтр нижних частот, как в Лабораторной работе 2. Выходы фильтров подключите к блокам *Build Waveform*, выходы которых нужно соединить с графическими индикаторами *Message 1 Out* и *Message 2 Out*.
2. Сохраните программу приемника в файл, в имени которого содержатся буквы "MuxRx" и ваши инициалы (например, *MuxRx_BAB.gvi*).

Вопрос

Начиная с уравнения (5), покажите аналитически, что ошибка фазы θ не влияет ни на один демодулированный выходной сигнал. Обратите внимание, что в этой лабораторной мы не берем реальную часть $\tilde{r}(t)$ до фильтрации полосовым фильтром.

3.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915 МГц

Частота IQ (IQ Rate): 2 МОтсчета/с

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): TX1

Длина сообщения (Message Length): 200 000 отсчетов - хороший блок данных для работы.

Индексы модуляции (Modulation Indices): Начните с 1.0 для каждой поднесущей.

Частоты поднесущих (Subcarrier frequencies): Возьмите 500 кГц для f_1 и 510 кГц для f_2 .

Начальная частота (Start Frequency), Дельта частоты (Delta Frequency),

Количество тонов (Number of Tones): Не критично, но максимальная частота не должна превышать 5 кГц. Представляется, что три тона – хорошо для каждого сообщения. Результаты будут более интересными, если использовать низкие частоты для одного сообщения и высокие для другого, чтобы сообщения по-разному выглядели на графике.

3. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915 МГц

Частота IQ (IQ Rate): 2 МОтсчета/с

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): RX2

Количество отсчетов: Такое же, что и длина передаваемого сообщения.

Обратите внимание, что в этой лабораторной между несущими частотами приемника и передатчика нет сдвига.

1. Запустите сначала передатчик, затем приемник. Через несколько секунд остановите приемник кнопкой STOP, потом остановите передатчик. Изучите графики Message Out и убедитесь, что приемник правильно демодулирует и отображает два сигнала сообщения.

2. *Спектр мощности*

Добавьте в приемник функцию *FFT Power Spectrum for 1 Chan (CDB)* из папки ExternalFiles . Подайте на вход "time signal" сигнал, формируемый функцией *Fetch Rx Data*. Подключите булевскую константу, равную True, ко входу "dB On". Подключите графический индикатор waveform graph к выходу "Power Spectrum/PSD". Измените метку горизонтальной оси индикатора на "Frequency," и настройте горизонтальную шкалу на отображение частотных компонент между 495 кГц и 515 кГц.

Запустите сначала передатчик, затем приемник. Остановите сначала приемник, затем передатчик. Сделайте скриншот спектра для отчета. Определите две поднесущие на вашей распечатке.

3. *Перекрестные искажения*

Перекрестные искажения (crosstalk) - феномен, при котором часть сигнала из одного канала "просачивается" в соседний канал. В качестве вступления к рассмотрению перекрестных искажений мы измерим среднюю мощность двух демодулированных выходных сигналов. Функция *AC & DC Estimator*, находящаяся в папке ExternalFiles, может измерять среднеквадратическое значение переменной составляющей сигнала. Если возвести выходной сигнал "AC estimate" функции *AC & DC Estimator* в квадрат, результат будет оценкой средней мощности сигнала, за вычетом постоянной составляющей. Подключите функцию *AC & DC Estimator*, функцию возведения в квадрат и числовой индикатор к выходу каждого ФНЧ приемника.

Задайте в передатчике одинаковые настройки двух сигналов сообщения. Установите оба коэффициента модуляции равными единице. Задайте частоту дискретизации ("IQ rate") приемника равной 10 МОтчетов/с. Запустите приемник и передатчик, и запишите мощность каждого демодулированного сигнала. Теперь вы готовы измерять перекрестные искажения.

Измерение А: Отключите один из блоков *AMonSubcarrier.gvi*, чтобы передавалось только сообщение 1. Запустите приемник и передатчик и запишите мощность каждого демодулированного сигнала. Назовите эти измерения P_{1A} and P_{2A} .

Измерение В: Теперь настройте передатчик так, чтобы передавалось только сообщение 2. Запустите приемник и передатчик и запишите мощность каждого демодулированного сигнала.

Назовите эти измерения P_{1B} and P_{2B} . Вычислите перекрестные искажения

$$XT_{12} = 10\log_{10} \frac{P_{2A}}{P_{2B}} \text{ dB} \text{ и } XT_{21} = 10\log_{10} \frac{P_{1B}}{P_{1A}} \text{ dB}.$$

Величина перекрестных искажений сигнала с частотным уплотнением каналов во многом зависит от избирательности фильтров, используемых для разделения каналов в приемнике. Для иллюстрации этого измените порядок обоих полосовых фильтров Чебышева с 5 до 2 и повторите измерение перекрестных искажения. Сравните значения XT_{12} и XT_{21} при использовании фильтра 2 порядка и фильтра 5 порядка.

3.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию по *AMonSubcarrier.gvi* и по созданным вами функциям для передатчика и приемника. Предоставьте также документацию для подпрограмм, если вы их создавали. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Ответьте на вопросы из раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте *AMonSubcarrier.gvi* и созданные вами функции для передатчика и приемника. Предоставьте также все дополнительные подпрограммы, если вы их создавали. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше.

Повторно предоставьте документацию для любых функций, которые вы модифицировали во время выполнения лабораторной работы.

Предоставьте график спектра из пункта 4 и всю информацию об искажениях, требуемую в пункте 5 из раздела "Порядок выполнения лабораторной работы".

Подавление помех от зеркального канала

Предварительные требования: Лабораторная работа 2 – Амплитудная модуляция

4.1 Цель

Упражнения этой лабораторной работы иллюстрируют главную проблему супергетеродинных приемников. Подавление помех от зеркального канала выполняется при помощи сложной фильтрации. В этой лабораторной работе представляется техника обработки, не представляющая затруднений в программно-определяемых радиосистемах, но практически невозможная в обычных аппаратно реализованных приемниках.

4.2 Вводная информация

Преобразование частоты

Как мы видели в лабораторной работе 2, в большинстве приемников систем связи перед демодуляцией полученный сигнал с частотой несущей f_c для усиления и фильтрации преобразуют в сигнал с промежуточной частотой f_{IF} (IF - intermediate frequency). В USRP преобразование частоты может быть выполнено путем сдвига несущей частоты приемника относительно несущей частоты переданного сигнала. Во избежание путаницы, генератор несущей приемника обычно называется местным (local) генератором, а его частота - частотой местного генератора f_{LO} . В лабораторной работе 2 мы устанавливали несущую частоту передатчика равной $f_c = 915.1$ МГц, а частоту местного генератора равной $f_{LO} = 915$ МГц. При таких настройках промежуточная частота равна $f_{IF} = f_c - f_{LO} = 100$ кГц.

В приемнике USRP преобразование частоты выполняется аппаратно путем умножения принятого сигнала на $\cos(2\pi f_{LO}t)$ и $-\sin(2\pi f_{LO}t)$. Например, пусть полученный АМ-модулированный сигнал равен

$$r(t) = A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_c t + \theta). \quad (1)$$

Приемник формирует

$$\begin{aligned}
r(t)\cos(2\pi f_{LO}t) &= A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_c t + \theta) \cos(2\pi f_{LO}t) \\
&= \frac{1}{2} A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos[2\pi(f_c - f_{LO})t + \theta] \\
&\quad + \frac{1}{2} A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos[2\pi(f_c + f_{LO})t + \theta].
\end{aligned} \tag{2}$$

Из двух термов в уравнении (2) первый имеет промежуточную частоту $f_{IF} = f_c - f_{LO}$, а второй - куда более высокую частоту $f_c + f_{LO}$. Высокочастотная компонента удаляется при помощи фильтрации USRP, формирующей синфазный сигнал

$$r_I(t) = A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_{IF}t + \theta). \tag{3}$$

Приемник формирует и второй сигнал,

$$\begin{aligned}
-r(t)\sin(2\pi f_{LO}t) &= -A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \cos(2\pi f_c t + \theta) \sin(2\pi f_{LO}t) \\
&= \frac{1}{2} A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \sin[2\pi(f_c - f_{LO})t + \theta] \\
&\quad - \frac{1}{2} A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \sin[2\pi(f_c + f_{LO})t + \theta].
\end{aligned} \tag{4}$$

И снова терм с высокой частотой удаляется, и приемник предоставляет квадратурный сигнал

$$r_Q(t) = A_r \left[1 + \mu \frac{m(t)}{m_p} \right] \sin(2\pi f_{IF}t + \theta). \tag{5}$$

Обычные аппаратные радиоприемники, как правило, работают с синфазным сигналом из уравнения (3). USRP комбинирует синфазный и квадратурный сигналы для формирования комплексного IF-сигнала $\tilde{r}(t)$, который имеет вид

$$\begin{aligned}
\tilde{r}(t) &= r_I(t) + jr_Q(t) \\
&= A_r \left[1 + \mu \frac{m(t)}{m_p} \right] [\cos(2\pi f_{IF}t + \theta) + j \sin(2\pi f_{IF}t + \theta)] \\
&= A_r \left[1 + \mu \frac{m(t)}{m_p} \right] e^{j(2\pi f_{IF}t + \theta)}.
\end{aligned} \tag{6}$$

Именно этот комплексный сигнал получается на выходе функции *Fetch Rx Data*.

В частотной области спектр $R(f)$ полученного сигнала $r(t)$, описанного уравнением (1), показан на рисунке 1.

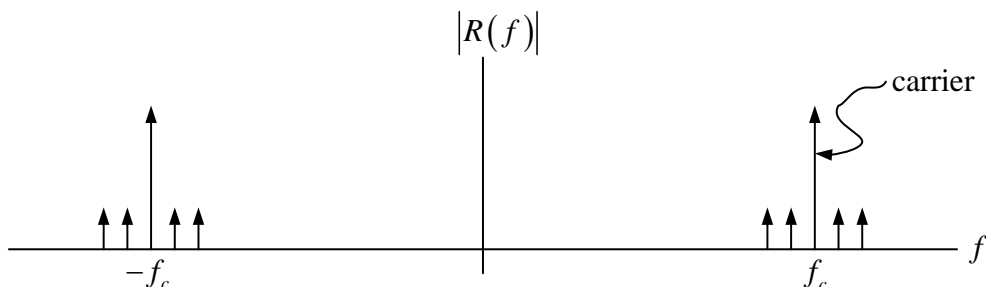


Рисунок 1. Спектр принятого АМ-сигнала

Поскольку $r(t)$ - сигнал с действительной составляющей, его спектр содержит как положительные, так и отрицательные частотные компоненты. После преобразования частоты комплексный IF-сигнал $\tilde{r}(t)$ имеет спектр $\tilde{R}(f)$, показанный на рисунке 2 Рисунок 2.

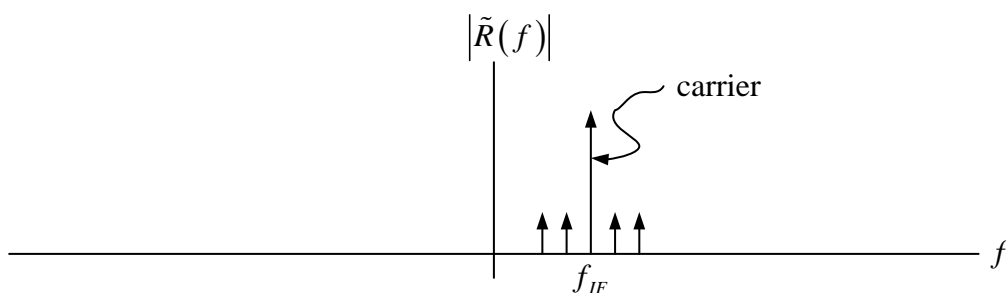


Рисунок 2. Спектр комплексного IF-сигнала

Обратите внимание, что $\tilde{r}(t)$ содержит только положительные частотные компоненты.

В лабораторной работе 2 мы пропускали комплексный IF-сигнал через полосовой фильтр. На рисунке 3 показана частотная характеристика полосового фильтра. Предполагается, что сигналы с несущими, отличающимися от f_{IF} , не будут пропущены фильтром.

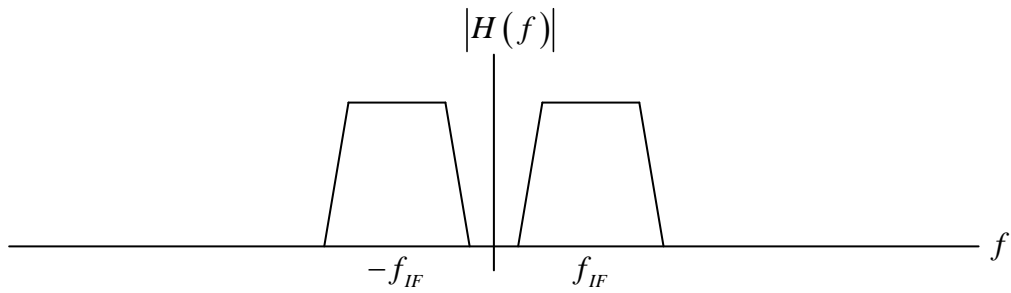


Рисунок 3. Частотная характеристика фильтра промежуточной частоты

Зеркальный сигнал

Предположим, вместе с сигналом из уравнения (1) получен еще один сигнал, имеющий вид

$$r_2(t) = A_{r2} \left[1 + \mu_2 \frac{m_2(t)}{m_{2p}} \right] \cos(2\pi f_{IM} t + \theta_2), \quad (7)$$

где несущая частота

$$f_{IM} = f_{LO} - f_{IF}. \quad (8)$$

Проанализировав уравнения (7) и (8), мы обнаружим, что второй сигнал формирует комплексный IF-сигнал $\tilde{r}_2(t)$, имеющий вид

$$\tilde{r}(t) = A_{r2} \left[1 + \mu_2 \frac{m_2(t)}{m_{2p}} \right] e^{j(-2\pi f_{IF} t + \theta_2)}. \quad (9)$$

Спектр $\tilde{R}_2(f)$ этого сигнала показан на рисунке 4.

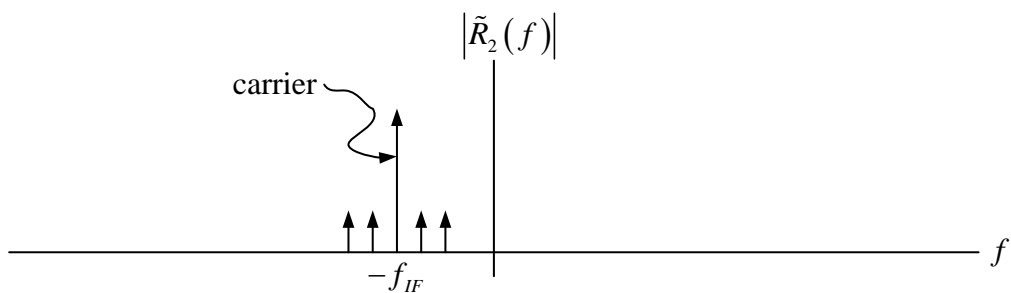


Рисунок 4. Спектр зеркального сигнала

Говорят, что сигналы $r(t)$ и $r_2(t)$ являются зеркальными по отношению друг к другу. Посмотрев на частотную характеристику на рисунке 3, мы увидим, что оба сигнала $r(t)$ и $r_2(t)$ проходят через фильтр промежуточной частоты, и два сигнала будут интерферировать друг с другом в демодуляторе, следующем за этим фильтром. Стоит обратить внимание на соотношение частот двух зеркальных сигналов. Один сигнал, $r(t)$, имеет несущую частоту $f_c = f_{LO} + f_{IF}$, а второй сигнал, $r_2(t)$, имеет несущую частоту $f_{IM} = f_{LO} - f_{IF}$. Эти несущие частоты симметричны относительно частоты приемника f_{LO} , таким же образом, как физический объект и его отражение симметричны относительно поверхности зеркала.

Подавление помех от зеркального канала

В обычном приемнике нежелательный зеркальный сигнал, как правило, удаляется до преобразования частоты. Для этого требуется поместить аналоговый фильтр во входном каскаде. Ясно, что у нас нет возможности добавить аналоговое оборудование в USRP, так что мы воспользуемся "искусственной" обработкой сигнала для удаления нежелательного зеркального сигнала после преобразования частоты. Обратите внимание, что на рисунке 2 нужный сигнал появляется после преобразования на промежуточной частоте f_{IF} , а на рисунке 4 мы видим, что нежелательный зеркальный сигнал появляется на промежуточной частоте $-f_{IF}$. Традиционный полосовой фильтр пропускает оба эти сигнала, но если бы мы могли создать фильтр, пропускающий только положительную частоту, то могли бы пропустить нужный сигнал, и подавить зеркальный. На рисунке 5 показана необходимая нам частотная характеристика.

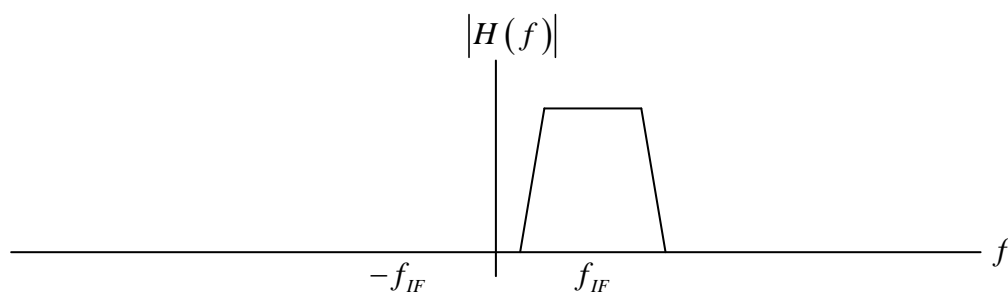


Рисунок 5. Частотная характеристика комплексного фильтра промежуточной частоты

Показанная на рисунке 5 частотная характеристика не обладает симметрией относительно вертикальной оси, что подразумевает комплексность импульсной характеристики фильтра. Фильтр с комплексной импульсной характеристикой трудно создать аппаратно, но, как мы увидим, это легко делается программно.

На практике аналоговый режекторный фильтр должен быть перестраиваемым, чтобы приемник мог принимать сигналы в диапазоне несущих частот f_c . Чтобы сделать фильтр перестраиваемым, обычно он должен быть простым, то есть низкого порядка. Как правило, используются фильтры второго порядка. Фильтр низкого порядка не очень хорошо способен подавить нежелательные зеркальные сигналы. Напротив, комплексный режекторный фильтр, показанный на рисунке 5 Рисунок 5, отцентрирован по фиксированной промежуточной частоте, и потому не должен быть перестраиваемым. Качество подавления зеркального канала ограничено только способностью фильтра подавлять сигналы на отрицательных частотах.

4.3 Подготовка к лабораторной работе

Комплексный фильтр уже подготовлен в файле `ChebyshevHilbert.gvi`. Ваша задача - создать программу для поиска частотной характеристики фильтра. Для этого существуют несколько способов, и вы можете сами выбрать метод.

1. Задайте фильтру:

- частоту дискретизации 1 МОтсчет/с
- верхнюю частоту среза 105 кГц
- нижнюю частоту среза 95 кГц
- порядок 5
- можно принять неравномерность, равную по умолчанию 0.1 дБ.

2. Выведите на график частотную характеристику в децибелах в диапазоне частот -200 кГц - 200 кГц. (Ось частот должна быть линейна, это не график Боде).

Примечание:

Вход и выход фильтра представляют собой массивы комплексных чисел. Если вы решите измерять частотную характеристику путем подачи синусоидальных сигналов различных частот, вам необходимо будет использовать комплексные синусоиды $e^{j2\pi ft}$, а не обычные $\cos(2\pi ft)$, чтобы отделить отрицательные частоты от положительных.

4.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание.
2. Откройте программы AM-передатчика и AM-приемника, созданные вами в лабораторной работе 2.

Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.1 МГц

Частота IQ (IQ Rate): Неважно. 200 кГц

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): TX1

Длина сообщения (Message Length): 200 000 отсчетов - хороший блок данных для работы.

Коэффициент модуляции (Modulation Index): Начните с 1.0

Начальная частота (Start Frequency), Дельта частоты (Delta Frequency), Количество тонов (Number of Tones): Не критично, но максимальная частота не должна превышать 5 кГц. Представляется, что три тона – неплохо для работы.

Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915 МГц

Частота IQ (IQ Rate): 1 МГц

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): RX2

Количество отсчетов (Number of Samples): Такое же, что и длина передаваемого сообщения.

Запустите передатчик и приемник и убедитесь, что на выходе приемника появляется демодулированное сообщение.

3. Модифицируйте функции приемника, добавив компоненты для индикации мощности принятой несущей. Для этого вспомните, что выход АМ-демодулятора имеет вид

$$r_o(t) = D \left[1 + \mu \frac{m(t)}{m_p} \right]. \quad (10)$$

Поскольку среднее значение сообщения $m(t)$ равно нулю, средним значением $r_o(t)$ будет принятая несущая D . Вы можете усреднить $r_o(t)$ при помощи полосового фильтра с частотой среза ниже самой низкочастотной компоненты $m(t)$, или же найдя подходящую функцию усреднения.

Запустите передатчик и приемник и запишите значение принятой несущей D .

4 Спектр мощности

Добавьте в приемник функцию *FFT Power Spectrum for 1 Chan (CDB)* из папки ExternalFiles. Подайте на вход "time signal" сигнал, формируемый функцией *Fetch Rx Data*. Подключите булевскую константу, равную True, ко входу "dB On". Подключите графический индикатор waveform graph к выводу "Power Spectrum/PSD". Измените метку горизонтальной оси индикатора на "Frequency."

Запустите передатчик и приемник. Сделайте скриншот спектра полученного сигнала. Спектр должен соответствовать показанному на рисунке 2. Рисунок 2

5. Воспользуйтесь уравнением (8) для определения частоты зеркального канала f_{IM} . Задайте частоту передатчика равной f_{IM} . Запустите передатчик и приемник и запишите значение принятой несущей D_2 . Вычислите коэффициент ослабления по зеркальному каналу (*image rejection ratio, IRR*) по формуле

$$IRR = 20 \log_{10} \frac{D}{D_2}. \quad (11)$$

(Вы должны получить результат около 0 дБ).

Сделайте еще один скриншот спектра полученного сигнала. На этот раз спектр должен соответствовать показанному на рисунке 4. Рисунок 4

6. Замените в приемнике полосовой фильтр на `ChebyshevHilbert.gvi`. Задайте несущую частоту передатчика равной $f_c = 915.1$ МГц, запустите передатчик и приемник и измерьте D . Задайте несущую частоту передатчика равной f_{IM} , запустите передатчик и приемник и измерьте D_2 . Рассчитайте IRR и сравните с результатом, полученным в пункте 5.
7. Сохраните программу приемника в файл, в имени которого содержатся буквы "AMImageRx" и ваши инициалы (например, `AMImageRx_BAB.gvi`).
8. *Вопрос повышенной сложности* (требуется знакомство с преобразованием Гилберта)

Предположим, в пункте 6 вы хотите получить сигнал на несущей частоте f_{IM} , а не на f_c .

Модифицируйте `ChebyshevHilbert.gvi` соответствующим образом. (Подсказка: потребуется всего лишь изменить знак в одном месте блока `mathScript`).

4.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию к программе, созданной вами для измерения частотной характеристики `ChebyshevHilbert.gvi`. Предоставьте документацию для модифицированного приемника. Предоставьте также документацию для подпрограмм, если вы их создавали. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Предоставьте график частотной характеристики из пункта 2 из раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте функции, созданные вами для измерения частотной характеристики `ChebyshevHilbert.gvi`. Предоставьте функции для модифицированного приемника. Предоставьте также все дополнительные подпрограммы, если вы их создавали. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше.

Повторно предоставьте документацию для любых функций, которые вы модифицировали в лабораторной работе.

Предоставьте графики спектра для пунктов 4 и 5.

Предоставьте вычисления коэффициента ослабления зеркального сигнала и результаты, полученные при выполнении пунктов 5 и 6.

Если вы выполнили задание повышенной сложности (пункт 8), покажите, как вы модифицировали *ChebyshevHilbert.gvi* и опишите, как вы проверяли результат.

Модуляция с двумя боковыми полосами и подавлением несущей

Предварительные требования: Лабораторная работа 2 – Амплитудная модуляция

5.1 Цель

В этой лабораторной работе рассматривается модуляция с подавлением несущей. Реализация демодулятора представлена простой схемой синхронизации фазы и частоты.

5.2 Вводная информация

Модуляция с двумя боковыми полосами и подавлением несущей

Амплитудная модуляция по своей сути неэффективна, поскольку большая часть передаваемой мощности сосредоточена в несущей. В схемах с подавлением несущей несущая просто не передается. На практике используются два распространенных метода подавления несущей, с двумя боковыми полосами и подавлением несущей (double-sideband suppressed-carrier, DSB-SC) и с одной боковой полосой (single-sideband, SSB). Модуляция с двумя боковыми полосами и подавлением несущей похожа на амплитудную модуляцию, за исключением того, что несущая не передается.

Если $m(t)$ - модулирующий сигнал сообщения, а $\cos(2\pi f_c t)$ является сигналом несущей с частотой f_c , тогда можно записать DSB-SC-сигнал $g(t)$ как

$$g(t) = Am(t)\cos(2\pi f_c t). \quad (1)$$

Для тех случаев, когда $m(t) = m_p \cos(2\pi f_m t)$, можно записать

$$\begin{aligned} g(t) &= Am_p \cos(2\pi f_m t)\cos(2\pi f_c t) \\ &= \frac{Am_p}{2} \cos[2\pi(f_c - f_m)t] + \frac{Am_p}{2} \cos[2\pi(f_c + f_m)t]. \end{aligned} \quad (2)$$

Два термина в уравнении (2) обозначают нижнюю и верхнюю боковые полосы, соответственно. Терма несущей частоты f_c нет. На рисунке 1 Рисунок 1 показан график несущей в 20 кГц, модулированной синусоидой в 1 кГц по методу DSB-SC.

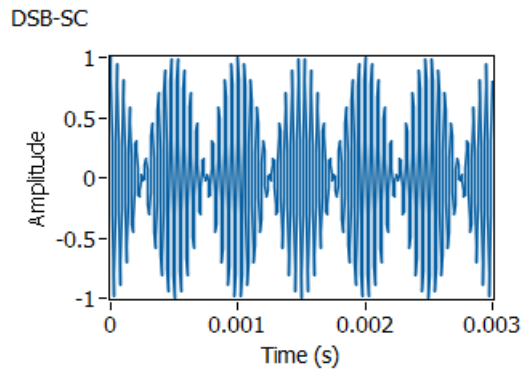


Рисунок 1. Модуляция с двумя боковыми полосами и подавлением несущей

DSB-SC-сигнал, поступающий на приемник, имеет вид

$$r(t) = Dm(t)\cos(2\pi f_c t + \theta), \quad (3)$$

где D является константой, как правило, значительно меньше A , а угол θ представляет собой разность фаз между генераторами несущей передатчика и приемника. Если генератор несущей приемника (местный генератор) настроен на ту же частоту, что и генератор передатчика, USRP сгенерирует два демодулированных сигнала

$$\begin{aligned} r_I(t) &= \frac{D}{2} m(t) \cos(\theta), \text{ и} \\ r_Q(t) &= \frac{D}{2} m(t) \sin(\theta). \end{aligned} \quad (4)$$

Функция *Fetch Rx Data* представляет эти демодулированные сигналы, как один комплексный сигнал $\tilde{r}(t)$ вида

$$\begin{aligned} \tilde{r}(t) &= \frac{D}{2} m(t) \cos(\theta) + j \frac{D}{2} m(t) \sin(\theta) \\ &= \frac{D}{2} m(t) e^{j\theta}. \end{aligned} \quad (5)$$

Заманчиво предположить, что можно извлечь сообщение $m(t)$ из $\tilde{r}(t)$, взяв амплитуду комплексного сигнала. К сожалению, амплитуда $\tilde{r}(t)$

$$|\tilde{r}(t)| = \frac{D}{2} |m(t)|, \quad (6)$$

где абсолютное значение представляет нежелательное искажение сигнала сообщения. Более продуктивно использовать синфазный (действительный) сигнал $r_I(t)$ из уравнения (4). Коэффициент $\cos(\theta) r_I(t)$ представляет собой константу усиления. К сожалению, значение этой константы пользователь не может контролировать, и оно может быть маленьким, если θ окажется вблизи $\pm\pi/2$. Более того, если частоты генераторов приемника и передатчика слегка отличаются, погрешность фазы θ будет изменяться со временем, заставляя $r_I(t)$ появляться и пропадать. В следующем разделе обсуждается, как мы можем скомпенсировать терм $\cos(\theta)$.

Синхронизация фаз

Существуют различные методы, которые могут быть использованы для устранения терма $\cos(\theta)$, ответственного за фазовую погрешность. Представленный здесь метод прост и легко реализуем в LabVIEW. Основные операции метода:

- Оценить θ
- Умножить $\tilde{r}(t)$ на $e^{-j\theta}$, чтобы получить $\tilde{r}(t)e^{-j\theta} = \frac{D}{2}m(t)e^{j\theta}e^{-j\theta} = \frac{D}{2}m(t)e^{j0}$
- Взять действительную часть: $\text{Re}\left\{\frac{D}{2}m(t)e^{j0}\right\} = \frac{D}{2}m(t)\cos(0) = \frac{D}{2}m(t)$.

Оценка θ требует нескольких шагов. Сначала обратите внимание, что фазовый угол $\tilde{r}(t)$ изменяется скачком на $\pm\pi$ каждый раз, когда $m(t)$ меняет знак. Для устранения этих скачков фазы, вначале $\tilde{r}(t)$ возводят в квадрат:

$$\tilde{r}^2(t) = \frac{D^2}{4}m^2(t)e^{j2\theta}. \quad (7)$$

Поскольку квадрат значения $m^2(t)$ никогда не изменяет знак, скачки фазы исключены. Угол 2θ можно извлечь функцией Complex to Polar из палитры Data Types → Numeric → Complex. На данной стадии будет полезно сгладить отклонения в 2θ , вызванные помехами. Для этого хорошо подходит фильтр *median filter* из папки ExternalFiles. Для параметров "left rank" и "right rank" можно принять значения по умолчанию. Затем функция Unwrap Phase из палитры Analysis → Signal Processing → Signal Operations удалит скачки $\pm 2\pi$. Наконец, деление на два даст нам желаемую оценку фазовой погрешности θ . На блок-диаграмме, приведенной на рисунке 2, показан весь процесс синхронизации фаз.

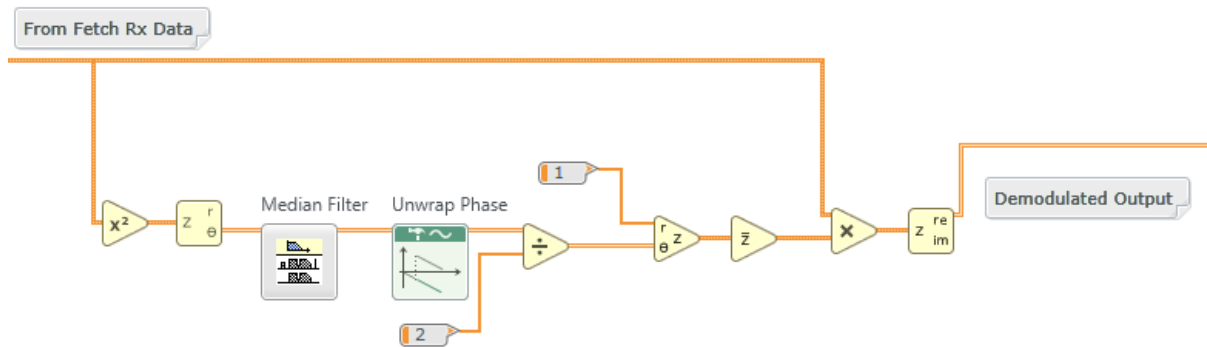


Рисунок 2. Синхронизация фаз

5.3 Подготовка к лабораторной работе

Передатчик

1. Шаблон для передатчика подготовлен в файле *Lab5TxTemplate.gvi*. Этот шаблон содержит четыре функции интерфейса, а также генератор сообщений, настроенный на генерацию трехтонального сигнала. Три тона изначально настроены на 1,2 и 3 кГц, но эти частоты можно изменять при помощи элементов управления на лицевой панели. Ваша задача - добавить необходимые блоки для генерации DSB-SC-сигнала, после чего передать DSB-SC-сигнал в цикл *while* на функцию *Write Tx Data*.

Подсказка: DSB-SC-сигнал, сгенерированный вами, будет иметь вид $g_I(nT)$. Для $g_Q(nT)$ задайте массив, содержащий нули, той же длины, что и $g_I(nT)$. Потом объедините их в единый комплексный массив $\tilde{g}(nT) = g_I(nT) + jg_Q(nT)$.

Примечания:

- а. Генератор сообщений создает сигнал, являющийся суммой набора синусоид одинаковой амплитуды. Вы можете выбрать число синусоид, включаемых в набор, их частоты и их общую амплитуду. В этом шаблоне для генератора сигнала сообщения задано начальное значение (*seed*). Поэтому начальные фазовые углы синусоид будут одинаковы при каждом запуске программы. В результате каждый раз будет генерироваться одно и то же сообщение, что

полезно при отладке. Чтобы вернуться к случайной характеристике сообщения, задайте параметр `seed` равным -1 .

- b. Цифро-аналоговые преобразователи USRP налагают одно практическое ограничение: масштабируйте генерируемые сигналы так, чтобы пиковое значение $|\tilde{g}(nT)|$ не превышало ± 1 . (Проверьте функцию *Quick Scale 1Dfunction* в папке ExternalFiles).
- c. Сохраните программу передатчика в файл, в имени которого содержатся буквы "DSBSCTx" и ваши инициалы (например, *DSBSCTx_BAB.gvi*).

Приемник

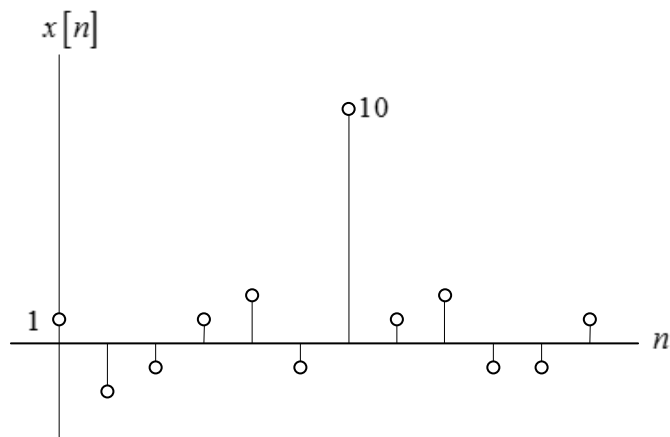
- 2 Шаблон для приемника подготовлен в файле *Lab5RxTemplate.gvi*. Шаблон содержит шесть функций интерфейса, а также графический индикатор *waveform graph*, на котором вы будете отображать демодулированный выходной сигнал.

Завершите программу, демодулировав возвращаемый функцией *Fetch Rx Data* комплексный массив и отобразив результат. Включите синхронизацию фаз с рисунка 2 Рисунок 2. Кроме того, для облегчения отладки, добавьте графический индикатор для отображения зависимости фазовой ошибки θ от времени.

Сохраните программу приемника в файл, в имени которого содержатся буквы "DSBSCRx" и ваши инициалы.

Вопросы

1. На приведенном ниже графике показана последовательность отсчетов $x = [1 \ -2 \ -1 \ 1 \ 2 \ -1 \ 10 \ 1 \ 2 \ -1 \ -1 \ 1]$. Предположим, что значения $x[n]$ для непоказанных значений индексов равны нулям. Обратите внимание на единственный выделяющийся отсчет при $n = 6$.



Пусть у нас есть простой *медианный фильтр*, выходной сигнал которого $y[n]$

$$y[n] = \text{median}\{x[n-2], \dots, x[n+2]\}, \quad n = 0, \dots, 11.$$

Найдите $y[n]$ для приведенной последовательности $x[n]$.

- Предположим, что частота генератора приемника отличается от частоты генератора передатчика на небольшую величину Δf . Модифицируйте уравнения (4) и (5), чтобы они включали этот сдвиг по частоте.

5.4 Порядок выполнения лабораторной работы

- Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.
- Убедитесь, что для передатчика заданы следующие настройки:
 Несущая частота (Carrier Frequency): 915 МГц
 Частота IQ (IQ Rate): Неважно. 200 кГц
 Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): TX1

Длина сообщения (Message Length): 200 000 отсчетов - хороший блок данных для работы.

Начальная частота (Start Frequency), Дельта частоты (Delta Frequency), Количество тонов (Number of Tones): Не критично, но максимальная частота не должна превышать 5 кГц. Три тона – неплохо для работы.

3. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915 МГц

Частота IQ (IQ Rate): 200 кГц

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): RX2

Количество отсчетов: Такое же, что и длина передаваемого сообщения.

Запустите сначала передатчик, затем приемник. Через несколько секунд остановите приемник кнопкой STOP, потом аналогично остановите передатчик. Используйте опцию масштабирования по горизонтали из палитры графического индикатора, чтобы растянуть сигнал сообщения "message" передатчика и сигнал "demodulated output" приемника. Сигналы должны быть одинаковы, за исключением масштабирования.⁷

4. Модифицируйте приемник, чтобы рассчитать $r_I(t)$ без синхронизации по фазе и $r_I(t)$ с синхронизацией фаз. Выведите оба сигнала на один график. Запустите приемник несколько раз и наблюдайте за выходами. Можете увидеть эффект, оказываемый термом $\cos(\theta)$ на несинхронизированный выход?
5. Попробуйте использовать АМ-приемник из Лабораторной работы 2 для демодуляции DSB-SC-сигнала. Обратите внимание, что вам понадобится сдвинуть частоту передатчика на 915.1 МГц. Запустите передатчик и приемник. Сделайте скриншот передаваемого сообщения и

⁷ Демодулированный сигнал может быть инвертирован. Это следствие возведения сигнала в квадрат на стадии синхронизации фаз. Ошибка $\pm 2\pi$ угла 2θ вовсе не является ошибкой, но когда значение угла делится на два, ошибка становится равна $\pm\pi$.

демодулированного сигнала. Растяните графики по временной оси, чтобы четко видеть сигналы. Смог ли детектор огибающей АМ-приемника корректно демодулировать DSB-SC-сигнал?

6. Синхронизатор фаз может также корректировать небольшой сдвиг по частоте. Используйте приемник и передатчик DSB-SC-сигнала и сдвиньте частоту передатчика на 10 Гц. Запустите передатчик и приемник. Сделайте скриншот передаваемого сообщения, несинхронизированного демодулированного выхода и синхронизированного демодулированного выхода. Растяните графики по временной оси, чтобы четко видеть сигналы. Убедитесь в правильности синхронизированного демодулированного выходного сигнала, возможно, с точностью до инверсии.

Повторите для сдвигов по частоте в 100 Гц и 1 кГц. Может ли синхронизатор фаз справиться со сдвигом в 1 кГц?

5.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию для передатчика и приемника. Предоставьте также документацию для подпрограмм, если вы их создавали. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Предоставьте ответы на вопросы, приведенные в конце раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте документацию по созданным вами функциям для DSB-SC-передатчика и DSB-SC-приемника. Предоставьте также все дополнительные подпрограммы, если вы их создавали. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше. Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы.

В соответствии с пунктом 4 опишите влияние на демодулированный сигнал термина $\cos(\theta)$.

Предоставьте графики для пункта 5. Обсудите, может ли DSB-SC-сигнал быть правильно демодулирован детектором огибающей.

Предоставьте графики для пункта 6. Прокомментируйте, смог ли синхронизатор фаз скомпенсировать сдвиги по частоте на 100 Гц и 1 кГц.

Частотная МОДУЛЯЦИЯ

6.1 Цель

В этой лабораторной работе рассматривается частотная модуляция. Эта работа является наглядной иллюстрацией применения программно-определяемых радиосистем, поскольку принцип программной реализации алгоритмов создания и демодуляции FM-сигналов гораздо проще тех, что используются в традиционном аппаратном подходе.

6.2 Вводная информация

Частотная модуляция

Частотную модуляцию (FM) изобрел Э.А. Армстронг в 1930-ых в качестве альтернативы AM, в то время широко применяемой в радиовещании. Преимущество частотной модуляции в том, что при заданной передаваемой мощности отношение сигнал-шум на выходе приемника значительно выше, чем для AM. Цифровая версия частотной модуляции, частотная манипуляция (frequency-shift keying) используется с еще более ранних времен.

При частотной модуляции частота несущей модулируется в зависимости от амплитуды сигнала сообщения. Точнее, если $m(t)$ - сигнал сообщения с пиковым значением m_p , тогда *мгновенное значение частоты (instantaneous frequency)* $f(t)$ несущей равно

$$f(t) = f_c + k_f m(t), \quad (1)$$

где f_c - частота несущей, а k_f - коэффициент пропорциональности, называемый частотной чувствительностью (frequency sensitivity). Терм $k_f m(t)$ называется девиацией мгновенного значения частоты (frequency deviation) относительно несущей частоты, а *пиковая девиация частоты (peak frequency deviation)* $\Delta f = k_f m_p$ - важный параметр FM-систем. Зная мгновенное значение частоты, мы можем найти полный мгновенный угол $\theta(t)$ несущей, проинтегрировав мгновенное значение частоты. То есть,

$$\begin{aligned} \theta(t) &= 2\pi \int_0^t f(\alpha) d\alpha \\ &= 2\pi f_c t + 2\pi k_f \int_0^t m(\alpha) d\alpha + \theta(0). \end{aligned} \quad (2)$$

Поскольку начальный угол $\theta(0)$ неважен, мы можем упростить выражение, приняв $\theta(0) = 0$. Тогда передаваемый FM-сигнал имеет вид

$$\begin{aligned}
g(t) &= A_c \cos[\theta(t)] \\
&= A_c \cos\left[2\pi f_c t + 2\pi k_f \int_0^t m(\alpha) d\alpha\right] \\
&= A_c \cos\left[2\pi f_c t + 2\pi \Delta f \int_0^t \left[m(\alpha)/m_p\right] d\alpha\right].
\end{aligned}
\tag{3}$$

На рисунке 1 показана несущая частотой 2 кГц, модулированная синусоидальным сигналом сообщения частотой 200 Гц.

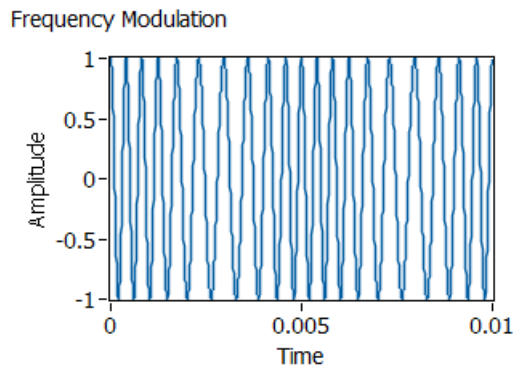


Рисунок 1. Частотно-модулированный сигнал

Для создания FM-сигнала в USRP сигнал сообщения нормализуется относительно пикового значения, умножается на $2\pi\Delta f$ и интегрируется для получения $2\pi\Delta f \int_0^t [m(\alpha)/m_p] d\alpha$. Затем формируется комплексный сигнал $\tilde{g}(t)$, где

$$\tilde{g}(t) = A_c e^{j2\pi\Delta f \int_0^t [m(\alpha)/m_p] d\alpha}.
\tag{4}$$

Комплексный сигнал $\tilde{g}(t)$ посылается на функцию Write Tx Data, и USRP создает FM-сигнал.

Единственным сложным шагом при генерации FM-сигнала является интегрирование сообщения. В дискретном времени мы имеем

$$\int_0^t x(\alpha) d\alpha \cong \sum_{k=0}^n x(nT)T,
\tag{5}$$

где T является величиной, обратной частоте дискретизации IQ. Если записать

$$y[n] = \sum_{k=0}^n x(nT)T, \quad (6)$$

то

$$\begin{aligned} y[n] - y[n-1] &= \sum_{k=0}^n x(nT)T - \sum_{k=0}^{n-1} x(nT)T \\ &= x(nT)T. \end{aligned} \quad (7)$$

Уравнение (7) является разностным уравнением IIR-фильтра. Этот фильтр можно реализовать при помощи функции IIR Filter из палитры Analysis→Signal Processing→Filters→IIR Filtering. Используйте массив "forward coefficients", равный $[T]$ и массив "reverse coefficients", равный $[1 \ -1]$.

Демодуляция

FM-демодуляцию гораздо легче выполнить в USRP, чем на традиционном оборудовании. Сигнал на выходе функции Fetch Rx Data $\tilde{r}(t)$ имеет вид

$$\tilde{r}(t) = A_r e^{j2\pi\Delta f \int_0^t [m(\alpha)/m_p] d\alpha}. \quad (8)$$

Это аналогично выражению (4), за исключением амплитуды A_r и присутствия шума (не показанного в этом выражении). Угол $\tilde{r}(t)$ легко извлекается при помощи функции *Complex to Polar*.⁸ Разверните угол, прежде чем переходить к следующему шагу. Далее развернутый угол дифференцируется, давая $2\pi\Delta f [m(t)/m_p] = 2\pi k_f m(t)$. Результат необходимо передать через ФНЧ, поскольку дифференцирование, как правило, усиливает высокочастотные помехи.

Для реализации дифференцирующего звена обратим внимание, что в дискретном времени

$$\frac{dx(t)}{dt} \cong \frac{x[n] - x[n-1]}{1}. \quad (9)$$

Дифференцирующее звено можно реализовать при помощи функции IIR Filter из палитры Analysis→Signal Processing→Filters→IIR Filtering. Для массива "FIR coefficients" используйте $[1 \ -1]$.

⁸ Для знакомых с традиционной FM-демодуляцией, эту операцию реализует ограничитель.

6.3 Подготовка к лабораторной работе

Передатчик

- 1 Шаблон для передатчика подготовлен в файле *Lab6TxTemplate.gvi*. Этот шаблон содержит четыре функции интерфейса, а также генератор сообщений, настроенный на генерацию трехтонального сигнала. Три тона изначально настроены на 1,2 и 3 кГц, но эти частоты можно изменять при помощи элементов управления лицевой панели. Ваша задача - добавить необходимые блоки для формирования сигнала $\tilde{g}(t)$ из уравнения 4, после чего передать сигнал на функцию Write Tx Data. Задайте уровень несущей A_c постоянным и равным 0.9. На рисунке 2 показано, как реализовать FM-модулятор, описанный в уравнении (4).

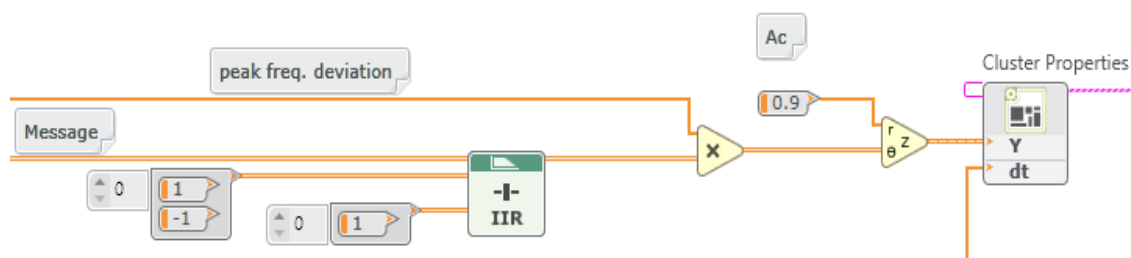


Рисунок 2. FM-модулятор

Сохраните программу передатчика в файл, в имени которого содержатся буквы "FMTx" и ваши инициалы (например, *FMTx_BAB.gvi*).

Приемник

- 2 Шаблон для приемника подготовлен в файле *Lab6RxTemplate.gvi*. Шаблон содержит шесть функций интерфейса, а также графический индикатор waveform graph, на котором вы будете отображать демодулированный выходной сигнал.

Завершите программу для демодуляции возвращаемого функцией Fetch Rx Data комплексного массива и отображения результата. На рисунке 3 показана реализация FM-демодулятора, включая извлечение угла, развертывание, дифференцирование и низкочастотную фильтрацию.

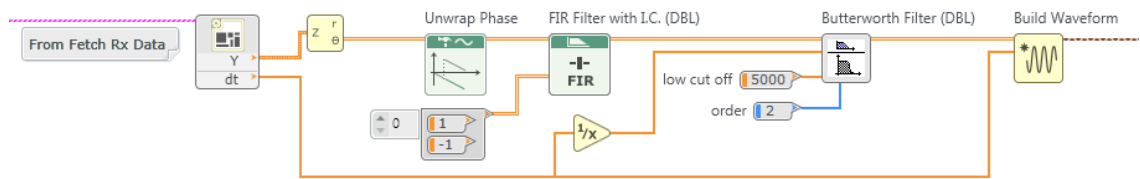


Рисунок 3. FM-демодулятор

Сохраните программу приемника в файл, в имени которого содержатся буквы "FMRx" и ваши инициалы (например, *FMRx_BAB.gvi*).

Вопросы

1. Найдите частотную характеристику интегрирующего звена, описанного уравнением (7). Сравните с частотной характеристикой идеального интегрирующего звена. Когда дискретный интегратор больше похож на идеальный - при низкой или при высокой входной частоте?
2. Найдите частотную характеристику дифференцирующего звена, описанного уравнением (9). Сравните с частотной характеристикой идеального дифференцирующего звена. Когда дискретное дифференцирующее звено больше похоже на идеальное - при низкой или при высокой входной частоте?

6.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенуатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.

Обратите внимание, что все графики в этой лабораторной работе отображаются в линейном масштабе (dB on = False).

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915 МГц

Частота IQ (IQ Rate): Неважно; 1 МГц

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): TX1

Длина сообщения (Message Length): 200 000 отсчетов - хороший блок данных для работы.

Пиковая девиация частоты (Peak Frequency Deviation): 30 кГц - хорошее значение для начала.

Начальная частота (Start Frequency), Дельта частоты (Delta Frequency), Количество тонов (Number of Tones): Не критично, но максимальная частота не должна превышать 5 кГц. Три тона - неплохо для работы, но возможно, вам захочется начать с одного тона для проверки функционирования.

3. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915 МГц

Частота IQ (IQ Rate): 1 МГц

Коэффициент усиления (Gain): Неважно. 0 дБ

Активная антенна (Active Antenna): RX2

Количество отсчетов (Number of Samples): Такое же, что и длина передаваемого сообщения.

Запустите сначала передатчик, затем приемник. Через несколько секунд остановите приемник кнопкой STOP, потом аналогично остановите передатчик. Используйте опцию масштабирования по горизонтали из палитры графического индикатора, чтобы растянуть сигнал "message" передатчика и сигнал "demodulated output" приемника. Сигналы должны быть одинаковы, за исключением масштабирования.

4. Полосу пропускания FM-сигнала весьма трудно рассчитать аналитически. Дж. Р. Карсон в 1920 году предложил практический способ оценки полосы пропускания:

$$B_{FM} \cong 2(\Delta f + B), \quad (10)$$

где B_{FM} - полоса частот FM-сигнала, Δf - пиковая девиация частоты сигнала, и B - полоса частот сообщения.

Добавьте в передатчик функцию *FFT Power Spectrum and PSD* из папки *ExternalFiles*. На вход "time signal" подайте комплексный сигнал, посылаемый на функцию *USRP Write Tx Data*. Подключите выход "Power Spectrum/PSD" к графическому индикатору. Измените метку горизонтальной оси индикатора на "Frequency". Полученный вами спектр будет идентичен спектру фактически переданного FM-сигнала, за исключением того, что несущая появится на частоте 0 Гц, нижняя боковая полоса будет в области отрицательных частот, а верхняя – в области положительных частот.

Для получения классического FM-спектра задайте однотональное сообщение частотой 1 кГц. Запустите передатчик и получите спектр мощности переданного сигнала для пиковой девиации частоты 1 кГц, 5 кГц и 30 кГц. Сделайте скриншот спектра для каждого случая. Не забудьте отмасштабировать горизонтальную ось, чтобы был виден каждый спектр. Снабдите ваши графики примечаниями, чтобы показать полосу частот, вычисленную по правилу Карсона из уравнения (10) для каждого случая.

Для получения более реалистичных спектров задайте трехтональное сообщение с частотами 1 кГц, 2 кГц и 3 кГц. Запустите передатчик и получите спектр мощности переданного сигнала для пиковой девиации частоты 1 кГц, 5 кГц и 30 кГц. Снова сделайте скриншоты спектров для каждого случая. Не забудьте отмасштабировать горизонтальную ось, чтобы был виден каждый спектр. Снабдите ваши графики примечаниями, чтобы показать полосу частоты, вычисленную по правилу Карсона для каждого случая.

5. Одним из наиболее любопытных и очень полезных явлений, связанных с FM, является так называемый эффект захвата частоты (capture effect). Для наблюдения этого эффекта вам потребуется модифицировать передатчик таким образом, чтобы он одновременно генерировал два FM-сигнала с различными уровнями несущей.

В программе передатчика скопируйте функцию *Basic Multitone* для генерации второго сигнала сообщения. Две ваших функции *Basic Multitone* могут иметь общий вход "sampling info", но должны иметь индивидуальные наборы входов *start frequency*, *stop frequency* и *#tones*. Константы "T" и "1" тоже могут быть общими.

Внутри цикла *while* передатчика создайте второй модулятор, добавив второе интегрирующее звено. Создайте на лицевой панели два элемента управления "Carrier 1" и "Carrier 2" для задания уровней несущей A_c для каждого модулятора. Для простоты оба модулятора могут иметь общую

пиковую девиацию частоты. Сложите два FM-сигнала с модуляторов и передайте сумму на функцию *Build Waveform*, данные с которой поступают на *USRP Write Tx Data*.

Настройте первый генератор сигналов сообщения на генерацию трех тонов с частотами 1 кГц, 2 кГц и 3 кГц. Настройте второй генератор сигналов сообщения на генерацию трех тонов с частотами 100 Гц, 200 Гц и 300 Гц. Эти два сигнала должны быть легко различимы в приемнике. Задайте пиковую девиацию частоты равной 30 000 Гц.

Протестируйте вашу систему, задав несущую Carrier 1 равной 0.9 и несущую Carrier 2 равной нулю. Запустите передатчик и приемник и убедитесь, что выход приемника совпадает с сообщением 1. Далее обнулите несущую Carrier 1, задайте несущую Carrier 2 равной 0.9 и убедитесь, что выход приемника совпадает с сообщением 2.

В приемнике добавьте функцию *FFT Power Spectrum and PSD* для просмотра спектра сигнала. Назовите горизонтальную ось "Frequency" и задайте диапазон от 0 до 5000 Гц. Сообщения будет легче различить в частотной области, чем во временной.

Теперь вы готовы наблюдать эффект захвата! Начните с того, что задайте несущую Carrier 1 равной 0.4 и несущую Carrier 2 равной 0.6. Запустите передатчик и приемник. Сделайте скриншот спектра выходного сигнала приемника. Повторите эксперимент, задав несущую Carrier 1 равной 0.5 и несущую Carrier 2 равной 0.5. Повторите эксперимент третий раз, задав несущую Carrier 1 равной 0.6 и несущую Carrier 2 равной 0.4. В первом и третьем случае вы должны обнаружить, что приемник демодулирует (захватывает) только более сильный сигнал. В этом и заключается эффект захвата: если два FM-сигнала принимаются на одной и той же несущей частоте, приемник демодулирует только более мощный сигнал, даже если его несущая лишь немного мощнее, чем у второго сигнала.

6.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию для программ передатчика и приемника. Предоставьте также документацию для подпрограмм, если вы их создавали. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Предоставьте ответы на вопросы, приведенные в конце раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте программы, созданные вами для реализации FM-передатчика и FM-приемника.

Предоставьте также все дополнительные подпрограммы, если вы их создавали. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше. Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы.

Предоставьте графики для пункта 4. Не забудьте отметить на каждом графике ширину полосы частот, определенную по правилу Карсона. Конкретно для спектра трехтонального сообщения: всегда ли боковые полосы FM-сигнала симметричны относительно несущей?

Предоставьте графики для пункта 5. Кратко объясните, как эта последовательность графиков демонстрирует эффект захвата.

ЛАБОРАТОРНАЯ РАБОТА 7

Амплитудная манипуляция

Предварительные требования: Лабораторная работа 2 – Амплитудная модуляция

7.1 Цель

Амплитудная манипуляция (Amplitude-shift keying, ASK) - самый простой вид цифровой модуляции. Мы используем ее для введения в цифровые системы связи, а также для ознакомления с некоторыми общими свойствами цифровых систем связи, такими, как кодирование символов, формирование импульсов, согласованная фильтрация, сравнение с порогом и синхронизация импульсов.

В этой лабораторной работе и разработка передатчика, и разработка приемника – интересные задачи, которые предоставляют возможности для исследования. Лабораторная работа разделена на несколько частей. В части, относящейся к передатчику, рассматривается формирование ASK-сигнала и влияние формы передаваемого импульса на полосу частот передаваемого сигнала. В части, относящейся к приемнику, рассматриваются демодуляция, согласованная фильтрация и детектирование сигнала. В третьей части рассматривается выравнивание битовых потоков приемника и передатчика.

7.2 Часть 1: Передатчик

Вводная информация

Амплитудная манипуляция, по сути - это амплитудная модуляция двоичного сигнала сообщения. Для иллюстрации этого предположим, что $m(t)$ - двоичное сообщение, представленное в биполярном формате без возврата к нулю (NRZ), как показано на рисунке 1.

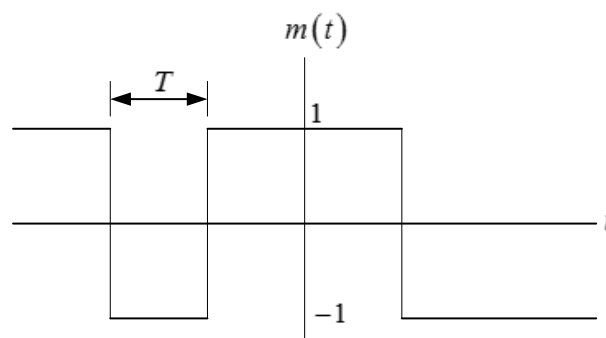


Рисунок 1. Двоичный сигнал сообщения

В биполярном формате без возврата к нулю двоичная 1 представлена импульсом амплитуды +1, а двоичный 0 - импульсом амплитуды -1. Каждый импульс имеет длительность T секунд. Определим АМ-сигнал, как

$$g_{ASK}(t) = A[1 + m(t)]\cos(2\pi f_c t), \quad (1)$$

где f_c - несущая частота, и A - амплитуда несущей. Из уравнения (1) очевидно, что или $g_{ASK}(t) = 2A\cos(2\pi f_c t)$, или $g_{ASK}(t) = 0$, в зависимости от того, равен ли соответствующий бит сообщения 1 или 0. Таким образом, несущая "включается" для передачи 1 и "выключается" для передачи 0. Иногда такой режим ASK называют "манипуляцией вкл-выкл" ("on-off" keying).

Длительность импульса T на рисунке 1 будет называться длительностью символа в этой и последующих лабораторных работах. Тогда скорость передачи символов равна $1/T$. В двоичных методах модуляции, подобных ASK, скорость передачи символов и скорость передачи в битах (bit rate) идентичны. Однако мы ознакомимся и с другими методами модуляции, например, с фазовой манипуляцией, где один символ может передавать несколько бит, и потому скорость передачи в битах может превышать скорость передачи символов.

Какой бы простой не казалась ASK, для формирования модулированного сигнала требуется выполнить несколько четко определенных операций. А именно:

1. Кодирование символов. Входные данные появляются в виде потока битов. Биты могут быть представлены в различных форматах. Далее мы увидим, что функция *MT Generate Bits* формирует массив, содержащий числа 0 и 1 (8-разрядных целых чисел). Поток битов может быть также представлен булевским массивом. На стадии кодирования символов биты заменяются числовыми значениями. Для ASK мы представим двоичную 1 комплексным числом $1 + j0$ двойной точности, а двоичный 0 - комплексным числом $-1 + j0$ двойной точности. Обратите внимание, что мы представляем биты комплексными числами, несмотря на то, что мнимая часть равна нулю. Это нужно потому, что для USRP требуются комплексные входные данные, а также потому, что в последующей работе мы используем мнимую часть для передачи дополнительной информации. Комплексные числа, представляющие входные биты, называются символами. В таблице 1 показано кодирование символов ASK.

Таблица 1: Кодирование символов ASK

Значение бита	Символ
0	$-1 + j0$
1	$1 + j0$

2. Передискретизация. Далее мы увидим, что символы передаются импульсами, форма которых важна для определения полосы частот передаваемого сигнала. Первый шаг в замене символов импульсами - размещение $L - 1$ нулей после каждого символа. Это даст интервал дискретизации

$$T_x = \frac{T}{L}, \quad (2)$$

или частоту дискретизации

$$\frac{1}{T_x} = L \frac{1}{T}. \quad (3)$$

Более высокий коэффициент передискретизации L упростит цифро-аналоговое преобразование в передатчике, но потребует более быстрой цифровой обработки. В данной лабораторной работе мы используем $L = 20$. Для реализации этой операции специально изготовлена функция *Upsample*.

3. Формирование импульсов. Если передискретизированный сигнал в виде прямоугольных импульсов с единичной амплитудой и длительностью L отсчетов пропустить через фильтр с импульсной характеристикой $g_{TX}[n]$, то на выходе фильтра каждый символ будет представлен прямоугольным импульсом. На рисунке 2 показано влияние передискретизации и фильтрации. Сигнал (а) представляет последовательность символов, где символы появляются каждые T секунд. Сигнал (b) показывает последовательность символов после передискретизации. Сигнал (с) показывает передискретизированную последовательность символов после прохождения через фильтр, формирующий импульсы. Обратите внимания, что сигнал (с) является дискретизированной во времени версией биполярного сигнала сообщения без возврата к нулю, показанного на рисунке 1. Важное преимущество представленной здесь последовательности операций для генерации сигнала сообщения в том, что импульсная характеристика $g_{TX}[n]$

фильтра, формирующего импульсы, необязательная должна иметь прямоугольную форму. Далее мы увидим, что в использовании других форм импульса могут быть свои преимущества.

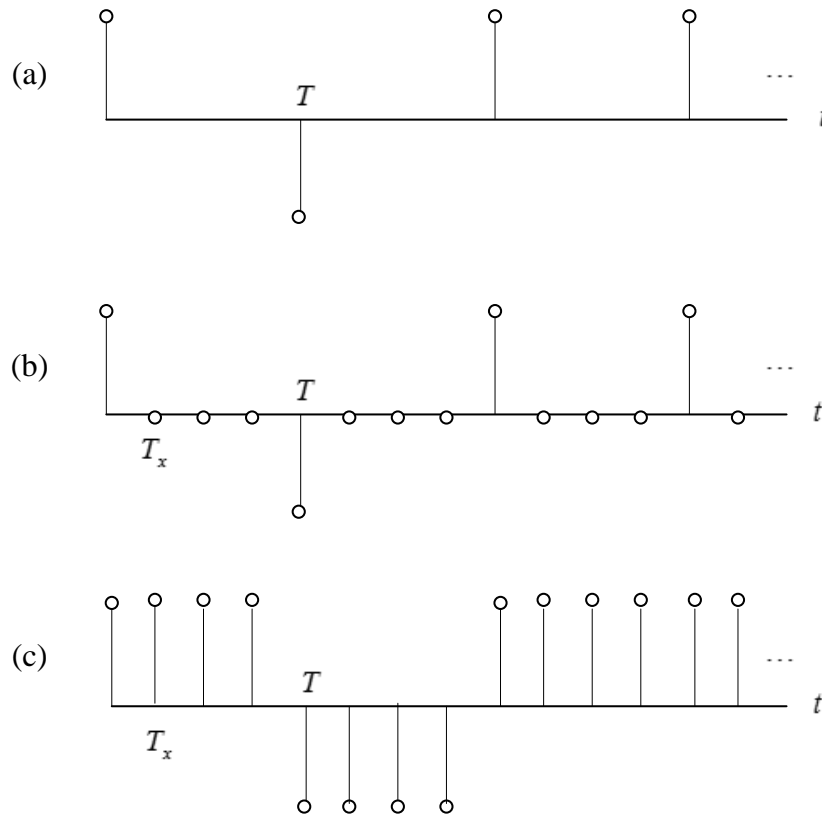


Рисунок 2. (а) Последовательность символов, (б) Передискретизированная последовательность символов, (с) Отфильтрованная передискретизированная последовательность символов

4. Модуляция. Если $m[n]$ представляет собой сигнал сообщения на выходе фильтра передатчика, то для выполнения завершающего шага необходимо применить уравнение (4):

$$\tilde{g}_{ASK}[n] = A(1 + m[n]), \quad (4)$$

где константа A выбрана, чтобы сохранить амплитуду $\tilde{g}_{ASK}[n]$ меньше единицы. Этот сигнал может быть передан в передатчик USRP. Передатчик USRP выполнит цифро-аналоговое преобразование и умножение на несущую $\cos(2\pi f_c t)$.

7.3 Подготовка к лабораторной работе

1. Создайте программу для генерации ASK-сигнала при помощи USRP. Шаблон для передатчика подготовлен в файле *ASKTxTemplate.gvi*. Этот шаблон содержит четыре функции интерфейса с USRP, а также функцию *MT Generate Bits* из комплекта Modulation Toolkit. Функция *MT Generate Bits* сгенерирует псевдослучайную последовательность битов, которая может служить последовательностью данных для тестирования вашей ASK-системы. Обратите внимание, что по умолчанию *MT Generate Bits* будет выдавать одну и ту же последовательность при каждом запуске программы. Это полезно для отладки, однако если вы захотите, чтобы каждый раз генерировалась новая последовательность, подключите функцию random number (случайное число) ко входу "seed in" (начальное значение). В последующих пунктах подробно описывается, как создать требуемый передатчик.
2. Сначала выполните кодирование символов, как показано в таблице 1. При этом целые числа (0 и 1) с выхода функции *MT Generate Bit* преобразуются в комплексные числа двойной точности $\pm 1 + j0$. В последующих лабораторных работах, где кодирование символов может быть более сложным, оно может быть реализовано в subVI. В этой лабораторной работе кодирование сравнительно простое, поэтому создавать subVI не обязательно.
3. Выполните передискретизацию функцией *Upsample* из субпалитры Analysis→Signal Processing→Signal Operation. В этой лабораторной работе у вас есть управляющие входы для задания скорости передачи символов $1/T$ и частоты IQ $1/T_x$. Задайте скорость передачи символов равной 10000 символов/с и частоту IQ 200×10^3 отсчетов/с. Используйте эти два входа для вычисления коэффициента передискретизации L .
3. Используйте функцию *MT Generate Filter Coefficients* из комплекта Modulation Toolkit для создания фильтра, формирующего импульсы. (Функцию *MT Generate Filter Coefficients* можно найти в субпалитре Analysis→Communications→Digital→Utilities.) Задайте тип модуляции ASK, а вход "samples per symbol" (количество отсчетов на символ) равным рассчитанному значению L . Создайте на лицевой панели элемент управления для входа "pulse shaping filter" (фильтр, формирующий импульсы) и изначально установите его в состояние "none" (нет). В таком режиме будут генерироваться прямоугольные импульсы. ("none" не означает отсутствия фильтра!) Подключите выход "pulse shaping filter coefficients" ко входу "Y" функции *Convolution*. Эта функция находится в субпалитре Analysis→Signal Processing→Signal Operation. Подключите выходной передискретизированный сигнал ко входу "X" функции *Convolution*.

4. Нормализуйте амплитуду отфильтрованного сигнала сообщения к максимальному абсолютному значению, равному единице. Это будет важно позднее, когда мы рассмотрим альтернативные формы импульсов. Возьмите функцию *Quick Scale 1D* (*Быстрое масштабирование*) из папки ExternalFiles, чтобы найти максимум абсолютного значения. Для гарантии того, что отмасштабированное сообщение останется комплексным, используйте отдельную функцию деления для выполнения масштабирования. (То есть, не используйте выход $Y[i] = X[i]/\text{Max}|X|$ функции *Quick Scale 1D*.)

5. Реализуйте уравнение (4). Пусть константа A равна $1/2$, тогда $\tilde{g}_{ASK}[n]$ будет изменяться в диапазоне от 0 до 1. Объедините в шаблоне $\tilde{g}_{ASK}[n]$ с T_x с помощью функции *Build Waveform* для формирования модулирующего сигнала ("Baseband Signal"). Обратите внимание, что величина $1/T_x$ – это реальная частота IQ ("actual IQ rate"). Подключите также $\tilde{g}_{ASK}[n]$ ко входу "data" функции *Write Tx Data*.

6. В шаблоне предоставлена функция *FFT Power Spectrum for 1 Chan*, которая позволит наблюдать спектр передаваемого сигнала.

Вы завершили создание ASK-передатчика. Сохраните программу передатчика в файл, в имени которого содержатся буквы "ASKTx" и ваши инициалы (например, *ASKx_BAB.gvi*).

Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданную вами программу передатчика.

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.1 МГц (Примечание: сдвиг в 100 кГц относительно несущей частоты приемника выбран не случайно).

Частота IQ (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_x$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): TX1

Частота символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): None

Запустите передатчик. Используйте большую кнопку STOP на лицевой панели для отключения выходных разъемов передатчика.

3. После запуска передатчика наблюдайте спектр передаваемого сигнала. Вы должны ясно видеть несущую на нулевой частоте. Важны еще следующие две характеристики: ширина спектра (bandwidth) и скорость затухания спектра (rate of spectral rolloff). Используя опцию Capture Data или курсоры на графике Power Spectrum, измерьте ширину спектра переданного сигнала по первым нулям. Соотнесите ее со скоростью передачи символов $1/T$. Скорость затухания спектра является мерой помех, которые ваш сигнал создаст для сигналов с близкими частотами несущих. Распечатайте копию спектра для сравнения со спектром, который получите в пункте 4.
4. На практике прямоугольные импульсы используются редко из-за очень малой скорости затухания спектра. Измените фильтр, формирующий импульсы, на "Root Raised" для получения фильтра с характеристикой приподнятого косинуса (root-raised-cosine). Запустите программу и снова изучите спектр. Измерьте ширину спектра переданного сигнала по первым нулям. Распечатайте копию спектра и сравните его скорость затухания со скоростью затухания спектра, полученного вами при использовании прямоугольных импульсов.

Вопросы

1. В пункте 2 вы задали $1/T = 10000$ символов/с и $1/T_x = 200 \times 10^3$ отсчетов/с. Найдите соответствующее значение для количества отсчетов на символ L .
2. Определите отношение ширины спектра по первым нулям ASK-сигнала для (а) прямоугольных импульсов и (б) импульсов с косинусным сглаживанием.
3. Сравните скорость затухания спектра переданного сигнала для прямоугольных импульсов и импульсов с косинусным сглаживанием.

7.4 Часть 2: Приемник

Вводная информация

Создание ASK-приемника начинается так же, как и создание аналогового AM-приемника. Мы сдвинем несущие частоты передатчика и приемника на 100 кГц, чтобы возвращенный USRP-приемником сигнал был AM-сигналом с промежуточной несущей частотой 100 кГц. Полученный AM-сигнал пропускается через полосовой фильтр промежуточной частоты и демодулируется детектором огибающей. Детектор огибающей реализуется путем извлечения амплитуды сигнала с выхода полосового фильтра и последующей фильтрации с помощью ФНЧ 2 второго порядка. Если вы выполняли лабораторную работу 2, Амплитудная модуляция, эти операции должны быть вам знакомы.

Для систем цифровой связи ФНЧ должен быть таким, чтобы минимизировать влияние шума и межсимвольных помех, которые могут появиться при наложении отфильтрованных принятых импульсов. Лучший фильтр для устранения помех - так называемый согласованный (matched) фильтр. Согласованный фильтр имеет частотную характеристику, амплитуда которой совпадает с амплитудой частотной характеристики фильтра, формирующего импульсы в передатчике. То есть, если $g_{RX}[n]$ - импульсная характеристика фильтра приемника, то нужно, чтобы $|G_{RX}(e^{j\omega})| = |G_{TX}(e^{j\omega})|$. Используя коэффициенты *MT Generate Filter Coefficients* и в передатчике, и в приемнике, мы гарантируем, что в приемнике выбран подходящий фильтр для согласования с фильтром, формирующим импульсы в передатчике.

Для завершения создания цифрового приемника необходимо выполнить еще несколько операций после AM-демодулятора.

1. Синхронизация импульсов. С выхода AM-приемника получают аналоговый сигнал, отсчеты которого необходимо брать по одному на символ, то есть каждые T секунд. Из-за задержек при фильтрации и распространении сигнала, а также из-за искажений в канале связи, необходимо определить оптимальное время для взятия этих отсчетов. В папке *BasicUSRPLabs* предоставлена функция *PulseAlign(real)*, которая выравнивает сигнал так, чтобы отсчет с индексом o был правильным первым отсчетом.
2. Взятие отсчетов. Функция *Decimate* берет отсчеты выровненного сигнала, начиная с индекса o и далее каждые T секунд.

3. Детектирование. После дискретизации сигнала необходимо и определить, какое значение представляет каждый отсчет – символ 0 или символ 1.
4. Кодирование символов. Значения детектированных символов необходимо преобразовать в биты. Для ASK эта операция легко включается в процесс детектирования.

Подготовка к лабораторной работе

1. Шаблон для приемника подготовлен в файле *ASKRxTemplate.gvi*. Этот шаблон содержит шесть функций интерфейса с USRP.

Мы хотим, чтобы приемник захватывал два кадра данных при каждом запуске. Поскольку приемник начинает работу в произвольное время, это гарантирует обязательное наличие в захваченных данных одного полного кадра. Используя значения *message length* (длина сообщения) и *symbol rate* (скорость передачи символов), доступные на лицевой панели, а также выход "Actual IQ Rate" функции *Configure Signal*, рассчитайте число отсчетов в кадре для приемника. Затем удвойте это число и подайте его на вход "number of samples" (количество отсчетов) функции *Fetch Rx Data*.

Функция *Fetch Rx Data* возвращает кластер комплексных чисел двойной точности, в котором доступны компоненты "Y" и "dt". Пропустите комплексный массив "Y" через полосовой фильтр. Фильтры находятся в папке *ExternalFiles* → *Files Pane* на панели файлов (*Files Pane*). Используйте фильтр Чебышева (CDB) пятого порядка с верхней частотой среза 110 кГц и нижней частотой среза 90 кГц. Заданная по умолчанию неравномерность в полосе пропускания, равная 0.1 дБ, приемлема. На вход "sampling frequency" фильтра можно подать величину, обратную компоненте "dt" сигнала, возвращаемого функцией *Fetch Rx Data*. Обратите внимание, что значение dt совпадает со значением выхода "actual IQ rate" функции *Configure Signal*. Обозначим этот параметр $1/T_z$.

2. Извлеките действительную часть комплексного массива, получаемого на выходе "Filtered X" полосового фильтра Чебышева. Для получения огибающей возьмите абсолютное значение и пропустите результат через согласованный фильтр. Взятие абсолютного значения заменяет двухполупериодный выпрямитель. Для согласованного фильтра используйте коэффициенты *MT Generate Filter Coefficients*, так же, как вы это делали в передатчике. Задайте тип модуляции ASK и

рассчитайте количество согласованных отсчетов на символ ("matched samples per symbol") M из реальной частоты IQ ("actual IQ rate" ($1/T_z$)) и скорости передачи символов (symbol rate ($1/T$)), взятых из элементов управления на лицевой панели. Создайте на лицевой панели элемент управления для входа "pulse shaping filter" и установите его в исходное состояние "none". При такой настройке будет сгенерирован согласованный фильтр с прямоугольной импульсной характеристикой (это не означает отсутствие фильтра, как вы могли подумать). Подключите выход "matched filter coefficients" (коэффициенты согласованного фильтра) ко входу "Y" функции *Convolution*. Эта функция находится в субпалитре Analysis→Signal Processing→Signal Operation. Выход согласованного фильтра подключите в шаблоне к функции *Cluster Properties*. Функцию *Cluster Properties* подключите к графическому индикатору Baseband Output.

3. Выходной сигнал цифрового демодулятора удобно визуализировать в виде так называемой "глазковой диаграммы" (eye diagram). Глазковая диаграмма - это график выходного сигнала, где горизонтальная ось отмасштабирована на один или два символа, а последовательно следующие символы наложены друг на друга.

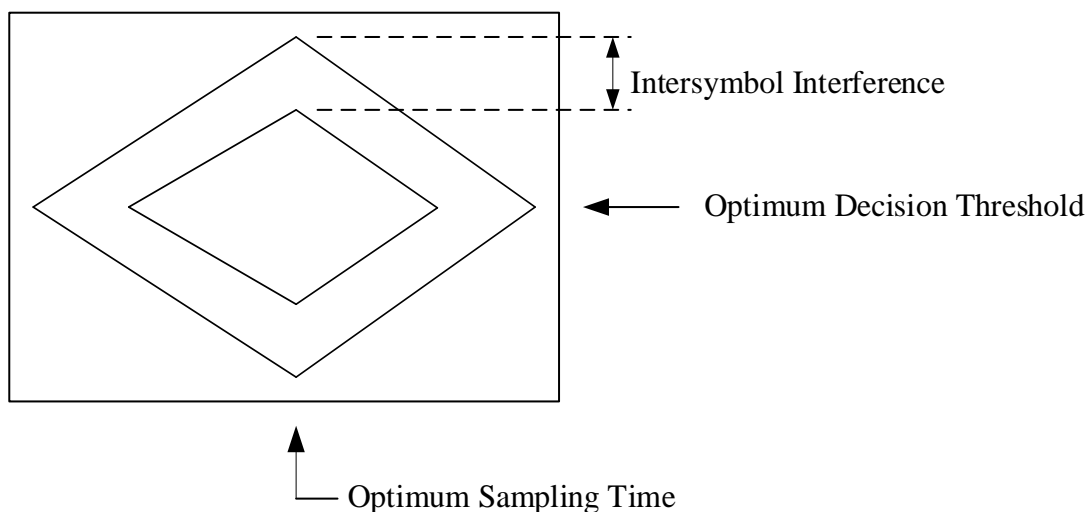


Рисунок 3. Стилизованная глазковая диаграмма

Intersymbol Interference – межсимвольная интерференция, Optimum Decision Threshold – оптимальный порог принятия решения, Optimum Sampling Time – оптимальное время выборки

На рисунке 3 показана стилизованная глазковая диаграмма, а также полезная информация, которую можно узнать из такой диаграммы.

Функция *MT Format Eye Diagram* из комплекта Modulation Toolkit есть в шаблоне приемника. Подключите выходной сигнал ко входу "waveform". Возьмите входное значение "symbol rate (Hz)"

из элемента управления на лицевой панели. Задайте параметр "eye length" (длина диаграммы) равным 2.

4. Поместите на блок-диаграмму функцию *PulseAlign(real)* и подключите выходной сигнал ко входу "input waveform" (входной сигнал), а рассчитанное значение M отсчетов/символ ко входу "receiver sampling factor" (коэффициент дискретизации приемника).

Как только сигнал выровнен, можно брать отсчеты. Возьмите функцию *Decimate (single shot)* из субпалитры Analysis→Signal Processing→Signal Operation. Коэффициент прореживания ("decimating factor") равен M .

5. Чтобы определить, похож ли полученный отсчет на 0 или 1, необходимо сравнить его с пороговым значением. Используйте функцию *Mean DBL* из папки ExternalFiles для вычисления порогового значения. Результат сравнения - цифровой выход приемника. На выходе функции сравнения будет булевский массив. Вы можете преобразовать его в массив целых числе, используя функцию *Boolean To Integer*.

Вы завершили создание ASK-приемника. Сохраните программу приемника в файл, в имени которого содержатся буквы "ASKRx" и ваши инициалы (например, *ASKRx_BAB.gvi*).

Порядок выполнения лабораторной работы

1. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

Частота IQ (IQ Rate): 1 МГц

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): RX2

Скорость передачи символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): None

2. Запустите сначала передатчик, затем приемник. После получения приемником блока данных можно остановить передатчик.

3. Используйте масштабирование по горизонтали из палитры графического индикатора Baseband Output, чтобы растянуть демодулированный сигнал и можно было рассмотреть отдельные импульсы. В идеале прямоугольные импульсы, пропущенные через согласованный фильтр приемника, должны преобразовываться в треугольные выходные импульсы. Обратите внимание, имеют ли демодулированные импульсы ожидаемую форму.

4. Рассмотрите глазковую диаграмму. Обратите внимание на оптимальное время выборки и наличие межсимвольных помех.

5. Выберите в приемнике и передатчике фильтр с косинусным сглаживанием "Root Raised" с помощью элементов управления "pulse shaping filter". Такая настройка и фильтра приемника, и фильтра передатчика гарантирует, что фильтры будут согласованными и обеспечат оптимальную эффективность в присутствии помех. Каскад из двух фильтров с косинусным сглаживанием выдает импульс с косинусным сглаживанием на выходе фильтра приемника. Импульсы с косинусным сглаживанием призваны минимизировать или, в идеале, устранять межсимвольные помехи.

Запустите сначала передатчик, затем приемник. После получения приемником блока данных можно остановить приемник. Изучите выходной сигнал и глазковую диаграмму.

Прокомментируйте изменения в глазковой диаграмме. Уменьшились ли межсимвольные помехи?

6. Для наблюдения эффекта синхронизации импульсов подайте на вход функции MT Format Eye Diagram сигнал с выхода "aligned waveform" (выровненный сигнал) функции PulseAlign(real). Снова запустите передатчик и приемник. Изучите глазковую диаграмму. Изменилось ли оптимальное время выборки?

Вопросы

1. Напишите формулу связи частоты дискретизации IQ, скорости передачи символов $1/T$ и количества отсчетов на символ M . Определите значение M при частоте дискретизации IQ в 1 МГц и скорости передачи символов 10 000 символов/с.
2. Обратите внимание, что частота дискретизации IQ в приемнике отличается от таковой в передатчике. Использование более высокой частоты дискретизации IQ требует более быстрой цифровой обработки. В чем преимущество использования более высокой частоты дискретизации IQ в приемнике? (Подсказка: это как-то связано с действием функции *PulseAlign(real)* и значением M).
3. Сравните формы прямоугольных импульсов и импульсов с косинусным сглаживанием по глазковой диаграмме. Какие признаки межсимвольных помех вы видите в каждом случае?
4. Используя импульсы с косинусным сглаживанием и фильтрацию в приемнике, изучите глазковую диаграмму, когда вход функции *MT Format Eye Diagram* связан с выходом "baseband output" и когда он связан с выходом "aligned waveform" функции *PulseAlign(real)*. В чем назначение функции *PulseAlign(real)*?
5. Передатчик запрограммирован на периодическую генерацию одного и того же кадра из 1000 символов. Приемник захватывает один блок из 2000 символов при каждом запуске. Можете ли вы, изучив график выходного сигнала приемника, определить, где заканчивается и начинается заново последовательность символов? Синхронизация кадров в приемнике – необходимая составляющая цифровой системы связи. Мы изучим синхронизацию кадров в следующей части этой лабораторной работы.

7.4 Часть 3: Выравнивание принятых битов

На данный момент у вас есть работающие передатчик и приемник, но сложно сказать, правильно ли они работают как система, пока вы не сравните принятые биты с переданными и не убедитесь в их совпадении. Чтобы провести такое сравнение необходимо, чтобы приемник распознавал начало переданной последовательности. Функция *AddFrameHeader(real)* из папки *BasicUSRPLabs* вставляет особую последовательность из 26 бит в начало передачи. В приемнике функция *FrameSync(real)* из той

же папки ищет эту последовательность и удаляет все биты, принятые до этого заголовка кадра. Функция *FrameSync(real)* удаляет также и заголовок кадра. Таким образом, последовательность битов на выходе приемника должна совпадать с последовательностью, переданной передатчиком.

1. Добавьте в передатчик функцию *AddFrameHeader(real)*. Поместите ее после кодирования символов, но до преобразования символов в комплексные числа
2. Добавьте в приемник функцию *FrameSync(real)*. Поместите ее сразу за функцией *Decimate*. Подключите выход функции *Decimate* ко входу "Sampled Input" функции *FrameSync(real)*. Остальные входы *FrameSync(real)* оставьте неподключенными. Подключите выход "Aligned Samples" функции *FrameSync(real)* к функции сравнения с порогом и к функции *Mean*, описанной в разделе "Приемник. Подготовка к лабораторной работе", пункт 5.

Подключите массив битов с выхода функции сравнения с порогом ко входу "array" функции *Array Subset*. Задайте на входе "index" нуль, а на входе "length" длину, определенную элементом управления "message length". Подключите к индикатору "Output bits" на лицевой панели приемника выход функции *Array Subset*.

Обратите внимание, что в этой лабораторной работе не используются выходы "Output Signal" и "max index" функции *FrameSync(real)*.

3. Запустите передатчик; затем запустите приемник; затем остановите передатчик. Сравните первую дюжину (или близкое количество) принятых битов с соответствующими переданными битами.
4. Измерение вероятности ошибок (bit error rate, BER) можно автоматизировать функцией *MT Calculate BER* из комплекта *Modulation Toolkit (Analysis→Communications→Digital→Measurements)*. На ленте *Configure* выберите "PN Fibonacci." Задайте параметр "BER trigger threshold" (порог обнаружения ошибки) равным 0.4. Подключите индикаторы к выходам "BER" и "trigger found?" При запуске программы на выходе "trigger found?" будет логическая единица каждый раз, как измеренный коэффициент ошибки меньше заданного порога. Запустите передатчик и приемник. Измеренный коэффициент ошибок должен быть равен нулю, если вы не добавляли помехи.

Вопросы

1. В пункте 3 совпадают ли принятые биты с переданными битами?

2. Чему было бы равно измеренное значение вероятности ошибок BER, если бы с выравниванием битов приемника было что-то неправильно? Обратите внимание, что когда измеренное значение вероятности ошибок выше порогового значения, на выходе "trigger found?" будет логический ноль, и все полученные значения BER не будут иметь смысла.

7.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также документацию для любых созданных вами функций. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Выполнение лабораторной работы

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также все созданные вами функции. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше.

Повторно предоставьте документацию для любых функций, которые вы модифицировали во время лабораторной работы.

Ответьте на все вопросы в каждом разделе "Вопросы".

Частотная манипуляция

Предварительные требования: Лабораторная работа 6 - Частотная модуляция, Лабораторная работа 7 - Амплитудная манипуляция.

8.1 Цель

При частотной манипуляции (frequency-shift keying, FSK), 1 представляется тоном определенной частоты, называемой обычно частотой посылки (mark frequency), а 0 - тоном другой частоты, называемой частотой паузы (space frequency). Отчасти своей популярностью FSK обязана тому, что тон передается всегда, даже когда источник генерирует длинную последовательность нулей. Это упрощает для приемника определение того, работает ли передатчик в режиме ожидания или завершил передачу. Кроме того, при FSK передаваемый сигнал всегда имеет постоянную амплитуду. Это позволяет использовать для передачи очень эффективный нелинейный усилитель мощности, что очень важно, когда передатчик питается от батарей.

FSK является цифровой версией частотной модуляции. Так же, как ASK-система строится на основе АМ-передатчика и АМ-приемника, FSK-система строится на основе FM-передатчика и FM-приемника. Такие дополнительные свойства, как кодирование символов, формирование импульсов, согласованная фильтрация, сравнение с порогом и синхронизация импульсов, применимы к FSK аналогично ASK.

8.2 Вводная информация

Передатчик

Полоса частот передаваемого сигнала - важное свойство любого метода модуляции. У полосы частот есть два компонента, "главный лепесток" (main lobe), который определяет ширину канала, необходимую для передаваемого сигнала, а также полоса пропускания фильтра во входном каскаде приемника. Скорость затухания спектра является определяющим фактором искажения, которые создаст ваш сигнал для сигналов в соседних каналах. Она определяет также, насколько близкими могут быть частоты аналогично модулированных сигналов. Полосу частот FSK-сигнала весьма трудно рассчитать аналитически. Дж. Р. Карсон в 1920 году предложил практический способ оценки полосы частот.

$$B_{FSK} \cong 2(\Delta f + B), \quad (1)$$

где B_{FSK} - полоса частот FSK-сигнала, Δf - пиковая девиация частоты сигнала, и B - полоса частот сообщения. Правило Карсона просто в применении, однако при работе с FSK-сигналами оно часто дает завышенные результаты.

Скорость затухания спектра FSK-сигнала во многом определяется гладкостью сигнала в моменты изменения частоты от посылки к паузе и наоборот. Если в эти моменты FSK-сигнал перестает быть непрерывным, скорость затухания спектра будет пропорциональна $1/f^2$, что сравнимо с ASK при использовании прямоугольных импульсов. Для FSK с *непрерывной фазой* скорость затухания спектра пропорциональна $1/f^4$. Вследствие этого в большинстве современных приложений используется именно вариант с непрерывной фазой. Еще более высоких значений скорости затухания спектра можно достичь сглаживанием сигнала сообщения перед подачей на FSK-передатчик.

Генерация аналогового FM-сигнала подробно рассматривалась в разделе "Вводная информация" лабораторной работы 6, *Частотная модуляция*. Настоятельно рекомендуется освежить в памяти этот материал. Вспомните, что для сообщения $m(t)$ мгновенная частота FM-сигнала определяется как

$$f(t) = f_c + k_f m(t), \quad (2)$$

где f_c - несущая частота и k_f - частотная чувствительность. Пиковая девиация частоты FSK-сигнала

$$\Delta f = k_f m_p, \quad (3)$$

где m_p - пиковое значение сообщения. Для формирования FSK-сигнала с непрерывной фазой двоичный сигнал сообщения $m(t)$ представлен в биполярном формате без возврата к нулю (NRZ), как показано на рисунке 1.

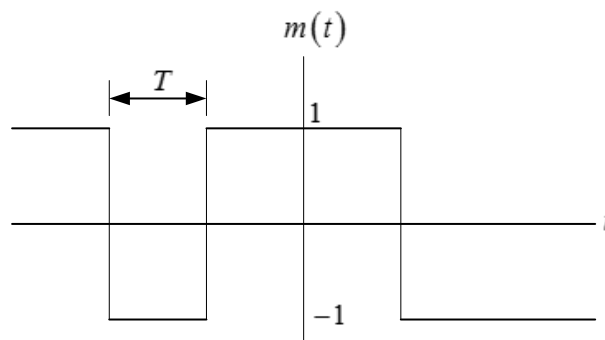


Рисунок 1. Двоичный сигнал сообщения

Длительность импульса T на рисунке 1 будем называть "длительностью символа", как и в предыдущей лабораторной работе, посвященной ASK. Пиковое значение сообщения составляет $m_p = 1$. Учитывая биполярный формат сообщения, из уравнения (2) мы видим, что частота посылки равна $f_c + \Delta f$, а частота паузы - $f_c - \Delta f$. Обращаясь к уравнению (1), важно отметить, что ширина полосы частот FSK-сигнала не равна $2\Delta f$, но включает ширину полосы частот сообщения. Поэтому нельзя сделать полосу частот произвольно малой, уменьшая пиковую девиацию частоты.

Как показано в лабораторной работе 6, сигнал, который должен быть отправлен в USRP для генерации FM-сигнала, равен

$$\tilde{g}(t) = A_c e^{j2\pi\Delta f \int_0^t m(\alpha) d\alpha}, \quad (4)$$

где A_c - амплитуда несущей. USRP сформирует FM-сигнал, имеющий вид

$$g(t) = \text{Re}[\tilde{g}(t) e^{j2\pi f_c t}] = A_c \cos\left[2\pi f_c t + 2\pi\Delta f \int_0^t m(\alpha) d\alpha\right]. \quad (5)$$

Следует отметить, что фаза FM-сигнала, описанного уравнением (5), пропорциональна $\int_0^t m(\alpha) d\alpha$.

Для сообщения в биполярном формате без возврата к нулю, показанному на рисунке 1, этот интеграл дает непрерывную по времени функцию.

Операции, необходимые для формирования FSK-сигнала после частотной модуляции, должны быть вам знакомы, если вы выполнили лабораторную работу, посвященную ASK.

1. Кодирование символов. Входные данные появляются в виде потока битов. Вспомните, что функция *MT Generate Bits* формирует на выходе массив, содержащий числа 0 и 1 (8-разрядных целых чисел). На стадии кодирования символов биты заменяются числовыми значениями. Для FSK мы представим двоичную 1 действительным числом двойной точности 1, а двоичный 0 - действительным числом двойной точности -1. В таблице 1 показано кодирование символов FSK.

Таблица 1. Кодирование символов FSK

Значение бита	Символ
0	-1
1	1

2. Передискретизация. Первый шаг в замене символов импульсами - помещение $L - 1$ нулей после каждого символа. Это даст интервал дискретизации

$$T_x = \frac{T}{L}, \quad (6)$$

или частоту дискретизации

$$\frac{1}{T_x} = L \frac{1}{T}. \quad (7)$$

Более высокий коэффициент передискретизации L упростит цифро-аналоговое преобразование в передатчике, но потребует более быстрой цифровой обработки. В данной лабораторной работе мы используем $L = 40$.

3. Формирование импульсов. Если передискретизированный сигнал – прямоугольный импульс единичной амплитуды и длительностью L отсчетов подать на фильтр с импульсной характеристикой $g_{TX}[n]$, на выходе фильтра каждый символ будет представлен прямоугольным импульсом. Операции по пунктам с 1 по 3 преобразуют входной поток битов в сигнал сообщения в биполярном формате без возврата к нулю, показанный на рисунке 1. Обратите внимание, что помимо прямоугольной можно использовать другие формы импульсов, просто изменив вид импульсной характеристики $g_{TX}[n]$.
4. Модуляция. Как только мы получили сигнал сообщения $m(t)$, применяется уравнение (4) для формирования сигнала, отправляемого в передатчик USRP.

Приемник

Разработка FSK-приемника начинается так же, как и FM-демодулятора. Сначала из комплексного сигнала, полученного из USRP, извлекается фаза. Затем фаза дифференцируется и фильтруется. Более подробно эти операции описаны в разделе "Вводная информация" лабораторной работы 6. Для завершения создания цифрового приемника после FM-демодуляции необходимо добавить еще несколько операций.

1. Синхронизация импульсов. Отсчет выходного аналогового сигнала FM-приемника необходимо снимать во время каждого символа, то есть каждые T секунд. Из-за задержек фильтрации и распространения сигнала, а также из-за искажений в канале связи нужно определить оптимальное

время для взятия этих отсчетов. В папке *BasicUSRPLabs* предоставлена функция *PulseAlign(real)*, которая выравнивает сигнал так, чтобы отсчет с индексом 0 был первым правильным отсчетом.

2. Дискретизация. Функция *Decimate* дискретизирует выровненный сигнал, начиная с индекса 0 и далее каждые T секунд.
3. Детектирование. После дискретизации сигнала необходимо проверить каждый отсчет, чтобы определить, какое он представляет значение символа - 0 или 1 .
4. Кодирование символов. Значения детектированных символов необходимо преобразовать в биты. Для FSK эта операция просто может быть включена в операцию детектирования.

8.3 Подготовка к лабораторной работе

Передатчик

1. Создайте программу для генерации FSK-сигнала с непрерывной фазой при помощи USRP. Шаблон для передатчика подготовлен в файле *FSKTxTemplate.gvi*. Этот шаблон содержит четыре функции интерфейса с USRP, а также функцию *MT Generate Bits* из комплекта *Modulation*. Функция *MT Generate Bits* сгенерирует псевдослучайную последовательность битов, которая может служить последовательностью данных для тестирования вашей FSK-системы. Обратите внимание, что по умолчанию *MT Generate Bits* будет выдавать одну и ту же последовательность при каждом запуске программы. Это полезно для отладки, однако если вы захотите, чтобы каждый раз генерировалась новая последовательность, подключите функцию *random number* ко входу "seed in". В последующих пунктах подробно рассматривается, как создать искомый передатчик.
2. Сначала выполните кодирование символов, как показано в таблице 1.
3. Выполните передискретизацию массива символов функцией *Upsample* из субпалитры *Analysis*→*Signal Processing*→*Signal Operation*. В этой лабораторной работе вам даются управляющие входы, чтобы задать скорость передачи символов $1/T$ и частоту IQ $1/T_x$. Задайте скорость передачи символов равной $10\,000$ символов/с и частоту IQ 400×10^3 отсчетов/с. Используйте эти два входа для вычисления коэффициента передискретизации L .

4. Используйте функцию *MT Generate Filter Coefficients* из комплекта Modulation Toolkit для генерации фильтра, формирующего импульсы. (Функцию *MT Generate Filter Coefficients* можно найти в субпалитре Analysis→Communications→Digital→Utilities). Задайте тип модуляции FSK, а вход "samples per symbol" (количество отсчетов на символ) равным рассчитанному значению L . Создайте на лицевой панели элемент управления для входа "pulse shaping filter" (фильтр, формирующий импульсы) и установите его в исходное состояние "none". Как и в лабораторной работе про ASK, при такой настройке будут сгенерированы прямоугольные импульсы. (Помните: "none" не означает отсутствия фильтра!) Подключите выход "pulse shaping filter coefficients" ко входу "Y" функции *Convolution*. Эта функция находится в субпалитре Analysis→Signal Processing→Signal Operation. Подключите выходной передискретизированный сигнал ко входу "X" функции *Convolution*.

5. Нормализуйте амплитуду отфильтрованного сигнала сообщения к максимальному абсолютному значению, равному единице. Это будет важно позднее, когда мы рассмотрим альтернативные формы импульсов. Функция *Quick Scale 1D* из папки ExternalFiles идеально подходит для выполнения этой задачи. (Вы можете использовать выход $Y[i] = X[i]/\text{Max}|X|$ функции *Quick Scale 1D*, поскольку входные данные "X" в этой работе только действительные). Подключите выход функции *Quick Scale* ко входу функции *Build Waveform*, выход которой соединен с графическим индикатором Message Signal шаблона.

6. Теперь реализуйте уравнение (4). Чтобы проинтегрировать, возьмите IIR-фильтр из палитры Analysis→Signal Processing→Filters→IIR Filtering. Используйте массив прямых коэффициентов ("forward coefficients"), равный $[1]$ и массив обратных коэффициентов ("reverse coefficients"), равный $[1 \quad -1]$. Умножьте проинтегрированное сообщение на $2\pi\Delta f$ и на dt . Пиковая девиация частоты Δf определяется элементом управления на лицевой панели; задайте ее равной 5000 Гц. Интервал дискретизации dt является обратной величиной реальной частоты IQ ("actual IQ rate") (dt и T_x описывают одну и ту же величину). Используйте функцию *polar to complex* для создания комплексного модулирующего сигнала. Пусть амплитуда несущей A_c будет равна 1. Комплексный модулирующий сигнал подключается ко входу "data" функции *Write Tx Data*.

7. Чтобы пронаблюдать спектр передаваемого сигнала, подключите комплексный модулирующий сигнал к функции *Build Waveform*, соединенной с графическим индикатором Baseband Power Spectrum шаблона.

Вы завершили создание FSK-передатчика. Сохраните программу передатчика в файл, в имени которого содержатся буквы "FSKTx" и ваши инициалы (например, *FSKx_BAB.gvi*).

Приемник

1. Создайте программу для реализации FSK-приемника при помощи USRP. Шаблон для приемника подготовлен в файле *FSKRxTemplate.gvi*. Этот шаблон содержит шесть функций интерфейса с USRP.

Рассчитайте количество отсчетов для функции *Fetch Rx Data* на основании длины сообщения и скорости передачи символов, указанных на лицевой панели. Удвойте количество отсчетов в кадре, чтобы приемник захватил два кадра данных. Поскольку приемник начинает работу в произвольное время, это обеспечит обязательное наличие в принятых данных одного полного кадра.

Передайте комплексный массив с выхода функции *Fetch Rx Data* на вход функции *Complex to Polar* для извлечения фазы. Разверните фазу функцией *Unwrap Phase* из субпанели Analysis→Signal Processing→Signal Operation для того, чтобы перед дифференцированием удалить скачки фазы на 2π . Для дифференцирования фазы используйте функцию *FIR Filter* с коэффициентами, взятыми из массива $[1 \quad -1]$. При дифференцировании на месте разрывов фазы возникнут большие выбросы. Это произойдет в начале сигнала и каждый раз, когда передатчик заново начинает передавать последовательность. Для сглаживания выбросов используйте медианный фильтр (*Median Filter*) из папки ExternalFiles. Задайте вход "left rank" равным 5 и оставьте вход "right rank" не подключенным.

2. Для реализации согласованного фильтра в приемнике используйте *MT Generate Filter Coefficients*, как вы это делали в передатчике. Задайте тип модуляции FSK и рассчитайте количество согласованных отсчетов на символ ("matched samples per symbol") M из реальной частоты IQ ("actual IQ rate" ($1/T_c$)) и скорости передачи символов (symbol rate ($1/T$)), полученных из элементов управления на лицевой панели. Создайте на лицевой панели элемент управления для входа "pulse shaping filter" и установите его в исходное состояние "none." (Помните: при такой настройке будет создан согласованный фильтр с прямоугольной импульсной характеристикой). Подключите выход "matched filter coefficients" ко входу "Y" функции *Convolution*. Выход согласованного фильтра

подключите в шаблоне к функции *Cluster Properties*. Функция *Cluster Properties* передает данные на графический индикатор Baseband Output.

3. Поместите на блок-диаграмму функцию *PulseAlign(real)* и подключите выходной сигнал фильтра ко входу "input waveform", а рассчитанное значение M отсчетов/символ ко входу "receiver sampling factor".

Как только модулирующий сигнал выровнен, можно брать с него отсчеты. Возьмите функцию *Decimate (single shot)* из субпалитры Analysis→Signal Processing→Signal Operation. Коэффициент прореживания ("decimating factor") равен M .

4. Функция *MT Format Eye Diagram* из комплекта Modulation Toolkit имеется в шаблоне приемника. Подключите демодулированный сигнал ко входу "waveform" глазковой диаграммы. Возьмите входное значение "symbol rate (Hz)" из элемента управления на лицевой панели. Задайте параметр "eye length"(размер диаграммы) равным 2.
5. Чтобы определить, представляет ли полученный отсчет 0 или 1, необходимо сравнить его с пороговым значением. Поскольку сообщение $m(t)$ - биполярный сигнал, в качестве порогового значения можно взять 0. Результат сравнения - цифровой выход приемника.

Вы завершили создание FSK-приемника. Сохраните программу приемника в файл, в имени которого содержатся буквы "FSKRx" и ваши инициалы (например, *FSKRx_BAB.gvi*).

Вопросы

1. В пункте 3 создания передатчика вы задали $1/T = 10000$ символов/с и $1/T_x = 400 \times 10^3$ отсчетов/с. Найдите соответствующее значение количества отсчетов на символ L .
2. Объясните, как IIR-фильтр с массивом прямых коэффициентов $[1]$ и обратных коэффициентов $[1 \quad -1]$ реализует интегратор.

3. Объясните, как FIR-фильтр с массивом коэффициентов $[1 \quad -1]$ реализует дифференцирующее звено, использованное в разделе "Приемник", пункт 1.
4. Объясните, что делает медианный фильтр. Для получения информации обратитесь к онлайн-справке LabVIEW 2014 по функции *Median Filter*.
5. В разделе "Приемник", пункт 2, значение реальной частоты IQ $1/T_z$ могло отличаться от частоты $1/T_x$, использованной в передатчике. (Обратите внимание, что скорость передачи символов $1/T$ должна быть одинакова и в приемнике, и в передатчике). Значение частоты IQ приемника определяет коэффициент выборки приемника M . В чем преимущество использования более высокого значения M ? В чем преимущество использования более низкого значения M ?
6. В FM-приемнике используется дифференциатор для отмены интегрирования в уравнении (4). Какое влияние окажет дифференциатор на любые помехи, которые могут поступать с сигналом? Для ответа на этот вопрос подумайте, что делает дифференциатор в частотной области. Что будет, если убрать фильтр, следующий за дифференциатором?

8.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2 USRP. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.
2. Убедитесь, что для передатчика заданы следующие настройки:
 - Несущая частота (Carrier Frequency): 915.0 МГц
 - Частота IQ (IQ Rate): 400 кГц Примечание: Это задает значение $1/T_x$.
 - Коэффициент усиления (Gain): 0 дБ
 - Активная антенна (Active Antenna): TX1
 - Частота символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): None

Пиковая девиация частоты (Peak Frequency Deviation): 5000 Гц

Запустите передатчик. Используйте большую кнопку STOP на лицевой панели для отключения разъемов передатчика.

3. После запуска передатчика наблюдайте спектр передаваемого сигнала. Важны следующие две характеристики: полоса частот и скорость затухания спектра. Измерьте ширину полосы частот переданного сигнала по первым нулям. Сравните измеренную ширину полосы частот с вычисленной по правилу Карсона из уравнения (1). Скорость затухания спектра является мерой помех, которые вызовет ваш сигнал для сигналов, использующих близкие частоты несущих. Установите на вертикальной шкале графического индикатора спектра диапазон от -100дБ до 0 дБ и распечатайте копию спектра для сравнения с пунктом 4.
4. Скорость затухания спектра FSK-сигнала определяется главным образом гладкостью передаваемого сигнала. У FSK с непрерывной фазой нет разрывов при изменении частоты передаваемого сигнала, но могут быть "углы", где резко изменяется наклон передаваемого сигнала. Для сглаживания этих углов сообщение $m(t)$ может быть пропущено через фильтр до передачи на FM-модулятор. Измените фильтр, формирующий импульсы, на гауссовский ("Gaussian") для создания очень плавных изменений фронтов импульсов. Запустите передатчик и наблюдайте сглаженный сигнал сообщения и его спектр мощности. Измерьте полосу частот передаваемого сигнала на 60 дБ ниже пика спектра. Установите на вертикальной шкале графического индикатора спектра диапазон от -100дБ до 0 дБ и распечатайте копию спектра для сравнения с пунктом 4. Сравните скорость затухания этого спектра и спектра, который был получен вами при использовании прямоугольных импульсов.
5. Измените фильтр, формирующий импульсы, обратно на "none". Задайте пиковую девиацию частоты равной 20000 Гц, 10000 Гц, 5000 Гц и 2500 Гц. В каждом случае измерьте ширину полосы частот передаваемого сигнала по первым нулям и сравните со значениями, вычисленными по правилу Карсона.

Для заданной скорости передачи символов 2500 Гц - минимальная девиация частоты. Ниже нее пострадает эффективность, поскольку сигналы посылки и паузы будут отличаться слишком мало.

6. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

Частота IQ (IQ Rate): 400 кГц Примечание: Это задает значение $1/T_z$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): RX2

Скорость передачи символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): None

7. Запустите сначала передатчик, затем приемник. После получения приемником блока данных можно остановить передатчик.

8. Используйте масштабирование по горизонтали из палитры графического индикатора Baseband Output, чтобы растянуть демодулированные сигналы и чтобы можно было рассмотреть отдельные импульсы. В идеале прямоугольные импульсы, пропущенные через согласованный фильтр приемника, должны преобразовываться в треугольные выходные импульсы. Обратите внимание, имеют ли демодулированные импульсы ожидаемую форму. (Вы можете также щелкнуть правой кнопкой мыши по графику и выбрать Capture Data, после чего проанализировать данные на панели Data).

9. Изучите глазковую диаграмму. Обратите внимание на оптимальное время выборки и межсимвольные помехи. Чтобы увидеть эффект от синхронизации импульсов подайте на вход функции *MT Format Eye Diagram* сигнал с выхода "aligned waveform" функции *PulseAlign(real)*. Запустите передатчик и приемник. Наблюдайте глазковую диаграмму. Каково теперь оптимальное время выборки?

10. Задайте пиковую девиацию частоты передатчика равной 20000 Гц, 10000 Гц, 5000 Гц и 2500 Гц. Для каждого случая запустите сначала передатчик, затем приемник. Запишите пиковые значения полосы частот выходного сигнала.

11. Задайте пиковую девиацию частоты равной 5000 Гц. Установите элемент управления "pulse shaping filter" в приемнике и передатчике в состояние "Gaussian". При такой настройке фильтр приемника - просто кусок проводника, то есть фильтр отсутствует. Запустите сначала передатчик, затем приемник. После получения приемником блока данных можно остановить передатчик. Изучите выходной сигнал и глазковую диаграмму.

Вопросы

1. При пиковой девиации частоты, равной 5000 Гц, сравните скорость затухания спектра прямоугольных импульсов и импульсов с фильтрацией по Гауссу. Кроме того, изучив график переданного сигнала сообщения, можете ли вы найти признаки межсимвольных помех при использовании фильтра Гаусса?
2. Для каждого значения пиковой девиации частоты, перечисленных в пункте 5, сравните ширину полосы частот переданного сигнала по первым нулям с шириной полосы частот, полученной по правилу Карсона из уравнения (1). Для каждого случая рассчитайте разницу в процентах. Для каких значений пиковой девиации частоты правило Карсона работает более точно – для больших или маленьких?
3. Изучите глазковую диаграмму на лицевой панели приемника. Сравните на ней выровненный и невыровненный выходной сигнал. Опишите, что делает функция *PulseAlign(real)*.
4. Создайте график зависимости амплитуды выходного сигнала приемника от пиковой девиации частоты. Какова связь между этими двумя величинами?
5. Как описано в лабораторной работе 7, Амплитудная манипуляция, на глазковой диаграмме легко различить местоположение оптимального порога принятия решения. В данной лабораторной работе вам было дано указание установить этот порог равным нулю. Запустите передатчик и приемник несколько раз с разными значениями пиковой девиации частоты и разными параметрами фильтра ("none" и "Gaussian"). Наблюдайте глазковую диаграмму и прокомментируйте целесообразность использования нуля в качестве порога принятия решения.

Дополнительно

Чтобы убедиться в правильности принятой последовательности битов, вы можете добавить в передатчик функцию *AddFrameHeader(real)*, а в приемник функции *FrameSync(real)* и *MT Calculate BER*. Следуйте инструкциям из лабораторной работы 7, Амплитудная манипуляция. Функции *AddFrameHeader(real)* и *FrameSync(real)* находятся в папке *BasicUSRPLabs*.

8.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также документацию для любых созданных вами функций. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Выполнение лабораторной работы

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также все созданные вами функции. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше.

Повторно предоставьте документацию для любых функций, которые вы модифицировали во время лабораторной работы.

Ответьте на все вопросы из каждого раздела "Вопросы".

Двоичная фазовая манипуляция (BPSK)

Предварительные требования: Лабораторная работа 5 - Частотная модуляция, лабораторная работа 7 - Амплитудная манипуляция.

9.1 Цель

В фазовой манипуляции (phase-shift keying, PSK), информация кодируется фазой передаваемой несущей, а не амплитудой и не частотой, как в ASK, в FSK. В двоичной фазовой манипуляции (BPSK) существуют два значения фазы, 0° и 180° , это означает, что для представления одного двоичного значения используется немодифицированная несущая, а для передачи другого она инвертируется.

BPSK является цифровой версией аналоговой модуляции с двумя боковыми полосами и подавлением несущей (DSB-SC). Мы увидим, что в основе приемника и передатчика BPSK лежат приемник и передатчик DSB-SC. Такие дополнительные свойства, как кодирование символов, формирование импульсов, согласованная фильтрация, сравнение с порогом, синхронизация импульсов и кадров, необходимые для ASK и FSK, требуются и в PSK-системе. Как и в DSB-SC, отсутствие передаваемой несущей приводит к снижению мощности, необходимой для обеспечения заданной производительности. Среди двоичных систем BPSK обеспечивает наименьшую среднюю мощность, необходимую для получения заданной вероятности ошибок передачи. Как мы увидим в последующих лабораторных работах, BPSK легко расширить до квадратурной фазовой манипуляции (QPSK) и квадратурной амплитудной модуляции (QAM). Эти расширения позволяют передавать несколько битов за импульс, что очень эффективно для передачи данных на высоких скоростях по каналу с ограниченной полосой пропускания.

Демодуляция BPSK-сигнала требует синхронизации фазы принятого сигнала. Синхронизация фазы также рассматривалась в лабораторной работе, посвященной DSB-SC. Мы увидим, что сложностей синхронизации фазы можно избежать, немного пожертвовав производительностью, если использовать дифференциальное кодирование передаваемых данных. Дифференциальная двоичная фазовая манипуляция (DPSK) - надежный и эффективный метод модуляции, широко используемый на практике.

9.2 Вводная информация

Генерация и детектирование аналогового DSB-SC-сигнала подробно рассматривалась в разделе "Вводная информация" лабораторной работы 7, *Модуляция с двумя боковыми полосами и подавлением несущей*. Настоятельно рекомендуется освежить в памяти этот материал.

Передатчик

Сигнал, подвергнутый двоичной фазовой манипуляции, является последовательностью импульсов вида

$$Ag_{TX}(t)\cos(2\pi f_c t + \theta). \quad (1)$$

В уравнении (1) A - константа, задающая передаваемый уровень мощности, $g_{TX}(t)$ - фиксированная форма импульса, f_c - несущая частота, а θ принимает значение или 0° , или 180° для передачи требуемой информации. Заметим, что мы можем записать уравнение (1) как

$$\pm Ag_{TX}(t)\cos(2\pi f_c t), \quad (2)$$

где знак плюс соответствует $\theta = 0^\circ$, а знак минус - $\theta = 180^\circ$. Мы предположим, как и в предыдущих лабораторных работах, посвященных цифровым системам связи, что новый импульс передается каждые T секунд, так что скорость передачи символов равна $1/T$ символов/с. Для двоичных схем, подобных BPSK, скорость передачи битов равна скорости передачи символов. Поскольку импульс $g_{TX}(t)$ не переносит информацию, его форму можно выбирать для удовлетворения другим критериям. Как и с ASK, нам требуется форма импульса, обеспечивающая высокую скорость затухания спектра и минимизирующая межсимвольные помехи. Мы используем импульсы с косинусным сглаживанием, как в лабораторной работе, посвященной ASK.

Операции, необходимые для формирования BPSK-сигнала должны быть вам знакомы, если вы выполнили лабораторную работу, посвященную ASK.

1. Кодирование символов. Входные данные появляются в виде потока битов. Вспомните, что функция *MT Generate Bits* формирует массив, содержащий числа 0 и 1. На стадии кодирования символов биты заменяются числовыми значениями. Для BPSK мы представим двоичную 1 комплексным числом $1 + j0$ двойной точности, а двоичный 0 - комплексным числом $-1 + j0$ двойной точности. Обратите внимание, что мы представляем биты комплексными числами, несмотря на то, что мнимая часть равна нулю. Это потому, что USRP требует комплексных входов,

а также потому, что в следующей работе мы используем мнимую часть для передачи дополнительной информации. В таблице 1 показано кодирование символов BPSK.

Таблица 1: Кодирование символов BPSK

Значение бита	Символ
0	$-1 + j0$
1	$1 + j0$

2. Передискретизация. Первый шаг в замене символов импульсами - помещение $L - 1$ нулей после каждого символа. Это даст интервал дискретизации

$$T_x = \frac{T}{L}, \quad (3)$$

или частоту дискретизации

$$\frac{1}{T_x} = L \frac{1}{T}. \quad (4)$$

Более высокий коэффициент передискретизации L упростит цифро-аналоговое преобразование в передатчике, но потребует более быстрой цифровой обработки.

3. Формирование импульсов. Если передискретизированный сигнал пропустить через фильтр с импульсной характеристикой $g_{TX}[n]$ в виде импульсов с косинусным сглаживанием, на выходе фильтра каждый символ будет представлен импульсом с косинусным сглаживанием. Операции с 1 по 3 преобразуют входной поток битов в биполярный сигнал сообщения. Обратите внимание, что помимо импульсов с косинусным сглаживанием, можно использовать другие формы импульсов, просто изменив форму импульсной характеристики $g_{TX}[n]$. Импульсы с косинусным сглаживанием имеют очень высокую скорость затухания спектра, поэтому передаваемый сигнал не вызовет помех у сигналов с близкими значениями частот несущей.

4. Модуляция. Биполярный сигнал сообщения, состоящий из последовательности импульсов вида $(\pm 1 + j0)g_{TX}[n]$, можно послать непосредственно в передатчик USRP. USRP преобразует сигнал в непрерывный и добавит несущую, как показано в уравнении (2).

Приемник

PSK-приемник начинается с DSB-SC-демодулятора. Когда передаваемый BPSK-сигнал поступает на приемник, он выглядит как последовательность импульсов вида

$$r(t) = \pm Dg_{TX}(t) \cos(2\pi f_c t + \phi), \quad (5)$$

где D является константой, как правило, значительно меньше константы A передаваемого сигнала, а угол ϕ представляет собой разность фаз между генераторами несущей передатчика и приемника. Если генератор несущей приемника настроен на ту же частоту, что и генератор передатчика, приемник USRP выполнит большую часть работы по демодуляции BPSK-сигнала. Функция *Fetch Rx Data* приемника выдает последовательность выходных импульсов, имеющих вид

$$\tilde{r}[n] = \pm \frac{D}{2} g_{TX}[n] e^{j\phi}. \quad (6)$$

Частота дискретизации в уравнении (6), $1/T_c$, задается параметром "IQ rate" приемника. Эта частота обеспечивает взятие M отсчетов каждые T секунд, где $1/T$ - скорость передачи символов.

Мы можем убрать сдвиг фаз ϕ одним из нескольких способов. Хорошо работает вариант, рассмотренный в лабораторной работе по DSB-SC. Подытоживая сказанное, начнем с возведения $\tilde{r}[n]$ в квадрат, что даст

$$\tilde{r}^2[n] = \frac{D^2}{4} g_{TX}^2[n] e^{j2\phi}. \quad (7)$$

Эта операция исключает изменения фазы, вызванные данными или изменением полярности $g_{TX}[n]$. Угол 2ϕ можно извлечь функцией *Complex to Polar* из палитры *Data Types* → *Numeric* → *Complex*. На данной стадии будет полезно сгладить отклонения в 2ϕ , вызванные помехами. Для этого пригодится фильтр *median filter* из папки *ExternalFiles*. Для параметров "left rank" и "right rank" можно принять значения по умолчанию. Далее, функция *Unwrap Phase* из палитры *Analysis* → *Signal Processing* → *Signal Operations* удалит скачки $\pm 2\pi$. Наконец, деление на два даст нам желаемую оценку погрешности фаз ϕ . На блок-диаграмме, приведенной на рисунке 1, показан весь процесс синхронизации фаз.

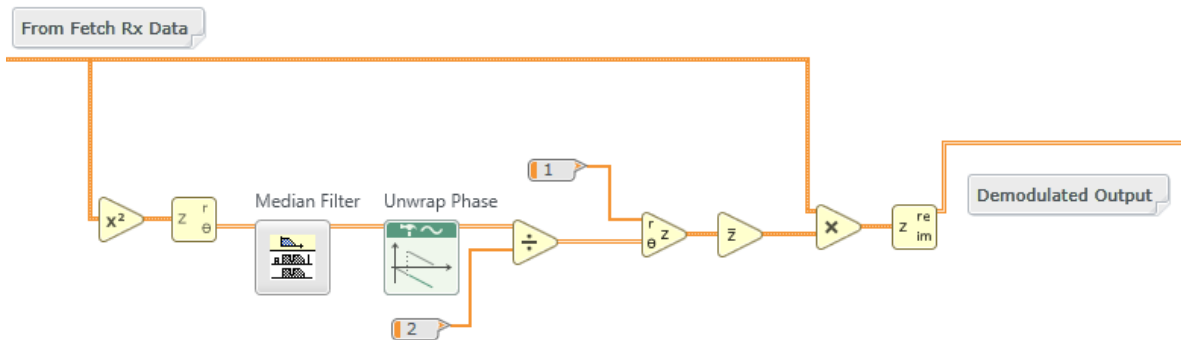


Рисунок 1. Синхронизация фазы несущей

Остальные операции, выполняемые приемником, должны быть вам знакомы из лабораторных работ по ASK и FSK. А именно:

1. Согласованная фильтрация. Мы используем в приемнике фильтр со сглаживанием спектра по закону приподнятого косинуса. Его импульсная характеристика $g_{RX}[n]$ согласована с формой $g_{TX}[n]$ принятых импульсов. Согласованный фильтр дает оптимальную производительность в присутствии аддитивного белого гауссова шума. Более того, каскад из двух фильтров со сглаживанием спектра по закону приподнятого косинуса $g_{TX}[n]$ и $g_{RX}[n]$ формирует импульсы с косинусным сглаживанием, не подверженные межсимвольным помехам.
2. Синхронизация импульсов. С выхода согласованного фильтра снимают аналоговый сигнал, отсчеты которого необходимо снимать по одному на каждый символ, то есть каждые T секунд. Из-за задержек при фильтрации и распространении сигнала, а также из-за искажений в канале связи, необходимо определить оптимальное время для взятия этих отсчетов. Функция $PulseAlign(real)$ выравнивает сигнал так, чтобы отсчет с индексом 0 был правильным первым отсчетом.
3. Взятие отсчетов. Функция $Decimate$ дискретизирует выровненный сигнал, начиная с индекса o и далее каждые T секунд.

4. Детектирование. После дискретизации сигнала необходимо определить, какое значение представляет каждый отсчет – символ 0 или символ 1.
5. Кодирование символов. Значения детектированных символов необходимо преобразовать в биты. Для двоичной PSK эта операция легко включается в процесс детектирования.

9.3 Подготовка к лабораторной работе

Передатчик

1. Создайте программу для генерации BPSK-сигнала при помощи USRP. Шаблон для передатчика подготовлен в файле *BPSKTxTemplate.gvi*. Этот шаблон содержит четыре функции интерфейса с USRP, а также функцию *MT Generate Bits* из комплекта Modulation. Функция *MT Generate Bits* сгенерирует псевдослучайную последовательность битов, которая может служить последовательностью данных для тестирования вашей BPSK-системы. Обратите внимание, что по умолчанию *MT Generate Bits* будет выдавать одну и ту же последовательность при каждом запуске программы. Это полезно для отладки, однако если вы захотите, чтобы каждый раз генерировалась новая последовательность, подключите функцию random number ко входу "seed in". В последующих пунктах подробно рассматривается, как создать искомый передатчик.
2. Сначала выполните кодирование символов, как показано в таблице 1.
3. Выполните передискретизацию массива символов функцией *Upsample* из субпалитры Analysis→Signal Processing→Signal Operation. В этой лабораторной работе вам даются управляющие входы, чтобы задать скорость передачи символов $1/T$ и частоту IQ $1/T_x$. Задайте скорость передачи символов равной 10 000 символов/с и частоту IQ равной 200×10^3 отсчетов/с. Используя скорость передачи символов и приведенную частоту IQ, рассчитайте коэффициент передискретизации L .
4. Используйте функцию *MT Generate Filter Coefficients* из комплекта Modulation Toolkit для генерации фильтра, формирующего импульсы. (Функцию *MT Generate Filter Coefficients* можно найти в

субпалитре Analysis→Communications→Digital→Utilities.) Задайте тип модуляции PSK, а вход "samples per symbol" равным рассчитанному вами значению L . Создайте на лицевой панели элемент управления для входа "pulse shaping filter" и установите его в состояние "Root Raised". Подключите выход "pulse shaping filter coefficients" ко входу "Y" функции *Convolution*. Эта функция находится в субпалитре Analysis→Signal Processing→Signal Operation. Подключите выходной передискретизированный сигнал ко входу "X" функции *Convolution*.

5. Нормализуйте амплитуду отфильтрованного сигнала сообщения к максимальному абсолютному значению, равному единице. Возьмите функцию *Quick Scale 1D* из папки ExternalFiles, чтобы найти максимум абсолютного значения. Чтобы гарантировать, что отмасштабированное сообщение останется комплексным, используйте отдельную функцию деления для выполнения масштабирования. Подключите отмасштабированное сообщение ко входу функции *Build Waveform*, соединенной в шаблоне с графическим индикатором Baseband Waveform. Отмасштабированное сообщение подайте также на функцию *Write Tx Data*.
6. Для наблюдения спектра передаваемого сигнала, подключите комплексный модулирующий сигнал к функции *FFT Power Spectrum function*, соединенной в шаблоне с графическим индикатором Power Spectrum.

Вы завершили создание BPSK-передатчика. Сохраните программу передатчика в файл, в имени которого содержатся буквы "BPSKTx" и ваши инициалы (например, *BPSKx_BAB.gvi*).

Приемник

1. Создайте программу для генерации BPSK-сигнала при помощи USRP. Шаблон для приемника подготовлен в файле *BPSKRxTemplate.gvi*. Этот шаблон содержит шесть функций интерфейса с USRP.

Рассчитайте количество отсчетов для функции *Fetch Rx Data* на основании длины сообщения и скорости передачи символов, взятых из элементов управления на лицевой панели, и приведенной частоты IQ. Удвойте это количество, чтобы приемник захватывал два кадра данных. Поскольку приемник начинает работу в произвольное время, мы тем самым гарантируем получение одного полного кадра в принятом блоке отсчетов.

Реализуйте синхронизацию фазы несущей, как показано на рисунке 1.

2. Для реализации в приемнике согласованного фильтра используйте *MT Generate Filter Coefficients* так же, как вы это делали в передатчике. Задайте тип модуляции PSK и рассчитайте количество согласованных отсчетов на символ ("matched samples per symbol") M из реальной частоты IQ ("actual IQ rate" ($1/T_z$)) и скорости передачи символов (symbol rate ($1/T$)), полученных из элементов управления на лицевой панели. Создайте на лицевой панели элемент управления для входа "pulse shaping filter" и установите его в состояние "Root Raised". Подключите выход "matched filter coefficients" ко входу "Y" функции *Convolution*. Выход фильтра, формирующего импульсы, подключите в шаблоне к функции *Cluster Properties*. Функцию *Cluster Properties* соедините с графическим индикатором Baseband Output.
3. Поместите на блок-диаграмму функцию *PulseAlign(real)* из папки ExternalFiles и подключите выходной сигнал ко входу "input waveform", а рассчитанное значение M отсчетов/символ ко входу "receiver sampling factor".

Как только модулирующий сигнал выровнен, можно брать его отсчеты. Возьмите функцию *Decimate (single shot)* из субпалитры Analysis→Signal Processing→Signal Operation. Коэффициент прореживания ("decimating factor") равен M .

4. Функция *MT Format Eye Diagram* из комплекта Modulation Toolkit имеется в шаблоне приемника. Подключите выходной сигнал ко входу "waveform" глазковой диаграммы. Возьмите входное значение "symbol rate (Hz)" из элемента управления на лицевой панели. Задайте параметр "eye length" равным 2.
5. Чтобы определить, представляет полученный отсчет 0 или 1, необходимо сравнить его с пороговым значением. Поскольку сообщение - биполярный сигнал, в качестве порогового значения можно взять 0. Результат сравнения - цифровой выход приемника. На выходе функции сравнения будет булевский массив. Вы можете преобразовать его в массив целых чисел, используя функцию *Boolean To Integer*.

Вы завершили создание BPSK-приемника. Сохраните программу приемника в файл, в имени которого содержатся буквы "BPSKRx" и ваши инициалы (например, *BPSKRx_BAB.gvi*).

Вопросы

1. В пункте 3 создания передатчика вы задали $1/T = 10000$ символов/с и $1/T_x = 200 \times 10^3$ (отсчетов/с).
Найдите соответствующее значение количества отсчетов на символ L .
2. В уравнении (6) раздела "Вводная информация" показано, что принятый сигнал $\tilde{r}[n]$ содержит множитель $e^{j\phi}$, где ϕ - некоторая разность фаз, которая может быть между генераторами несущих приемника и передатчика. Объясните, что произойдет, если исключить из приемника шаг синхронизации фаз. А именно, что будет на выходе приемника, если ϕ примет значение $\pi/2$?
3. В разделе "Приемник", пункт 2, значение реальной частоты IQ $1/T_z$ могло отличаться от частоты $1/T_x$, использованной в передатчике. (Обратите внимание, что скорость передачи символов $1/T$ должна быть одинакова и в приемнике, и в передатчике). Значение частоты IQ приемника определяет коэффициент M выборки приемника. В чем преимущество использования более высокого значения M ? В чем преимущество использования более низкого значения M ?
4. Хотя BPSK представляет собой версию ASK с подавлением несущей, мы не используем детектор огибающей для демодуляции BPSK, как делали это с ASK. Почему? Что будет на выходе приемника, если использовать для демодуляции детектор огибающей?
5. Вероятность ошибок - мера эффективности цифровой системы связи. Как изменится вероятность ошибок, если передатчик выйдет из строя, и приемник будет принимать только шум? Объясните, почему вы так считаете.

9.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

Частота IQ (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_x$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): TX1

Скорость передачи символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): Root Raised

Запустите передатчик. Используйте большую кнопку STOP на лицевой панели для отключения разъемов передачи.

3. После запуска передатчика наблюдайте спектр передаваемого сигнала. Измерьте ширину главного лепестка спектра передаваемого сигнала. Измените состояние элемента управления фильтром, формирующим импульсы, на "none" для генерации прямоугольных импульсов, и снова запустите передатчик. Сравните спектр передаваемого сигнала со спектром, полученным для импульсов с косинусным сглаживанием. Верните прежнюю настройку фильтра "Root Raised".

4. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

Частота IQ (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_x$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): RX2

Скорость передачи символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): Root Raised

5. Запустите сначала передатчик, затем приемник. После получения приемником блока данных можно остановить передатчик.

6. Наблюдайте глазковую диаграмму. Обратите внимание на оптимальное время выборки и присутствие или отсутствие межсимвольных помех. Для наблюдения эффекта синхронизации импульсов подайте на вход функции *MT Format Eye Diagram* сигнал с выхода "aligned waveform" функции *PulseAlign(real)*. Запустите передатчик и приемник. Наблюдайте глазковую диаграмму. Каково теперь оптимальное время выборки?

7. Модифицируйте передатчик, включив в него функцию *AddFrameHeader(real)* из папки *BasicUSRPLab*. Поместите ее после кодирования символов, но до преобразования символов в комплексные числа. Модифицируйте приемник, включив в него функцию *FrameSync(real)* из папки *BasicUSRPLab*. Поместите ее сразу за функцией *Decimate*. Подключите выход функции *Decimate* ко входу "Sampled Input" функции *FrameSync(real)*. Остальные входы *FrameSync(real)* оставьте неподключенными. Подключите выход "Aligned Samples" (выровненные отсчеты) функции *FrameSync(real)* к функции сравнения с порогом.

Подключите массив битов с выхода функции сравнения с порогом ко входу "array" функции *Array Subset*. Задайте вход "index" равным нулю, а на вход "length" подайте длину, определенную элементом управления "message length". Значение с выхода функции *Array Subset* подайте на индикатор "Output bits" на лицевой панели приемника.

Обратите внимание, что в этой лабораторной работе не используются выходы "Output Signal" и "max index" функции *FrameSync(real)*.

8. Измерение вероятности ошибок (BER) можно автоматизировать функцией *MT Calculate BER* из комплекта Modulation Toolkit (субпалитра Analysis → Communications → Digital → Measurements). На ленте Configure выберите "PN Fibonacci". Задайте параметр "BER trigger threshold" равным 0.4.

Подключите индикаторы к выходам "BER" и "trigger found?" При выполнении программы на выходе "trigger found?" будет истинное значение каждый раз, когда измеренное значение вероятности ошибок будет меньше заданного порога.

Запустите передатчик и приемник. Если все работает правильно, вероятность BER должна быть равна 0 или 1.0. Запустите передатчик, затем запустите приемник не меньше десяти раз. Убедитесь, что примерно в половине случаев вероятность BER равна 0, а в половине - 1.0.

Вопросы

1. Как соотносятся ширина главного лепестка спектра и скорость затухания спектра импульсов с косинусным сглаживанием с этими же величинами при использовании прямоугольных импульсов?
2. Что означает BER, равная 1.0? Объясните, почему в половине случаев BER равна 0, а в половине - 1.0.

Дифференциальное кодирование

Предположим, что до отправки в передатчик биты данных проходят через логическую схему, показанную на рисунке 2. В этой схеме вентиль исключающее ИЛИ выдает 0, когда текущее значение входа отличается от предыдущего значения выхода, и 1 - когда они совпадают.

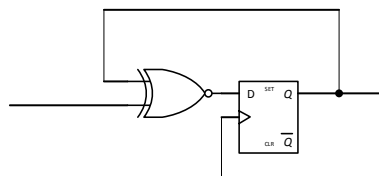


Рисунок 2. Дифференциальный кодер

Например, из входной последовательности

1 0 1 1 0 0 0 1 ...

сформируется

0 0 1 1 1 0 1 0 0 ... ,

при условии, что исходное состояние триггера – 0. Теперь предположим, что принятая последовательность битов передается через схему, показанную на рисунке 3 Рисунок 3. На выходе будет 1 каждый раз, когда текущее значение

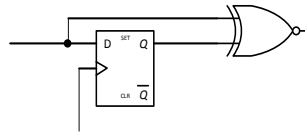


Рисунок 3. Дифференциальный декодер

входа и предыдущее значение входа совпадают, и 0 - когда различаются. Например, из принятой последовательности битов

0 0 1 1 1 0 1 0 0 ...

сформируется

1 0 1 1 0 0 0 1 ...

Как видно, декодер на рисунке 3 выполняет преобразование, обратное тому, что выполняется в кодере на рисунке 2. Важно обратить внимание на то, что происходит при инвертировании принятой комбинации битов. Если принятая последовательность битов равна

1 1 0 0 0 1 0 1 1 ...,

"одинаковые" и "разные" биты не изменяются, то есть на выходе декодера все равно будет

1 0 1 1 0 0 0 1 ...

9. Добавьте дифференциальный кодер в передатчик BPSK, а дифференциальный декодер - в приемник. Поместите кодер сразу после функции *MT Generate Bits*, перед кодированием символов. Поместите декодер после сравнения с порогом. Обратите внимание, что передаваемую последовательность необходимо будет удлинить на 1 бит, чтобы включить начальное состояние триггера кодера.

Подсказка: D-триггер создает задержку в один отсчет. Эту задержку легко реализовать в LabVIEW функцией *Feedback Node* из палитры *Programming Flow*.

10. Запустите передатчик, затем запустите приемник не меньше десяти раз. Вы должны увидеть, что вероятность ошибок BER всегда равна 0.

Дифференциальное кодирование, часть 2

Дифференциальное кодирование можно также выполнять после кодирования символов.

Таблица 2

В таблице 2 приведена таблица истинности, показывающая взаимосвязь между операцией исключающего ИЛИ-НЕ и обычным умножением. Ниже мы увидим, что дифференциальное кодирование позволяет исключить из схемы приемника операцию синхронизации фазы.

Таблица 2: Исключающее ИЛИ-НЕ и умножение чисел

В двоичном виде			В числовом виде		
Вход		Выход	Вход		Выход
0	0	1	-1	-1	1
0	1	0	-1	1	-1
1	0	0	1	-1	-1
1	1	1	1	1	1

11. Реализуйте дифференциальный кодер в числовом виде. В передатчике поместите кодер после функции *AddFrameHeader(real)*. (При этом заголовок кадра тоже будет закодирован). Не забудьте добавить начальное состояние кодера в начало передаваемой последовательности.

Сохраните программу передатчика в файл, в имени которого содержатся буквы "BPSKTx" и ваши инициалы (например, *BPSKx_BAB.gvi*).

12. В приемнике удалите синхронизатор фазы и подайте принятые данные непосредственно на функцию *Convolution*, реализующую согласованный фильтр. В приемнике поместите дифференциальный декодер сразу после функции *Decimate*, предназначенной для взятия отсчетов, и перед функцией *FrameSync(real)*. Поскольку на данной стадии принятые данные

являются комплексными, разрабатываемый вами декодер должен формировать результат на основе текущего отсчета и комплексно-сопряженной величины предыдущего отсчета. Затем возьмите действительную часть результата. Вам может также понадобиться заменить функцию *PulseAlign(real)* на *PulseAlign(Complex)*.

Сохраните программу приемника в файл, в имени которого содержатся буквы "BPSKRx" и ваши инициалы (например, *BPSKRx_BAB.gvi*).

12. Запустите сначала передатчик, затем приемник несколько раз. Убедитесь, что вероятность ошибок BER всегда равна 0.

Вопросы

1. Покажите, что дифференциальный кодер на рисунке 2 и дифференциальный декодер на рисунке 3 продолжают правильно работать в системе, если исходное состояние триггера кодера равно 1.
2. Начиная с уравнения (6), покажите, почему в DPSK-приемнике не требуется синхронизация фазы.
3. В BPSK-системе амплитуда глазковой диаграммы одинакова при каждом запуске приемника. В DPSK-системе эта амплитуда при каждом запуске изменяется. Объясните, почему так происходит. Объясните, будет ли BER тоже изменяться при каждом запуске DPSK-приемника в присутствии помех.
4. Дифференциальная фазовая манипуляция должна обладать устойчивостью при ненулевой разности фаз между генераторами передатчика и приемника. Возможно также, что у этих генераторов будут отличаться частоты. Предположим, что между генераторами приемника и передатчика существует небольшая разность частот Δf . Модифицируйте уравнение (6), чтобы эта разность принималась во внимание. Как изменится сигнал на выходе DPSK-приемника из-за этой разницы частот? Используя значения параметров из данной лабораторной работы, определите, насколько большой должна быть разность частот, чтобы считать это "значимым". Обязательно уточните, что вы подразумеваете под словом "значимое".

9.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию по созданным вами функциям для передатчика и приемника. Предоставьте также документацию для любых созданных вами функций. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Ответьте на все вопросы раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте созданным вами программы передатчика и приемника. Включите программы для BPSK и DPSK. Кроме того, включите в отчет все созданные вами функции. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше.

Предоставьте документацию для программ *DPSKTx* и *DPSKRx*. Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы.

Ответьте на все вопросы из каждого раздела "Выполнение лабораторной работы".

Глазковая диаграмма

Предварительные требования: Лабораторная работа 9 - Двоичная фазовая манипуляция (BPSK)

10.1 Цель

Хотя мы отображали глазковую диаграмму в каждой выполненной до сих пор лабораторной работе по цифровым методам модуляции, для непосредственного ее создания мы полагались на комплект Modulation Toolkit. В этой лабораторной работе мы научимся создавать глазковую диаграмму с нуля. Мы рассмотрим, как изменяется глазковая диаграмма при наличии в канале связи межсимвольных искажений (intersymbol interference ISI) и научимся измерять уровень межсимвольных искажений (ISI) по глазковой диаграмме.

В этой лабораторной работе в качестве тестовой системы используются BPSK-передатчик и BPSK-приемник, разработанные в лабораторной работе 9. Для создания межсимвольных искажений предоставлена функция *Channel.gvi*. Однако, если у вас есть второе устройство USRP, вы можете попробовать реализовать передачу от одного устройства к другому через антенны и посмотреть, сколько межсимвольных помех возникает на самом деле из-за многолучевого распространения волн в реальной среде.

10.2 Вводная информация

Если рассматривать сообщение, генерируемое BPSK-передатчиком, как комплексный сигнал, можно записать

$$\tilde{x}(t) = A \sum_{n=-\infty}^{\infty} a_n g_{TX}(t - nT), \quad (1)$$

где A - константа, определяющая среднюю передаваемую мощность, $a_n = \pm 1$ - значения символов, несущие передаваемую информацию, $g_{TX}(t)$ - форма передаваемых импульсов и T - время между символами. Тильда над \tilde{x} подчеркивает, что это узкополосный сигнал; а сигнал, который на самом деле передается, равен

$$\begin{aligned} x(t) &= \text{Re} \left[\tilde{x}(t) e^{j2\pi f_c t} \right] \\ &= A \sum_{n=-\infty}^{\infty} a_n g_{TX}(t - nT) \cos(2\pi f_c t). \end{aligned} \quad (2)$$

Обратите внимание, предполагается, что импульсы $g_{TX}(t)$ действительны.⁹

Передаваемый сигнал проходит по каналу связи, после чего демодулируется приемником. Смоделируем канал связи, как линейный инвариантный во времени фильтр с импульсной характеристикой $h(t)$. Когда сигнал поступает в приемник и демодулируется, он пропускается через фильтр приемника с импульсной характеристикой $g_{RX}(t)$. Обозначим импульсную характеристику тракта фильтр передатчика, канал и фильтр приемника, как $g(t)$. То есть,

$$g(t) = g_{TX}(t) * h(t) * g_{RX}(t). \quad (3)$$

Тракт $g_{TX}(t)$ и $g_{RX}(t)$, как правило, разрабатывают так, чтобы он был свободен от межсимвольных искажений. В настоящей лабораторной работе для этого мы используем тракт, в котором объединенный фильтр формирует спектр по закону приподнятого косинуса. К сожалению, проектировщик никак не может контролировать отклик канала $h(t)$, который может внести межсимвольные искажения в принятый сигнал. После фильтрации в приемнике принятый сигнал имеет вид

$$\tilde{y}(t) = D \sum_{n=-\infty}^{\infty} a_n g(t - nT) + n(t), \quad (4)$$

где D - константа амплитуды, $g(t)$ - форма импульсов, определяемая уравнением (3), и $n(t)$ - белый гауссов шум, проходящий через фильтр приемника.

После того, как принятый сигнал отфильтрован, берутся отсчеты по одному на символ каждые T секунд. Если используется синхронизация импульсов, и отсчеты берутся в правильные моменты времени, дискретизированный сигнал равен

$$\tilde{y}(kT) = D \sum_{n=-\infty}^{\infty} a_n g[(k - n)T] + n(kT). \quad (5)$$

Эти отсчеты сравниваются с порогом, равным нулю, чтобы определить, представляет каждый полученный символ $+1$ или -1 . Перепишем уравнение (5) как

⁹ Сигнал $\tilde{x}(t)$ и импульсные характеристики $g_{TX}(t)$ и $g_{RX}(t)$ на самом деле реализуются в дискретном времени. Однако запись их в виде непрерывных функций значительно облегчает чтение.

$$\tilde{y}(kT) = Da_k g(0) + D \sum_{\substack{n=-\infty \\ n \neq k}}^{\infty} a_n g[(k-n)T] + n(kT). \quad (6)$$

Первый терм в уравнении (6) представляет желаемый отсчет в момент времени $t = kT$, второй терм представляет межсимвольные искажения, а третий терм представляет шум.

Чтобы оценить уровень межсимвольных искажений, порождаемых $g(t)$ в худшем случае, предположим, что для $a_k = 1$ и $a_n, n \neq k$ формируется последовательность, в которой каждый искаженный элемент становится отрицательным. То есть, если последовательность данных

$$a_n = \begin{cases} -1, & g[(k-n)T] \geq 0 \\ 1, & g[(k-n)T] < 0. \end{cases} \quad (7)$$

то уравнение (6) становится равным

$$\begin{aligned} \tilde{y}(kT) &= Dg(0) - D \sum_{\substack{n=-\infty \\ n \neq k}}^{\infty} |g[(k-n)T]| + n(kT) \\ &= Dg(0) - D \sum_{n \neq 0} |g(nT)| + n(kT). \end{aligned} \quad (8)$$

Без учета помех пример из уравнения (8) представляет собой "наихудший случай". Запишем

$$V_w = Dg(0) - D \sum_{n \neq 0} |g(nT)|. \quad (9)$$

Мы можем также определить "наилучший случай". Если каждый искажающий элемент увеличивает отсчет, то получим

$$V_b = Dg(0) + D \sum_{n \neq 0} |g(nT)|. \quad (10)$$

По традиции, уровень искажений символов на выходе фильтра приемника измеряется одним из двух параметров. *Максимальное искажение сигнала (peak distortion)* определяется как

$$\text{максимальное искажение} = \frac{D \sum_{n \neq 0} |g(nT)|}{Dg(0)}. \quad (11)$$

Раскрытие глазка (*eye opening*) определяется, как

раскрытие глазка = 1- максимальное искажение

(12)

Пример глазковой диаграммы показан на рисунке 1.

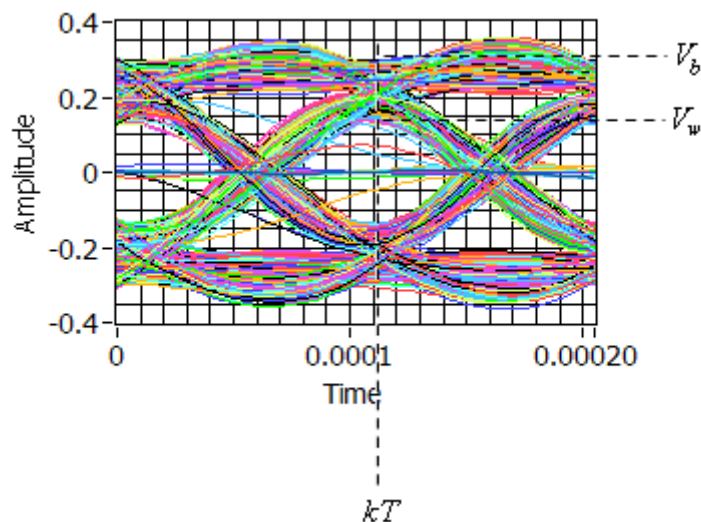


Рисунок 1. Глазковая диаграмма

Показаны время взятия выборки kT , наилучшее и наихудшее напряжения V_b и V_w , соответственно.

Обратите внимание, что V_b и V_w должны быть измерены в один и тот же момент времени kT . Как вы

видите из рисунка, величины $D \sum_{n \neq 0} |g(nT)|$ и $Dg(0)$, необходимые для нахождения максимального искажения диаграммы или раскрытия глазка не сразу очевидны. Однако, если переписать уравнения (11) и (12), мы увидим, что

$$\text{максимальное искажение} = \frac{V_b - V_w}{V_b + V_w} \quad (13)$$

и

$$\text{раскрытие глазка} = \frac{2V_w}{V_b + V_w}. \quad (14)$$

Так, например, мы можем измерить V_b и V_w по глазковой диаграмме и использовать уравнение (14)

для вычисления раскрытия глазка. На рисунке 1 $V_b = 0.296$ и $V_w = 0.124$, что дает раскрытие глазка 0.59, или около 60%.

Межсимвольные искажения причиняют неудобство, потому что снижают эффективность системы связи. Для BPSK-системы с согласованным фильтром приемника, в присутствии аддитивного белого гауссова шума и при отсутствии межсимвольных искажений, вероятность ошибки составляет

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \quad (15)$$

где E_b - среднее количество энергии на бит в приемнике, $N_0/2$ - спектр мощности шума на входе приемника, а Q - интеграл гауссовской вероятности. Если есть межсимвольные искажения, вероятность ошибки определяется по наихудшему варианту. Для раскрытия глазка η наихудшая вероятность ошибки имеет вид

$$P_e = Q\left(\eta\sqrt{\frac{2E_b}{N_0}}\right), \quad (16)$$

где E_b - количество энергии на бит, без межсимвольных искажений, как и в уравнении (15).

10.3 Подготовка к лабораторной работе

Глазковая диаграмма

1. Модифицируйте BPSK-приемник, включив в него собственный функционал построения глазковой диаграммы. Как это сделать:

Требуемая глазковая диаграмма представляет собой график выровненного сигнала на выходе функции *PulseAlign(real)*. Протяженность горизонтальной оси (оси времени) составляет удвоенную длительность символа. Графики, относящиеся к последовательным парам символов, накладываются друг на друга. Чтобы построить такой график, извлеките из выровненного сигнала массив Y . Извлеките также dt , далее это вам пригодится. Потом используйте функцию *Reshape Array* из субпалитры *Data Types* → *Array* и сформируйте двумерный массив, каждая строка которого содержит два символа данных. Первый вход "dimension size" функции *Reshape Array* отвечает за количество строк, а второй - за количество столбцов. Количество строк определяет количество кривых графика, а количество столбцов - число отсчетов в двух символах данных. Вам понадобится рассчитать эти значения в вашей программе. Выход функции *Reshape Array* представляет собой двумерный массив, который необходимо подключить к функции "2D DBL to 1D Cluster" до вывода на график. Используйте значение dt и выход функции *Reshape Array*. Подключите результирующий кластер к графическому индикатору *Waveform Graph*. Когда на графический индикатор *Waveform Graph* подается двумерный массив, каждая строка массива

выводится на график в виде отдельной кривой. Именно так должна вести себя глазковая диаграмма.

2. Сохраните модифицированный BPSK-приемник в файл, в имени которого содержатся буквы "BPSKEyeRx" и ваши инициалы (например, *BPSKEyeRx_BAB.gvi*).

Модель канала

Чтобы протестировать глазковую диаграмму, вам понадобится внести некоторые межсимвольные искажения. Для этого служит функция *Channel.gvi* из папки *BasicUSRPLabs*. Поместите в приемнике *Channel.gvi* непосредственно после функции *Fetch Rx Data*. Подключите входы "Use Channel" (используемый канал) и "Propagation Delay" (задержка распространения) к элементам управления на лицевой панели. В приемнике подключите ко входу "Samples per Symbol" значение количества отсчетов на символ, рассчитанное из реальной частоты IQ и скорости передачи символов. Остальные входы *Channel.gvi* можно оставить неподключенными.

Channel.gvi симулирует многолучевой канал. Заданные по умолчанию параметры канала создают импульсную характеристику, имеющую вид

$$h(t) = \delta(t) + 0.2\delta(t - \tau) - 0.08\delta(t - 2\tau), \quad (17)$$

где τ - задержка распространения, заданная элементом управления на лицевой панели. *Channel.gvi* может также добавлять в сигнал шум, но мы не будем использовать эту возможность в данной лабораторной работе.

Многолучевое распространение сигнала создает пики и впадины в частотной характеристике канала связи. Из-за этого возникают искажения принятого сигнала, что приводит к межсимвольным искажениям. Некоторые авторы используют так называемую *ширину полосы когерентности* (*coherence bandwidth*) в качестве меры нерегулярности частотной характеристики канала. В грубом приближении ширина полосы когерентности - это интервал частот, в котором коэффициент усиления канала остается примерно постоянным. Межсимвольные искажения становятся серьезными, когда ширина полосы частот сигнала превышает ширину полосы когерентности канала. Ширина полосы когерентности обратно пропорциональна разбросу задержек распространения между первым и последним принятыми сигналами. По умолчанию для канала из уравнения (17) ширина полосы когерентности обратно пропорциональна параметру τ . Их отношение составляет

$$B_{coh} = \frac{0.816}{\tau},$$

где B_{coh} - ширина полосы когерентности в герцах, а τ - в секундах.

Вопросы

1. Воспользуйтесь уравнением (15), чтобы найти E_b/N_0 для вероятности ошибки $P_e = 10^{-6}$. Теперь используйте уравнение (16), чтобы найти P_e для раскрытия глазка $\eta = 0.8$. Повторите для $\eta = 0.5$. Используя то же значение E_b/N_0 , найдите раскрытие глазка η , которое даст $P_e = 10^{-5}$.
2. Предположим, $g(0) = 1$, $g(T) = 0.2$, $g(2T) = -0.08$ и $g(nT) = 0$ в противном случае. Используя уравнения (9) и (10), найдите V_w and V_b . Затем используйте уравнение (14) для вычисления раскрытия глазка.
3. Используя уравнение (18), найдите ширину полосы когерентности для $\tau = 50 \mu s$, $\tau = 100 \mu s$ и $\tau = 150 \mu s$.

10.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2. Подключите USRP к вашему компьютеру и включите питание USRP. Запустите LabVIEW и откройте созданные вами программы приемника и передатчика.

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

Частота IQ (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_x$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): TX1

Частота символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): Root Raised

Запустите передатчик. Используйте большую кнопку STOP на лицевой панели для отключения разъемов передачи.

3. После запуска передатчика наблюдайте спектр передаваемого сигнала. Измерьте ширину главного лепестка спектра передаваемого сигнала. Ширина полосы частот сигнала составляет половину ширины главного лепестка, если считать только компоненты с положительной частотой.

4. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

Частота IQ (IQ Rate): 200 кГц. Примечание: Это задает значение $1/T_z$. Обратите внимание, что параметры T_z и dt совпадают.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): RX2

Частота символов (Symbol Rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): Root Raised

Использовать канал (Use Channel): off

5. Запустите сначала передатчик, затем приемник. После получения приемником данных можно остановить передатчик. Вероятность ошибок BER приемника должна составлять 0.0 или 1.0. Пусть в данной работе вас не беспокоит значение BER, равное 1.0.

6. Наблюдайте глазковую диаграмму, полученную с помощью программы, которую вы создали при выполнении пункта 1 подготовки к лабораторной работе. Измерьте V_w и V_b и рассчитайте раскрытие глазка. Оно не должно быть равным 100% из-за фильтрации в USRP.

Наиболее эффективный способ измерения V_w и V_b по глазковой диаграмме - использовать курсоры. Щелкните по графическому индикатору глазковой диаграммы, а в ленте щелкните по кнопке Graph Parts. В диалоговом окне щелкните по иконке отображения курсоров Cursor Legend. Для создания нового курсора щелкните по кнопке New Cursor в окошке Cursors рядом с полем глазковой диаграммы.

Для измерения V_w и V_b поместите курсор X в момент времени, в котором раскрытие глазка максимально. Эта точка будет находиться вблизи центра временной оси. Потом используйте

курсор Y для измерения V_w и V_b . Не меняйте положение X между измерениями. Более точную пару результатов измерений вы получите, если прежде, чем использовать курсор Y, увеличите на графическом индикаторе необходимую область глазковой диаграммы (кнопка Graph Parts, Graph Tools).

7. Задайте задержку распространения τ равной $50 \mu s$, задайте Use Channel равным "on" и повторите операции по пунктам 5 и 6. Повторите для $\tau = 100 \mu s$ и $\tau = 150 \mu s$. Составьте таблицу со значениями раскрытия глазка для каждого значения τ .

Вопросы

1. Почему в этой лабораторной работе нас не беспокоит то, что вероятность ошибок BER равна 1.0?
2. Многолучевое распространение радиоволн – типичное явление в сотовой телефонии. Задержки распространения могут отличаться в зависимости от местоположения базовой станции, мобильного устройства и соседних зданий, но эти задержки не зависят от скорости передачи символов. Однако с каждым новым поколением сотовой связи скорость передачи символов растет. На основании результатов, полученных вами в ходе выполнения лабораторной работы, обсудите соотношение между скоростью передачи символов, шириной полосы когерентности, раскрытием глазка и вероятностью ошибок BER. Что происходит при увеличении скорости передачи символов?

10.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию по модифицированному вами приемнику, включая глазковую диаграмму. Предоставьте также документацию для любых созданных вами функций. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Ответьте на все вопросы из раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте модифицированный приемник, включая глазковую диаграмму. Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы. Предоставьте таблицу, составленную при выполнении пункта 7 лабораторной работы.

Ответьте на все вопросы из раздела "Выполнение лабораторной работе".

Эквалайзинг

Предварительные требования: Лабораторная работа 10 – Глазковая диаграмма

11.1 Цель

В большинстве цифровых систем связи, фильтр формирования импульсов передатчика и согласованный фильтр приемника разработаны таким образом, что импульсы, получаемые от согласованного фильтра приемника, не содержат ошибок, вызванных межсимвольными помехами (intersymbol interference). Однако, как мы уже обсуждали в лабораторной работе по глазковой диаграмме, в канале связи сигнал зачастую подвергается дополнительной фильтрации, что может привести к межсимвольным искажениям.

Одним из способов борьбы с межсимвольными помехами при создании канала связи является использование дополнительного фильтра в приемнике. В идеальном случае, дополнительный фильтр должен иметь характеристику, "обратную" характеристике фильтра в канале. Фильтр, предназначенный для борьбы с негативными побочными эффектами, обусловленными фильтрацией в канале, называется эквалайзер.

В беспроводных системах причиной искажения сигнала в канале часто является многолучевое распространение. То есть, принятый сигнал может включать в себя отражения от зданий и других объектов окружающей среды. Искажение, которое наблюдается в конкретной беспроводной линии будет зависеть от местоположения передатчика, приемника и расположенных рядом отражающих объектов. Поскольку передатчик, приемник и даже некоторые из отражающих объектов могут перемещаться, искажение сигнала в канале со временем может понемногу изменяться. В этом случае было бы полезно использовать *адаптивный* эквалайзер, который способен отслеживать изменения в канале и с учетом этих изменений изменять свои свойства.

В данной лабораторной работе вам потребуется передатчик и приемник BPSK, созданный в лабораторной работе 9 в качестве испытательного стенда системы. *Channel.gvi*, который вы использовали в лабораторной работе по глазковой диаграмме, будет источником межсимвольных помех. Эквалайзер, который вам предстоит исследовать, входит в состав Modulation Toolkit.

11.2 Вводная информация

Эквалайзеры, в частности, адаптивные эквалайзеры, почти всегда реализуются в виде FIR фильтров. Это обусловлено двумя причинами. Во-первых, и это самое главное, FIR фильтр всегда устойчив. В результате, мы не должны быть обеспокоены тем, что эквалайзер может стать нестабильным в процессе адаптации. Во-вторых, иногда оказывается, что "оптимальный" эквалайзер является некаузальным. Некаузальный FIR фильтр всегда можно сделать каузальным, добавив конечную задержку. Такая простая корректировка не может быть применена к некаузальному IIR фильтру.

Как показывает опыт, зачастую канал связи можно с успехом моделировать как FIR фильтр. В условиях многолучевой окружающей среды последовательные отражения становятся слабее, и в конце концов могут стать ниже уровня шума. Как правило, лишь немногие из сильных отражений вызывают существенные межсимвольные искажения. К сожалению, инверсный FIR фильтр всегда является IIR фильтром, а это значит, что FIR эквалайзер никогда не будет в состоянии полностью устранить межсимвольные помехи, вызванные FIR фильтром канала. Мы обнаружили, что даже очень длинный FIR эквалайзер всегда оставляет небольшое количество остаточных межсимвольных искажений.

Предположим, что $\tilde{y}[k] = \tilde{y}(kT)$ представляет дискретизированный выходной сигнал согласованного фильтра приемника. Параметр T представляет собой интервал времени между отсчетами, а также время между передаваемыми символами. Вспомним из лабораторной работы, в которой рассматривалась глазковая диаграмма, что каждый отсчет $\tilde{y}[k]$ является суммой значения требуемого отсчета, межсимвольной помехи и шума. Блок-схема FIR фильтра в качестве эквалайзера представлена на рисунке 1.

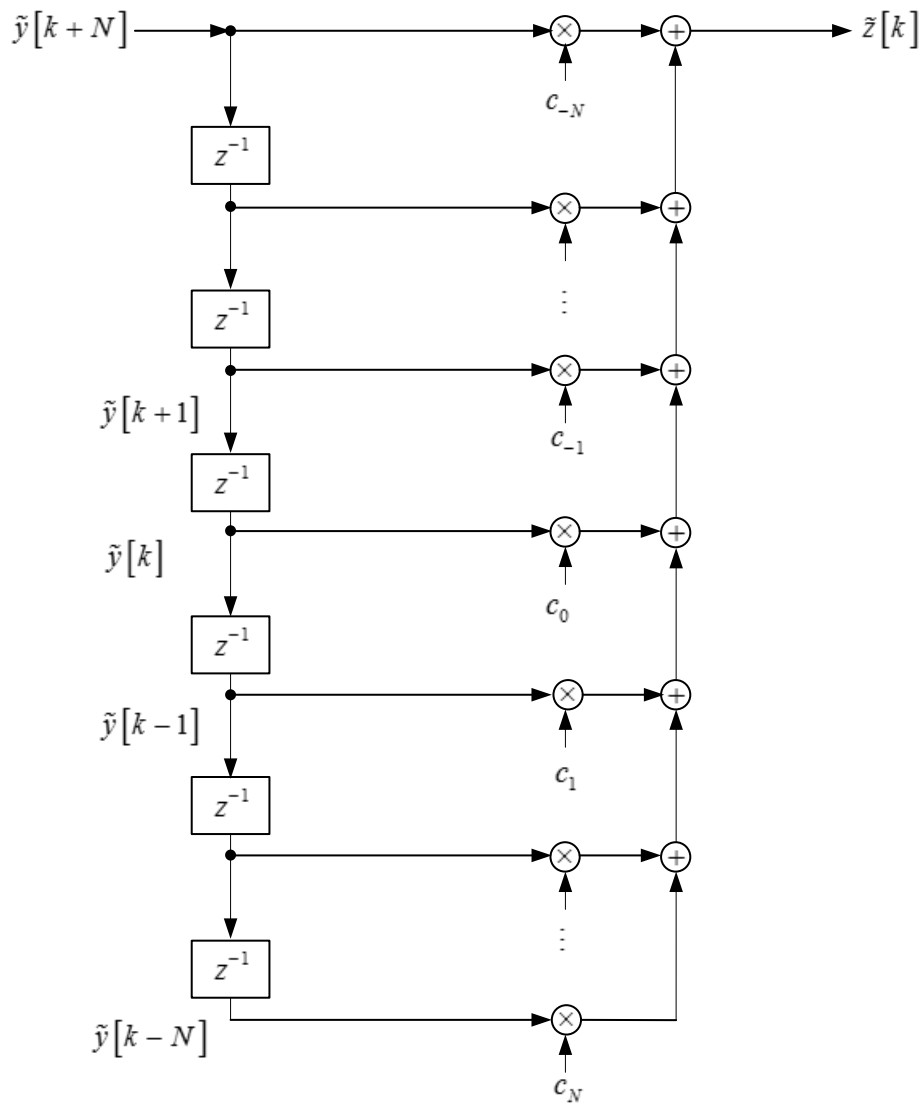


Рисунок 1. FIR эквалайзер

Выход эквалайзера обозначен $\tilde{z}[k]$. На схеме, блоки, обозначенные z^{-1} , представляют элементы задержки на один отсчет. Стрелки, ведущие к элементам умножения на коэффициенты c_n , обычно называют "отводами", а сами коэффициенты называют "весовыми коэффициентами отводов". Эта блок-схема может быть описана уравнением

$$\tilde{z}[k] = \sum_{n=-N}^N c_n y[k-n]. \quad (1)$$

Наша цель при проектировании эквалайзера – найти такие значения весовых коэффициентов c_{-N}, \dots, c_N , которые позволят минимизировать остаточные межсимвольные искажения. А в лучшем

случае, найти способ автоматической подстройки весовых коэффициентов для уменьшения искажений.

Для подстройки весовых коэффициентов эквалайзера c_{-N}, \dots, c_N , приемник должен знать, какие данные отправляет передатчик. Общепринятой практикой является начинать передачу с "обучающей выборки» $b_k, k = 0, \dots, N_t - 1$, которая известна приемнику. В данной лабораторной работе мы будем использовать 26-символьную выборку, которую уже использовали для синхронизации кадров.

Предполагается, что принятой обучающей выборки, состоящей из $N_t = 26$ символов, достаточно для того, чтобы весовые коэффициенты подстроились как можно ближе к оптимальным значениям. В результате в выборке $\tilde{z}[k]$ останутся минимальные межсимвольные искажения. После эквалайзера BPSK приемник выполняет детектирование и декодирование символов путем сравнения $\tilde{z}[k]$ с пороговым значением, равным нулю. Если \hat{b}_k представляют выходные данные, то

$$\hat{b}_k = \begin{cases} 1, & \text{if } \tilde{z}[k] \geq 0 \\ 0, & \text{if } \tilde{z}[k] < 0. \end{cases} \quad (2)$$

На основе каких данных приемник будет выполнять подстройку весовых коэффициентов эквалайзера после окончания обучающей выборки? В эквалайзере с управлением по решению вместо обучающей последовательности битов b_k используются решения \hat{b}_k . Даже когда вероятность ошибки относительно высока (например, 0.01), решения \hat{b}_k вряд ли будут неверными. Оказывается, что иногда неправильная подстройка весовых коэффициентов вызывает небольшое изменение межсимвольных искажений на выходе эквалайзера. На рисунке Рисунок 2 изображена структура эквалайзера с управлением по решению.

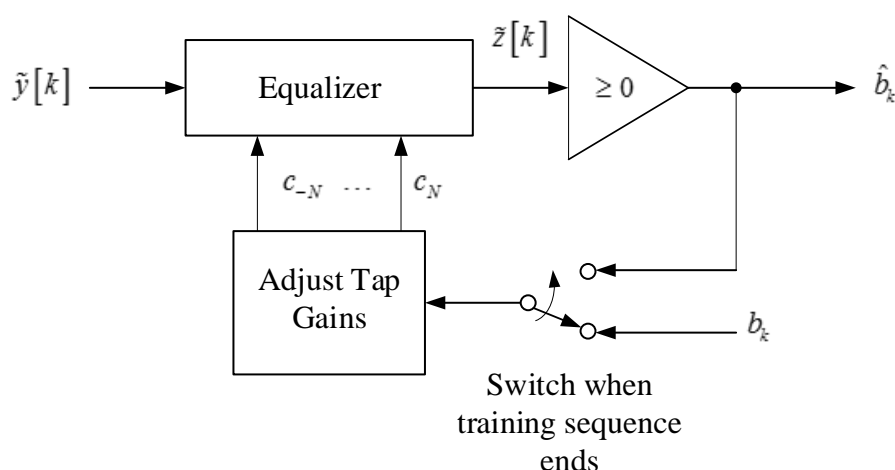


Рисунок 2. Эквалайзер с управлением по решению

Equalizer – эквалайзер, Adjust Tap Gains – подстройка весовых коэффициентов, Switch when training sequence ends – переключить по завершении обучающей выборки

Имеется несколько альтернативных критериев принятия решений для оптимальной подстройки весовых коэффициентов отводов эквалайзера. Одна из возможностей заключается в подстройке весовых коэффициентов с целью максимизации раскрытия глазка. Однако известно, что эквалайзеры имеют тенденцию усиливать шум, который является частью входной последовательности $\tilde{y}[k]$. Максимальное раскрытие глазка также приводит к увеличению шума. Альтернативный критерий, основываясь на b_k , минимизирует среднеквадратическую ошибку между последовательностью $\tilde{z}[k]$ и последовательностью символов a_k . Поскольку среднеквадратическая ошибка включает в себя межсимвольные искажения и шум, то этот критерий менее склонен к увеличению шума, чем методы, которые не учитывают шумовую составляющую. Таким образом, если a_k - последовательность символов, соответствующая обучающей последовательности битов b_k , мы можем записать среднеквадратическую ошибку E , как

$$E = E(\tilde{z}[k] - a_k)^2, \quad (3)$$

где E - оператор математического ожидания. Подставляя в уравнение (1), получаем

$$E = E\left(\sum_{n=-N}^N c_n \tilde{x}[k] - a_k\right)^2. \quad (4)$$

Если бы мы разрабатывали неадаптивный эквалайзер, мы бы искали весовые коэффициенты c_{-N}, \dots, c_N , которые минимизируют E в уравнении (4). Вместо этого мы хотим найти способ подстройки коэффициентов. Предположим, что $c_{-N}^{(q)}, \dots, c_N^{(q)}$ представляют собой значения весовых коэффициентов после q -го обновления. Далее мы можем обновлять весовые коэффициенты в соответствии с алгоритмом

$$c_n^{q+1} = c_n^q - \Delta \frac{\partial E}{\partial c_n}, \quad n = -N, \dots, N, \quad q = 0, 1, \dots \quad (5)$$

Этот алгоритм является примером использования метода наискорейшего спуска, суть которого заключается в изменении коэффициентов в направлении отрицательного градиента среднеквадратической ошибки относительно значений коэффициента. Это направление наискорейшего спуска к минимуму среднеквадратической ошибки. Параметр Δ определяет величину шага. Мы можем оценить производную $\partial E / \partial c_n$ по формуле (4):

$$\begin{aligned}
\frac{\partial E}{\partial c_n} &= E \left[2 \left(\sum_{i=-N}^N c_i \tilde{y}[k-i] - a_k \right) \tilde{y}[k-n] \right] \\
&= 2E \left[(\tilde{z}[k] - a_k) \tilde{y}[k-n] \right] \\
&= 2E \left[e[k] \tilde{y}[k-n] \right], \quad n = -N, \dots, N,
\end{aligned} \tag{6}$$

где $e[k] = \tilde{z}[k] - a_k$ ошибка сигнала (включающего шум и межсимвольные искажения) с индексом k .

Оказывается, что оператор математического ожидания в уравнении (6) представляет собой некоторую проблему, т.к. не существует способа, чтобы оценить его в фактической реализации фильтра. Мы можем избежать этой проблемы путем аппроксимации математического ожидания по текущему значению. То есть,

$$E[e[k] \tilde{y}[k-n]] \cong e[k] \tilde{y}[k-n]. \tag{7}$$

Тогда правило обновления коэффициентов принимает вид

$$c_n^{q+1} = c_n^q - \Delta e[k] \tilde{y}[k-n], \quad n = -N, \dots, N, \quad q = 0, 1, \dots \tag{8}$$

Обычно нам необходимо обновлять коэффициенты эквалайзера после каждого принятого символа. При этом индекс k и индекс q инкрементируются одновременно. Мы имеем

$$c_n^{k+1} = c_n^k - \Delta e[k] \tilde{y}[k-n], \quad n = -N, \dots, N, \quad k = 0, 1, \dots \tag{9}$$

Процедура, выраженная уравнением (9), называется алгоритмом *наименьших средних квадратов* (LMS, *least mean square*) для подстройки весовых коэффициентов эквалайзера. Мы инициализируем процедуру, задав

$$c_n = \begin{cases} 1, & n = 0 \\ 0, & \text{иначе.} \end{cases} \tag{10}$$

Блок-схема адаптивного эквалайзера на основе алгоритма наименьших средних квадратов представлена на рисунке 3.

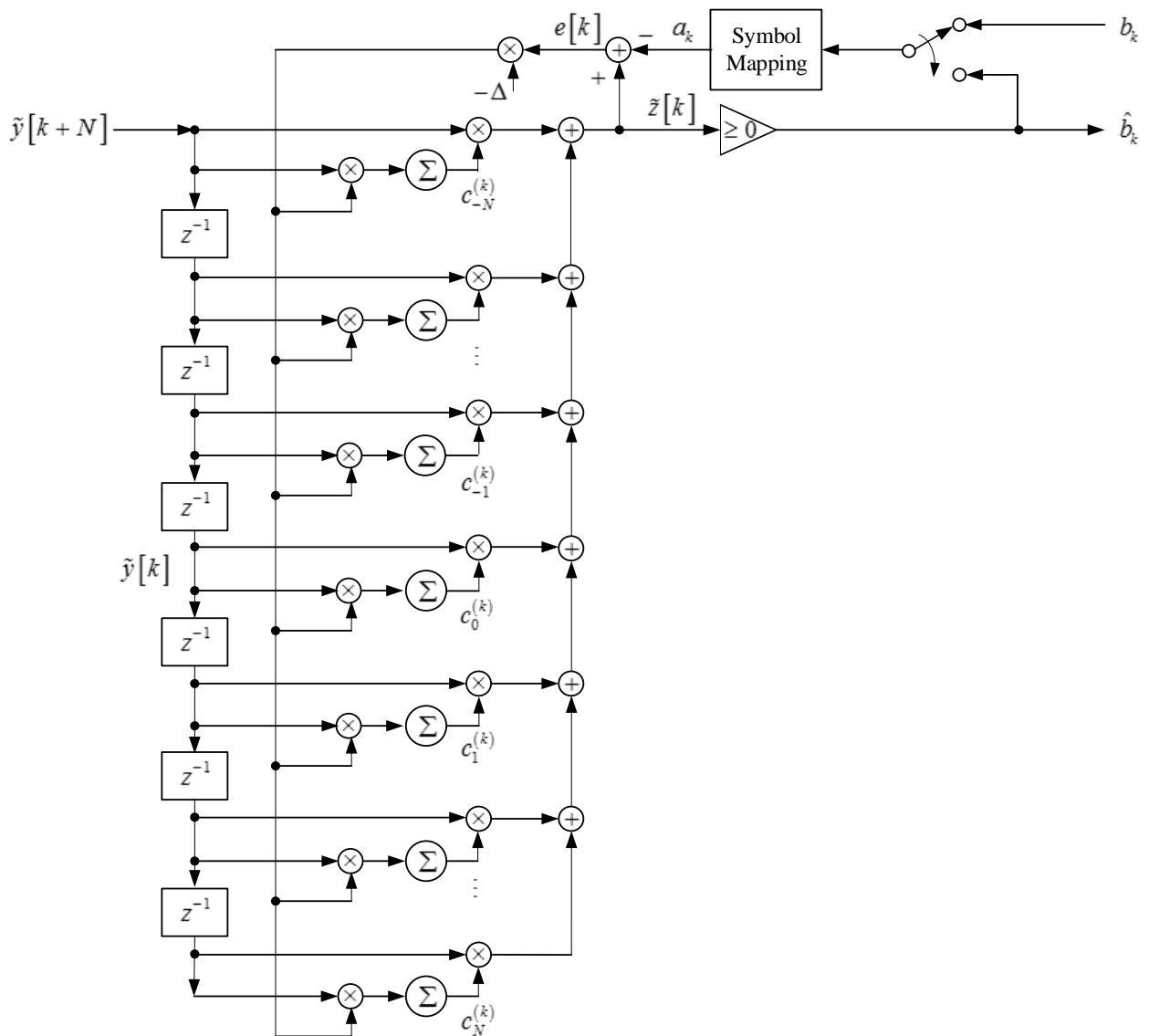


Рисунок 3. Адаптивный эквалайзер на основе алгоритма наименьших средних квадратов

Заметим что если размер шага Δ в уравнении (9) небольшой, то значения весовых коэффициентов никогда не претерпят значительных изменений за одно обновление. Это означает, что значения весовых коэффициентов в любой момент времени являются накопленной суммой небольших корректировок. Поскольку накопление является формой усреднения, уравнение (9) объясняет аппроксимацию оператора математического ожидания по одному отсчету в уравнении (7). Кроме того, поскольку весовые коэффициенты в каждый момент времени изменяются незначительно, ошибка в одном символе при использовании обратной связи по решению не выведет эквалайзер далеко из режима подстройки.

11.3 Подготовка к лабораторной работе

1. В этой лабораторной работе вы будете использовать BPSK передатчик и приемник для реализации фазовой манипуляции. Убедитесь, что в вашем приемнике есть `Channel.gvi`, содержащий модель канала, а также есть возможность отображения глазковой диаграммы. Вы можете использовать либо *MT Format Eye Diagram* из Modulation Toolkit, либо программу глазковой диаграммы, созданную в предыдущей лабораторной работе. Подключите вход "ISI Channel Model" *Channel.gvi* к элементу управления на лицевой панели вашего приемника.
2. Создайте программу, реализующую адаптивный эквалайзинг. Для начала работы вам предлагается воспользоваться шаблоном эквалайзера *EqualizerTemplate.gvi*. Входы эквалайзера должны быть подключены следующим образом:
 - a. Вход Input Complex Waveform *Equalizer.gvi* должен быть соединен с выходом Output Complex Waveform *FrameSync(real)*. Этот сигнал был выровнен по началу кадра без субдискретизации. Сигнал включает также символы заголовка, которые будут использованы при обучении эквалайзера.
 - b. На вход Receiver Sampling Factor эквалайзера должен быть подан параметр, определяющий количество отсчетов на символ, это значение рассчитывается в BPSK приемнике.
 - c. Вход Equalizer Length эквалайзера должен быть подключен к элементу управления на лицевой панели вашего приемника.

В дополнение к имеющимся входам и выходам, шаблон эквалайзера содержит копию обучающей выборки, массив, содержащий карту символов, и кластер параметров коррекции с прямой связью "feedforward equalizer parameters". Эти параметры включают количество отводов эквалайзера на символ (один), значение размера шага Δ , используемого в процессе обучения и процессе принятия решения (0.05 при обучении, 0.001 в режиме принятия решения).

В завершении вам необходимо добавить две функции из Modulation Toolkit: функцию *MT Generate System Parameters* из субпалитры Analysis→Communications→Digital→Utilities и функцию *MT PSK Feedforward Equalizer* из субпалитры Analysis→Communications→Digital→Equalization.

Щелкните по функции MT PSK Feedforward Equalizer, затем на ленте Configure выберите PSK(M) для настройки параметров *MT Generate System Parameters*. Создайте константу для входа "PSK type" и выберите "Normal". Подключите "Receiver Sampling Factor" ко входу "samples per symbol". Подключите константу со значением 2 ко входу "M-PSK". Единственный выход *MT Generate System Parameters*, который вы будете использовать – это кластер "PSK system parameters". Подключите его к функции *Cluster Properties*, имеющейся в шаблоне для замены карты символов, используемой по умолчанию, на нужную карту символов. Модифицированный кластер параметров PSK системы будет использован в *MT PSK Feedforward Equalizer*.

На ленте Configure выберите Specify Length для *MT PSK Feedforward Equalizer.gvi*. Подключите входы "input complex waveform" и "PSK system parameters". Подключите вход "equalizer length" к соответствующему элементу управления. Подключите вход "training bits" к массиву битов обучающей выборки, доступной в шаблоне. Подключите "feedforward equalizer (LMS) parameters" к кластеру, в предоставленном шаблоне. Вход "reset" можно оставить неподключенным, т.к. его значение по умолчанию "true" - правильное. Подключите все выходы к соответствующим индикаторам и можно приступать к работе.

3. Сохраните ваш эквалайзер в файл, имя которого включает в себя слово "Equalizer" и ваши инициалы (например, Equalizer_BAB.gvi).
4. Откройте блок-диаграмму вашего BPSK приемника. Подключите оставшиеся входы *FrameSync(real)*, а именно, подключите вход "Input Signal" к выходу "Aligned Signal" *PulseAlign(real)* и вход "Receiver Sampling Factor" к соответствующему элементу. Вход "Sampled Input" *FrameSync(real)* должен оставаться подключенным к выходу *Decimate*.
5. Добавьте ваш эквалайзер к BPSK приемнику. Подключите входы, как описано в пункте 2 выше. В течение лабораторной работы вам будет предложено получить данные для сравнения глазковых диаграмм до и после эквалайзинга. Подключите выход "Equalized Complex Waveform" эквалайзера ко второму индикатору глазковой диаграммы. Подключите также выход "Squared Error" к индикатору "waveform graph" так, чтобы иметь возможность наблюдать уменьшение ошибки при подстройке эквалайзера. Подключите выходы "Output Bits" и "equalizer coefficients out" к индикаторам.

Вопросы

1. При задержке распространения сигнала в 100 мкс импульсная характеристика по умолчанию, реализуемая *Channel.gvi*, равна $h[n] = \delta[n] + 0.2\delta[n-1] - 0.08\delta[n-2]$. z-преобразование позволяет получить характеристику канала $H(z) = 1 + 0.2z^{-1} - 0.08z^{-2}$. Характеристика идеального эквалайзера должна быть обратна характеристике канала. То есть,

$$H_{eq}(z) = \frac{1}{1 + 0.2z^{-1} - 0.08z^{-2}} \quad (11)$$

Найдите выражение для идеальной импульсной характеристики эквалайзера $h_{eq}[n]$.

Коэффициенты $h_{eq}[n]$ являются весовыми коэффициентами идеального эквалайзера (т.е. бесконечной длины). Найдите численные значения для первых пяти весовых коэффициентов c_0, \dots, c_4 .

2. Вы должны определить, что идеальный эквалайзер в вопросе 1 имеет каузальную импульсную характеристику. При необходимости эквалайзер на рисунке 3 может быть некаузальным. Ответьте на вопрос 1 для $h[n] = 0.1\delta[n] + \delta[n-1] + 0.3\delta[n-2]$, если эквалайзер некаузальный. Найдите

численные значения C_{-4}, \dots, C_4 . (Подсказка: Идеальный эквалайзер является IIR фильтром. Если эквалайзер некаузальный, он должен быть устойчивым).

11.4 Порядок выполнения лабораторной работы

1. Подключите тестовый кабель и аттенюатор между разъемами TX 1 и RX 2 USRP. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте передатчик, созданный вами при подготовке к выполнению лабораторной работы.

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

IQ частота (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_x$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): TX1

Скорость передачи символов (Symbol rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр формирования импульса (Pulse shaping filter): Root Raised

Запустите передатчик. Используйте большую кнопку STOP на лицевой панели для отключения разъемов передатчика.

3. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

IQ частота (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_z$. Обратите внимание, что значение T_z совпадает со значением dt .

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): RX2

Скорость передачи символов (Symbol rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр формирования импульса: Root Raised

Использование канала: off

4. Запустите сначала передатчик, затем приемник. После того, как приемник получит данные, вы можете завершить работу передатчика. Приемник должен показать значение BER равное 0.0 или 1.0. В этой лабораторной работе вас не должно беспокоить значение BER, равное 1.0.

5. Задайте в модели канала значение по умолчанию 100 мкс для задержки распространения сигнала. Установите Use Channel в состояние "on". Задайте длину эквалайзера "Equalizer Length" равной 11. Запустите сначала передатчик, затем приемник. После того, как приемник получит данные, вы можете завершить работу передатчика. Сравните глазковые диаграммы до и после эквалайзинга. Используйте курсоры для измерения V_w и V_b и рассчитайте раскрытие глазка для каждого случая.

6. Повторите действия по пункту 5 для задержки распространения сигнала со значениями 50 мкс и 100 мкс. Сравните раскрытие глазка диаграммы до и после эквалайзинга для каждого случая. Подготовьте таблицу, отражающую раскрытие глазков до и после эквалайзинга для каждой из трех задержек распространения.

7. Задайте значение задержки распространения сигнала в канале 100 мкс и запустите передатчик и приемник. Наблюдайте значения на выходе эквалайзера "Squared Error". Какое приблизительное значение квадратичной ошибки получено в установившемся режиме?

8. Запишите значения весовых коэффициентов отводов c_{-4}, \dots, c_4 при задержке распространения сигнала 100 мкс и длине эквалайзера 11. Сравните полученные значения со значениями, вычисленными при ответе на вопрос 1 в разделе "Подготовка к лабораторной работе". (Не ожидайте очень близкого совпадения. При подготовке к работе расчеты выполнялись для бесконечной длины эквалайзера. Кроме того, весовые коэффициенты адаптивного эквалайзера постоянно подстраиваются, изменяясь вблизи оптимальных значений. Попробуйте запустить приемник несколько раз, чтобы увидеть, насколько велики изменения весовых коэффициентов относительно установившихся значений).

9. Измените модель канала с межсимвольными искажениями на модель, полученную при ответе на вопрос 2 из раздела "Подготовка к лабораторной работе". Оставьте без изменения значения задержки распространения сигнала 100 мкс и длины эквалайзера равной 11. Запустите передатчик и приемник. Проверьте глазковые диаграммы, чтоб удостовериться в работоспособности эквалайзера. Запишите значения весовых коэффициентов c_{-4}, \dots, c_4 . Сравните полученные значения со значениями, вычисленными при ответе на вопрос 2 из раздела "Подготовка к лабораторной работе".
10. При значении задержки распространения сигнала равной 100 мкс измерьте раскрытие глазка после эквалайзинга, используя любую из двух моделей канала и увеличивая длину эквалайзера с шагом, равным нечетным числам, начиная с 1. При какой длине эквалайзера раскрытие глазка стабилизируется?
11. Еще раз о синхронизации фазы. Вы, возможно, помните из лабораторной работы по BPSK, что любая разность фаз ϕ между генераторами передатчика и приемника порождает множитель $\cos(\phi)$ в демодулированном сигнале. Этот множитель влияет на амплитуду и полярность демодулированного сигнала. Оказывается, что эквалайзер может устранить фазовую ошибку так же, как и искажения в канале. Чтобы увидеть, как это происходит, внесите следующие изменения в приемник:
- Функция *MT PSK Feedforward Equalizer* включает в себя компаратор порога и идентификатор символа. Выходные биты эквалайзера доступны на выходе *Output Bits Equalizer.gvi*. В вашем приемнике необходима функция *Array Subset*, следующая за компаратором порога и позволяющая ограничить последовательность принятых битов в соответствии с длиной кадра. Замените проводник от компаратора порога на проводник от выхода *Output Bits* эквалайзера к функции *Array Subset*.
 - Последовательность битов с выхода эквалайзера включает в себя 26-битный заголовок кадра. Чтобы удалить этот заголовок, подайте на вход *index* функции *Array Subset* 26 вместо 0.

Запустите передатчик и приемник. Вы должны получить вероятность ошибок BER, равную 0. Попробуйте запустить приемник, устанавливая *Use Channel* в состояния *on* и *off*. Вы должны получать вероятность BER равную 0 в обоих случаях, тем самым доказывая, что фазовая ошибка, вызванная алгоритмом коррекции фазы, удалена.

- с. Полностью удалите элементы синхронизации фаз. На Рисунок 4 показан целесообразный способ сделать это.

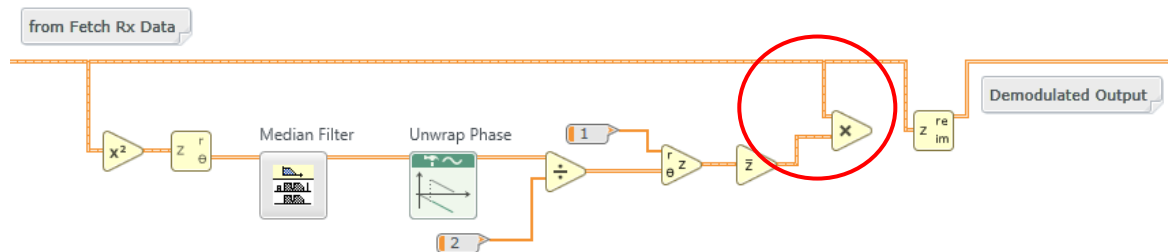


Рисунок 4. Удаление синхронизации фаз

Запустите передатчик и приемник. Use Channel может быть в состоянии on или off. Вероятность ошибок BER должна быть равна нулю в каждом случае.

Вопросы

1. Ответьте на вопросы, приведенные в пунктах 6-10 раздела "Порядок выполнения лабораторной работы".

11.5 Отчет

Подготовка к лабораторной работе

Предоставьте документацию для вашей программы эквалайзера и вашего модифицированного приемника, который включает эквалайзер и вторую глазковую диаграмму. Кроме того, включите в отчет документацию для всех созданных вами функций. При подготовке документации используйте разборчивые скриншоты лицевой панели и блок-диаграммы.

Ответьте на все вопросы раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Предоставьте программы эквалайзера и приемника. Предоставьте также созданные вами функции. Убедитесь, что в ваших файлах соблюдается соглашение об именовании, описанное в инструкциях выше.

Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы.

Ответьте на вопросы, приведенные в пунктах 6-10 раздела "Порядок выполнения лабораторной работы". Прокомментируйте ваши достижения при выполнении пункта 11 с использованием эквалайзера с синхронизацией фаз.

Квадратурная фазовая манипуляция

12.1 Цель

Все системы модуляции, рассматриваемые до сих пор, выполняли кодирование одного бита данных в один символ. Для приложений, требующих высоких скоростей передачи данных при ограниченной полосе пропускания канала, рекомендуется в одном символе кодировать несколько бит данных. В этой лабораторной работе мы рассмотрим квадратурную фазовую манипуляцию (QPSK, quadrature phase-shift keying), вариант двоичной фазовой манипуляции, которая кодирует два бита в каждом символе. Используя QPSK, мы сможем передавать данные с удвоенной скоростью без расширения пропускной способности канала. Однако это приведет к некоторой потере производительности, а также потребует большей мощности передатчика для поддержания заданной вероятности битовой ошибки при наличии шума.

В данной лабораторной работе рассматриваются новые принципы кодирования сигналов, более сложные, чем кодирование, используемое в BPSK. В этой работе будет также рассмотрено сигнальное созвездие, которое является инструментом визуализации, очень полезным для понимания высокоэффективных методов модуляции.

QPSK просто расширяется на методы модуляции, которые позволяют передавать даже более двух бит в одном символе. Одним из таких методов является "квадратурная амплитудная модуляция" (QAM, quadrature amplitude modulation). Например, 16-QAM может передавать 4 бита данных в одном символе. Однако, большее количество бит, передаваемых в одном символе, требует большей мощности передатчика для поддержания заданной вероятности битовой ошибки.

12.2 Вводная информация

В лабораторной работе, в которой рассматривалась BPSK, PSK сигнал представлял собой последовательность импульсов следующей формы

$$x(t) = Ag_{TX}(t)\cos(2\pi f_c t + \theta), \quad (1)$$

где A - константа, $g_{TX}(t)$ - импульс определенной формы, предназначенный для ограничения полосы частот сигнала и управления межсимвольными искажениями, f_c - частота несущей.

Информация передается фазой θ . В BPSK достаточно было взять два значения фазы θ , при которых каждый импульс содержал один бит информации. Для QPSK необходимо взять четыре значения фазы θ . Поскольку для различения одного из четырех состояний необходимо два бита, каждый импульс QPSK несет в себе два бита информации. Мы будем использовать четыре возможных фазовых угла $\theta = \pm\pi/4, \pm3\pi/4$.

Другой способ представления импульсов дает уравнение разложения по косинусу (1). Можно записать

$$x(t) = A \cos(\theta) g_{TX}(t) \cos(2\pi f_c t) - A \sin(\theta) g_{TX}(t) \sin(2\pi f_c t). \quad (2)$$

Подстановка четырех возможных значений θ дает

$$x(t) = \pm (A/\sqrt{2}) g_{TX}(t) \cos(2\pi f_c t) \mp (A/\sqrt{2}) g_{TX}(t) \sin(2\pi f_c t). \quad (3)$$

Уравнение (3) можно интерпретировать, предположив, что один бит данных определяет полярность $(A/\sqrt{2}) g_{TX}(t) \cos(2\pi f_c t)$ (синфазный терм), в то время как второй бит данных определяет полярность $(A/\sqrt{2}) g_{TX}(t) \sin(2\pi f_c t)$ (квадратурный терм).

Чтобы создать передаваемый импульс, представленный уравнениями (1) или (3), мы должны передать в USRP импульс в комплексной форме

$$\tilde{x}(t) = A g_{TX}(t) e^{j\theta}, \quad \theta = -3\pi/4, -\pi/4, \pi/4, 3\pi/4. \quad (4)$$

В этом случае символ является комплексным числом $Ae^{j\theta}$. USRP будет генерировать передаваемый сигнал, выраженный уравнением (1):

$$x(t) = \text{Re}[\tilde{x}(t) e^{j2\pi f_c t}] = A g_{TX}(t) \cos(2\pi f_c t + \theta). \quad (5)$$

Удобный способ визуализации сигнала в комплексной форме, представленной формулой (4), - нарисовать возможные значения символов $Ae^{j\theta}$ на комплексной плоскости. Сигнальное созвездие QPSK показано на Рисунке 1.

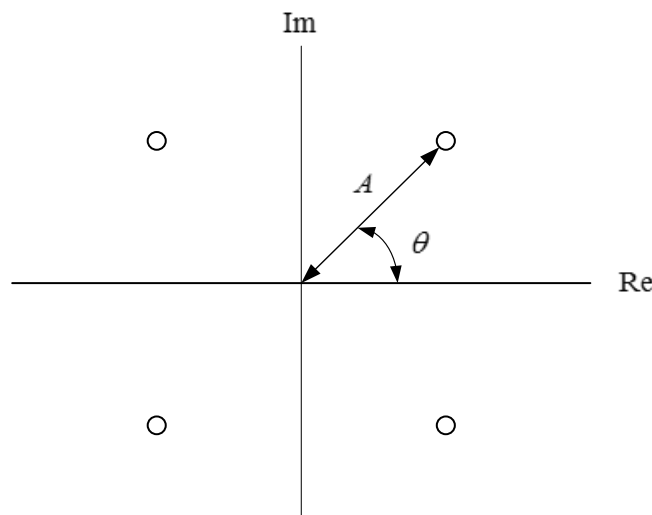


Рисунок 1. Сигнальное созвездие QPSK

Из сигнального созвездия легко увидеть, что существуют четыре возможных передаваемых сигнала (отсюда два бита на символ), каждый из возможных символов имеет одну и ту же амплитуду A , и возможны четыре фазовых угла $\theta = \pm\pi/4, \pm3\pi/4$.

Давайте проверим связь между парой входных бит и точками на сигнальном созвездии (т.е. значениями символа). Отображение пары бит точкой на сигнальном созвездии фактически произвольно, при условии, что передатчик и приемник используют одно и то же кодирование. Если один бит ассоциируется с полярностью на действительной части комплексной плоскости (синфазный) и один бит – с полярностью на мнимой части (квадратурный), мы получим созвездие, изображенное на рисунке 2. Как показано на рисунке, бит справа определяют полярность действительной части, в то время как бит слева определяют полярность мнимой части. Это будет удобно при последующей интерпретации пары бит двоичными числами. При этом левый бит будем интерпретировать, как младший значащий бит, а правый бит – как старший значащий бит. Подобная интерпретация может показаться интуитивно противоречивой, но эти последовательности бит будут храниться в виде массивов, и естественно записывать массивы с увеличением индекса вправо. Символьное кодирование может быть также представлено таблицей. В таблице 1 символы закодированы в соответствии с рисунком 2.

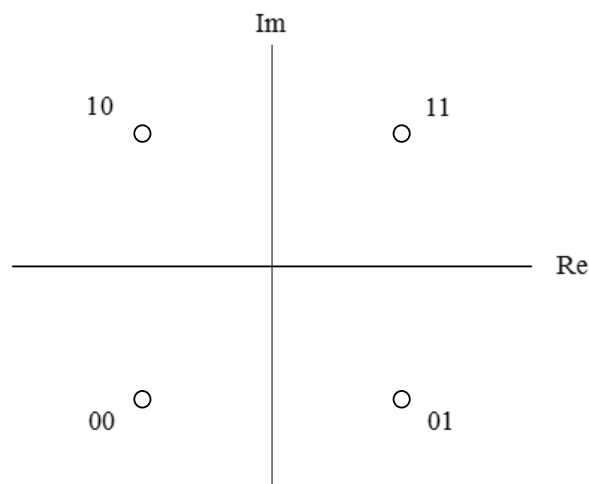


Рисунок 2. Вариант символьного кодирования

Таблица 1. Кодирование символов

Номер индекса	Комбинация бит	Значение символа
0	00	$Ae^{-j3\pi/4} = -A/\sqrt{2} - jA/\sqrt{2}$
1	10	$Ae^{j3\pi/4} = -A/\sqrt{2} + jA/\sqrt{2}$
2	01	$Ae^{-j\pi/4} = A/\sqrt{2} - jA/\sqrt{2}$
3	11	$Ae^{j\pi/4} = A/\sqrt{2} + jA/\sqrt{2}$

Приемник QPSK начинается с DSB-SC демодулятора. Когда передаваемый QPSK сигнал поступает на приемник, он имеет форму последовательности импульсов, заданных формулой

$$r(t) = Dg_{TX}(t) \cos(2\pi f_c t + \theta + \phi), \quad (6)$$

где D константа (обычно много меньше константы A в передаваемом сигнале), угол θ несет в себе информацию, а угол ϕ представляет собой разность фаз между генераторами несущей передатчика и приемника. Если частота несущей приемника задана равной частоте несущей передатчика, то приемник USRP будет выполнять большую часть операций по демодуляции QPSK сигнала. *Fetch Rx Data* приемника обеспечит последовательность выходных импульсов, каждый из которых задается¹⁰

$$\tilde{r}(t) = \frac{D}{2} g_{TX}(t) e^{j(\theta+\phi)}. \quad (7)$$

Последовательность импульсов, представленных уравнением (7) пропускается через согласованный фильтр с импульсной характеристикой $g_{RX}(t)$. Если записать $g(t) = g_{TX}(t) * g_{RX}(t)$, то выходной сигнал согласованного фильтра может быть представлен

$$\tilde{y}(t) = \frac{D}{2} g(t) e^{j(\theta+\phi)}. \quad (8)$$

Выход согласованного фильтра затем дискретизируется по одному отсчету на символ. В идеальном случае $g(0) = 1$, и на выходе дискретизатора получаем последовательность комплексных чисел вида

¹⁰ Все сигналы представлены в LabVIEW, как сигналы, дискретизированные во времени. Представление их, как сигналы, непрерывные во времени, делает описание более удобным для чтения.

$$\tilde{y}[k] = \frac{D}{2} e^{j(\theta_k + \phi)}, \quad (9)$$

где θ_k значение фазы в момент выборки $t = kT$, а D и ϕ являются константами. На практике отсчеты $\tilde{y}[k]$ будут искажены из-за межсимвольных помех и шума. Как и в предыдущей лабораторной работе, межсимвольные искажения устраняются фильтром эквалайзера. Мы также видели, что использование эквалайзера позволяет устранить фазовую ошибку ϕ , и в результате получить

$$\tilde{y}[k] = \frac{D}{2} e^{j\theta_k} + \text{residual ISI} + \text{noise}. \quad (10)$$

Следующие операции, которые должны быть выполнены приемником, это детектирование и декодирование сигнала. Детектирование сигнала представляет собой процесс проверки каждого отсчета $\tilde{y}[k]$ и определения, какому переданному символу он вероятнее всего соответствует. Оказывается, что алгоритм детектирования, называемый *правилом принятия решения*, на удивление прост и имеет изящную геометрическую формулировку. На Рисунке 3 показано сигнальное созвездие с отсчетом $\tilde{y}[k]$. Вследствие остаточных межсимвольных искажений и шума $\tilde{y}[k]$ не может в точности соответствовать какому-либо из возможных значений символов.

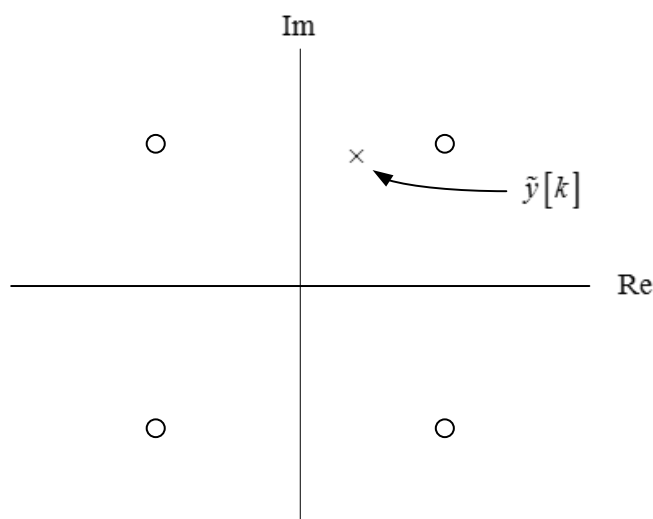


Рисунок 3. Сигнальное созвездие и полученный отсчет

Правило принятия решения для определения наиболее вероятного переданного символа $\tilde{y}[k]$ предполагает следующее: вычисление расстояния между $\tilde{y}[k]$ и каждым возможным символом.

Наименьшее расстояние от $\tilde{y}[k]$ определяет наиболее вероятный символ. На Рисунок 3 один наиболее вероятный символ с наименьшим расстоянием от $\tilde{y}[k]$ расположен в первом квадранте. После того, как символ определен, для преобразования символа в битовую последовательность

Таблица 1 можно воспользоваться таблицей кодирования 1. В примере на Рисунок 3 полученный отсчет соответствует комбинации бит 11.

12.3 Подготовка к лабораторной работе

1. Создайте функцию для кодирования символов в передатчике. (Детектирование и декодирование в приемнике будет осуществляться *MT PSK Feedforward Equalizer*). В таблице 2 приведены характеристики входных и выходных параметров вашей функции.

Таблица 2. Характеристики для кодирования данных

Входы			Выход		
Имя	Тип		Имя	Тип	
Data In (Входные данные)	Одномерный массив 8-ми битных целочисленных значений	Поток битовых данных	Symbols Out (Выходные символы)	Одномерный массив	Передаваемый поток символов
Symbol Map (Таблица кодирования символов)	Одномерный массив комплексных чисел двойной точности	Символы, индексированные парой бит, представлены в виде чисел			

Ваша функция должна соответствовать кодировке символов из

Таблица 1 с $A = 1$. В шаблоне *MapDataTemplate.gvi* входы и выходы уже подключены. Массив, содержащий коды символов, необходим вам в качестве одного из входов и доступен в шаблоне *QPSK Tx.gvi*, который описан ниже.

Сохраните вашу функцию кодирования символов в файл, имя которого включает в себя "MapData" и ваши инициалы (например, *MapData_BAB.gvi*).

2. Создайте программу, реализующую QPSK передатчик. Для начала работы вам предлагается воспользоваться шаблоном *QPSKTxTemplate.gvi*. Этот шаблон включает входы и выходы, массив кодов символов, функции для отображения сигнала и спектра мощности, а также функции, необходимые для интерфейса с USRP.

Вам нужно будет добавить

- функцию кодирования символов по пункту 1, описанному выше,
- функцию *AddFrameHeader(Complex)*
- передискретизацию
- формирование импульса, сглаженных по закону косинуса
- масштабирование амплитуды

Подсказка: Воспользуйтесь в качестве модели вашим BPSK передатчиком. QPSK передатчик отличается только методом кодирования символов.

Сохраните ваш приемник в файл, имя которого включает в себя "QPSKTx" и ваши инициалы (например, *QPSKTx_BAB.gvi*).

5. Создайте функцию эквалайзера для QPSK. Для этого вы можете скопировать функцию BPSK эквалайзера и выполнить следующие изменения:
 - Измените значение входа "M-PSK" *MT Generate System Parameters* с 2 на 4.
 - Замените таблицу кодирования символов путем копирования таблицы кодирования символов QPSK из шаблона *QPSKTxTemplate.gvi*.
 - Замените массив констант обучающей выборки на новую обучающую последовательность, в которой продублирован каждый бит. На Рисунок 4 показан простой способ дублирования с помощью функции *Interleave 1D Arrays*.

Сохраните ваш эквалайзер в файл, имя которого включает в себя слово "QPSK_Equalizer" и ваши инициалы (например, *QPSK_Equalizer_BAB.gvi*).

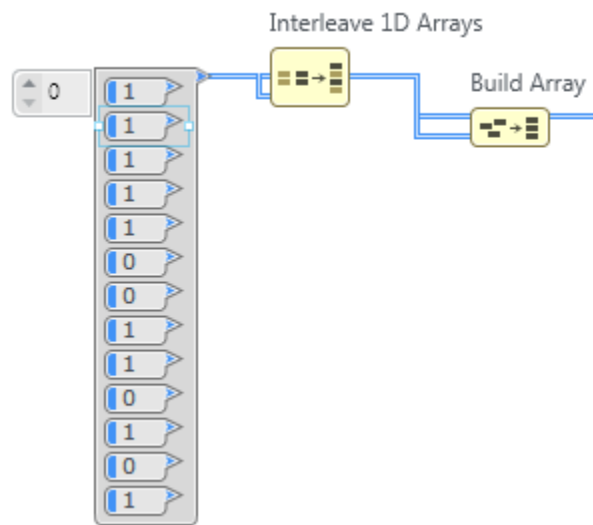


Рисунок 4. Изменение обучающей выборки

- Создайте программу, реализующую QPSK приемник. Для начала работы вам предлагается воспользоваться шаблоном QPSKRxTemplate.gvi. Шаблон включает в себя входы и выходы, функции, необходимые для взаимодействия с USRP и несколько функций для настройки индикаторов.

Подсказка: Воспользуйтесь в качестве модели вашим BPSK приемником. QPSK приемник очень похож.

Вам необходимо добавить

- *Channel.gvi*.
- Функцию *Cluster Properties* после *Channel.gvi*, чтобы получить доступ к компоненту "Y", массиву отсчетов в виде комплексных чисел двойной точности.
- Согласованный фильтр с характеристикой по закону приподнятого косинуса и функцию *Convolution* для реализации фильтра.
Примечание: Там нет синхронизатора фаз, так как эта функция будет выполняться эквалайзером. Кроме того, нет необходимости использовать действительную часть полученных значений выборок.
- Функцию *Cluster Properties* для создания кластера комплексных чисел двойной точности, получаемых с выхода функции *Convolution*.
- *PulseAlign (Complex)*.
- Функцию *Cluster Properties* после *PulseAlign(Complex)*, чтобы получить доступ к элементу "Y". Добавьте *Decimate (single shot)* для получения выборки выравненных импульсов.
- *FrameSync(Complex)*.
- Ваш эквалайзер, реализованный при выполнении пункта 3, потребуется позже. Выход "Output Signal" *FrameSync(Complex)* является входом эквалайзера. Не используйте выход "Aligned Samples" *FrameSync(Complex)*.
- Подключите выход эквалайзера "Output Bits" к функции *Array Subset*. Установите значение "index" равное 52 для того, чтобы удалить обучающую последовательность, и задайте значение "length" соответствующим количеству бит в сообщении. Выходные значения

функции *Array Subset* могут быть отображены как "Output Bits" приемника, а также могут быть переданы в *MT Calculate BER*.

- В шаблоне есть две функции *MT Format Eye Diagram*. Подключите выход "Output Signal" *PulseAlign(Complex)* к одной из этих функций. Это позволит визуализировать глазковую диаграмму синфазного компонента принимаемого сигнала прежде, чем будет выполнен эквалайзинг. Подключите выход эквалайзера "Output Complex Waveform" ко второй глазковой диаграмме. Это позволит визуализировать глазковую диаграмму синфазного элемента принимаемого сигнала после эквалайзинга.

Подключите также выход эквалайзера "Output Complex Waveform" ко входу "waveform" *MT Format Constellation*.

Сохраните ваш приемник в файл, имя которого включает в себя "QPSKRx" и ваши инициалы (например, *QPSKRx_BAB.gvi*).

Вопросы

1. Скорость передачи символов вашего передатчика и приемника равна 10 000 символов/с. Какова скорость передачи данных в бит/с?
2. Если приемник настроен на частоту IQ 200 кГц, сколько отсчетов на символ будет использоваться в приемнике?
3. *Спектральная эффективность* модулированного сигнала это отношение скорости передачи данных в бит/с к ширине спектра передаваемого сигнала. Вычислите спектральную эффективность BPSK сигнала, использующего импульсы, сглаженные по закону косинуса, которые вы сгенерировали в лабораторной работе по BPSK. Сравните со спектральной эффективностью сигнала QPSK, при условии, что ширина спектра сигнала остается той же самой.

12.4. Порядок выполнения лабораторной работы

1. Подсоедините тестовый кабель и аттенюатор между разъемами TX 1 и RX 2 USRP. Подключите USRP к вашему компьютеру и включите питание. Запустите LabVIEW и откройте передатчик, созданный вами при подготовке к выполнению лабораторной работы.

2. Убедитесь, что для передатчика заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

IQ частота (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_x$.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): TX1

Скорость передачи символов (Symbol rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): Root Raised

3. Запустите передатчик. Используйте большую кнопку STOP на лицевой панели для отключения разъемов передатчика.

4. Используя индикатор спектра мощности на лицевой панели передатчика, измерьте ширину спектра комплексного сигнала.

5. Убедитесь, что для приемника заданы следующие настройки:

Несущая частота (Carrier Frequency): 915.0 МГц

IQ частота (IQ Rate): 200 кГц Примечание: Это задает значение $1/T_z$. Обратите внимание, что параметры T_z и dt совпадают.

Коэффициент усиления (Gain): 0 дБ

Активная антенна (Active Antenna): RX2

Скорость передачи символов (Symbol rate): 10 000 символов/с

Длина сообщения (Message Length): 1000 бит

Фильтр, формирующий импульсы (Pulse shaping filter): Root Raised

Использовать канал (Use Channel): off

6. Запустите сначала передатчик, затем приемник. После того, как приемник получит данные, вы можете завершить работу передатчика. Приемник должен показать значение BER, равное 0.0.

7. Сравните глазковые диаграммы до и после эквалайзинга. Запустите приемник несколько раз, пока вы не определите, что изменяется, а что не изменяется от запуска к запуску. Иногда на глазковой диаграмме, полученной до эквалайзинга, будут отображаться не два отсчета, а четыре. Сделайте скриншоты глазковой диаграммы до и после эквалайзинга.

8. Установите в модели канала по умолчанию элемент управления Use Channel в состояние "on", а значение задержки распространения сигнала задайте равную 100 мкс. Запустите передатчик, а затем несколько раз запустите приемник. Сравните глазковые диаграммы до и после эквалайзинга. Измерьте раскрытие глазка до и после эквалайзинга. Попытайтесь получить наилучший результат измерений для случая "до эквалайзинга".

9. Установите элемент управления Use Channel в состояние "off", и снова запустите передатчик и приемник. В этот раз наблюдайте результат по индикатору сигнального созвездия. Сделайте скриншот созвездия.

10. Вместо эквалайзера подключите выход "Output Signal" *FrameSync(Complex)* к *MT Format Constellation*. Запустите передатчик, а затем несколько раз запустите приемник. Как изменилось значение угла на сигнальном созвездии? Как изменился радиус на сигнальном созвездии?

Вопросы

1. Какова измеренная ширина спектра переданного QPSK сигнала? (Примечание: не ширина спектра модулирующего сигнала). Сравните с шириной спектра BPSK сигнала, которую вы измерили в предыдущей лабораторной работе.
2. Зафиксируйте результаты измерения раскрытия глазка, выполненные в пункте 6.
3. Что явилось причиной поворота сигнального созвездия, который вы наблюдали при выполнении пункта 8? Как выглядит сигнальное созвездие, когда на глазковой диаграмме, полученной до эквалайзинга, отображаются четыре отсчета вместо 2? Объясните, почему это происходит с глазковой диаграммой. Подсказка: Запомните, что график глазковой диаграммы отображает только синфазную компоненту сигнала.
4. Как будет выглядеть сигнальное созвездие, если несущие частоты передатчика и приемника немного отличаются?
6. Какое вы ожидаете изменение вероятности ошибок BER при вращении сигнального созвездия? (Предположим, что эквалайзер отсутствует).
7. Понаблюдайте за переходными процессами на графическом индикаторе сигнального созвездия. Обратите внимание, что переходы сигналов между первым и третьим квадрантами проходят через начало координат графика, как и переходы сигналов между вторым и четвертым квадрантами. Объясните, что происходит с амплитудой принятого сигнала при этих переходах.

12.5 Отчет

Подготовка к лабораторной работе

Подготовьте документацию для созданных вами четырех функций: кодирования символов, передатчика, эквалайзера и приемника. Кроме того, включите в отчет документацию для других созданных вами функций. Для получения документации распечатайте разборчивые скриншоты лицевой панели и блок-диаграммы.

Ответьте на все вопросы раздела "Подготовка к лабораторной работе".

Выполнение лабораторной работы

Включите в отчет программы кодирования символов, передатчика, эквалайзера и приемника. Повторно предоставьте документацию для любых функций, которые вы модифицировали при выполнении лабораторной работы.

Включите в отчет графики, которые получены при выполнении пунктов 5 и 7.

Ответьте на вопросы из раздела "Порядок выполнения лабораторной работы".