

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 символы 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.

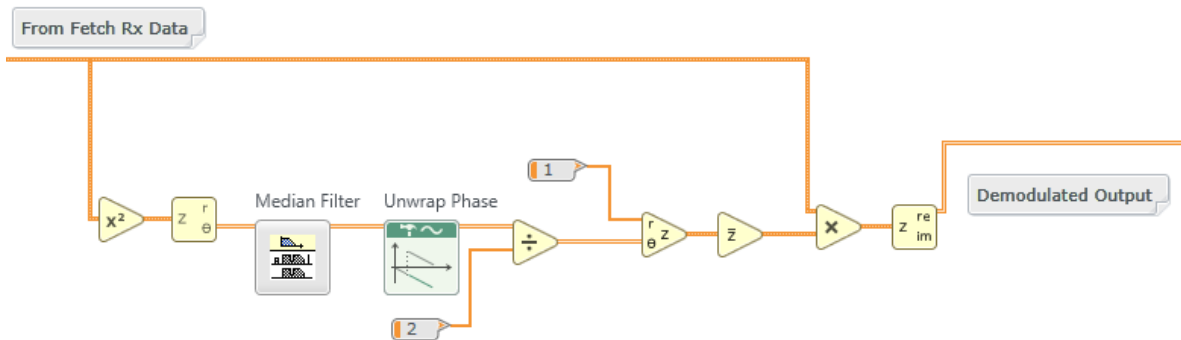


Рисунок 1. Синхронизация фазы несущей

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.

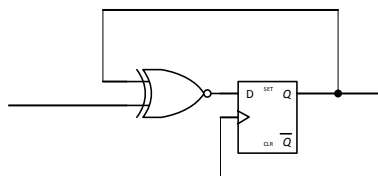


Рисунок 2. Дифференциальный кодер

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

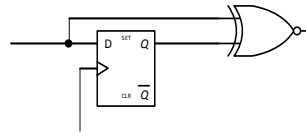


Рисунок 3. Дифференциальный декодер

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.

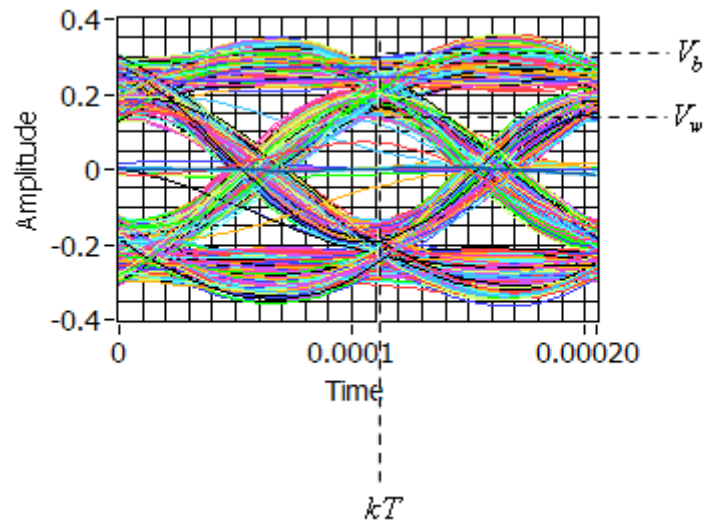


Рисунок 1. Глазковая диаграмма

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.

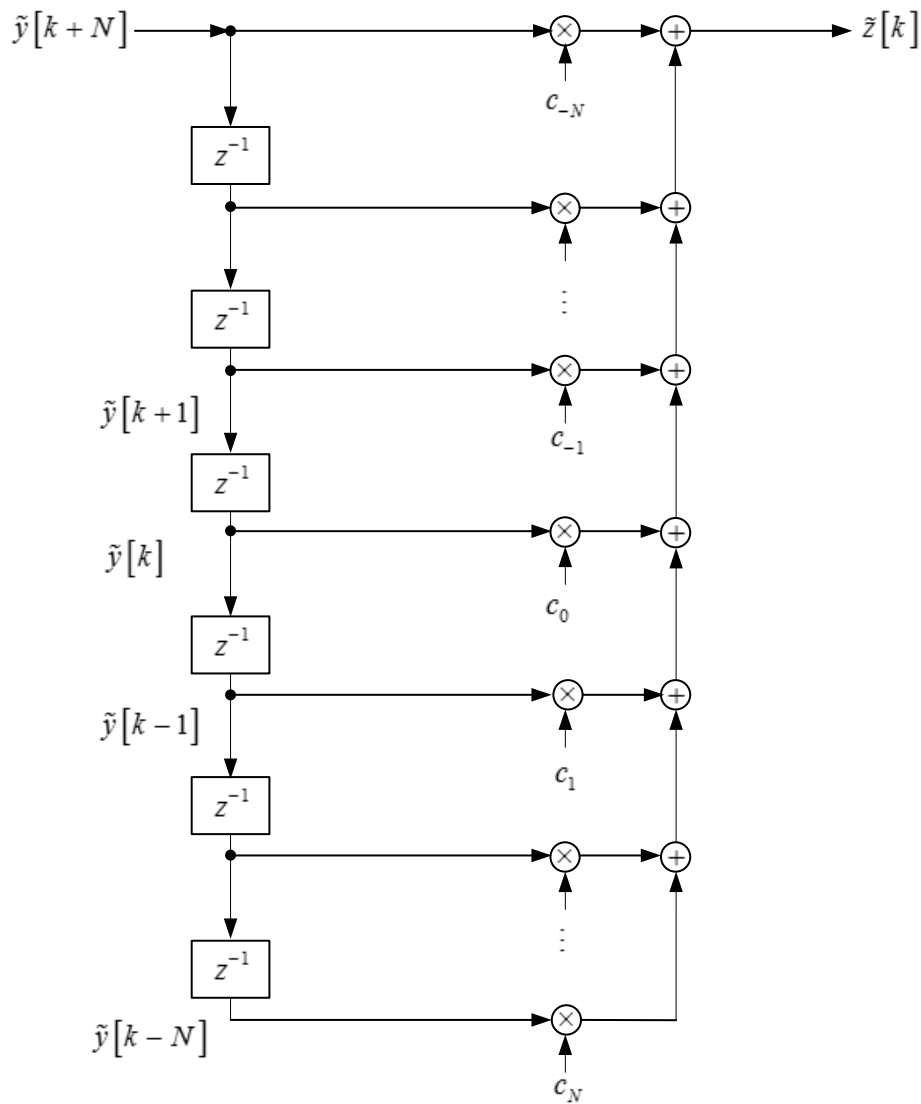


Рисунок 1. FIR эквалайзер

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.

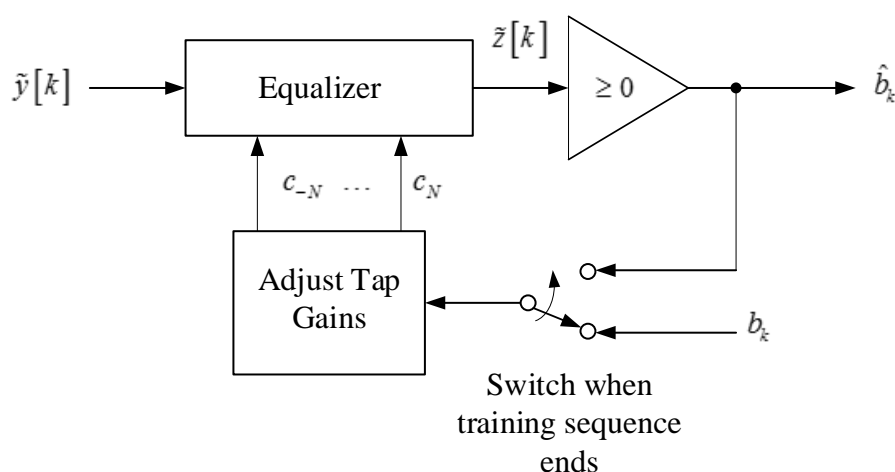


Рисунок 2. Эквалайзер с управлением по решению

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.

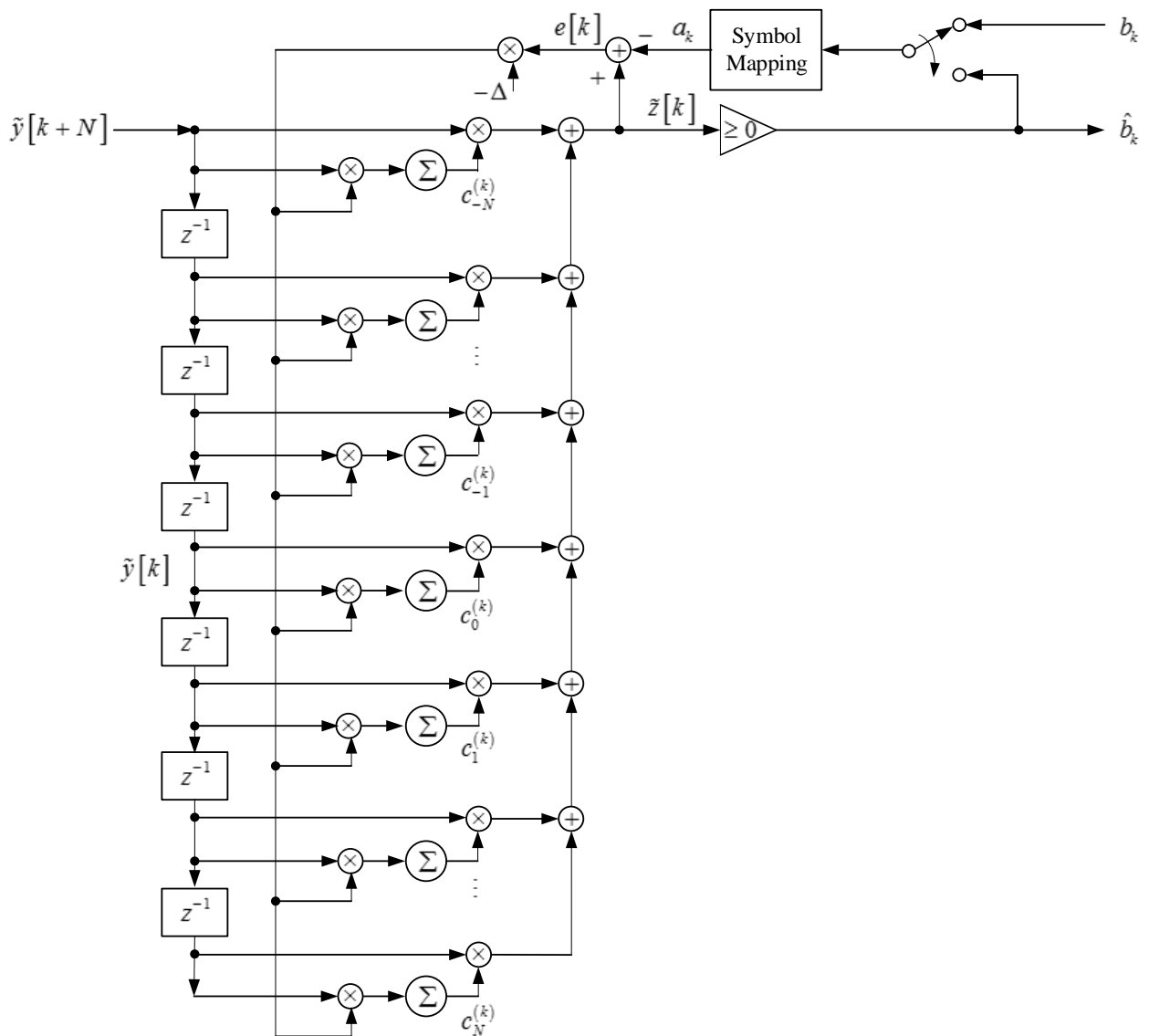


Рисунок 3. Адаптивный эквалайзер на основе алгоритма наименьших средних квадратов

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 обратной 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 Equalizer* 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.

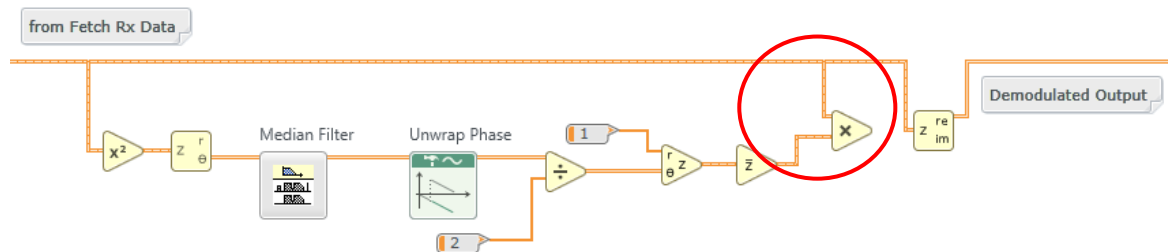


Рисунок 4. Удаление синхронизации фаз

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.