**СОДЕРЖАНИЕ**

Задание 2

Введение 3

Формирование сигнала 5

Аналого-цифровой преобразователь. 9

Модем 16

Воздействие шумов и помех 19

Цифро-аналоговый преобразователь 22

Сравнение сигналов 23

Литература 27

# Задание

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Входной сигнал, X,  (без случайной составляющей) | Точность, % | Интерфейс модема | Тип модуляции |
| 18 |  | 6/12 | параллельный | ФМ |

# Введение

Назначением системы связи является передача сообщения из одной точки в другую через канал связи, обладающий определенными свойствами (в частности, пропускающий лишь некоторую полосу частот). Для решения этой задачи приходится осуществлять целый ряд преобразований.

Прежде всего исходное сообщение подвергается первичному кодированию (кодированию источника), цель которого - преобразование аналогового сообщения в цифровое либо сжатие информации. Следующий этап - помехоустойчивое кодирование. Здесь в сообщение вносится избыточность с целью обеспечить возможность исправления на приемной стороне всех или некоторых возникших в процессе передачи ошибок. После применения помехоустойчивого кода сообщение поступает в модулятор, преобразующий цифровое сообщение в аналоговый модулированный сигнал, занимающий заданную полосу частот.

В процессе прохождения модулированного сигнала через канал связи сигнал подвергается воздействию шумов и помех. Искаженный сигнал поступает на вход приемника.

Структура приемной части является зеркальным отражением структуры передатчика - сигнал проходит через блоки, в обратном порядке осуществляющие преобразования, обратные по отношению к тем, что производились в передатчике.

Прежде всего сигнал подвергается демодуляции, в процессе которой аналоговый модулированный сигнал преобразуется в цифровое сообщение. Далее производится декодирование помехоустойчивого кода, при этом благодаря корректирующим свойствам кода возможно исправление части (или всех) ошибок, возникших в процессе передачи. После исправления ошибок следует декодирование источника - восстановление исходного сообщения.

Можно выделить следующие группы функций:

* функции моделирования и анализа сигналов;
* функции кодирования/декодирования источника;
* функции помехоустойчивого кодирования/декодирования;
* функции модуляции/демодуляции;
* функции моделирования каналов связи.

Кроме этих групп, непосредственно связанных со структурной схемой телекоммуникационной системы, имеется еще несколько групп общих и служебных функций:

* функции расчета специальных фильтров;
* функции вычислений в конечных полях (полях Галуа) ;
* вспомогательные функции.

## Формирование сигнала

Схема генерации входного сигнала изображена на рисунке 1.

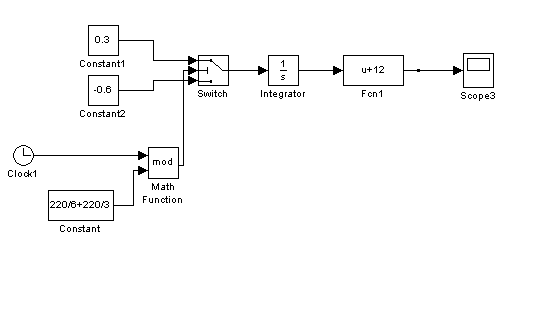


Рисунок 1 – Входной сигнал.

Согласно задания, мы должны получить треугольнообразный сигнал вида:



Таким образом, амплитуда сигнала равна 4, т.е. кривая сигнала равномерно возрастает (коэффициент 0,3) на величину равную 4, а затем равномерно падает (коэффициент - 0,6) на эту же величину. Определим период сигнала:



Настройки блока Constant (определяет период) показаны на рисунке 2.

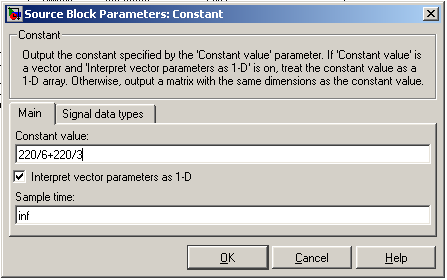


Рисунок 2 – Настройка блока Constant.

Блок Constant1 определяет коэффициент участка возрастания сигнала, его "constant value" равно 0,3. Блок Constant2 определяет коэффициент участка убывания сигнала, его "constant value" равно - 0,6.

Блок Switch реализует переключение между участком возрастания и убывания. Так как генерируется сигнал поочередно убывающий и возрастающий, то при условии "u2>=Threshold" Threshold должен быть равен 40/6 (диапазон участка убывания). Настройки блока Switch приведены на рисунке 3.

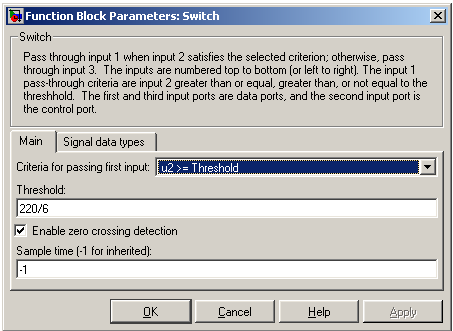


Рисунок 3 – Настройка блока Switch.

Для построение сигнала в диапазоне [-1; 3] необходимо использовать блок Fcn1 со значением параметра "expression", равным u+3. В результате получаем требуемый сигнал.

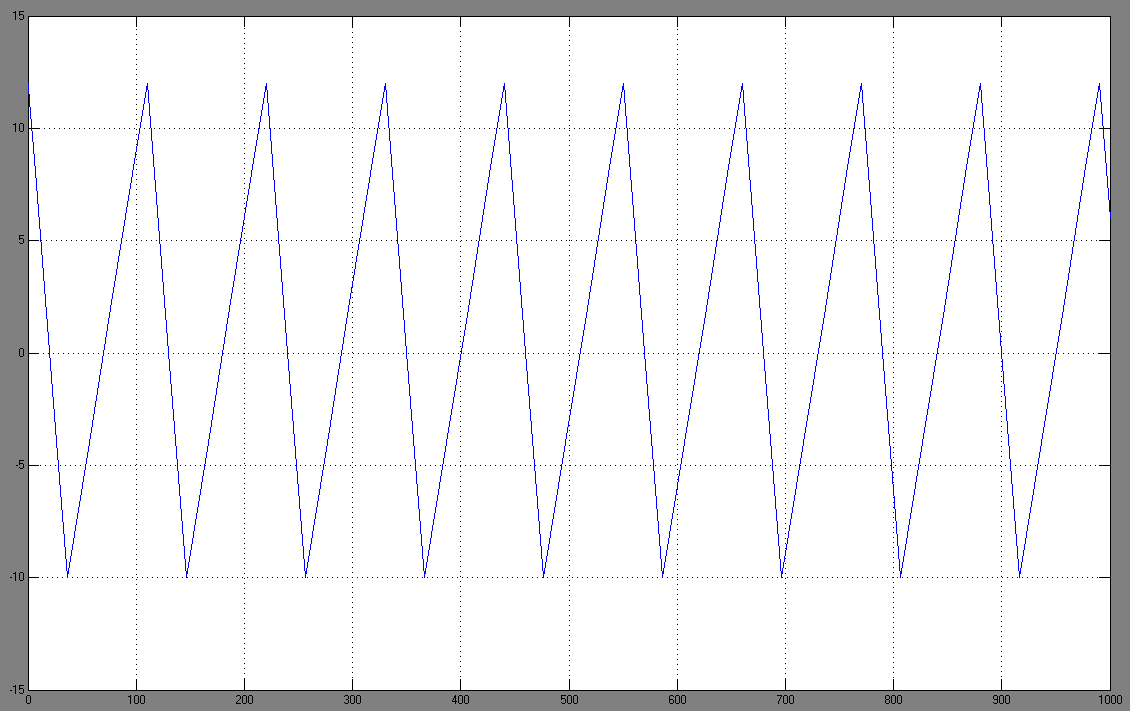


Рисунок 4 – График сгенерированного сигнала.

Данный сигнал можно сгенерировать, используя блок MATLAB Fcn и функцию:

function result=buildOfSignal(time)

global move;

global sdvig;

if time<2

sdvig=0;

move=1;

end;

if move==2

if (time-1-sdvig) \*0.3-1<3

result=(time-1-sdvig) \*0.3-1;

else

move=2;

sdvig=time;

%result=-0.6\*(time-1-sdvig) +3;

result=3;

end;

else

if - 0.6\*(time-1-sdvig) +3>-1

result=-0.6\*(time-1-sdvig) +3;

else

move=1;

sdvig=time;

%result=(time-1-sdvig) \*0.3-1;

result=-1;

end;

end;

## Аналого-цифровой преобразователь

Схема АЦП изображена на рисунке 5.

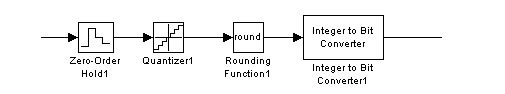


Рисунок 5 – АЦП.

Аналого-цифровой преобразователь состоит из блоков дискретизации (Zero-Order Hold1), квантователя (Quantizer1), целочисленного преобразователя (Rounding Function1) и конвертора (Integer to Bit Converter1).

Так как АЦП составляет в том числе и блок Integer to Bit Converter1, то на АЦП должны подаваться положительные значения. Поэтому между генератором сигнала и АЦП необходимо разместить блок Fcn3 со значением expression равным u+10.



Рисунок 6 – Блок смещения сигнала в положительный диапазон.

Блоков дискретизации Zero-Order Hold1 выполняет экстраполяцию входного сигнала на интервале дискретизации. Блок фиксирует значение входного сигнала в начале интервала дискретизации и поддерживает на выходе это значение до окончания интервала дискретизации. Затем входной сигнал изменяется скачком до величины входного сигнала на следующем шаге дискретизации. Параметр блока simple time настраивает такт дискретизации.

Рассчитаем такт дискретизации.



- верхняя граничная частота в спектре сигнала. Для нахождения этой величины строим спектральную характеристику, используя блок Averaging Power Spectral Density со следующими настройками:

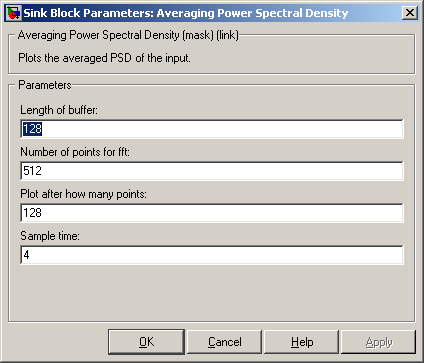


Рисунок 7 – Настройка блока Averaging Power Spectral Density.

В результате получим график, приведенный на рисунке 8:

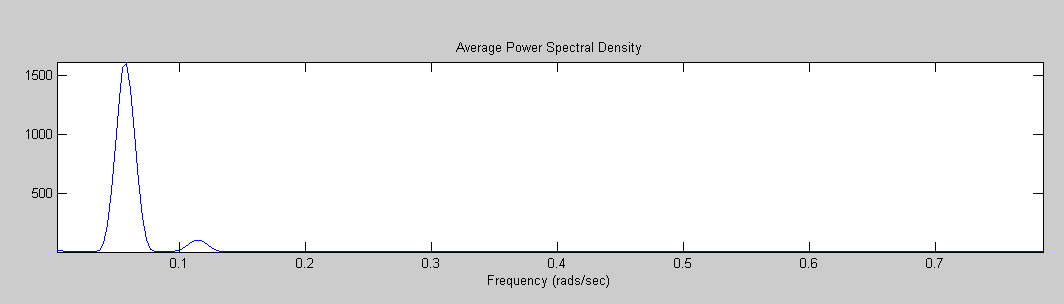


Рисунок 8 – Спектр сигнала.

На графике отчетливо видно две гармоники при ω=0,06 рад/с и при ω=0,11 рад/с. Расчет периода дискретизации ведется по второй гармонике.



Таким образом, блок дискретизации Zero-Order Hold1 будет иметь следующие настройки:

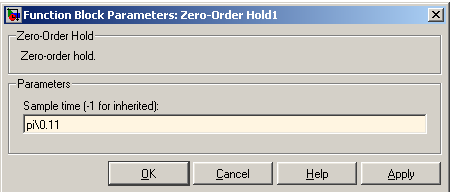


Рисунок 9 – Настройки блока Zero-Order Hold1.

Следующим элементом в АЦП является квантователь. Он обеспечивает квантование с одинаковым шагом по уровню. Рассчитаем значения его параметров.



Таким образом, настройки блока квантования Quantizer1 будут следующие:

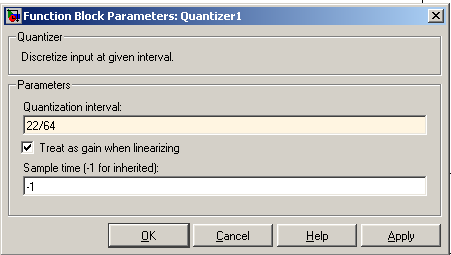


Рисунок 10 – Настройки блока Quantizer1.

Блок Rounding Function1 является простой математической функцией округления значения до ближайшего целого. Этот блок – подготовительное звено перед конвертированием квантованного сигнала в биты.

Блок Integer to Bit Converter1 выполняет преобразование целочисленных значений в биты. Настройки этого блока приведены на рисунке 11:

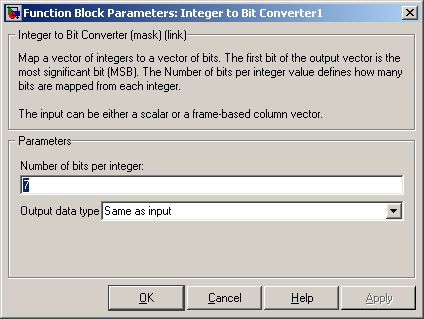


Рисунок 11 – Настройки блока Integer to Bit Converter1.

Параметр number of bits per integer (количество бит на число) определяется разрядностью АЦП (6) плюс 1.

При этом осциллограмма Scope1 покажет, что 4 битовых канала фактически не используются. Вероятнее всего это связано с несоответствием диапазона сигнала на входе АЦП и его разрядности, т.е. если используется АЦП 7 бит, то максимальное значение сигнала, на него подаваемое должно быть 128 (27) иначе его разрядность не используется при округлении к целым числам, т. о. сигнал, подаваемый на АЦП должен быть увеличен до этой величины, а после ЦАП на столько же уменьшен.

Для реализации увеличения и уменьшения сигнала используются блоки Gain (для увеличения) и Gain1 (для уменьшения) с коэффициентами 8 и 1/8 соответственно.

Блоки Gain выполняют умножение входного сигнала на постоянный коэффициент.

Параметры:

Gain – Коэффициент усиления.

Multiplication – Способ выполнения операции. Может принимать значения (из списка):

- Element-wise K\*u– Поэлементный.

- Matrix K\*u – Матричный. Коэффициент усиления является левосторонним операндом.

- Matrix u\*K – Матричный. Коэффициент усиления является правосторонним операндом.

Saturate on integer overflow (флажок) – Подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Параметр блока Gain может быть положительным или отрицательным числом, как больше, так и меньше 1. Коэффициент усиления можно задавать в виде скаляра, матрицы или вектора, а также в виде вычисляемого выражения.

В том случае если парметр Multiplication задан как Element-wise K\*u, то блок выполняет операцию умножения на заданный коэффициент скалярного сигнала или каждого элемента векторного сигнала. В противном случае блок выполняет операцию матричного умножения сигнала на коэффициент заданный матрицей.

По умолчанию коэффициент усиления является действительным числом типа double.

Для операции поэлементного усиления входной сигнал может быть скалярным, векторным или матричным любого типа, за исключением логического (boolean). Элементы вектора должны иметь одинаковый тип сигнала. Выходной сигнал блока будет иметь тот же самый тип, что и входной сигнал. Параметр блока Gain может быть скаляром, вектором или матрицей либого типа, за исключением логического (boolean).

При вычислении выходного сигнала блок Gain использует следующие правила:

Если входной сигнал действительного типа, а коэффициент усиления комплексный, то выходной сигнал будет комплексным.

Если тип входного сигнала отличается от типа коэффициента усиления, то Simulink пытается выполнить приведение типа коэффициента усиления к типу входного сигнала. В том случае, если такое приведение невозможно, то расчет будет остановлен с выводом сообщения об ошибке. Такая ситуация может возникнуть, например, если входной сигнал есть беззнаковое целое (uint8), а параметр Gain задан отрицательным числом.

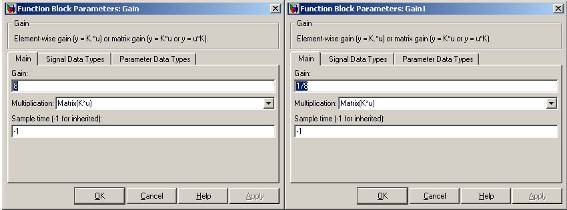


Рисунок 12 – Настройка блоков Gain и Gain1

Использование блоков Gain и Gain1 позволит оптимизировать загрузку канала.

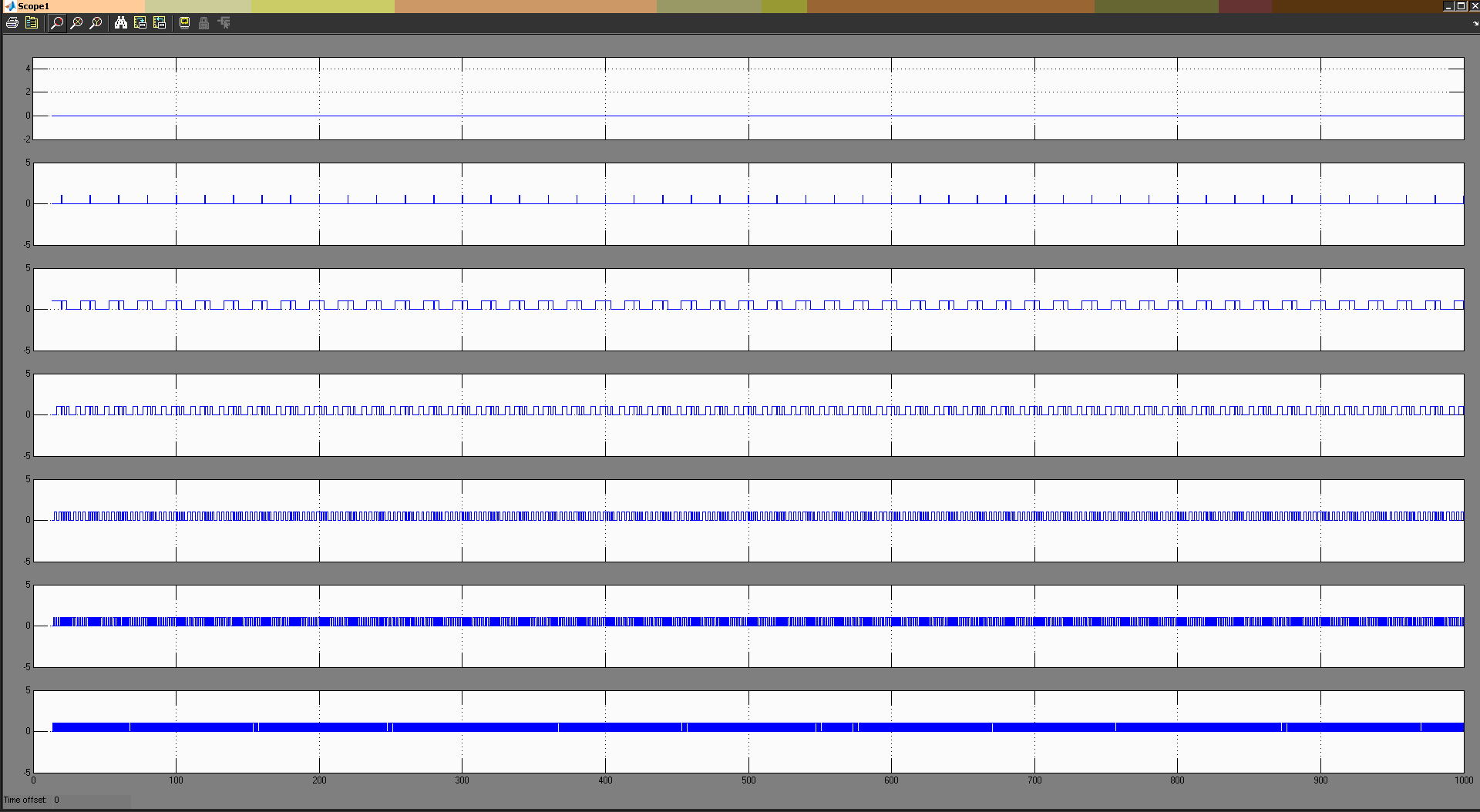


Рисунок 13 – Показания на осциллограмме Scope1

## 

## Модем

Для передачи по любому каналу связи цифровое сообщение, представляющее собой последовательность символов (чисел), необходимо преобразовать в аналоговый сигнал - изменяющуюся во времени физическую величину (например, напряжение). Кроме того, канал связи способен пропускать лишь определенную полосу частот, так что сформированный аналоговый сигнал должен этой полосе соответствовать. Указанное преобразование осуществляется путем модуляции. Обратный процесс носит название демодуляции.

Сущность процесса модуляции состоит в том, что передаваемое сообщение используется для изменения каких-либо параметров несущего колебания (чаще всего - синусоидального). Это позволяет создать модулированный сигнал, занимающий заданную полосу частот.

При цифровой модуляции возможны следующие ее виды:

* амплитудная манипуляция;
* частотная манипуляция;
* минимальная частотная манипуляция;
* фазовая манипуляция;
* квадратурная манипуляция.

Цифровая модуляция и демодуляция включают в себя две стадии. При модуляции цифровое сообщение сначала преобразуется в аналоговый модулирующий сигнал с помощью функции modmap, а затем осуществляется аналоговая модуляция. При демодуляции сначала получается аналоговый демодулированный сигнал, а затем он преобразуется в цифровое сообщение с помощью функции demodmap.

По заданию необходимо использовать фазовую манипуляцию (модуляцию цифрового сигнала). Фазовая манипуляция (англ. phase-shift keying (PSK)) была разработана в начале развития программы исследования дальнего космоса; сейчас схема PSK широко используется в коммерческих и военных системах связи. Фазоманипулированный сигнал имеет следующий вид:



Фазовый член φi(t) может принимать М дискретных значений.

Имеет много разновидностей: 2 фазы (BPSK, B-Binary, 1 бит на 1 смену фазы), 4 фазы (QPSK, Q-Quadro, 2 бита на 1 смену фазы), 8 фаз (8-psk, 3 бита на 1 смену фазы) и т.д. Таким образом, количество бит n, передаваемых одним перескоком фазы является степенью в которую возводится двойка при определении числа фаз, требующихся для передачи n-порядкового двоичного числа.

В построении модема будет использоваться QPSK.

Квадратурная фазовая манипуляция иногда называется 4-PSK. В QPSK (англ. Quadrature Phase Shift Keying) используются созвездие из четырех точек, размещенных на равных расстояниях на окружности. Используя 4 фазы, QPSK может кодировать два бита на символ, как показано на рисунке, для минимизации BER - удвоенная вероятность для BPSK. Анализ показывает, что скорость может быть увеличена в два раза относительно BPSK при той же полосе сигнала, либо оставить скорость прежней, но уменьшить полосу вдвое.

Хотя QPSK может рассматриваться как квадратурная модуляция, ее проще рассматривать в виде двух независимых модулированных несущих сдвинутых на 90°. При таком подходе четные (нечетные) биты используются для модуляции синфазной составляющей, а нечетные (четные) - квадратурной составляющей несущей. Так как BPSK используется для обеих составляющих несущей, то они могут быть демодулированы независимо.

В результате, вероятность ошибки на бит для QPSK такая же как и для BPSK:



Однако, так как в символе два бита, то величина ошибки на символ возрастает:



При высоком отношении с/ш (это необходимо для реальных QPSK систем) вероятность м. б. примерно оценена по следующей формуле:



Как и при BPSK, существует проблема неопределенности начальной фазы в приемнике. Поэтому QPSK с дифференциальным кодированием на практике используется чаще.

Отличие QPSK от АМн, ЧМн в том что плотность передаваемой информации в расчёте на частотную ширину канала (на символ, на герц) выше единицы.

Например, в АМн плотность много меньше единицы (0,1-0,001 бит на герц) - это связано с необходимостью накопления энергии в фильтрах в первых малочувствительных приёмниках (например, русский изобретатель радио Попов использовал АМн в первом в мире приёмнике). В ЧМн этот показатель приближается к единице (0,1-1) бит на символ (герц). Например, в GMSK, применяемому в GSM плотность информации равняется 1.

Этот вид модуляции используется например в стандарте сотовой связи CDMA2000 1X EV-DO.

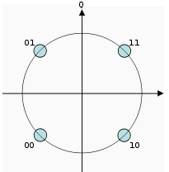


Рисунок 14 - Фазовое созвездие для квадратурной ФМн.

В качестве блоков модуляции будут использоваться блоки раздела Communications Blockset\Modulation\Digital Baseband Modulation\PM QPSK Modulator Baseband, а в качестве блоков демодуляции – элементы того же раздела QPSK Demodulator Baseband.

Так как, после АЦП аналоговый сигнал преобразуется в параллельный цифровой, то на входе модема необходимо установить демультиплексор с размерностью 7, а на выходе мультиплексор с размерностью 7.

Блок модема изображен на рисунке 13:

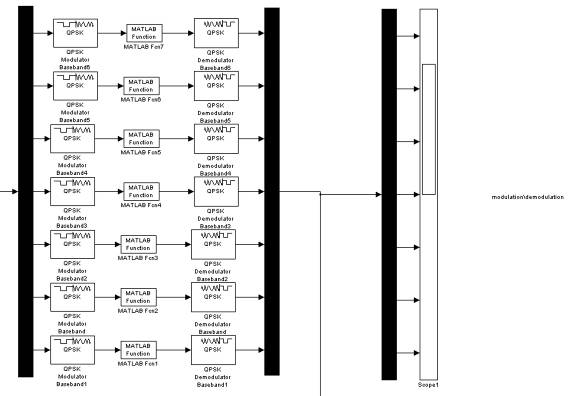


Рисунок 15 – Модем.

## Воздействие шумов и помех

Сформированный в результате модуляции сигнал поступает в канал связи, где он подвергается воздействию шумов и помех. Поэтому функции моделирования каналов связи должны обеспечивать внесение в сигнал искажений согласно используемым статистическим моделям.

Можно воспользоваться функцией awgn, которая позволяет добавить к сигналу аддитивный белый нормальный шум, реализовав при этом заданное отношение сигнал/шум. При этом мощность сигнала, знать которую необходимо для расчета требуемого уровня шума, может задаваться в явном виде либо измеряться автоматически.

Синтаксис:

y = awgn(x,snr);

y = awgn(x,snr,sigpower);

y = awgn(x,snr,'measured');

y = awgn(x,snr,sigpower,state);

y = awgn(x,snr,'measured',state);

y = awgn(...,powertype);

Описание:

y = awgn(x,snr)

Добавляет белый гауссов шум к вектору сигнала x. Скаляр snr задает отношение сигнал/шум в децибелах. Если значения x являются комплексными, функция awgn добавляет комплексный шум. При этом предполагается, что мощность сигнала x равна 0 дБ.

y = awgn(x,snr,sigpower)

То же, что и предыдущий вариант синтаксиса, но в данном случае параметр sigpower указывает мощность сигнала x в децибелах.

y = awgn(x,snr,'measured')

То же, что y = awgn(x,snr), но мощность сигнала не считается равной 0 дБ, а автоматически измеряется.

y = awgn(x,snr,sigpower,state)

То же, что y = awgn(x,snr,sigpower), но целочисленный параметр state позволяет задавать внутреннее состояние генератора гауссовых случайных чисел MATLAB (функция randn).

y = awgn(x,snr,'measured',state)

То же, что y = awgn(x,snr,'measured'), но целочисленный параметр state позволяет задавать внутреннее состояние генератора гауссовых случайных чисел MATLAB (функция randn).

y = awgn(...,powertype)

То же, что предыдущие варианты синтаксиса, но в данном случае строковый параметр powertype задает единицы измерения, использованные при указании параметров snr и sigpower. Возможные значения параметра powertype - 'dB' и 'linear'. При значении 'linear' мощность измеряется в ваттах.

Таким образом, для реализации помех воспользуемся блоком MATLAB Fcn с настройками, указанными на рисунке 15:

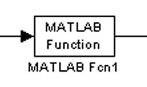


Рисунок 16 - Блок MATLAB Fcn.

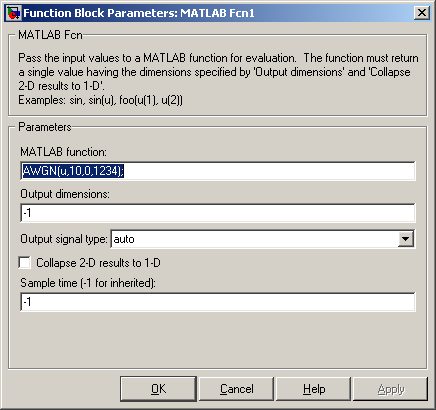


Рисунок 17 – Реализация белого шума.

## Цифро-аналоговый преобразователь

Схема ЦАП изображена на рисунке 16:

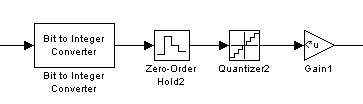


Рисунок 18 – Схема АЦП.

В состав АЦП входят блоки конвертора (Bit to Integer Converter), задержки исполнения на период дискретизации (Zero-Order Hold2) и квантования (Quantizer2).

Блок Bit to Integer Converter предназначен для преобразования цифрового сигнала в целочисленный аналоговый. Его настройки соответствуют блоку Integer to Bit Converter1 в АЦП:

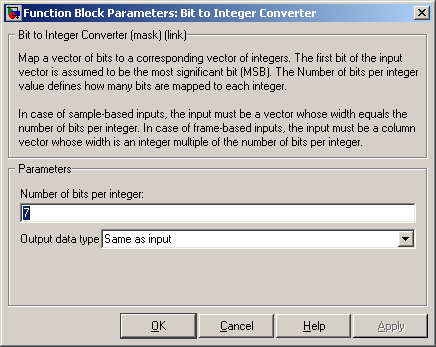


Рисунок 19 – Настройка блока Bit to Integer Converter.

Блока Zero-Order Hold2 обеспечивает задержку входного сигнала на период дискретизации. Основным назначением блока является экстраполяция входного сигнала на интервале дискретизации. Но он также может быть использован для согласования работы дискретных блоков, имеющие разные такты дискретности.

Следующим элементом в ЦАП является квантователь. Он обеспечивает квантование с одинаковым шагом по уровню. Рассчитаем значения его параметров.

Блок Quantizer2 будет иметь схожие настройки с Quantizer1 АЦП:

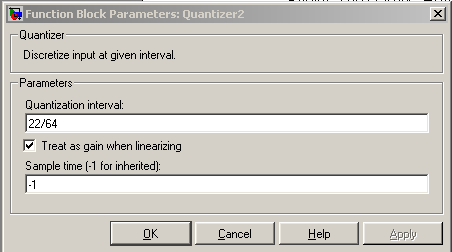


Рисунок 20 – Настройки блока Quantizer2.

## Сравнение сигналов

Использование блоков Averaging Power Spectral Density (на сгенерированный сигнал перед входом на АЦП) и Averaging Power Spectral Density1 (на полученный сигнал после выхода с ЦАП), позволяет сравнить входной и выходной сигнал.

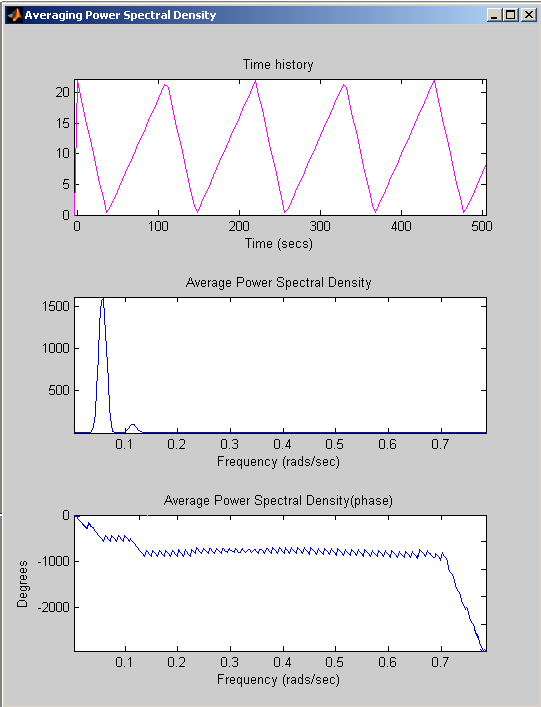


Рисунок 19 – Входной сигнал.

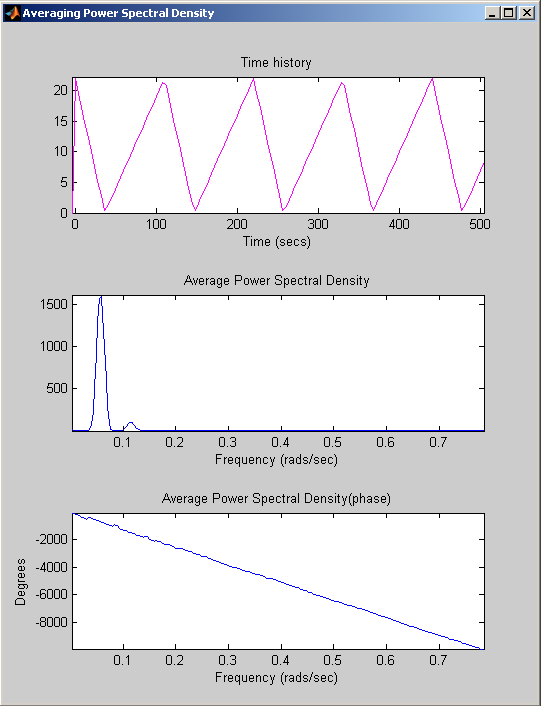


Рисунок 20 – Выходной сигнал.

Для оценки помехоустойчивости системы связи необходимо произвести сравнение исходного (передаваемого) сообщения с сообщением, полученным в результате приема, и определить число ошибок, возникших в процессе передачи, а также вероятность ошибки. Эти действия выполняются функциями symerr и biterr, первая из которых подсчитывает число несовпадающих символах в двух сообщениях, а вторая - число несовпадающих битов в двоичных представлениях этих символов. Кроме числа ошибок, обе функции могут возвращать долю ошибок в общем числе символов (битов) и индикаторы мест возникновения ошибок.

Воспользуемся блоками To Workspace и To Workspace1, установленными соответственно перед входом на АЦП и после выхода с ЦАП. Блоки записывают данные, поступающие на их вход в рабочую область Matlab.

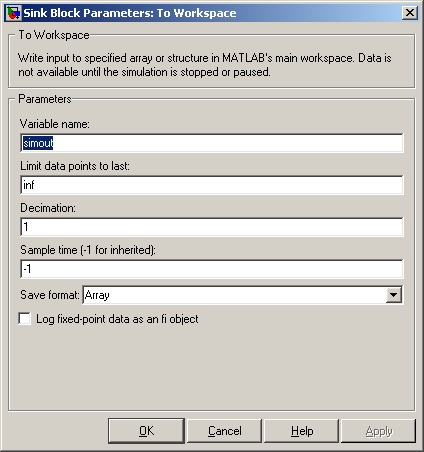


Рисунок 21 – Настройка блока To Workspace

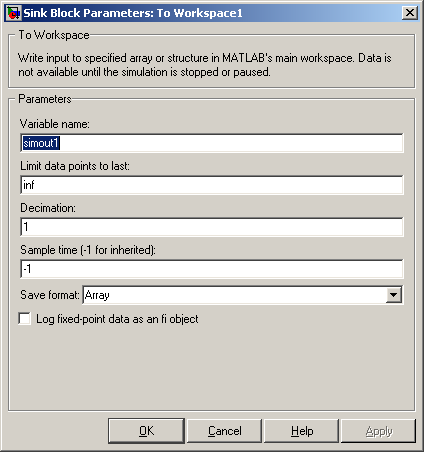


Рисунок 22 – Настройка блока To Workspace1

Первый блок записывает данные в массив simout, второй - в simout1. Воспользуемся функцией symerr для оценки вероятности возникновения ошибки.

>> for i=1: 100

a2(i) =simout1(i);

end

>> for i=1: 100

a1(i) =simout(i);

end

>> [n,r] =symerr(a1,a2);

>> r

r =

0.0900

Переменная r - это вероятность ошибки. Значение 0,09 соответствует 9%.

# Литература

1. Сергиенко А.Б. Цифровая обработка сигналов – СПб.: Питер, 2002.
2. Дьяконов В. Matlab. Обработка сигналов и изображений. Специальный справочник. – СПб.: Питер 2002.
3. Интернет http: // www. exponenta. ru/