Амплитудная модуляция

**Амплиту́дная модуля́ция** — вид [модуляции](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F), при которой изменяемым параметром [несущего сигнала](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D1%81%D1%83%D1%89%D0%B8%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) является его[амплитуда](https://ru.wikipedia.org/wiki/%D0%90%D0%BC%D0%BF%D0%BB%D0%B8%D1%82%D1%83%D0%B4%D0%B0).

% амплитудная модуляция

Fs = 4000; % частота дискретизации

f = 100; % частота сигнала

Fm = 1000; % частота модулирующего сигнала

t = ( 0 : 1 / Fs : 4 )'; % длительность сигнала 4 с

A = 10; % амплитуда сигнала, В

Am = 15; % амплитуда модулирующего сигнала, В

s = A .\* sin( 2 .\* pi .\* f .\* t ); % сигнал

y = ammod( s, Fm, Fs, 0, Am ); % модулированный сигнал,

plot( t, y ); % строим график

xlabel( 't, Время, с' );

ylabel( 'А, амплитуда сигнала, В' );

title( 'Амплитудная модуляция' );

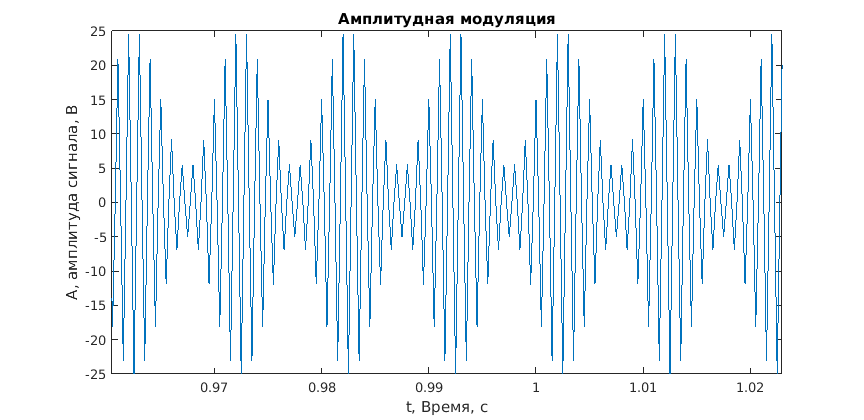


Рисунок 1 – АМ

Y = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр амплитудномодулированного сигнала' );

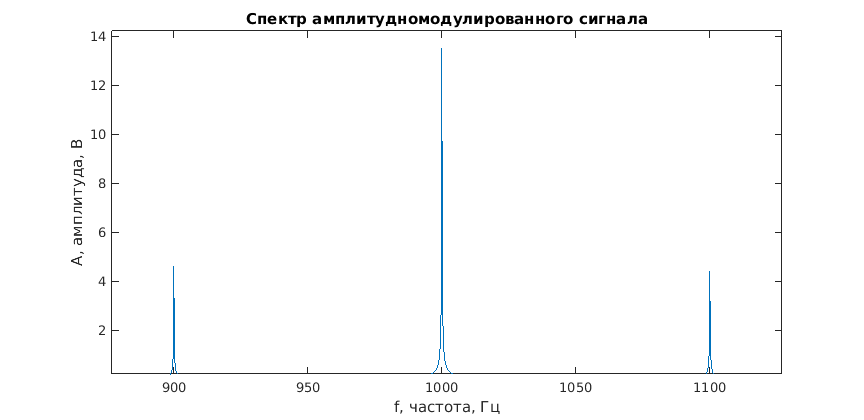


Рисунок 2 - Спектр АМ сигнала

% АМ с подавленной несущей

y = ammod( s, Fm, Fs ); % модулированный сигнал

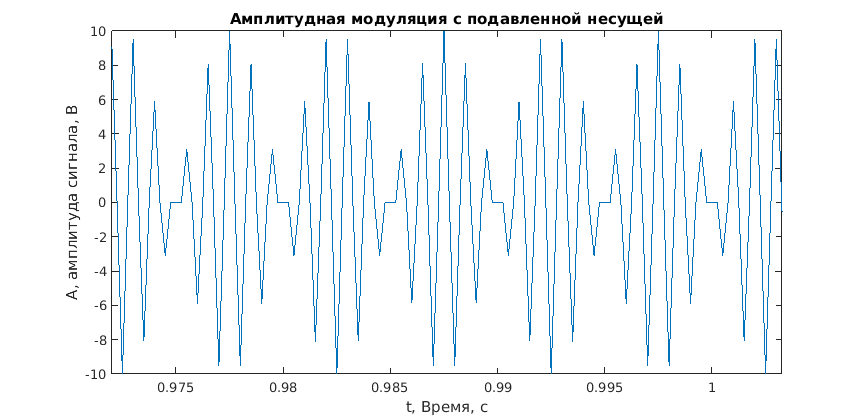


Рисунок 3 - АМ с подавленной несущей

Y = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр амплитудномодулированного сигнала с подавленной несущей' );

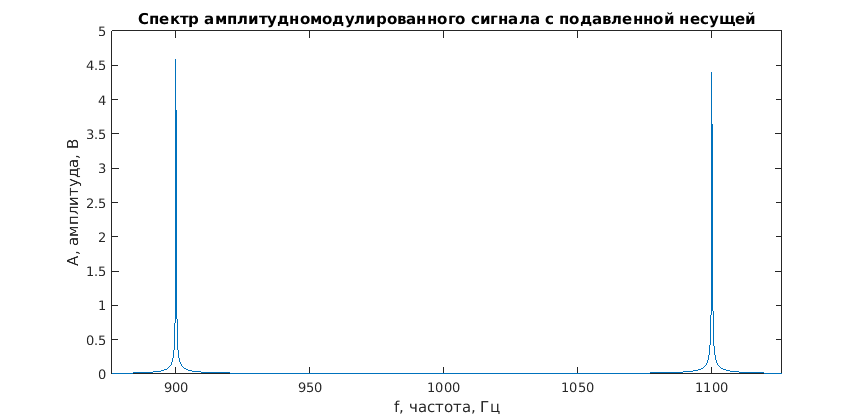


Рисунок 4

% АМ с подавленной боковой

y = ssbmod( s, Fm, Fs ); % модулированный сигнал

plot( t, y ); % строим график

xlabel( 't, Время, с' );

ylabel( 'А, амплитуда сигнала, В' );

title( 'Амплитудная модуляция с подавленной боковой' );

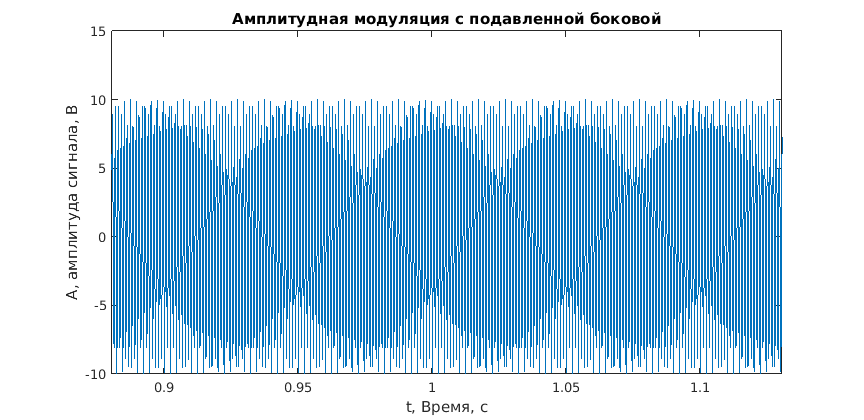


Рисунок 5

Y = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр амплитудномодулированного сигнала с подавленной боковой' );

% подавили верхнюю боковую



Рисунок 6

% амплитудно-импульсная модуляция, побитовая передача

M = 8; % глубина модуляции

k = log2( M ); % количество битов на символ

binInputData = randi( [ 0 1 ], 1024, k ); % формируем входные данные

decInputData = bi2de( binInputData ); % переводим в десятичные числа

plot( decInputData ); % строим график

xlabel( 'Отсчеты' );

ylabel( 'Сигнал' );

title( 'Дискретный сигнал' );

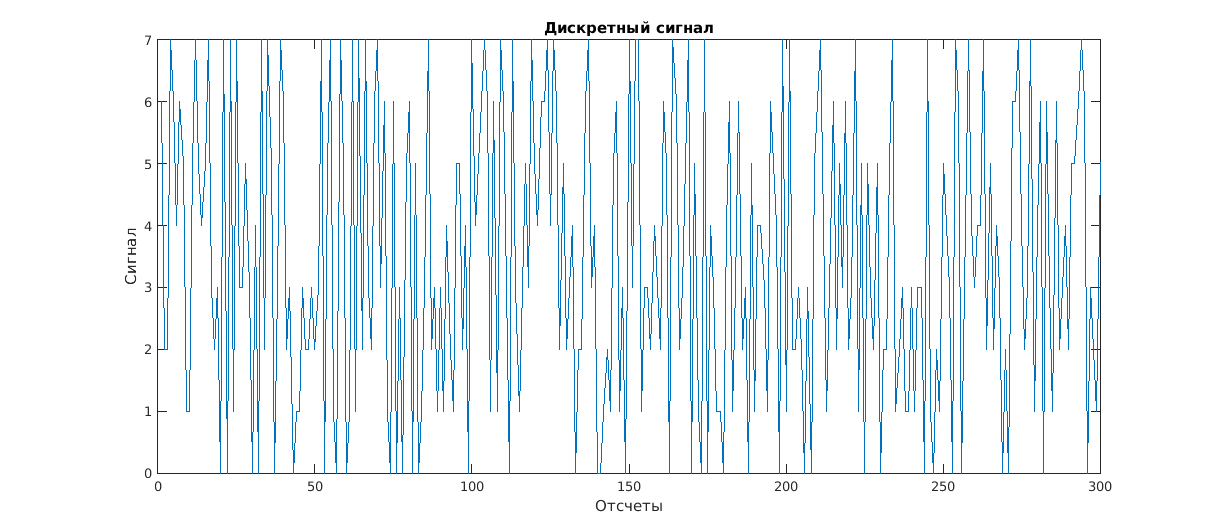


Рисунок 7

modData = pammod( decInputData, M ); % модулированный сигнал

scatterplot( modData ); % строим созвездие

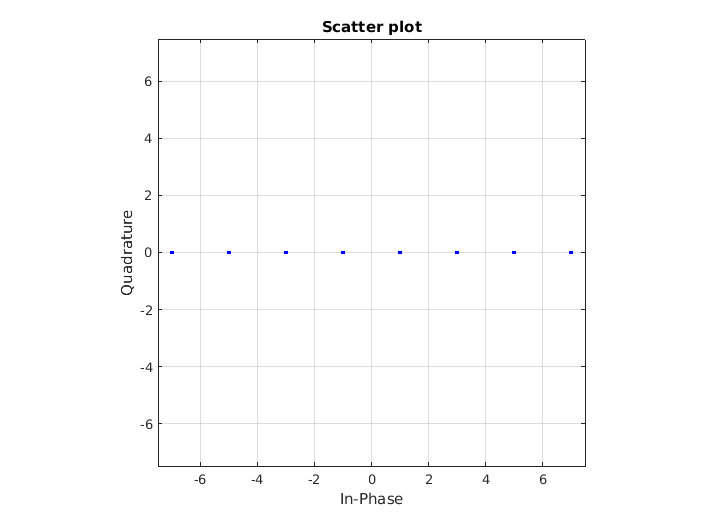


Рисунок 8

% моделируем передачи по каналу с переменным соотношением с/ш, считаем

% вероятности ошибок:

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

decErrors = []'; % количества символьных ошибок

decErrorProbabilities = []'; % и вероятностей возникновения символьных ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

decResData = pamdemod( chanData, M ); % демодуляция сигнала

binResData = de2bi( decResData ); % представляем битами

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( binInputData, binResData );

[ decErrors( end + 1 ), decErrorProbabilities( end + 1 ) ] = symerr( decInputData, decResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче амплитудноимпульсномодулированного сигнала' );

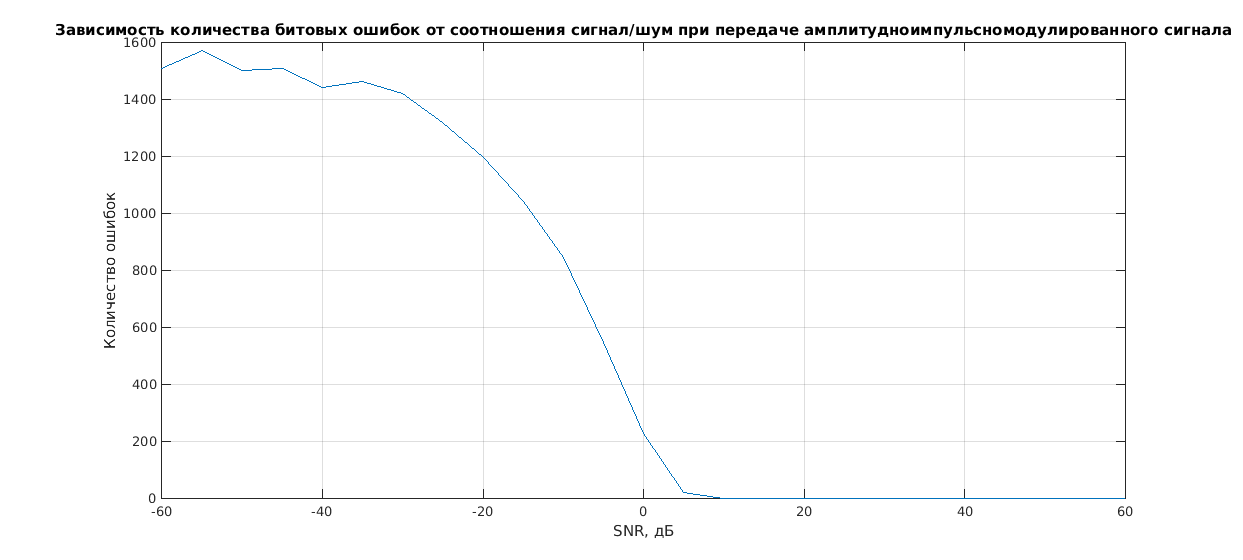


Рисунок 9

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения битовой ошибки от соотношения сигнал/шум' );



Рисунок 10

plot( snrs, decErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче амплитудноимпульсномодулированного сигнала' );

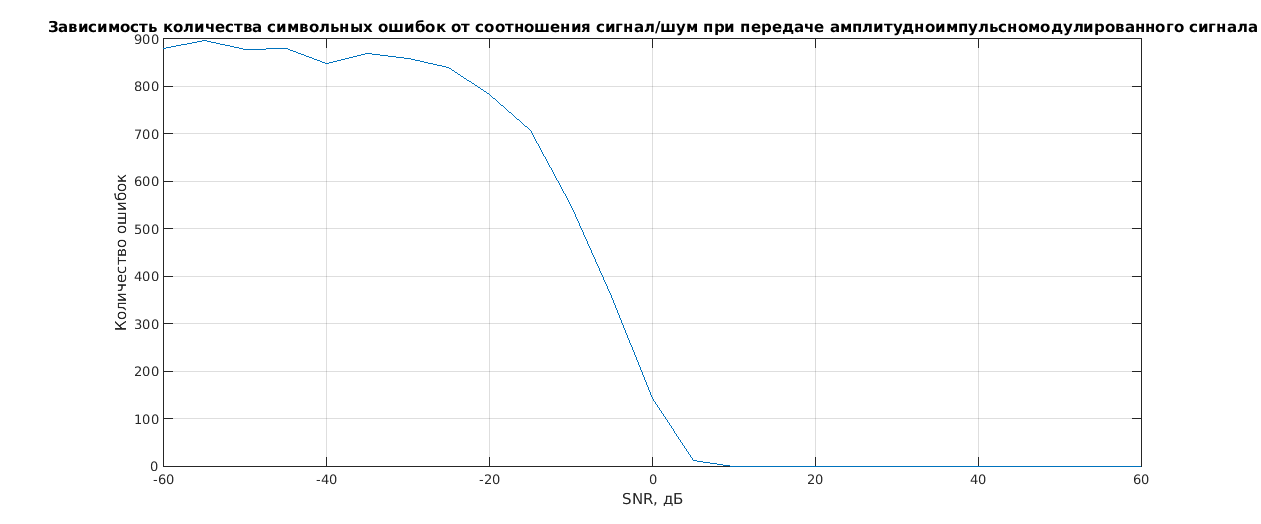


Рисунок 11

plot( snrs, decErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения символьной ошибки от соотношения сигнал/шум' );

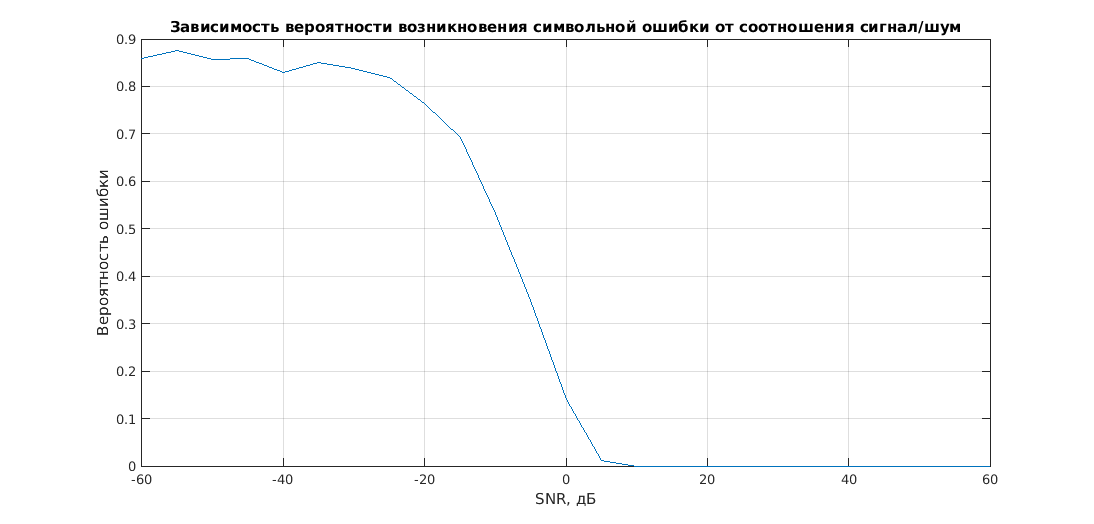


Рисунок 12

% Квадратурная модуляция

phase = pi / 4; % чтоб было поинтересней, добавим начальную фазу

modData = qammod( decInputData, M, phase ); % модулируем

scatterplot( modData ); % строим созвездие

grid on

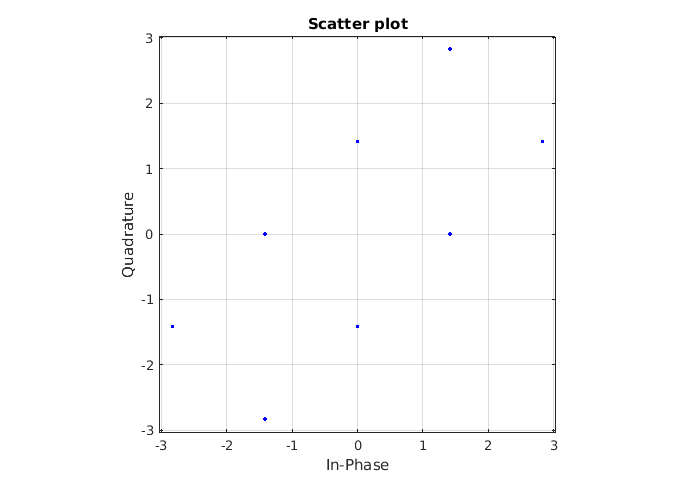


Рисунок 13

% моделируем то же для QAM:

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

decErrors = []'; % количества символьных ошибок

decErrorProbabilities = []'; % и вероятностей возникновения символьных ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

decResData = qamdemod( chanData, M, phase ); % демодуляция сигнала

binResData = de2bi( decResData ); % представляем битами

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( binInputData, binResData );

[ decErrors( end + 1 ), decErrorProbabilities( end + 1 ) ] = symerr( decInputData, decResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче QAM сигнала' );



Рисунок 14

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения битовой ошибки от соотношения сигнал/шум' );

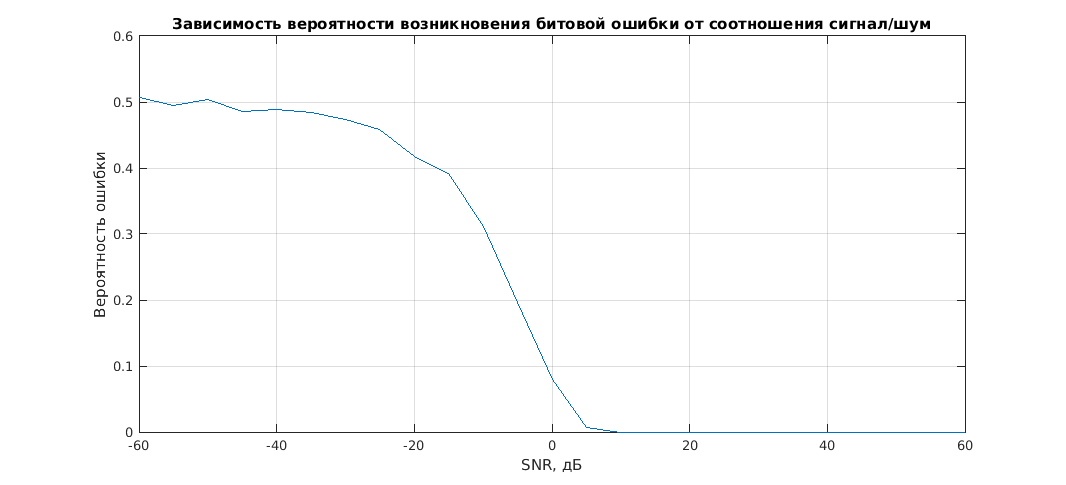


Рисунок 15

plot( snrs, decErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче QAM сигнала' );

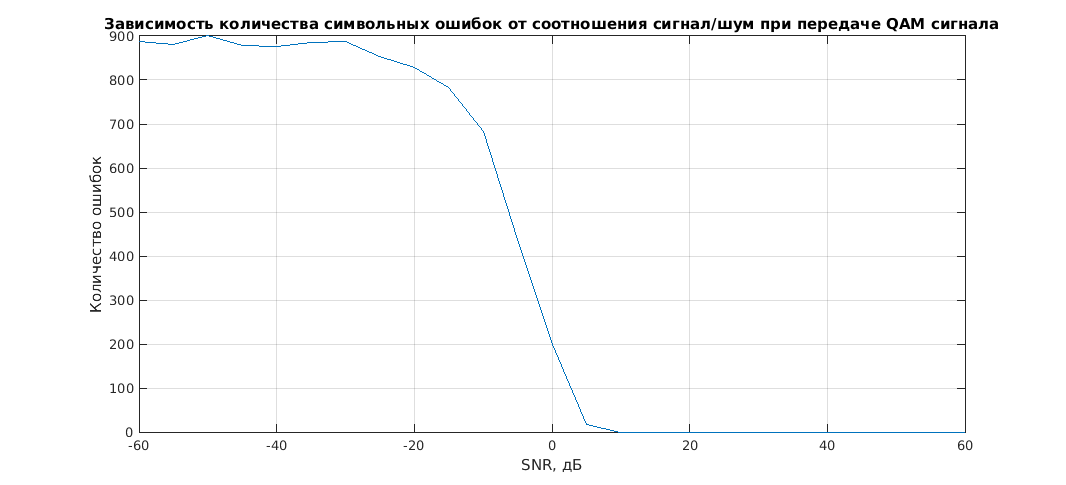


Рисунок 16

plot( snrs, decErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения символьной ошибки от соотношения сигнал/шум' );

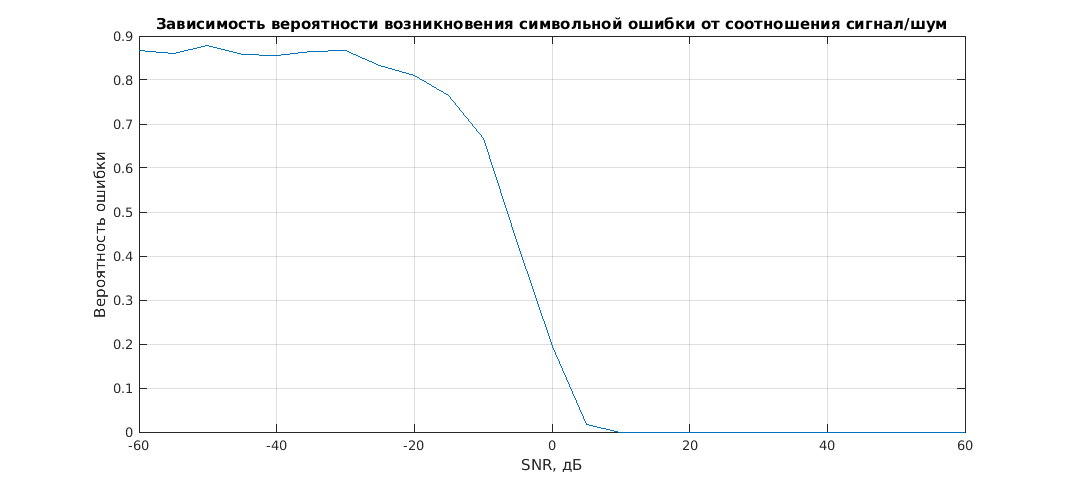


Рисунок 17

% амплитудная манипуляция методом топора

lengthOfBit = 20; % количество отчетов, которое будет длиться 1 символ

% сформируем сигнал из набора входных данных

sig = zeros( lengthOfBit \* length( decInputData ), 1 ); % инициализируем массив нового сигнала

for i = 0 : length( decInputData ) - 1

for j = 1 : lengthOfBit

sig( i \* lengthOfBit + j ) = decInputData( i + 1 ); % копируем данные

end

end

modSig = ammod( sig, Fm, Fs, 0, Am ); % модулированный сигнал

errors = []'; % количество символьных ошибок

errorProbabilities = []'; % и вероятностей возникновения символьных ошибокY = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр амплитудномодулированного сигнала' );

for snr = -60 : 5 : 60

chanSig = awgn( modSig, snr ); % сигнал в канале передачи

resSig = amdemod( chanSig, Fm, Fs, 0, Am ); % демодуляция сигнала

% вернемся к представлению данных

resData = zeros( length( decInputData ), 1 );

j = 1;

for i = 1 : lengthOfBit : length( resSig ) - lengthOfBit

resData( j ) = round( mean( resSig( i : i + lengthOfBit ) ) ); % возьмем среднее из lengthOfBit отсчетов

j = j + 1;

end

% подсчет ошибок

[ errors( end + 1 ), errorProbabilities( end + 1 ) ] = symerr( decInputData, resData );

end

% строим графики

plot( snrs, errors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче ASK сигнала' );

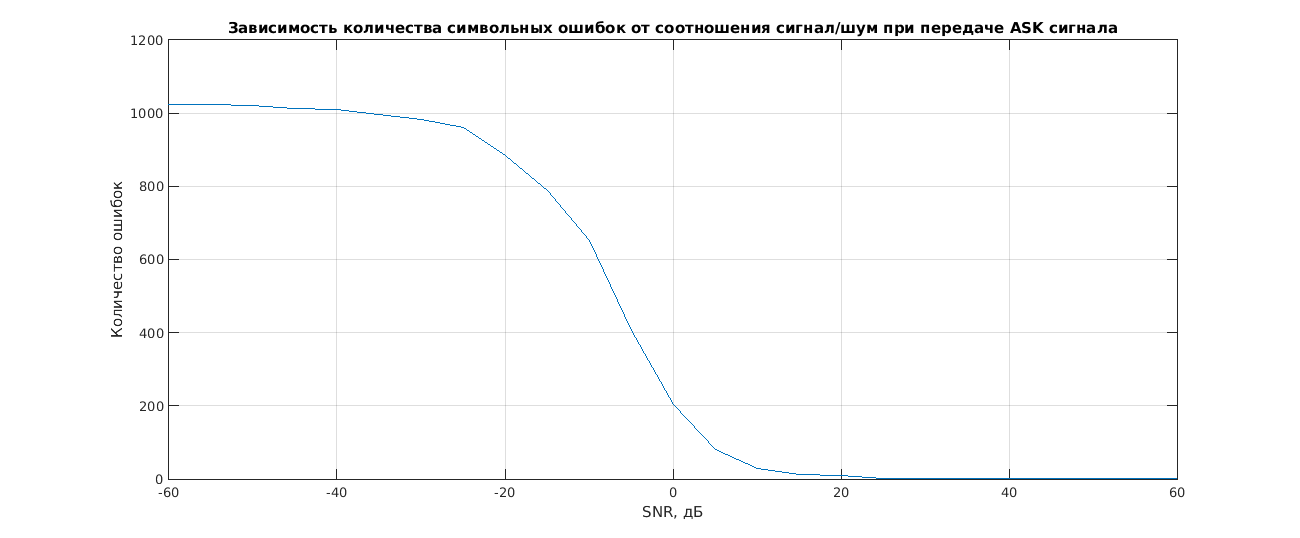


Рисунок 18

plot( snrs, errorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность возникновения ошибки' );

title( 'Зависимость вероятности возникновения символьных ошибок от соотношения сигнал/шум при передаче ASK сигнала' );

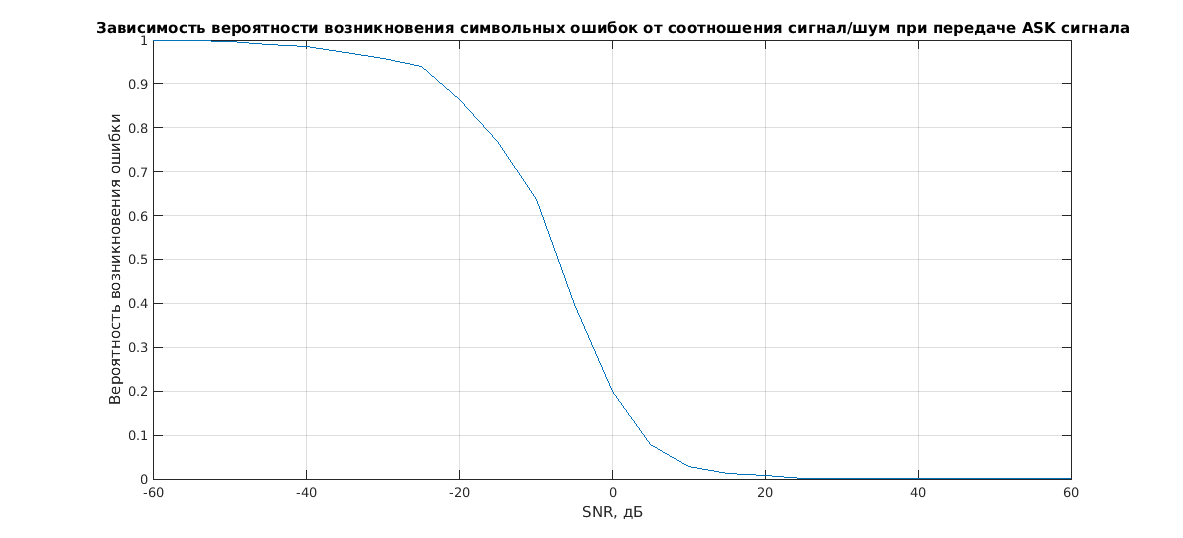


Рисунок 19

Частотная модуляция

% Частотная модуляция

% зададим новые параметры системы

Fs = 16000; % частота дискретизации

f = 100; % частота сигнала

Fm = 1000; % частота модулирующего сигнала

t = ( 0 : 1 / Fs : 4 )'; % длительность сигнала 4 с

A = 10; % амплитуда сигнала, В

Am = 15; % амплитуда модулирующего сигнала, В

s = A .\* sin( 2 .\* pi .\* f .\* t ); % сигнал

dev = 40; % максимальная девиация частоты

y = fmmod( s, Fm, Fs, dev ); % модулируем

plot( t, y ); % строим график

xlabel( 't, Время, с' );

ylabel( 'А, амплитуда сигнала, В' );

title( 'Частотная модуляция' );

grid on

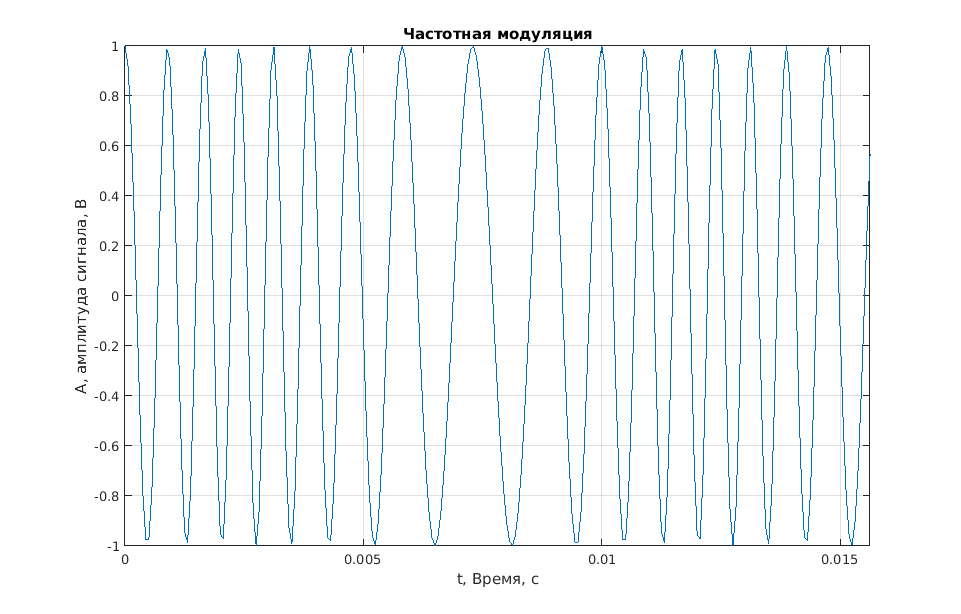


Рисунок 20 Частотная модуляция

Y = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр частотномодулированного сигнала' );

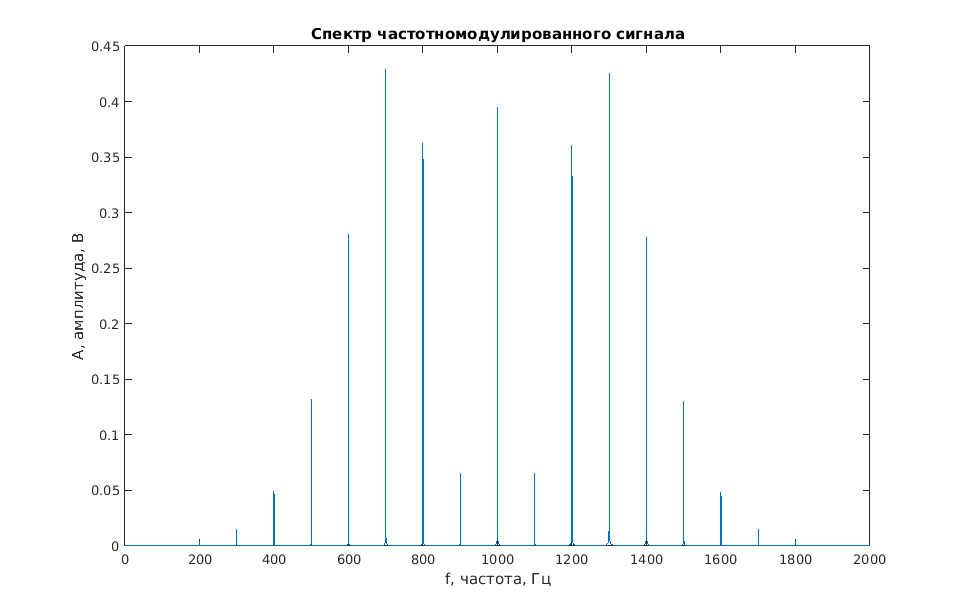


Рисунок 21 - Спектр ЧМ сигнала

% ЛЧМ

y = chirp( 0 : 1 / Fs : 0.5, 20, t( end ), 8000 );

plot( 0 : 1 / Fs : 0.5, y ); % строим график

xlabel( 't, Время, с' );

ylabel( 'А, амплитуда сигнала, В' );

title( 'Линейная частотная модуляция' );

grid on

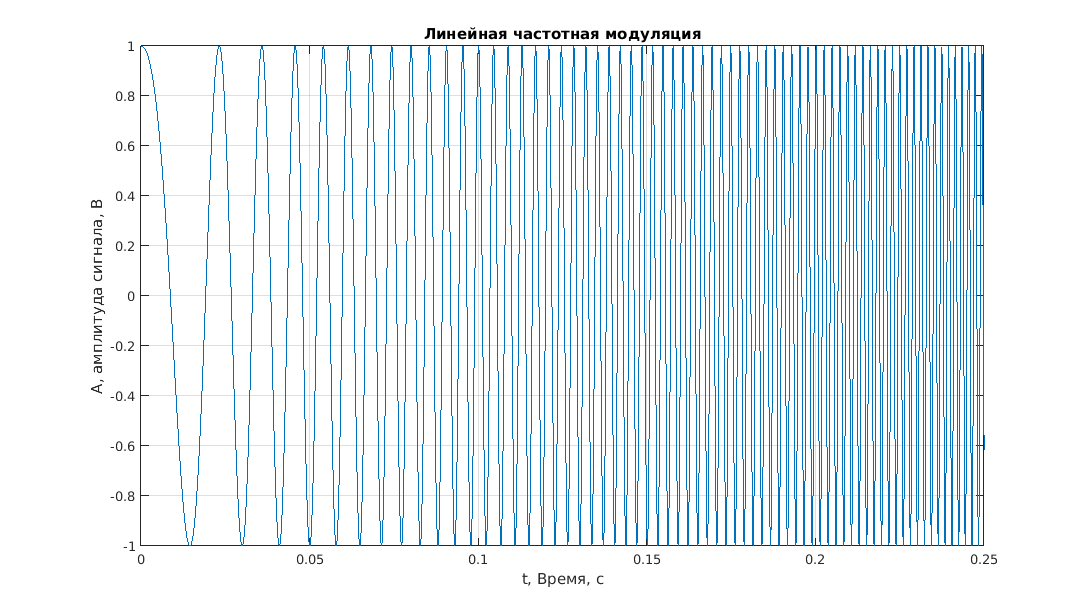


Рисунок 22 – ЛЧМ

Y = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр ЛЧМ-сигнала' );

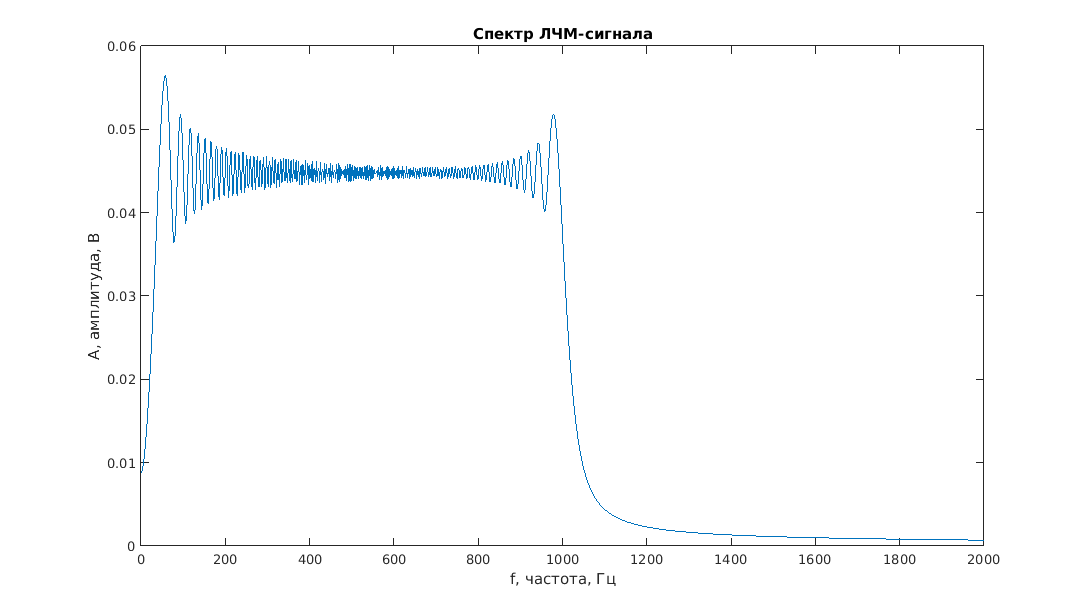


Рисунок 23 - Спектр ЛЧМ

specgram( y ) % сонограмма наглядней

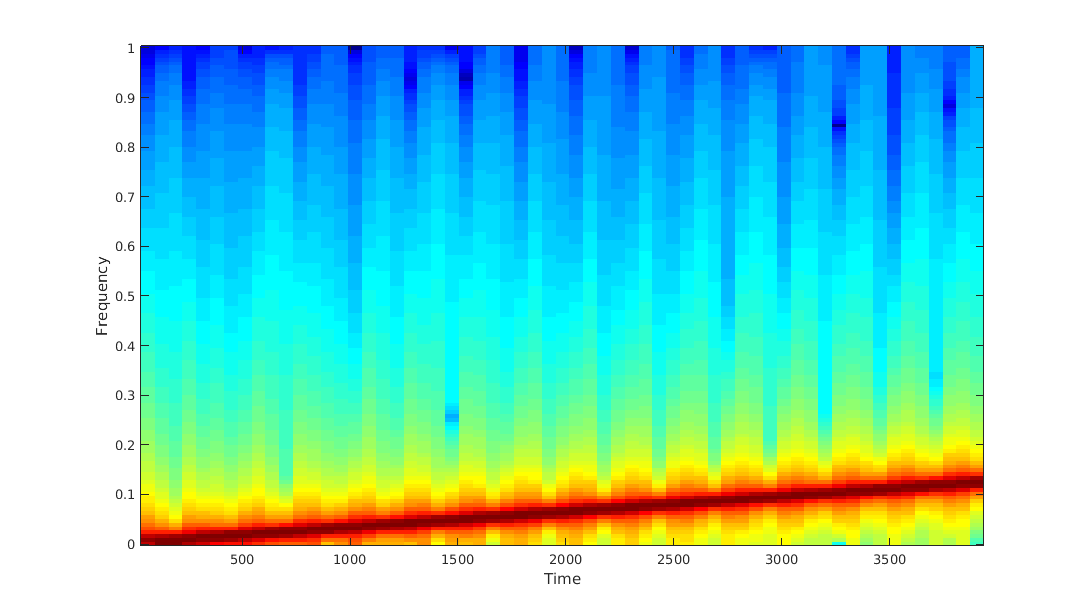


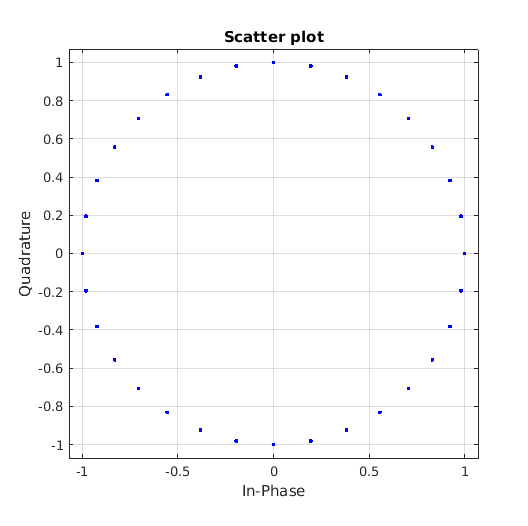
Рисунок 24 Сонограма ЛЧМ

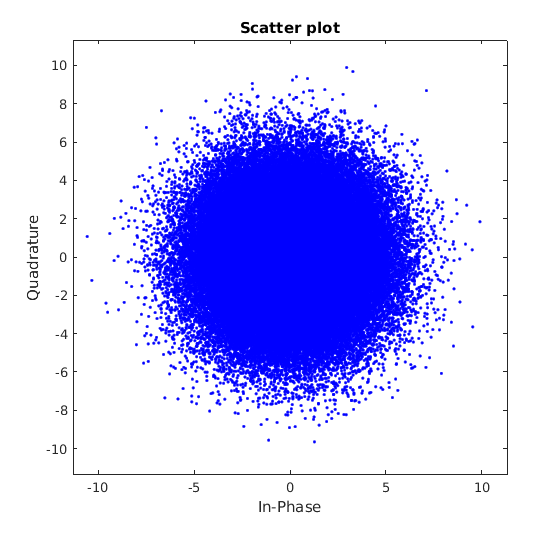
% FSK

lengthOfBit = 100; % количество отсчетов на символ

freqSep = 1000; % Валер, раз ты дрочишь теорию, то напиши, что это

modData = fskmod( decInputData, M, freqSep, lengthOfBit, Fs ); % модулируем сигнал

scatterplot( modData ) % строим созвездие модулированного сигнала



grid on

scatterplot( awgn( modData, -10 ) ) % созвездие зашумленного сигнала

scatterplot( fskdemod( awgn( modData, -10 ), M, freqSep, lengthOfBit, Fs ) ) % созвездие принятого сигнала

grid on

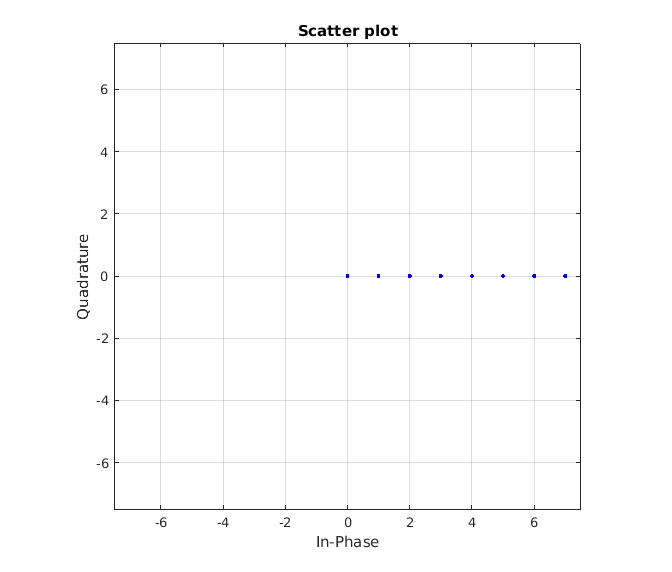


Рисунок 25 - Созвездие принятого сигнала

% моделируем передачу

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

decErrors = []'; % количества символьных ошибок

decErrorProbabilities = []'; % и вероятностей возникновения символьных ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

decResData = fskdemod( chanData, M, freqSep, lengthOfBit, Fs ); % демодуляция сигнала

binResData = de2bi( decResData ); % представляем битами

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( binInputData, binResData );

[ decErrors( end + 1 ), decErrorProbabilities( end + 1 ) ] = symerr( decInputData, decResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче FSK сигнала' );

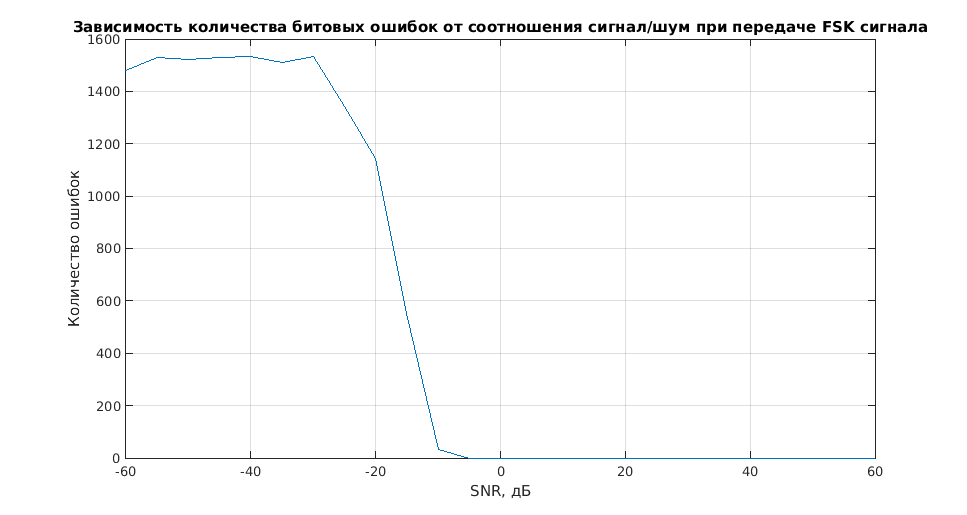


Рисунок 26 - Зависимость количества битовых ошибок

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения битовой ошибки от соотношения сигнал/шум' );

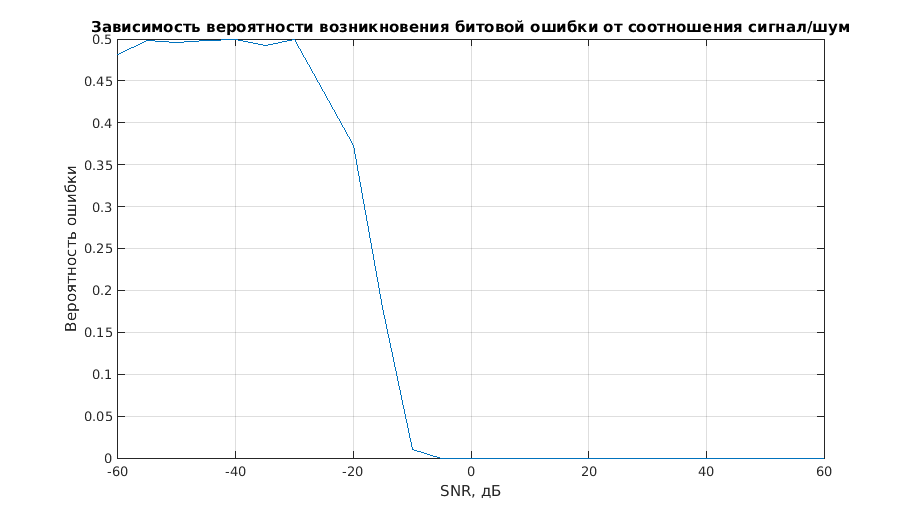


Рисунок 27 - Зависимость вероятности возникновения ошибки

plot( snrs, decErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче FSK сигнала' );

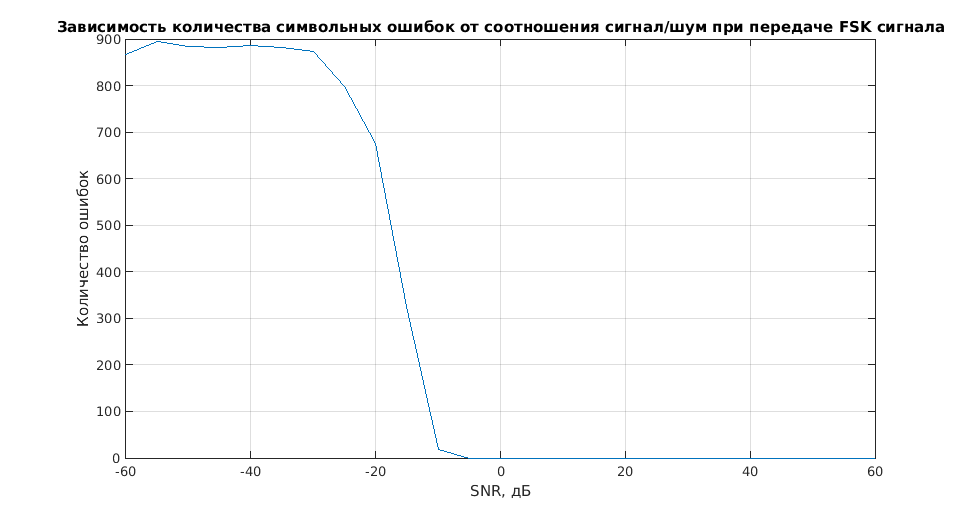


Рисунок 28 - Зависимость количества символьных ошибок от соотношения сигнал/шум FSK

plot( snrs, decErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения символьной ошибки от соотношения сигнал/шум' );

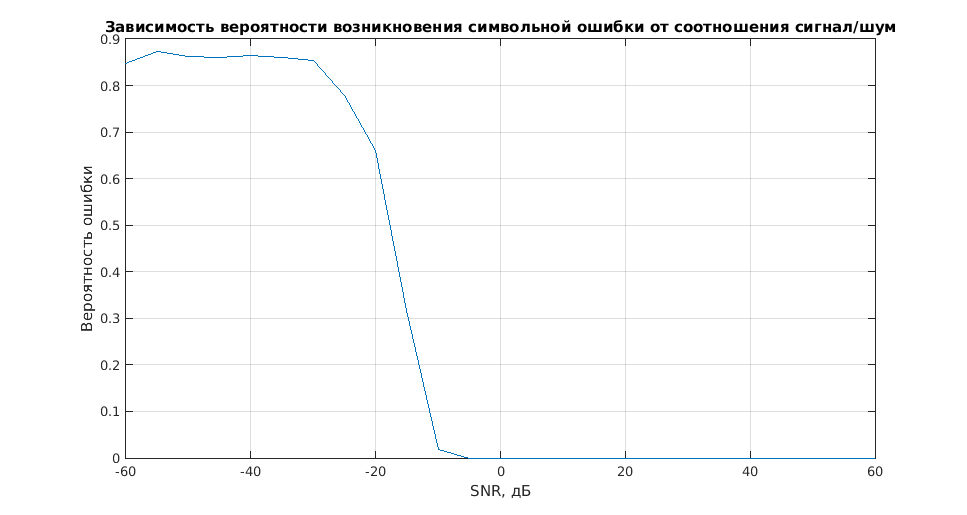


Рисунок 29 - Зависимость вероятности возникновения символьных ошибок от соотношения сигнал/шум

% MSK

lengthOfBit = 20; % количество отсчетов на символ

modData = mskmod( [binInputData( 1, : ) binInputData( 2, : ) binInputData( 3, : )], lengthOfBit ); % модулируем

scatterplot( modData ) % созвездие

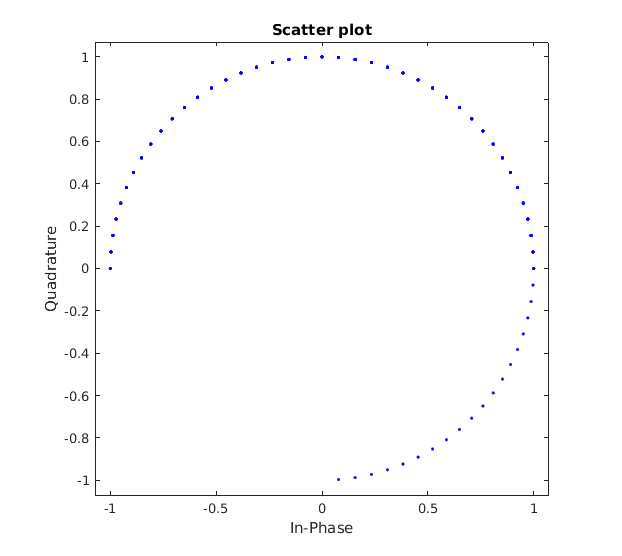


Рисунок 30

scatterplot( awgn( modData, 0 ) ) % созвездие сигнала в канале

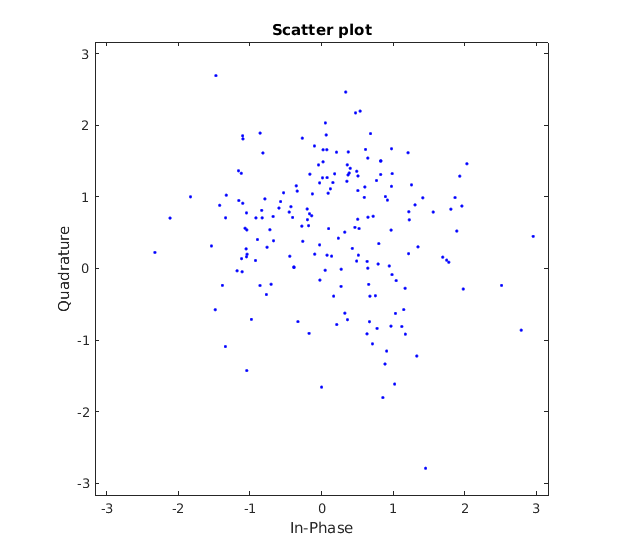


Рисунок 31

% моделируем передачу

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

binResData = mphaseVectorskdemod( chanData, lengthOfBit ); % демодуляция сигнала

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( [ binInputData( 1, : ) binInputData( 2, : ) binInputData( 3, : ) ], binResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче MSK сигнала' );

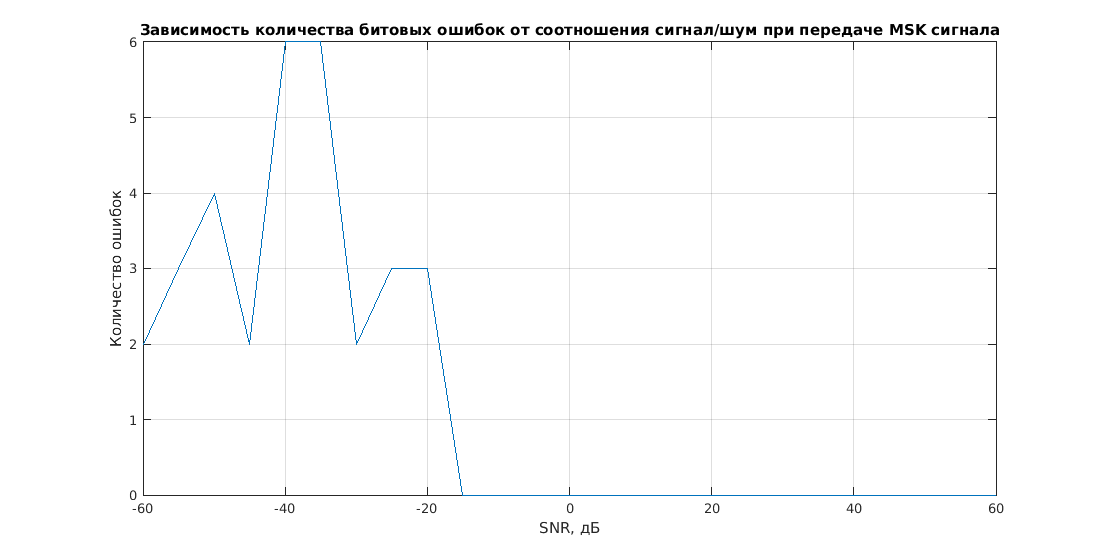


Рисунок 32 - Зависимость битовых ошибок

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки на бит' );

title( 'Зависимость вероятности битовых ошибок от соотношения сигнал/шум при передаче MSK сигнала' );

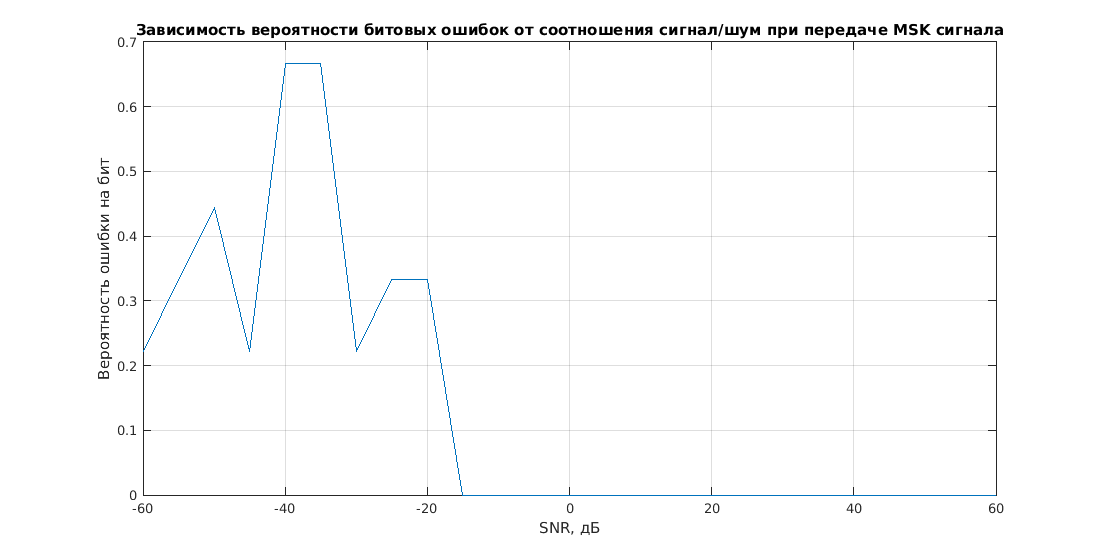


Рисунок 33 - Зависимость вероятности битовых ошибок

% Фазовая модуляция

phasedev = pi / 2; % задаем девиацию фазы

y = pmmod( s, Fm, Fs, phasedev ); % модулируем

plot( t, y ); % строим график

xlabel( 't, Время, с' );

ylabel( 'А, амплитуда сигнала, В' );

title( 'Фазовая модуляция' );

grid on

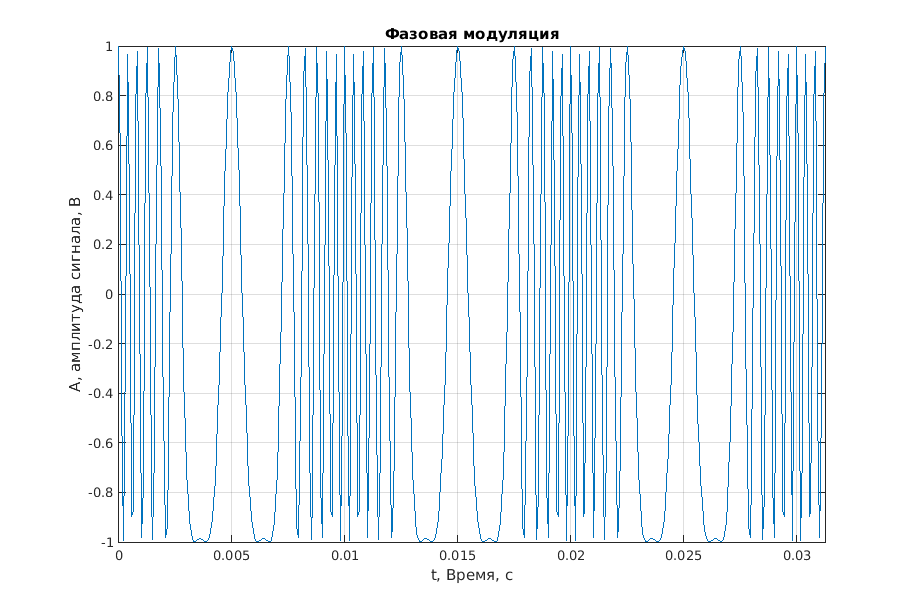


Рисунок 34 - Фазовая модуляция

Y = 2 \* abs( fft( y ) / length( y ) ); % строим спектр

freqs = Fs \* ( 0 : length( y ) / 2 ) / length( y ); % вектор частот

plot( freqs, Y( 1 : length( y ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр фазомодулированного сигнала' );

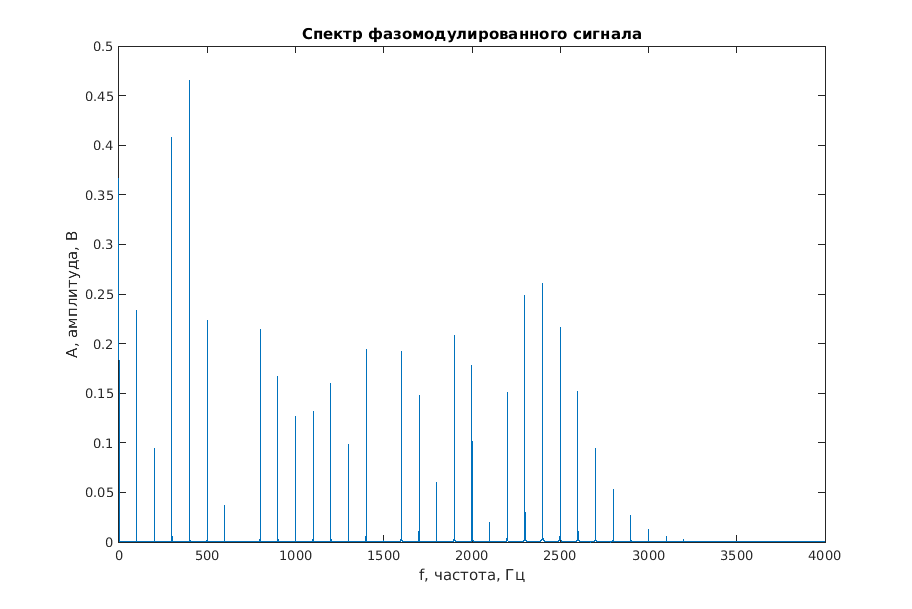


Рисунок 35 - Спектр ФМ-сигнала

% PSK

modData = pskmod( decInputData, M ); % модулируем

scatterplot( modData ) % созвездие

grid on

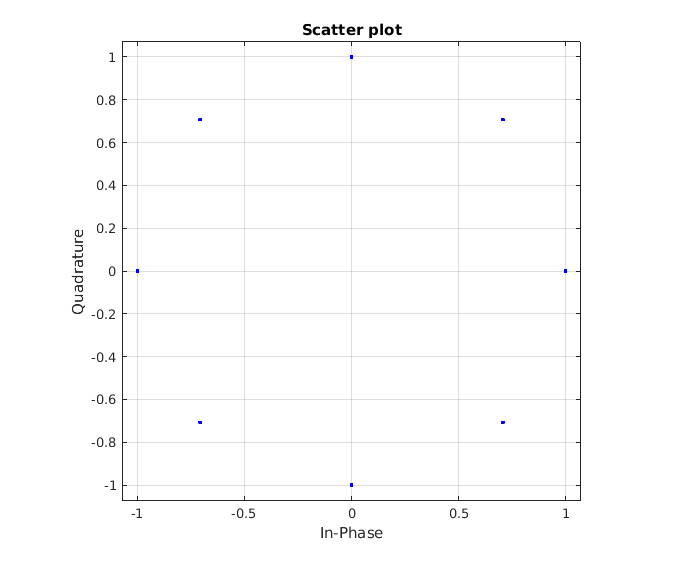


Рисунок 36 - PSK

scatterplot( awgn( modData, -10 ) ) % созвездие при передаче

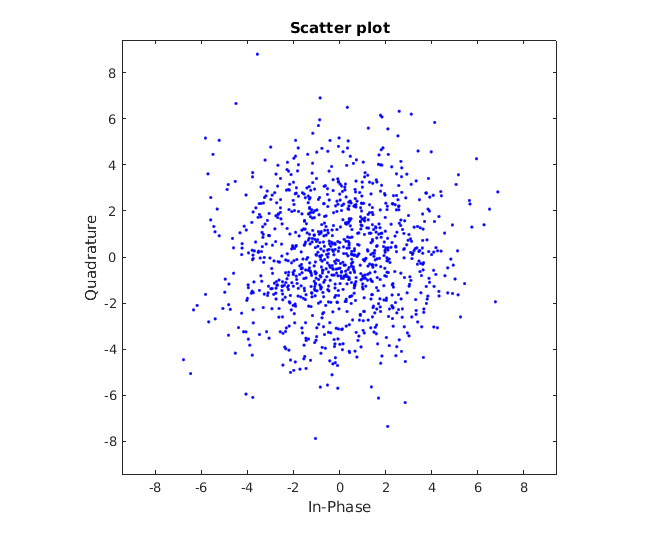


Рисунок 37

% моделируем передачу

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

decErrors = []'; % количества символьных ошибок

decErrorProbabilities = []'; % и вероятностей возникновения символьных ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

decResData = pskdemod( chanData, M ); % демодуляция сигнала

binResData = de2bi( decResData ); % представляем битами

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( binInputData, binResData );

[ decErrors( end + 1 ), decErrorProbabilities( end + 1 ) ] = symerr( decInputData, decResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче PSK сигнала' );

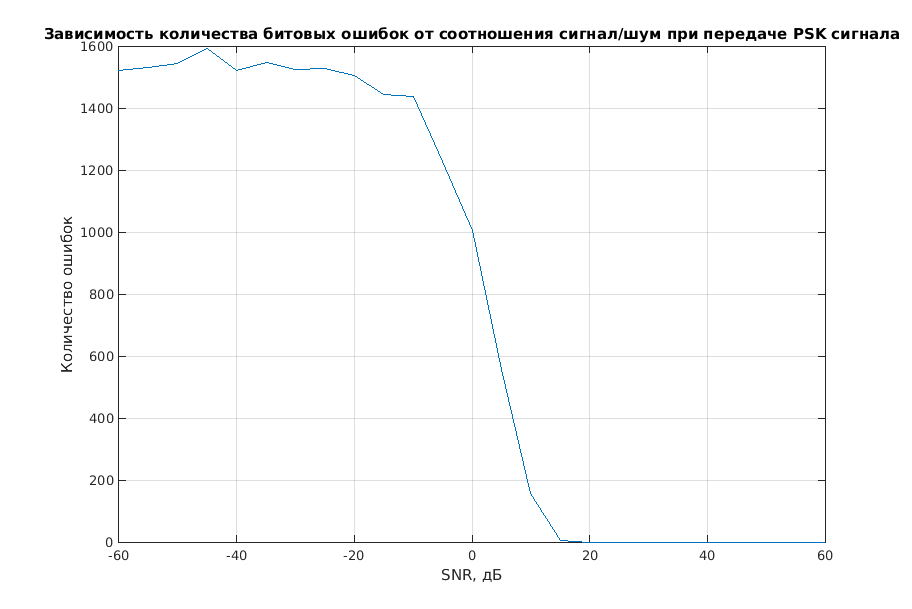


Рисунок 38 - Зависимость количества битовых ошибок

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения битовой ошибки от соотношения сигнал/шум' );

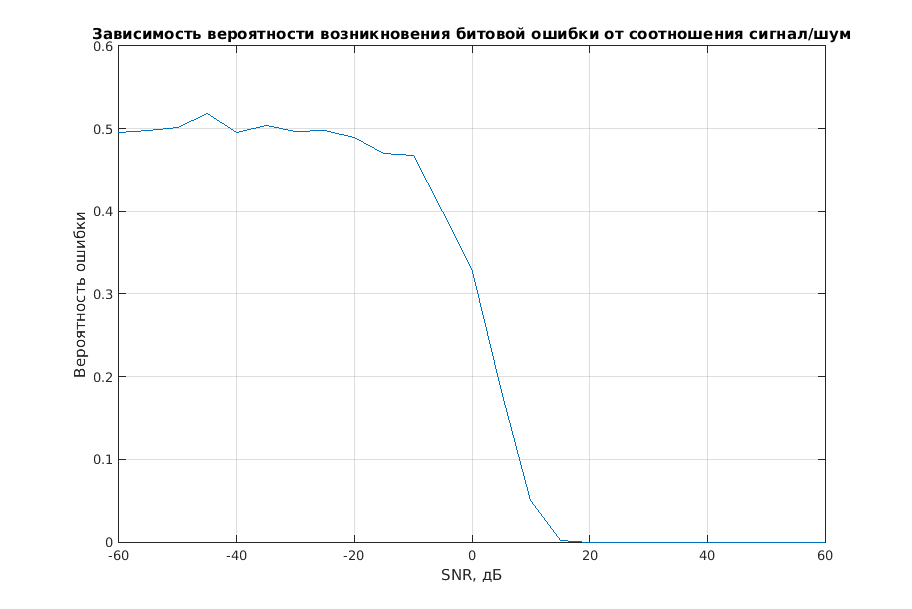


Рисунок 39

plot( snrs, decErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче РSK сигнала' );

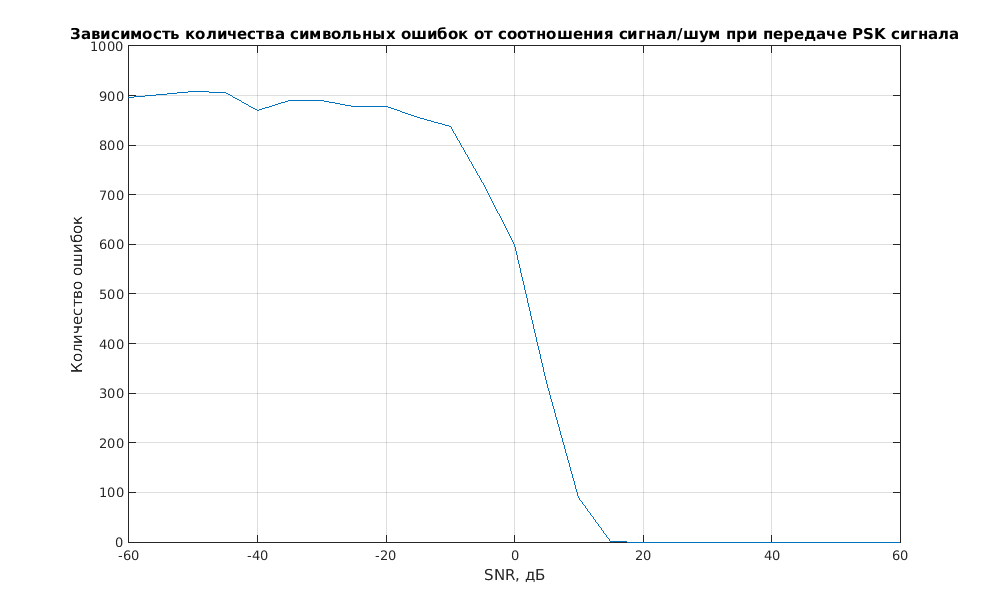


Рисунок 40

plot( snrs, decErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения символьной ошибки от соотношения сигнал/шум' );

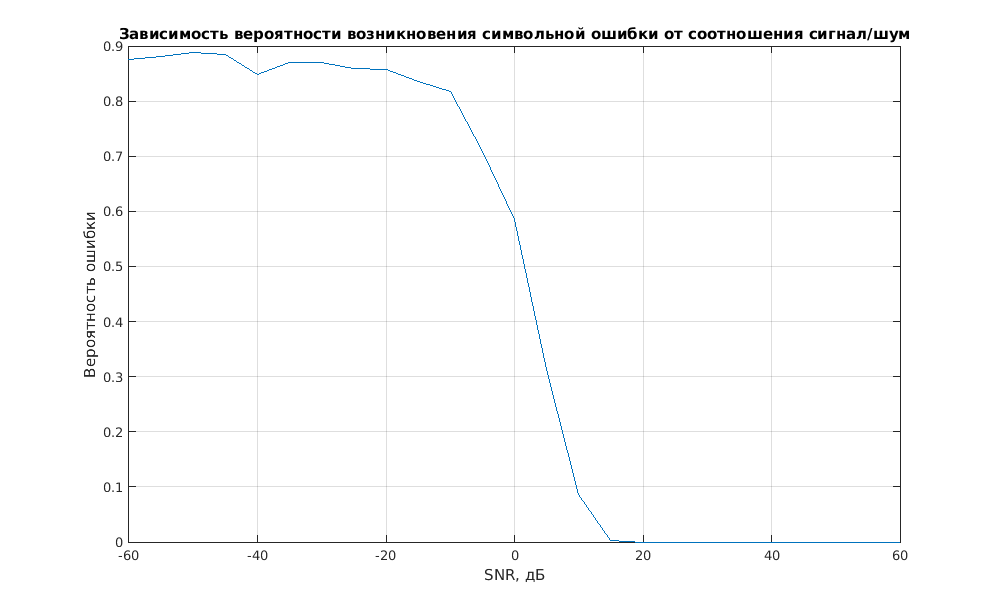


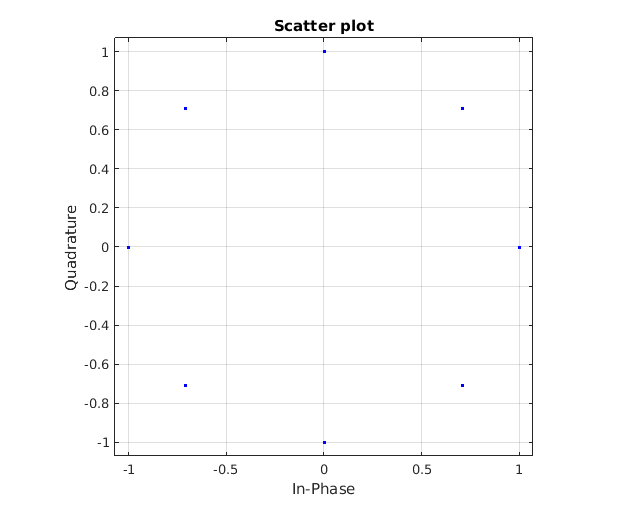
Рисунок 41

% PPM - это то же, что DPSK, насколько я понял

modData = dpskmod( decInputData, M ); % модулируем

scatterplot( modData )

grid on



% моделируем передачу

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

decErrors = []'; % количества символьных ошибок

decErrorProbabilities = []'; % и вероятностей возникновения символьных ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

decResData = dpskdemod( chanData, M ); % демодуляция сигнала

binResData = de2bi( decResData ); % представляем битами

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( binInputData, binResData );

[ decErrors( end + 1 ), decErrorProbabilities( end + 1 ) ] = symerr( decInputData, decResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче PPM сигнала' );

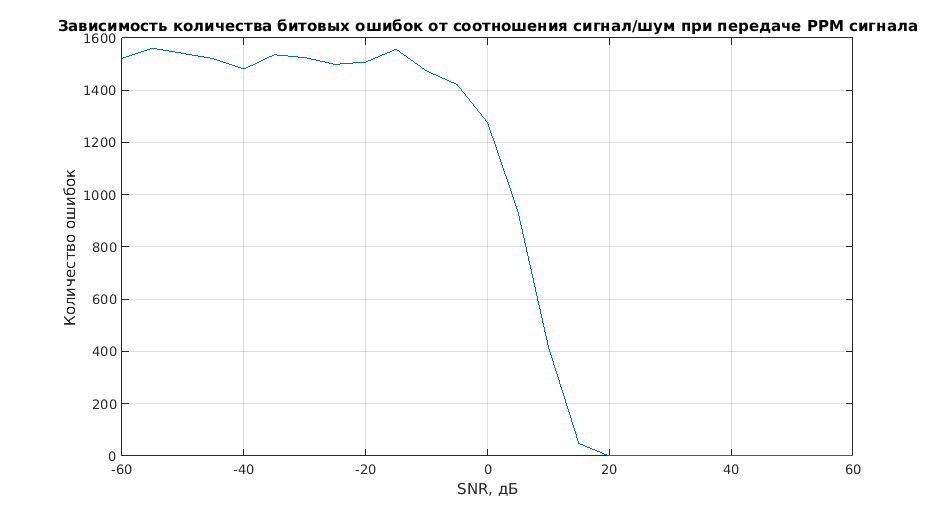


Рисунок 42

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения битовой ошибки от соотношения сигнал/шум' );

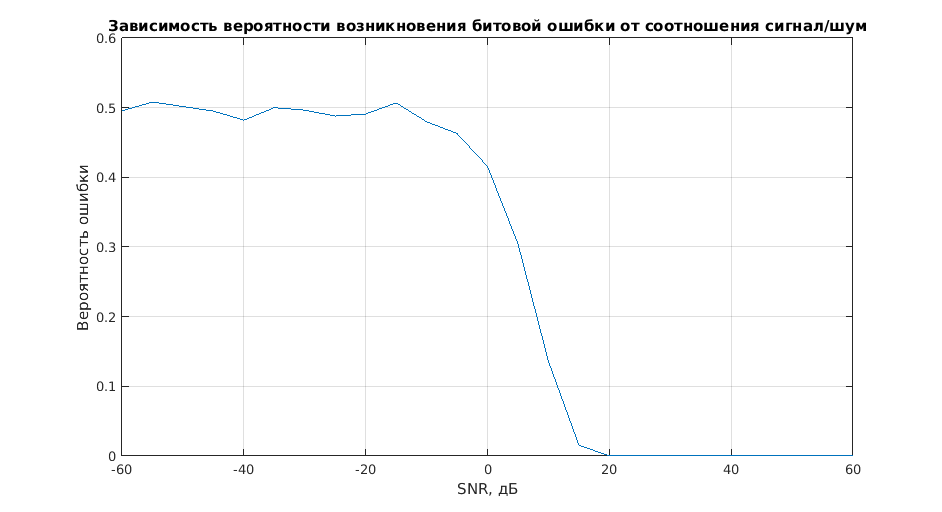


Рисунок 43

plot( snrs, decErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче РPM сигнала' );

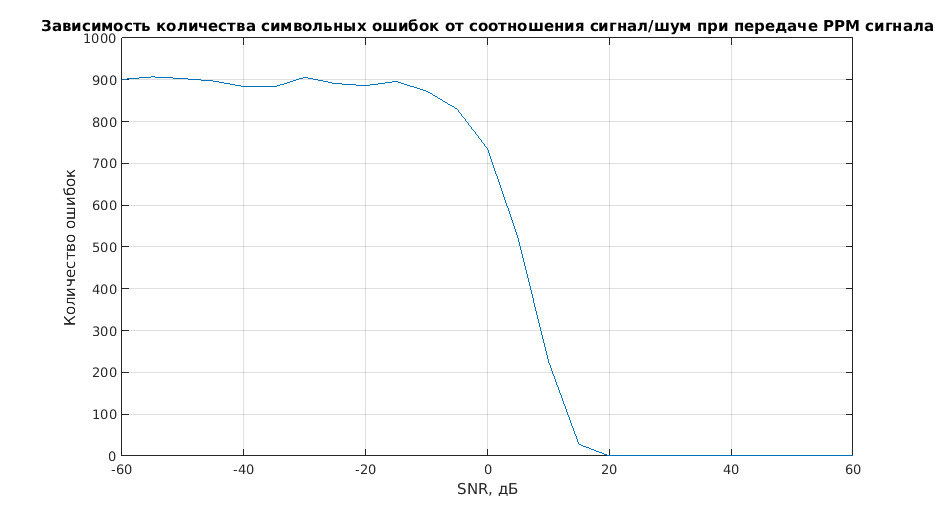


Рисунок 44

plot( snrs, decErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения символьной ошибки от соотношения сигнал/шум' );

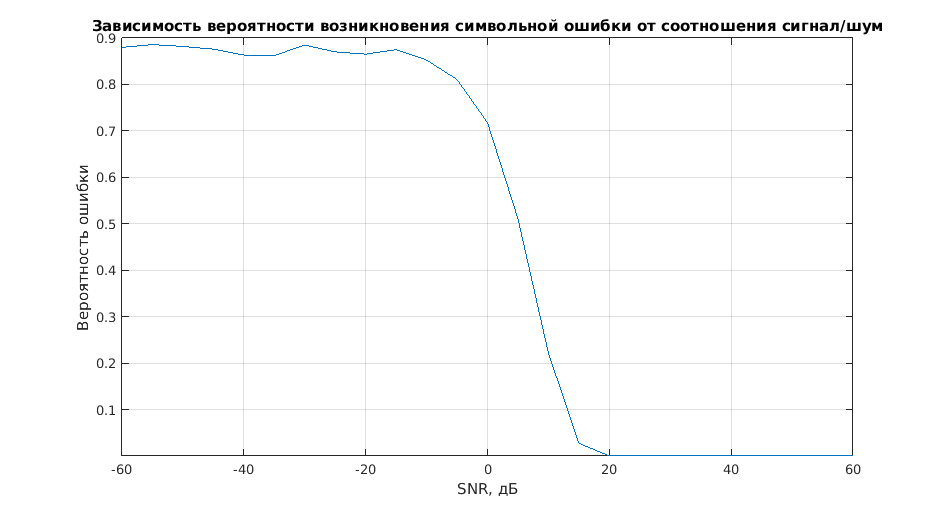


Рисунок 45

% OQPSK

modData = oqpskmod( mod( decInputData, 4 ) );

scatterplot( modData )

grid on

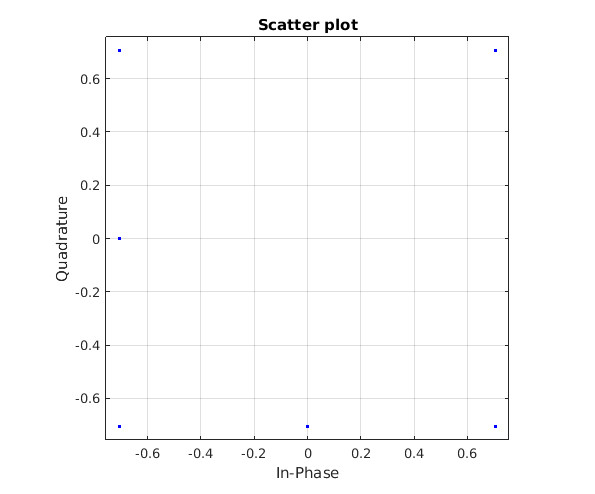


Рисунок 46

% моделируем передачу

snrs = []'; % инициализируем векторы соотношения с/ш,

binErrors = []'; % количества битовых ошибок

binErrorProbabilities = []'; % и вероятностей возникновения битовых ошибок

decErrors = []'; % количества символьных ошибок

decErrorProbabilities = []'; % и вероятностей возникновения символьных ошибок

for snr = -60 : 5 : 60

chanData = awgn( modData, snr ); % сигнал в канале передачи

decResData = oqpskdemod( chanData ); % демодуляция сигнала

binResData = de2bi( decResData ); % представляем битами

% подсчет ошибок

[ binErrors( end + 1 ), binErrorProbabilities( end + 1 ) ] = biterr( de2bi( mod( decInputData, 4 ) ) , binResData );

[ decErrors( end + 1 ), decErrorProbabilities( end + 1 ) ] = symerr( decInputData, decResData );

snrs( end + 1 ) = snr;

end

% строим графики

plot( snrs, binErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества битовых ошибок от соотношения сигнал/шум при передаче OQPSK сигнала' );

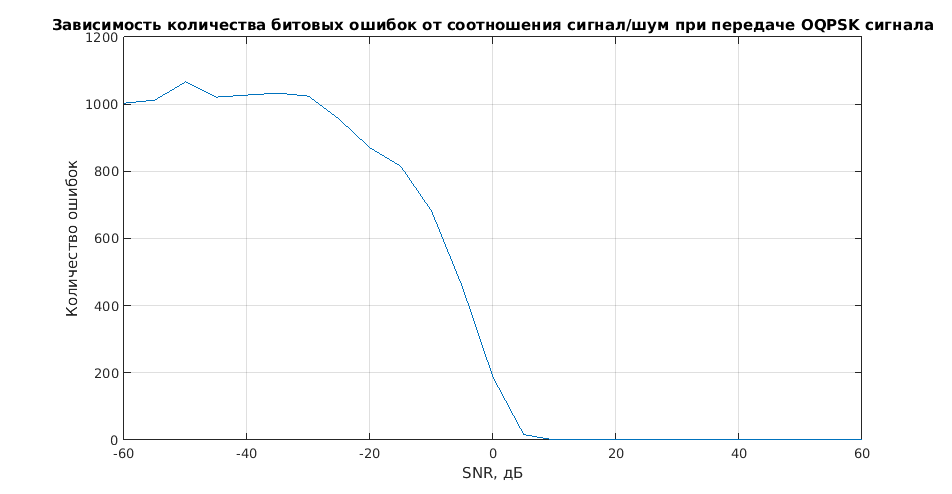


Рисунок 47

plot( snrs, binErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения битовой ошибки от соотношения сигнал/шум' );

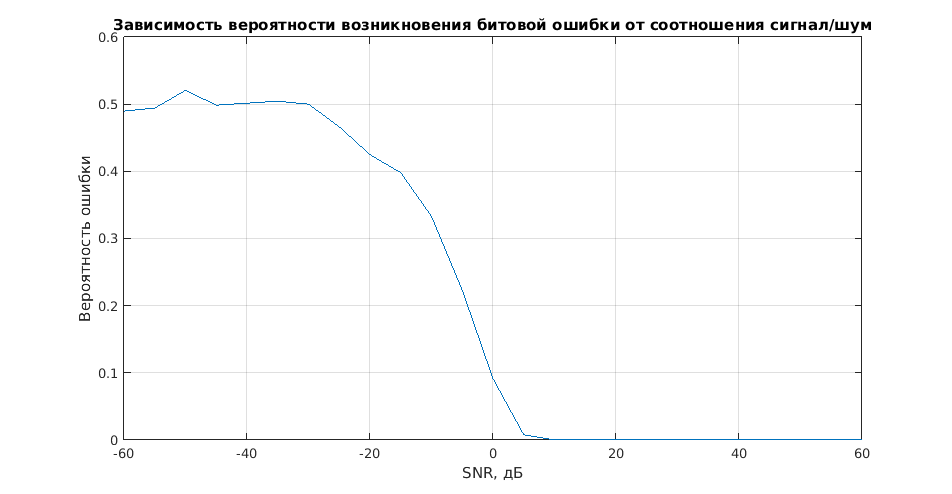


Рисунок 48

plot( snrs, decErrors )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Количество ошибок' );

title( 'Зависимость количества символьных ошибок от соотношения сигнал/шум при передаче OQPSK сигнала' );

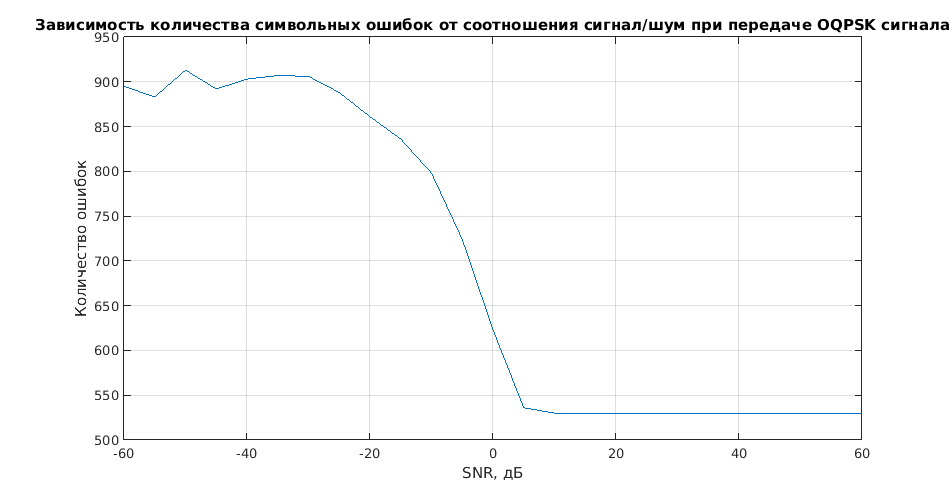


Рисунок 49

plot( snrs, decErrorProbabilities )

grid on

xlabel( 'SNR, дБ' );

ylabel( 'Вероятность ошибки' );

title( 'Зависимость вероятности возникновения символьной ошибки от соотношения сигнал/шум' );

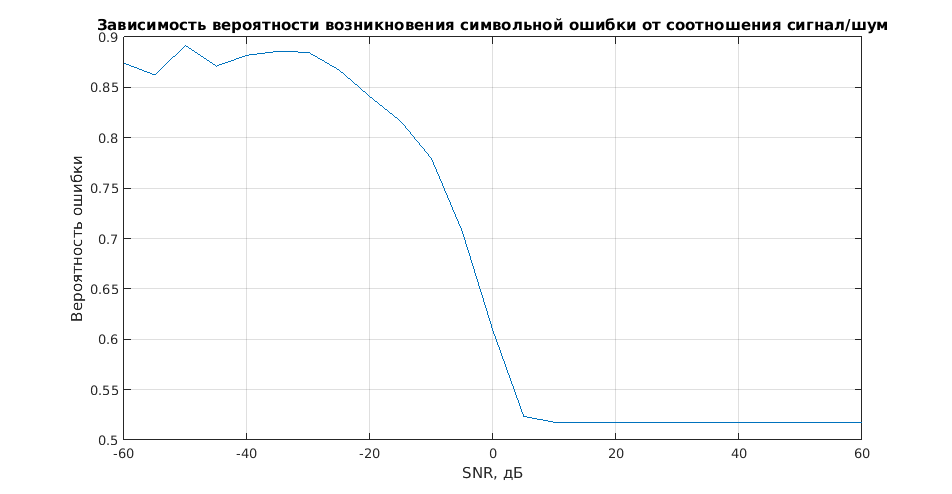


Рисунок 50

% Дельта-модуляция

partition = [ -10 : 10 ]; % параметры дельта-модуляции

codebook = [ -9.5 : 9.5 ];

[ partition, codebook ] = lloyds( s, codebook ); % оптимизация

predictor = [ 0 1 ]; % формула

y = dpcmenco( s, codebook, partition, predictor ); % модулируем

x = dpcmdeco( y, codebook, predictor ); % демодуляция

plot( t, s )

hold on

plot( t, x )

legend( 'До модуляции', 'После демодуляции' )

xlabel( 't, Время, с' )

ylabel( 'A, Амплитуда, В' )

title( 'Дельта-модуляция' )

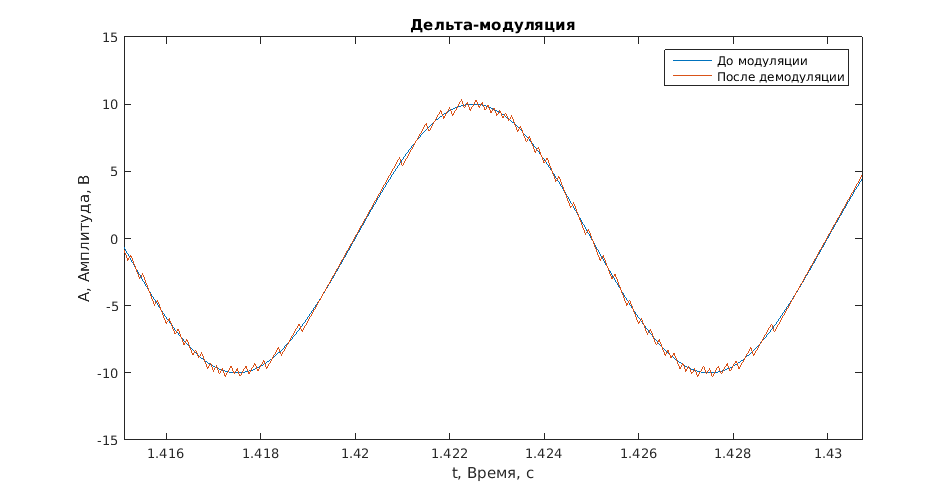


Рисунок 51

Y = 2 \* abs( fft( x ) / length( x ) ); % строим спектр

freqs = Fs \* ( 0 : length( x ) / 2 ) / length( x ); % вектор частот

plot( freqs, Y( 1 : length( x ) / 2 + 1 ) ); % график спектра

xlabel( 'f, частота, Гц' );

ylabel( 'A, амплитуда, В' );

title( 'Спектр дельтамодулированного сигнала' );

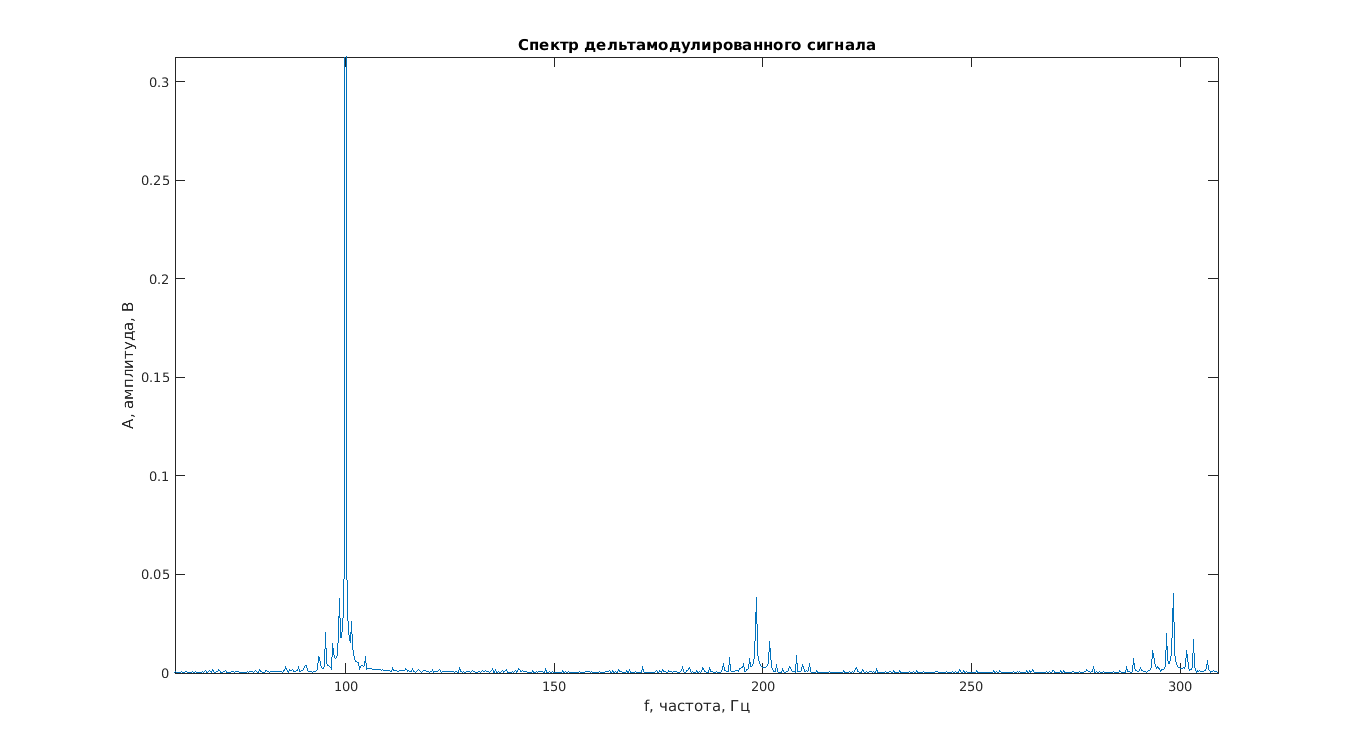


Рисунок 52

% PCM

[ index, quantized ] = quantiz( s, partition, codebook ); % квантуем

plot( t, s )

hold on

plot( t, quantized )

legend( 'Оригинальный сигнал', 'Квантованный сигнал' )

xlabel( 'Время t, с' )

ylabel( 'Амплитуда сигнала А, В' )

title( 'Импульсно-кодовая модуляция' )

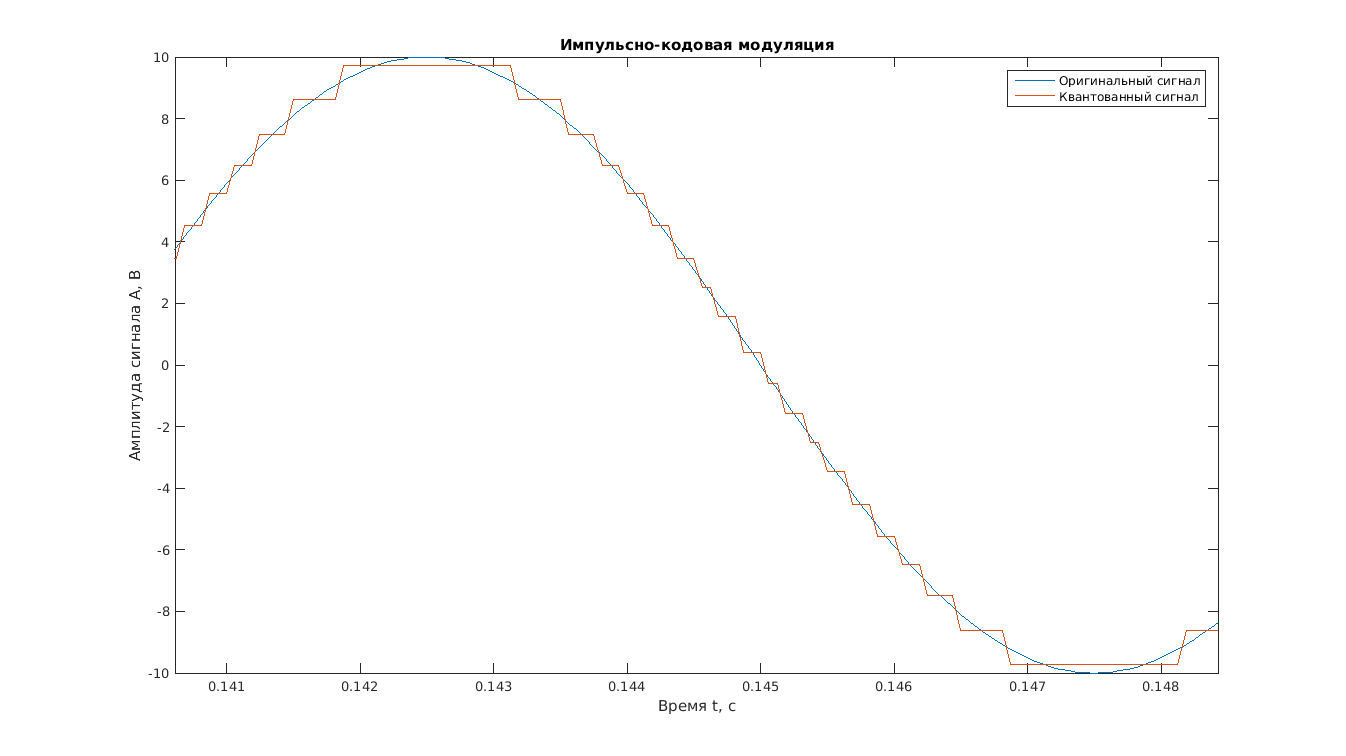


Рисунок 53

% PWM

m = Am \* sawtooth( 2 \* pi \* Fm \* t );

for i = 1 : length( s )

if s( i ) >= m( i )

sig( i ) = 1;

else

sig( i ) = 0;

end

end

plot( t, s )

hold on

plot( t, sig )

grid on

legend( 'Сигнал', 'Модулированный сигнал' )

xlabel( 'Время t, c' )

ylabel( 'Амплитуда А, В' )

title( 'Широтно-импульсная модуляция' )

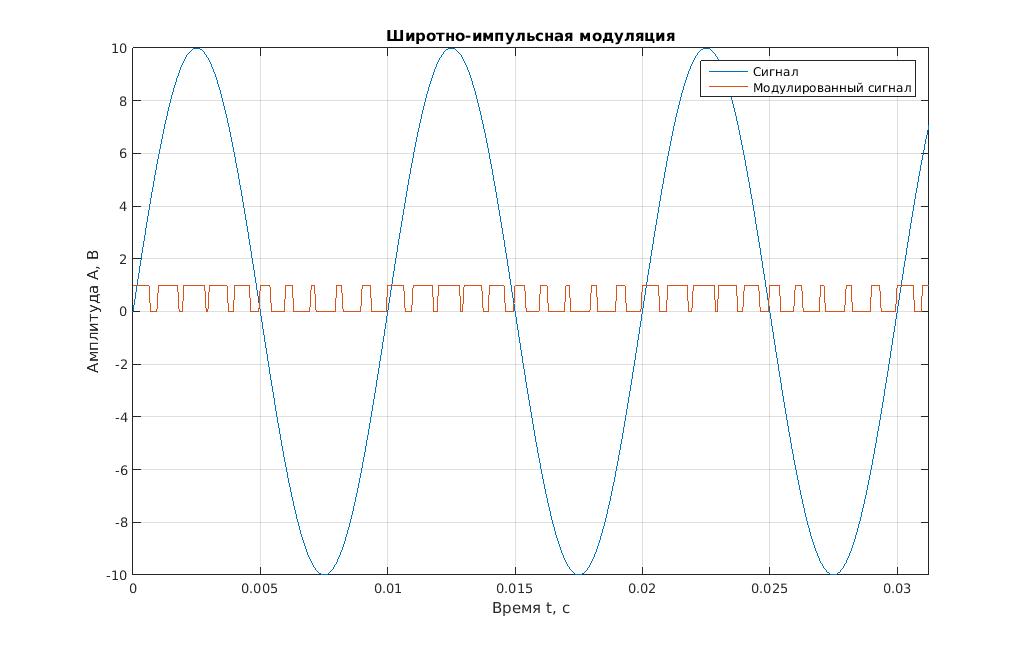


Рисунок 54 - ШИМ