Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Телекоммуникационные технологии

Лабораторная работа №6:
Цифровая модуляция

Работу выполнил:
Сергеев А.А.
Группа: 33531/2
Преподаватель:
Богач Н.В.

Санкт-Петербург
2019

Содержание

1 Цель

Изучение методов модуляции цифровых сигналов.

2 Постановка задачи

1. Получить сигналы BPSK, PSK, OQPSK, genQAM, MSK, MFSK модуляторов,
2. Построить их сигнальные созвездия.
3. Провести сравнение изученных методов модуляции цифровых сигналов.

3 Теоретический раздел

Числа при передаче информации в цифровой форме с периодом T поступают от источника информации и называются символами (symbol), а частота передачи символов – символьной скоростью (symbol rate) $f_T = \frac{1}{T}$. В практике передачи данных распространена двоичная последовательность символов, где числа передаются значениями 0 и 1.
Каждому из возможных символов устанавливается определенный набор параметров несущего колебания, которые поддерживаются постоянными на интервале T до прихода следующего символа. Это означает преобразование последовательности чисел в ступенчатый сигнал, который используется в качестве модулирующего сигнала. Соответственно, параметры несущего колебания, на которые переносится сигнал, меняются скачкообразно. Такой способ модуляции несущей обычно называется манипуляцией (keying).
В зависимости от изменяемых параметров манипуляцию разделяют на амплитудную, фазовую, частотную и квадратурную.
При частотной манипуляции (ЧМн; английский термин – frequency shift keying, FSK) каждому возможному значению передаваемого символа сопоставляется своя частота. В течение каждого символьного интервала передается гармоническое колебание с частотой, соответствующей текущему символу.
MSK (minimum shift key) – манипуляция с минимальным сдвигом частоты. Разность частот сигналов, соответствующих различным битам, равна половине скорости передачи информации. Манипуляция называется с минимальным сдвигом частоты, так как значение $\Delta f = \frac{1}{2T}$ является минимальной разностью частот, при котором сигналы с различными частотами, являются ортогональными.
MFSK - Многопозиционная частотная манипуляция. Метод манипуляции, при котором N дискретных состояних входного сигнала преобразуются в набор из N фиксированных частот, передаваемых параллельно или последовательно.
Амплитудная манипуляция (АМн; английский термин – amplitude shift keying, ASK), при которой скачкообразно меняется амплитуда несущего колебания, является частным случаем квадратурной манипуляции.
Фазовая манипуляция (ФМн; английский термин – phase shift keying, PSK), при которой скачкообразно меняется фаза несущего колебания, тоже является частным случаем квадратурной манипуляции.
Фазоманипулированный сигнал имеет следующий вид:

$$s_m(t) = g(t)cos[2\pi f_c t + \phi_m(t)],$$

где $g(t)$ определяет огибающую сигнала: $\phi_m(t)$ является модулирующим сигналом. $\phi_m(t)$ может принимать M дискретных значений. f_c – частота несущей; t – время.
Если $M = 2$, то фазовая манипуляция называется двоичной фазовой манипуляцией (BPSK, B-Binary – 1 бит на 1 смену фазы), если $M = 4$ – квадратурной фазовой манипуляцией (QPSK, Q-Quadro – 2 бита на 1 смену фазы).
ini_phase задает начальную фазу комплексной огибающей в радианах OQPSK (Offset QPSK) – Четырехфазная ФМ со сдвигом. Позволяет избежать скачков фазы на 180° и, следовательно, глубокой модуляции огибающей. Формирование сигнала в модуляторе OQPSK происходит так же, как и в модуляторе ФМ-4, за исключением того, что манипуляционные элементы информационных последовательностей $x(t)$ и $y(t)$ смещены во времени на длительность одного элемента T.
При квадратурной манипуляции (КАМн; английский термин - quadrature amplitude shift keying, QASK) каждому из возможных значений дискретного символа C_k ставится в соответствие пара величин – амплитуды синфазной и квадратурной составляющих либо, что эквивалентно, амплитуда и начальная фаза несущего колебания:

$$C_k -> (a_k, b_k), s(t) = a_k cos\omega_0 t + b_k sin\omega_0 t, kT <= t < (k+1)T$$

Параметры аналогового колебания, сопоставленные дискретному символу C_k, удобно представлять в виде комплексного числа в алгебраической $(a_k + jb_k)$ или экспоненциальной $(A_k exp(j\phi_k))$ форме. Совокупность

этих комплексных чисел для всех возможных значений дискретного символа называется сигнальным созвездием.
При представлении дискретного символа комплексным числом C_k сигнал с квадратурной манипуляцией можно записать следующим образом:

$$s(t) = Re(C_k exp(-j\omega_0 t)), kT <= t < (k+1)T.$$

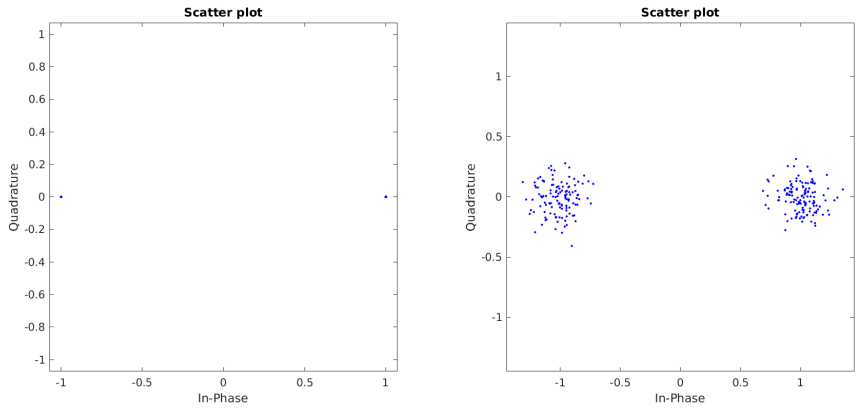
4 Ход работы

4.1 BPSK

Генерируем случайное сообщение, получаем сигнал от двоичного фазового модулятора и сигнальное созвездие $BPSK$:

```
function [SNR_BPSK, BER_SNR_BPSK] = bpsk
    %BPSK
    msg                             = randi([0 1], [1 256]);
    signal_modulation               = pskmod(msg, 2);
    signal_noise                    = awgn(signal_modulation, 15);
    draw_scatterplot(signal_modulation);
    draw_scatterplot(signal_noise);
    errors                          = randerr(1, 256, 13);
    signal_modulation_errors        = (signal_modulation + errors);
    signal_demodulation             = pskdemod(signal_modulation_errors, 2);
    [numberBPSK, ratioBPSK]         = symerr(msg, signal_demodulation);
    [BITnumberBPSK, BITratioBPSK] = biterr(msg, signal_demodulation);
    BER_SNR_BPSK                    = [];
    SNR_BPSK                        = [];
    for m = -15:1:15
        SNR       = m;
        ber_sum = 0;
        rep       = 100;
        for v = 1:rep
            signal_noise = awgn(signal_modulation, SNR, 'measured');
            demod_BPSK   = pskdemod(signal_noise, 2);
            [err, BER]   = biterr(msg, demod_BPSK);
            ber_sum      = ber_sum + BER;
        end
        BER           = ber_sum./rep;
        BER_SNR_BPSK = [BER_SNR_BPSK, BER];
        SNR_BPSK      = [SNR_BPSK, SNR];
    end
end
```

Полученное сигнальное созвездие и зашумлённое сигнальное созвездие $BPSK$:



Число ошибочных символов: 0, вероятность ошибки на символ: 0.

4.2 PSK

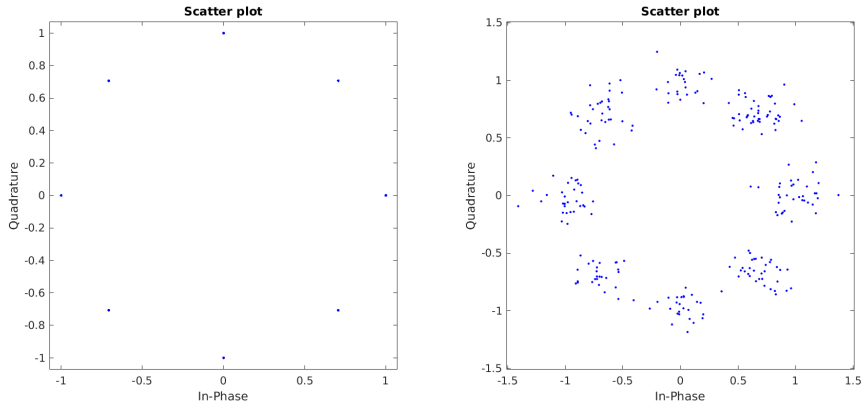
Генерируем случайное сообщение, получаем сигнал от двоичного фазового модулятора и сигнальное созвездие PSK:

```
function [SNR_PSK, BER_SNR_PSK] = psk
    %PSK
```

```
    msg                          = randi([0 7], [1 256]);
    signal_modulation            = pskmod(msg, 8);
    signal_noise                 = awgn(signal_modulation, 15);
    draw_scatterplot(signal_modulation);
    draw_scatterplot(signal_noise);
    errors                       = randerr(1, 256, 13);
    signal_modulation_errors     = signal_modulation+errors;
    signal_demodulation          = pskdemod(signal_modulation_errors, 8);
    [numberPSK, ratioPSK]        = symerr(msg, signal_demodulation);
    [BITnumberPSK, BITratioPSK] = biterr(msg, signal_demodulation);
    BER_SNR_PSK                  = [];
    SNR_PSK                      = [];

    for m = -15:1:15
        SNR      = m;
        ber_sum = 0;
        rep      = 100;
        for v = 1:rep
            signal_noise  = awgn(signal_modulation, SNR, 'measured');
            demod_PSK     = pskdemod(signal_noise, 8);
            [err, BER]    = biterr(msg, demod_PSK);
            ber_sum       = ber_sum + BER;
        end
        BER          = ber_sum./rep;
        BER_SNR_PSK = [BER_SNR_PSK, BER];
        SNR_PSK      = [SNR_PSK, SNR];
    end
end
```

Полученное сигнальное созвездие и зашумлённое сигнальное созвездие PSK:



Число ошибочных символов: 4, вероятность ошибки на символ: 0.0156.

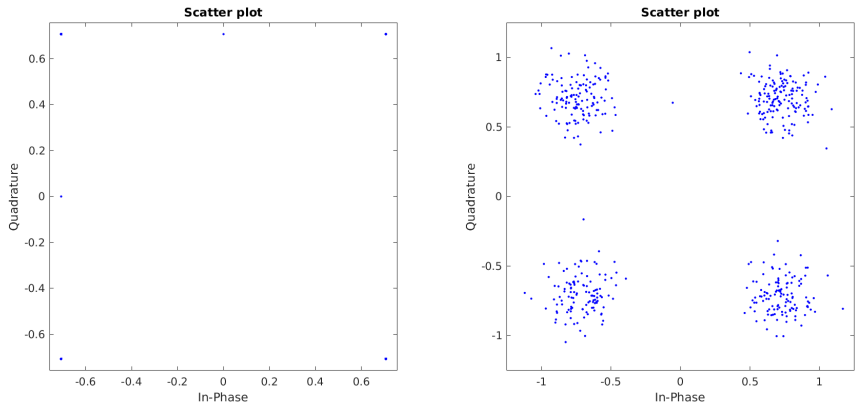
4.3 OQPSK

Генерируем случайное сообщение, получаем сигнал от двоичного фазового модулятора и сигнальное созвездие $OQPSK$:

```
function [SNR_OQPSK, BER_SNR_OQPSK] = oqpsk
    %OQPSK
    msg                          =randi([0 3], [1 256]);
    signal_modulation            = oqpskmod(msg, pi/2);
    signal_noise                 = awgn(signal_modulation, 15);
    draw_scatterplot(signal_modulation);
    draw_scatterplot(signal_noise);
    errors                       = randerr(1,513,13);
    signal_modulation_errors     = signal_modulation + errors;
    signal_demodulation          = oqpskdemod(signal_modulation_errors, pi/2);
    [numberOQPSK, ratioOQPSK] = symerr(msg, signal_demodulation);
    BER_SNR_OQPSK                = [];
    SNR_OQPSK                    = [];
    for m = -15:1:15
        SNR      = m;
        ber_sum = 0;
        rep      = 100;
        for v = 1:rep
            signal_noise = awgn(signal_modulation, SNR, 'measured');
            demod_OQPSK  = oqpskdemod(signal_noise, pi/2);
```

```
            [err, BER]    = biterr(msg, demod_OQPSK);
            ber_sum       = ber_sum + BER;
        end
        BER             = ber_sum ./ rep;
        BER_SNR_OQPSK = [BER_SNR_OQPSK, BER];
        SNR_OQPSK       = [SNR_OQPSK, SNR];
    end
end
```

Полученное сигнальное созвездие и зашумлённое сигнальное созвездие $OQPSK$:



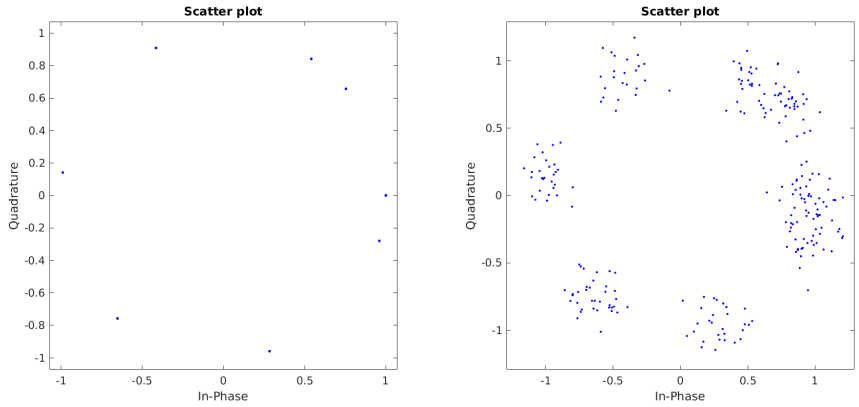
Число ошибочных символов: 0, вероятность ошибки на символ: 0.

4.4 genQAM

Генерируем случайное сообщение, получаем сигнал от двоичного фазового модулятора и получаем сигнальное созвездие $genQAM$:

```
function [SNR_QAM, BER_SNR_QAM] = genqam
    %genQAM
    M                                = 8;
    msg                              = randi([0 M-1], [1 256]);
    signal_modulation                = genqammod(msg, exp(j * [0:M-1]));
    signal_noise                     = awgn(signal_modulation, 15);
    draw_scatterplot(signal_modulation);
    draw_scatterplot(signal_noise);
    errors                           = randerr(1, 256, 13);
    signal_modulation_errors         = signal_modulation + errors;
    signal_demodulation              = genqamdemod(signal_modulation_errors, exp(j * [0:M-1])
    ↪ );
    [numbergenQAM, ratiogenQAM]      = symerr(msg, signal_demodulation);
    [BITnumbergenQAM, BITratiogenQAM] = biterr(msg, signal_demodulation);
    BER_SNR_QAM                      = [];
    SNR_QAM                          = [];
    for m = -15:1:15
        SNR      = m;
        ber_sum = 0;
        rep      = 100;
        for v = 1:rep
            signal_noise  = awgn(signal_modulation, SNR, 'measured');
            demod_QAM     = genqamdemod(signal_noise, exp(j * [0:M-1]));
            [err, BER]    = biterr(msg, demod_QAM);
            ber_sum       = ber_sum + BER;
        end
        BER          = ber_sum ./ rep;
        BER_SNR_QAM = [BER_SNR_QAM, BER];
        SNR_QAM      = [SNR_QAM, SNR];
    end
end
```

Полученное сигнальное созвездие и зашумлённое сигнальное созвездие $genQAM$:



Число ошибочных символов: 13, вероятность ошибки на символ: 0.0508.

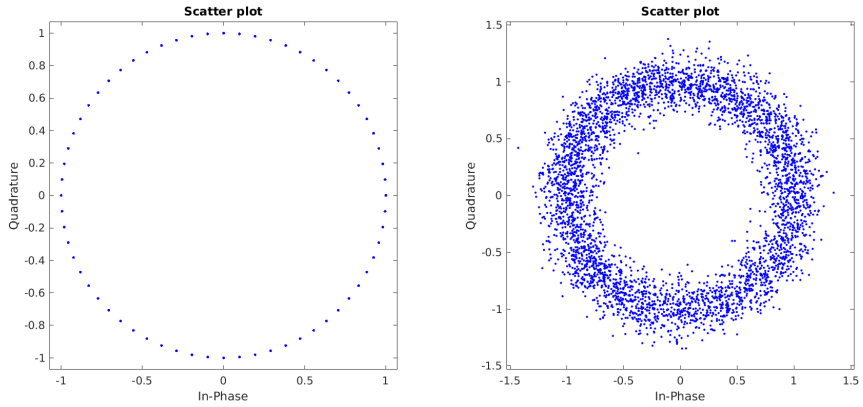
4.5 MSK

Генерируем случайное сообщение, получаем сигнал от двоичного фазового модулятора и сигнальное созвездие MSK:

```
function [SNR_MSK, BER_SNR_MSK] = msk
    %MSK
    msg                     = randi([0 1], [1 256]);
    signal_modulation       = mskmod(msg, 16);
    signal_noise            = awgn(signal_modulation, 15);
    draw_scatterplot(signal_modulation);
    draw_scatterplot(signal_noise);
    errors                  = randerr(1, 256 * 16, 13);
    signal_modulation_errors = signal_modulation + errors;
    signal_demodulation     = mskdemod(signal_modulation_errors, 16);
    [numberMSK, ratioMSK]   = symerr(msg, signal_demodulation);

    BER_SNR_MSK = [];
    SNR_MSK     = [];
    for m = -15:1:15
        SNR      = m;
        ber_sum  = 0;
        rep      = 100;
        for v = 1:rep
            signal_noise  = awgn(signal_modulation, SNR, 'measured');
            demod_MSK     = mskdemod(signal_noise, 16);
            [err, BER]    = biterr(msg, demod_MSK);
            ber_sum       = ber_sum + BER;
        end
        BER          = ber_sum ./ rep;
        BER_SNR_MSK = [BER_SNR_MSK, BER];
        SNR_MSK      = [SNR_MSK, SNR];
    end
end
```

Полученное сигнальное созвездие и зашумлённое сигнальное созвездие MSK:



Число ошибочных символов: 0, вероятность ошибки на символ: 0.

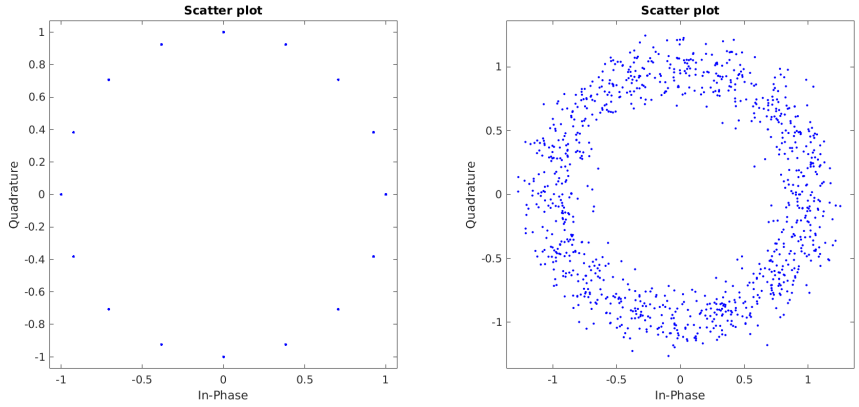
4.6 FSK

Генерируем случайное сообщение, получаем сигнал от двоичного фазового модулятора и сигнальное созвездие FSK:

```
function [SNR_FSK, BER_SNR_FSK] = fsk
    %FSK
    msg                         = randi([0 1], [1 256]);
    M                           = 2;
    freqsep                     = 4;
    nsamp                       = 4;
    Fs                          = 32;
    signal_modulation           = fskmod(msg, M, freqsep,nsamp, Fs);
    signal_noise                = awgn(signal_modulation, 15);
    draw_scatterplot(signal_modulation);
    draw_scatterplot(signal_noise);
    errors                      = randerr(1, 256 * nsamp, 13);
    signal_modulation_errors = signal_modulation + errors;
    signal_demodulation         = fskdemod(signal_modulation_errors, M, freqsep, nsamp, Fs);
    [numberFSK, ratioFSK]       = symerr(msg, signal_demodulation);

    BER_SNR_FSK                 = [];
    SNR_FSK                     = [];
    for m = -15:1:15
        SNR       = m;
        ber_sum   = 0;
        rep       = 100;
        for v = 1:rep
            signal_noise = awgn(signal_modulation, SNR, 'measured');
            demod_FSK    = fskdemod(signal_noise, M, freqsep, nsamp, Fs);
            [err, BER]   = biterr(msg, demod_FSK);
            ber_sum      = ber_sum + BER;
        end
        BER = ber_sum ./ rep;
        BER_SNR_FSK = [BER_SNR_FSK, BER];
        SNR_FSK = [SNR_FSK, SNR];
    end
end
```

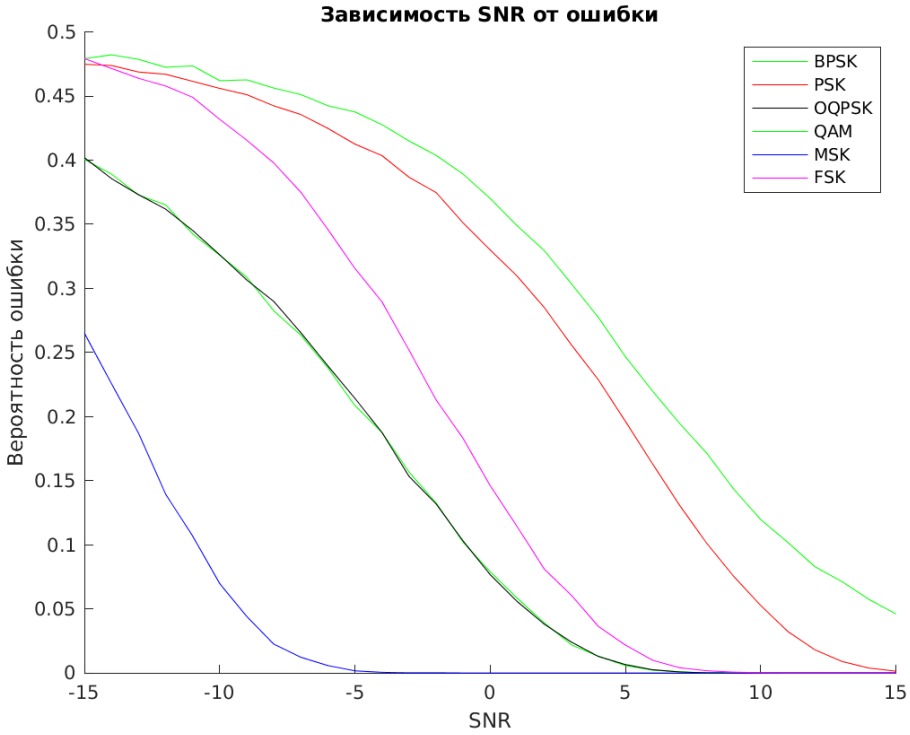
Полученное сигнальное созвездие и зашумлённое сигнальное созвездие FSK:



Число ошибочных символов: 0, вероятность ошибки на символ: 0.

5 Сравнение методов модуляции

Сравним изученные выше методы модуляции. Отобразим на графике зависимость SNR от вероятности ошибки для каждого рассмотренного метода модуляции. Результат сравнения:



6 Вывод

Цифровая модуляция - это процесс преобразования цифровых символов в сигналы с характеристиками канала. При низкочастотной модуляции эти сигналы обычно имеют вид импульсов заданной формы. В случае полосовой модуляции импульсы заданной формы модулируют синусоиду, насываемую несущей; для радиопередачи на нужное расстояние несущая преобразуется в электромагнитное поле.

В даной лабораторной работе были рассмотрены различные виды цифровой модуляции. Тип цифровой модуляции выбирается в зависимости от требований к скорости передачи и помехозащищенности. Самой надежной считается квадратурная манипуляция, так как информацию можно подавать сразу по двум параметрам. Для повцышения скорости передачи могут быть использованы PSK или QAM с большим количеством точек, что в свою очередь негативно скажется на помехоустойчивости вследствие их близкого расположения друг относительно друга на сигнальном созвездии. Число бит, передаваемых одним состоянием, определяется как log2N, где N - уровень модуляции. Таким образом, чем выше уровень модуляции, тем больше данных мы можем передать (или потерять) за единицу времени.