EE-442 | **Project 1**
Single Carrier Transmission

Francesco Cecchetti, Gloria Dal Santo

November 2020

1 Transmitter

This first section discusses the different components of the transmitter, whose structure is outlined by the following block diagram:

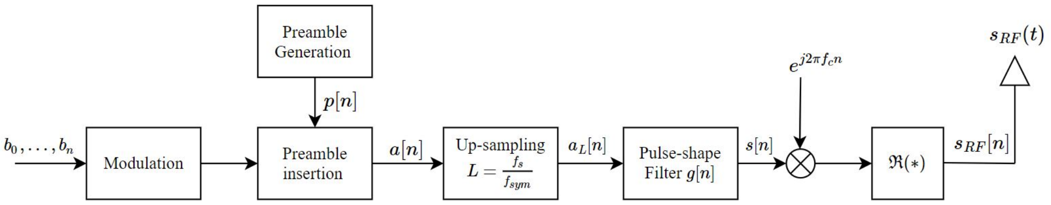


Figure 1: Transmitter - Block diagram

1.1 Modulation

The modulation process consists in translating the bit sequence $b_0, ..., b_n$ randomly generated with uniform distribution, into a sequence of symbols (or "constellation points"). The modulation considered in the following is the Phase-Shift Keying, where the chosen constellation points are positioned with uniform angular spacing around a circle, allowing them to be transmitted with the same energy.
Two types of modulation are considered in this context (Fig. 2 and 3):

- BPSK (Binary phase-shift keying): two phases separated by 180°. It can modulate at 1bit/symbol
- QPSK (Quadrature phase-shift keying): uses four points on the constellation diagram, equally spaced around a circle according to "Gray coding", to minimise the bit error rate (BER). With four phases, it can modulate two bits per symbol.

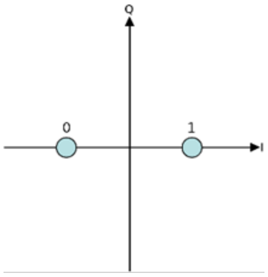


Figure 2: BPSK constellation

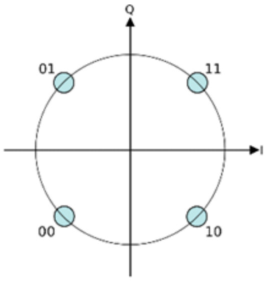


Figure 3: QPSK constellation

Assuming that all the symbols $a_1 \dots a_M$ are equally likely, the average transmitted energy is defined as:

$$\bar{P} = \frac{1}{M}\sum_{i=1}^{M} |a_i|^2$$

where M is the number of constellation points, *i.e.* $M = 2$ for BPSK and $M = 4$ for QPSK. In order to transmit each symbol with the same energy, the constellation points a_i must be normalized as follows:

$$\bar{a}_i = \frac{a_i}{\sqrt{\frac{2(M-1)}{3}}} \Rightarrow \bar{a}_i = \frac{a_i}{\sqrt{2}} \text{ for QPSK}$$

$$\bar{a}_i = a_i \text{ for BPSK}$$

Under these conditions, the symbols magnitude is always $|a_i| = 1$ and the values that they can take on are $a = \{1, -1\}$ in BPSK modulation and $a = \{\pm 1 \pm j\}/\sqrt{2}$ in QPSK.

1.2 Preamble generation and insertion

To allow the detection of the beginning of the data frame within the received signal, a preamble, known to both the transmitter and the receiver, is prepended to the symbol sequence. The way the receiver detects the preamble (the detection algorithm) is based on a correlation filter which will be discussed later, this section presents the reason why a particular type of preamble is required. Indeed, in order to have a very clear peak in the autocorrelation function and thus to allow the receiver to clearly detect the beginning of the data frame, the preamble should be a random-like sequence, a sequence which has similar characteristics to random noise, but still reproducible in the receiver. This sequence is created implementing a Linear Feedback Shift Register (LFSR), whose output is a Pseudo-Noise sequence of length $N_p = 100$, BPSK modulated, which eventually is prepended to the signal. The only information that the receiver must know in order to detect the preamble in the received sequence is the characteristic polynomial of the LFSR, which in this implementation is $p_8(x) = x^0 + x^2 + x^3 + x^4 + x^8$.

1.3 Up-sampling

At this point, the symbols sequence is up-sampled with up-sampling factor $L = f_s/f_{sym}$, where f_s is the sampling frequency and $f_{sym} = 1/T$ is the symbol rate defined by the symbol period T, i.e. the time between two transmitted symbols. The up-sampling process consists practically in adding $L - 1$ zeroes in between two symbols. One of the reasons why this operation is needed is the necessity of transmitting the symbols at a rate different from the sampling one. In this implementation there is indeed no freedom in choosing f_s and thus the only option is to add non informative bits (zero bits) in between the actual data. Moreover, the up-sampling process narrows the signal's bandwidth by a factor of L, which allows to occupy less bandwidth in the transmission channel. Another major reason of using the up-sampled signal lies in the up-conversion process, which will be discussed in section 1.5.

1.4 Pulse shape filtering

The up-sampled discrete-time symbols sequence $a_L[n]$ is then filtered in order to be transmitted through the channel. In particular, a pulse amplitude modulation is implemented: this process consists in defining a prototype waveform ("pulse shape") $g[n]$ and then mapping each of the symbols onto the pulses $g_i[n]$, with $g_i[n] = g[n]a_L[i]$. Therefore, the pulse-shaped signal s[n] is expressed as the superposition of the pulses delayed by multiples of the symbol period T:

$$s[n] = \sum_{k=-\infty}^{\infty} a_L[k]g[n - kT] \tag{1}$$

It is always the case that pulse $g[n]$ lasts longer than T and thus in eq.(1) the pulses overlap. For this reason, the choice of the filter $g[n]$ is made either to have an impulse response that converges rapidly to zero or to fulfil the Nyquist criterion:

$$g[kT] = 0 \;\; \forall k \neq 0$$

This condition assures that whenever the transmitted signal is sampled by the receiver at the time instances $t = nT$, only the n-th pulse contributes to the n-th sample and thus there are no ISI (inter symbol interference) problems. However, the receiver does not sample the signal exactly at multiples of the symbol period T and for this reason at the sample points the pulses are interfering with each other. Thus, the reason why a filter whose impulse response converges rapidly to zero is required.
The chosen filter $g[n]$ is the root-raised-cosine filter (RRC) whose impulse response is:

$$g_{RRC}[n] = \frac{\frac{4\alpha}{\pi}\cos\left(\frac{\pi(1+\alpha)n}{T}\right) + (1-\alpha)\mathrm{sinc}\left(\frac{\pi(1-\alpha)n}{T}\right)}{1-\left(\frac{4\alpha n}{T}\right)^2}$$

As discussed in section 3.2, in the receiver the signal is filtered by another filter matched to $g_{RRC}[n]$: in order to fulfil the Nyquist criterion and to achieve maximum SNR after the filtering operation in the receiver side, the matched filter should be the complex conjugate and time-reverted version of the transmit filter:

$$g_{MF}[n] = g^*[-n] \tag{2}$$

Since $g_{RRC}[n]$ is real-valued and symmetric, according to eq.(2), it is clear that the RRC filter is exactly equal to its matched filter. The total effective filter of the transmission system is then the convolution between the two, usually called RC (raised cosine), and it can be shown that the raised cosine filter fulfils the Nyquist criterion.

The two-sided bandwidth of the raised cosine filter is $B = R(1+\alpha)$ where $R = 1/T$ corresponds to the minimum bandwidth of the signal and α is the roll-off factor. α is a measure of the excess bandwidth of the filter with respect to the ideal low-pass (for which $\alpha = 1$) and at the same time it indicates how rapidly the impulse response of the filter decays to zero. In this implementation, the filter length is 21 symbols and a roll-off factor of 0.22 has been chosen. Figures (5) and (4) show the impulse and frequency responses of the filter $g_{RRC}[n]$ for $f_{sym} = 100$.
The filter output is thus:

$$s[n] = \sum_{k=-\infty}^{\infty} a_L[k] g_{RRC}[n-kT]$$

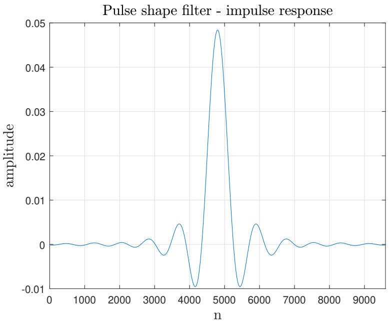


Figure 4: Impulse response of $g_{RRC}[n]$

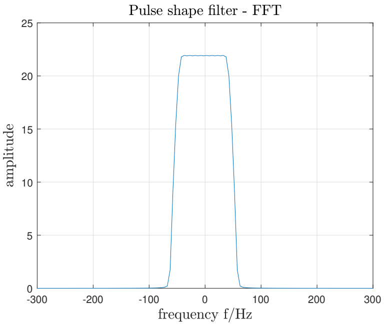


Figure 5: Frequency response of $g_{RRC}[n]$

1.5 Up-conversion

The discrete-time, band-limited, complex-valued sequence $s[n]$, output of the RRC filter, must be transformed into a real-valued sequence and its spectrum must be shifted to a carrier frequency f_c. These two steps are simply implemented as shown in the following formula:

$$s_{RF}[n] = \Re\{s[n]e^{j2\pi f_c n}\}$$

By taking the real part of the shifted signal, the spectrum of the signal becomes symmetrical with respect to $f = 0$.

At this point, one other reason that leads to implement the up-sampling process comes out: since the spectrum of the symbol sequence is band-unlimited (periodic) and the baseband component must be isolated prior to be shifted to the carrier frequency f_c, by up-sampling and interpolating the signal with filter $g_{RRC}[n]$, the spectral images become sufficiently distant, therefore it's easier to filter them avoiding problems in the transmitter's DAC's filter, which shows often relevant non-ideal behaviours and its parameters are not controllable.

2 AWGN Channel

The up-converted signal is then processed by the soundcard and sent to the USB speakers, that simulate the transmitter antenna. The transmission channel is the air, modeled as a AWGN (Additive White Gaussian Noise) channel which justifies the assumptions made throughout this implementation. Under this condition, the transmitted symbols are disturbed by a complex random variable $w = w_r + jw_i$ whose real and immaginary part are independent normal distributed variables, $w_r, w_i \sim \mathcal{N}(0, \sigma_w^2/2)$. The receiver is a USB microphone connected to the same soundcard and positioned around 30 cm far from the speakers.

3 Receiver

This section will analyse the receiver, whose structure is described by the following block diagram:

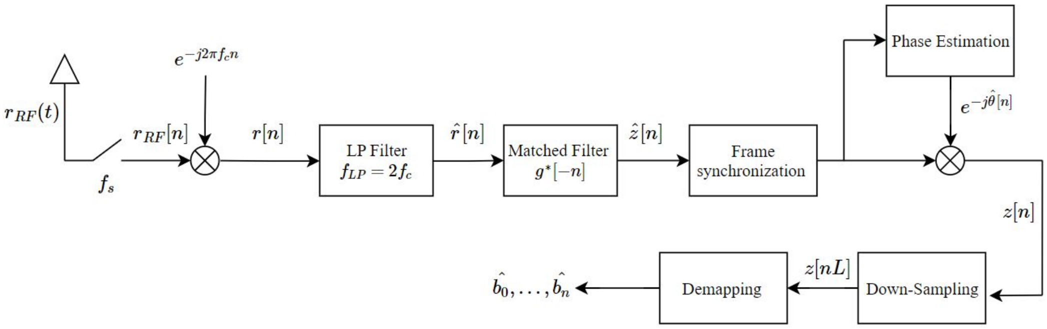


Figure 6: Receiver - Block diagram

3.1 Sampling and Down-conversion

The received continuous time signal $r_{RF}(t)$ is sampled at f_s and down-converted to the baseband by simply multiplying the signal with the complex phasor $e^{-j2\pi f_c n}$:

$$r[n] = r_{RF}[n]e^{-j2\pi f_c n}$$

where $r_{RF}[n]$ is the sampled $r_{RF}(t)$.
In a noiseless scenario, the received down-converted signal can be expressed as follows

$$r[n] = \frac{1}{2}s[n] + \frac{1}{2}\left(\Re\{s[n]\} - \Im\{s[n]\}\right) \cdot e^{-4\pi f_c t} \tag{3}$$

where it can be seen the presence of unwanted components at twice the carrier frequency. To get rid of these byproducts a low pass filter with cut off frequency $f_{LP} = 2 \cdot f_c$ has been applied. Figures (7) and (8) show the FFT of the received signal before and after both down-conversion and filtering.
The signal in eq.(3) represents an ideal scenario. In reality there is no such a noiseless signal but instead there are at least three sources of noise that distort the signal being transmitted: the unavoidable background noise recorded by the microphone, a time offset and a phase offset. The background noise

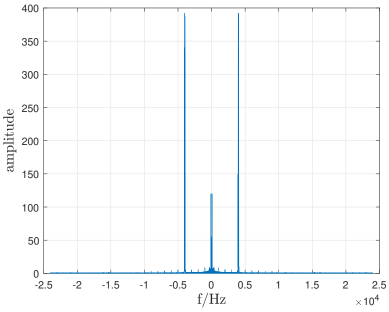


Figure 7: FFT of the received signal $r_{RF}[n]$

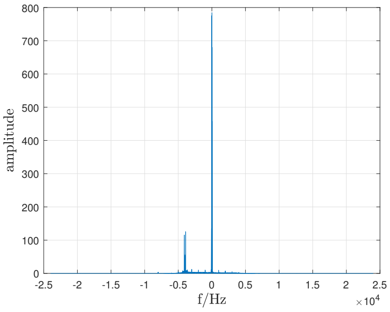


Figure 8: FFT of the down-sampled and filtered signal $\hat{r}[n]$

can be easily seen in Fig.(9) were the transmitted and received signals are compared (the amplitudes of the signals have been scaled in order to highlight the different shape of the waveform, in reality the amplitude of the transmitted signal is always greater that that of the received one for obvious reasons). In this implementation the amplitude of the noise is smaller with respect to the received signal and its effect will be further reduced by the matched filter (section 3.2).

The time offset is due to the time that the signal takes to travel along the channel and to reach the receiver. The continuous time signal that reaches the receiver can be expressed as follows:

$$r_{RF}(t) = s(t - \varepsilon T)e^{j2\pi f_c(t-\varepsilon T)}$$

where εT is the time delay (expressed by a fraction of the symbol period T). The complex phasor relative to the up-conversion process is shown explicitly to denote how the phase error is originated: after the down-conversion process the received signal is $s(t-\varepsilon T)e^{j2\pi f_c(t-\varepsilon T)}e^{-j2\pi f_c t} = s(n-\varepsilon T)e^{-2\pi f_c \varepsilon T} = s(t-\varepsilon T)e^{-j\theta}$ from which it can be seen how the time delay ε causes the phase-shift $e^{-j\theta}$. The time and phase offsets and their correction are further discussed respectively in sections 3.4 and 3.5.

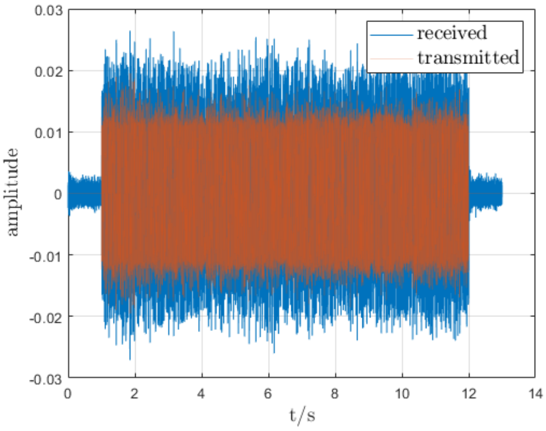


Figure 9: Down-converted received signal

3.2 Matched filter

As already discussed in section 1.4, assuming a AWGN channel the filter that maximizes the SNR is the matched filter, and it does so out of a trade-off between noise cutting and spectrum distortion. The impulse response of the filter follows eq.(2). Besides maximizing the SNR and reducing the ISI, the matched filter cuts out the unwanted frequency components at frequencies $|f| > \frac{(1+\alpha)}{T}$Hz. Since the cut off frequency of the pulse shape filter is lower than $2f_s$, the low pass filter mentioned in the previous section could be avoided. This is not the case in real receivers, where a filtering operation must be implemented before sampling the signal. To keep the analogy with real systems, the low pass filter has been included in the implementation.

3.3 Frame synchronization

At this point the receiver must detect the beginning of the data frame, by means of the preamble prepended to the symbols sequence. The detection problem is solved using a correlator, *i.e.* a filter matched to the complex-conjugate, time-reversed preamble. The pseudo-random structure of the preamble showed in section 1.2 allows the receiver to see a clear peak in the correlator's output and thus to detect the beginning of the data.
The correlator's output is:

$$c[n] = \sum_{i=0}^{N_p-1} = \tilde{p}^*[i]\hat{z}[n-i]$$

where $\tilde{p}^*[i] = p[N_p - 1 - i]$ is the reverted preamble with length N_p.
By virtue of the generalised likelihood test, the beginning of the frame is the index for which $\mathcal{T}$, the correlator's magnitude, normalised to the energy of the signal's portion inside the correlator, is bigger than a threshold γ:

$$\mathcal{T} = \frac{|c[n]|^2}{\sum_{i=o}^{N_p-1} |\hat{z}[n-i]|^2} > \gamma$$

It's worth noticing that since the signal is oversampled (with oversampling factor L), the index of the correlation's peak is not the one corresponding to the first time that $\mathcal{T} > \gamma$ is fulfilled, but instead, once a peak is detected all the L samples following the peak go inside the correlator and the returned index is the one corresponding to the highest $\mathcal{T}$. As for the choice of the threshold γ, a excessively low value brings to "false alarms", *i.e.* the receiver starts receiving the sequence which is not the actual sent data. Conversely, if the threshold is too high, a "detection miss" (the receiver misses the beginning of the data frame) would occur. After analysing the trade-offs between the two scenarios, $\gamma = 15$ has been set.
Once that the beginning of the data frame is detected, the phase of the correlator output at its peak is stored in order to be used as an initial phase estimator: $\theta_0 = \arg\{c[n_{max}]\}$, where n_{max} indicates the index of the correlator's peak. (cfr. section 3.5).

3.4 Time synchronization

Once the first symbol of the transmitted data has been detected, time recovery and interpolation steps are usually required. In this implementation these operations can be avoided thanks to the very high oversampling factor L. The absolute value of the maximum error that this approximation leads to is equal $|\mu| \leq \frac{T_s}{2 \cdot L}$ where T_s is the sampling period, 1/48 ms in this implementation. The maximum error that can be obtained throughout the simulations is $|\mu_{max}| \approx 0.9 \cdot 10^{-6}s$ given by $L = 24$ and it was verified that under these conditions the BER (bit error rate) is still acceptable.

3.5 Phase estimation

The received signal at this point can be represented as $z[n] = a[n]e^{j\theta[n]}$ where $\theta[n]$ is the phase offset that must be compensated. The phase offset can be modelled as slow varying signal $\theta[n] = \theta[n-1] + \Delta\theta[n]$ where $\Delta\theta[n] \sim \mathcal{N}(0, \sigma^2_{\Delta\theta})$. The initial offset θ_0 has been already estimated during the frame synchronization, while to estimate $\theta[n]$ it can be used the advantage that the non normalized symbols of the QPSK modulation, raised to the power of the modulation order M, lose all the information that they carry. Recalling the definition of the symbols

$$\begin{aligned} a_M[n] &\in \{e^{j2\pi \frac{1}{M} d_l + j\frac{\pi}{M}} | d_l \in \{0, 1, \ldots, M\}, M > 2\} \\ a_2[n] &\in \{1, -1 | M = 2\} \end{aligned}$$

then it can be obtained the signals $z^4[n] = (-1) \cdot e^{j4\theta[n]}$ for QPSK moduation and $z^2[n] = e^{j2\theta[n]}$ for the BPSK, and the estimated phase offset can be derived by computing the following argument:

$$\begin{aligned} \hat{\theta}_4[n] &= \frac{1}{4} \arg\{-z[n]^4\} \\ \hat{\theta}_2[n] &= \frac{1}{2} \arg\{z[n]^2\} \end{aligned} \tag{4}$$

Here the power normalization has been neglected since it scales the signal by a positive value without affecting the phase. To overcome the ambiguity of $\pm\frac{\pi}{M}$ that arises when computing the argument in eq.(4) it is necessary to keep track of the previously estimated phase as discussed earlier. With the initial information given by $\hat{\theta}[n-1]$, at each iteration $\hat{\theta}[n]$ corresponds to $\frac{1}{4}\arg\{-z[n]^4\} \cdot \exp\{j\frac{\pi}{2}k\}$ where k is the value that minimizes the distance between $\hat{\theta}[n-1]$ and $\hat{\theta}[n]$ (for QSPK $k \in \{-1, 0, 1, 2, 3, 4\}$) that is: for every $n-1$, k identifies the most probable phase at n assuming that the original phase does not change too quickly. This last assumption leads to a further correction: the new estimated phase is set as the result of the following interpolation, which can be seen as a low pass filtering operation,

$$\hat{\theta}'[n+1] = 0.99 \cdot \hat{\theta}[n] + 0.01 \cdot \hat{\theta}[n+1].$$

The algorithm that has been just stated is called the "Viterbi and Viterbi" algorithm.
It can be further noticed that the distinction made in eq.(4) is redundant. Since the algorithm, for the case in which QPSK is applied, checks in every quadrant the phase that is closer to the last estimated one among the four possibilities, it inherently includes the phases that would have be evaluated if the BPSK was considered.
Fig.(10) and Fig.(11) illustrates the effect of the phase offset and the result obtained by the algorithm for a bitstream with symbol rate $f_{sym} = 100$ Bd.

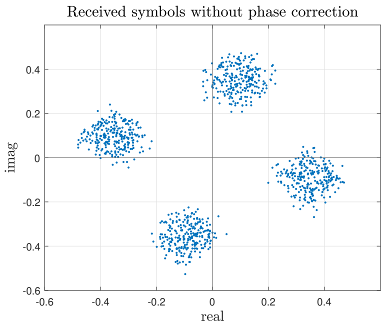


Figure 10: Received symbols

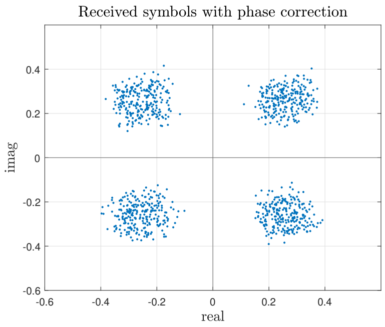


Figure 11: Received symbols with phase correction

3.6 Down-sampling and Demapping

At this stage all the major sources of error are corrected. The last steps consist in down-sampling the signal by keeping only one out of L samples and demapping it. The demapping operation is based on a hard decision between the possible symbols and it must be consistent with the used modulation scheme. To do so, the regions separated by the x-y axis in Fig.(3) (or simply di $y-$axis in BPSK) are considered as decision regions, and the classification of the symbols is made by analyzing the signs of their imaginary and real parts.

4 Results

4.1 Bit Error Rate

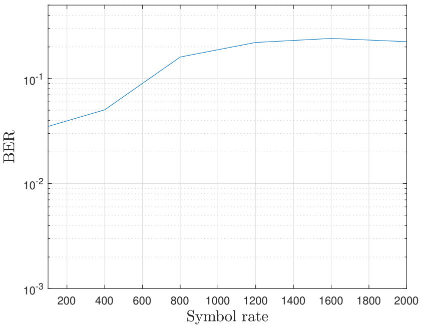
The system has been tested to different symbol rates, in the same room conditions and with the same position of the hardware. Fig.(12) shows the BER, defined as the number of bit errors divided by the total number of transferred bits, averaged on the transmission of six frames of 2000 bits each, in QPSK modulation. It can be seen that the error is always below 0.25 and it decreases for lower symbols rates. Table 1 summarises the value of the BER obtained for both QPSK and BPSK modulation and it can be noticed that for the latter the number of bit errors is much lower. Indeed, the BPSK is the most robust among all the PKS modulations, since the constellation points are separated by 180^o, the highest admissible angle. However, as discussed in section 1.1, it is able to modulate only 1 bit/symbol which makes it unsuitable for high rate transmissions.

f_{sym} $[sym/s]$	100	400	800	1200	1600	2000
QPSK	0.0350	0.0504	0.1603	0.2206	0.2409	0.2244
BPSK	0.0013	0.0013	0.0067	0.0218	0.0977	0.0932

Table 1: Bit error rates at different symbol rates

For what concerns the matched-filter, its length has been set as 21 times the symbol length and indeed, the greater the symbol rate the shorter the filter.

The sampling frequency of the implemented system was $f_s = 48$ kHz, limited by the soundcard, and the carrier frequency was set as $f_c = 4$ kHz. To further investigate the limitations in the frequency response of the hardware, f_c has been increased until 20 kHz, frequency above which the speakers were not able to reproduce any sound. It was observed that for higher carrier frequencies the BER is considerably lower (maximum values around 0.1).



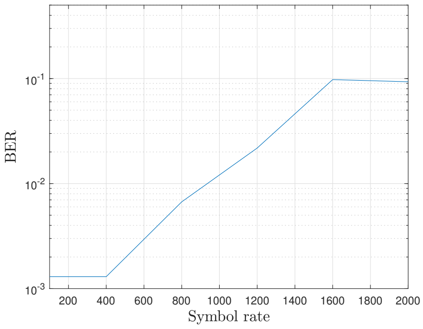


Figure 12: BER at different symbols rates, QPSK. Figure 13: BER at different symbols rates, BPSK.

4.2 Carrier frequency offset

When a carrier frequency offset $\Delta f_c \geq 60$ mHz is introduced, the BER jumps to its maximum value, i.e. BER ≈ 0.5. In this scenario the received signal is given by

$$r(t) = \sum_{k=-\infty}^{\infty} a_L[k] g_{RRC}(t - kT - \varepsilon T) \exp(j2\pi \Delta f_c t + j\theta) + w(t)$$

where $w(t)$ is the channel noise and ε is the time offset, previously neglected.
From this equation it can be seen that the carrier phase offset, due to carrier frequency mismatch, can not be treated as the phase noise θ that was discussed in section 3.5 since it is not a Gaussian random variable but it is rather a linear function of the time. The baseband received signal is shifted from the center axis by Δf_c, which gives rise to a distorted matched filter output pulse and to intersymbol interference. To avoid a degradation of the received signal a frequency estimation and correction must be made before the signal enters the matched filter.