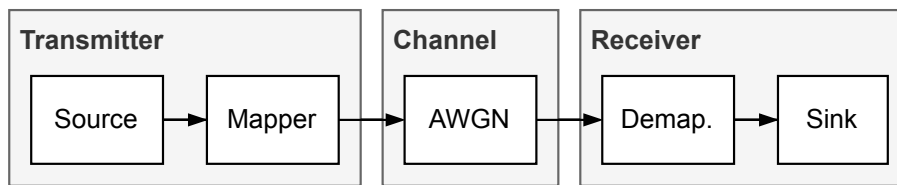# 1 Mapping and Demapping

1.1 The AWGN Channel

Figure 1.1: System model for Lab 1

We now replace the BSC by another very common but a bit more sophisticated channel model. In the additive white gaussian noise (AWGN) channel the symbols are complex numbers which represent the amplitude and phase of the transmitted signal and carry the digital data. After transmission over a channel which disturbs the symbols with additive noise, the demapper converts the received noisy symbols back to a bit stream.

In the following, the transmitter and channel blocks are explained in more detail.

- Source: Throughout the lab, we use grayscale images as payload data. The pixels are transmitted rowwise, starting with the upper left pixel. The brightness of each pixel is represented by an unsigned 8-bit value, and the least significant bit (LSB) is transmitted first.

- Mapper: The modulation scheme used by our system is quadrature phase shift keying (QPSK), which means that the bits are pairwise mapped onto symbols $a = (\pm 1 \pm j)/\sqrt{2}$, as shown in Figure 1.2. With this modulation scheme, the information lies solely in the phase, since all symbols have the same amplitude $|a| = 1$, hence the name *Phase Shift Keying*.

- AWGN channel: In the channel, the transmitted symbols $a[n]$ are disturbed by complex-valued *additive white Gaussian noise* (AWGN) $w[n]$ with zero mean and variance σ_w^2. (A complex-valued Gaussian distributed random variable w is obtained by taking two independent real-valued Gaussian random variables with half the variance, $w_1, w_2 \sim \mathcal{N}(0, \sigma_w^2/2)$[1], and combining them to a complex number $w = w_1 + jw_2$.) "White" in this

---

[1] The notation $x \sim N(\mu, \sigma^2)$ means that the random variable x is Gaussian distributed with mean μ and variance σ^2, i.e. the probability density function (PDF) is $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

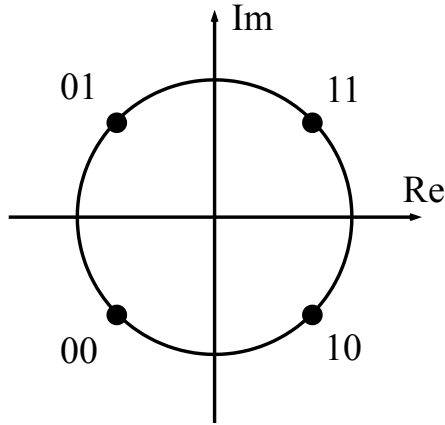context means that the noise samples are uncorrelated, i.e. $E\{w[m]w^*[n]\} = \sigma_w^2\delta[m-n]$.

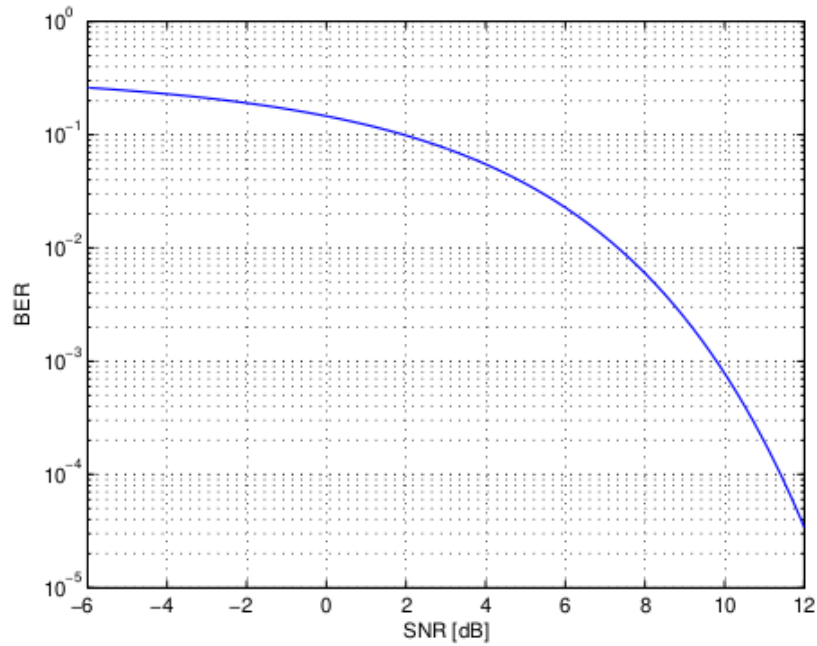Figure 1.2: Mapping of bit pairs b_0b_1 to QPSK symbols

Figure 1.3: BER for uncoded QPSK modulation, AWGN channel

The received symbols are then given as

$$r[n] = a[n] + w[n].\tag{1.1}$$

An important measure in communication systems is the ratio of average signal power to average noise power (*signal-to-noise ratio*, SNR), usually expressed in decibel:

$$\text{SNR[dB]} = 10\log_{10}\left(\frac{E\{|a[n]|^2\}}{E\{|w[n]|^2\}}\right).\tag{1.2}$$

$SNR = \dfrac{1}{\sigma_w^2}$

noise variance of each component:

$\sigma_w^2 = \dfrac{1}{SNR}$

$\Rightarrow \sigma_w^2 = \dfrac{1}{2 SNR}$

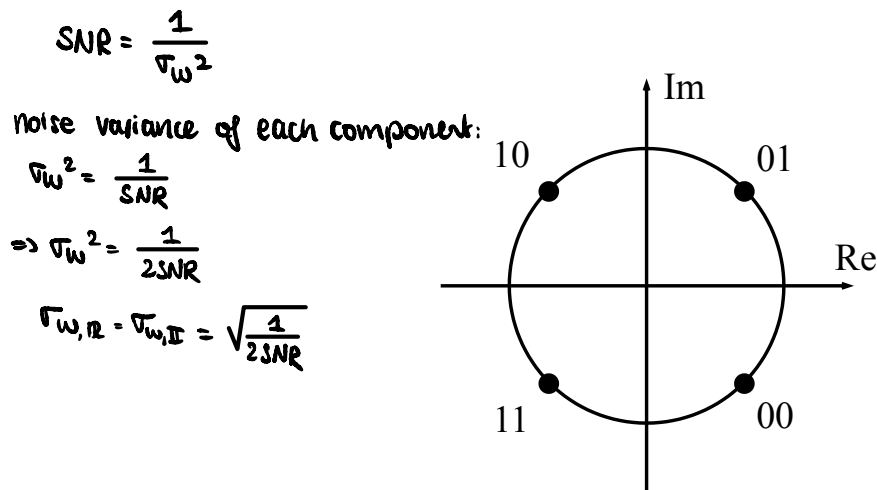$\sigma_{w,\mathbb{R}} \cdot \sigma_{w,\mathbb{I}} = \sqrt{\dfrac{1}{2 SNR}}$

Figure 1.4: Alternative Mapping of bit pairs $b_0 b_1$ to QPSK symbols

Since the power of the data symbols is normalized to $1 (|a[n]|^2 \equiv 1)$, the SNR of our system is $1/\sigma_w^2$. The best achievable *bit error rate* (BER) for an uncoded QPSK modulation is plotted in Figure 1.3 as a function of the SNR.

1.2 Your Tasks

In order to perform the labs on your own computer you need to install the Communications Toolbox. To install the toolbox on Matlab simply go to Home>Add-Ons>Get Add-Ons, search for Communications Toolbox and install it.

A1T1 You are given a transmitted signal, as well as the width and height of the transmitted image. The signal can be loaded into the MATLAB workspace with the command *load<filename>*.

Your task is to generate a noisy received signal by adding, dependent on a SNR value in dB, correctly scaled noise to the loaded signal.[2] Use the MATLAB function *randn* to generate the noise.[3] The received symbols are then demapped into bits and the received image is displayed. *image_decoder*.

Hint: Note that the signal length is not equal to the number of pixels in the image.

opt. Plot the received symbols as clouds of points in the complex plane. Compare plots of the noisy constellations of the received signal for different channel conditions.

A1T2 • For the signal space constellation diagram shown in Fig. 1.5, find the average energy of the red and blue constellations (independently).

---

[2] In later chapters, the signals that you get will already contain the impairments introduced by the channel, e.g., a timing or phase error. However, you will always have to add the noise yourself, so that you can simulate your system at arbitrary SNR values.

[3] The command *help <function>* can be used in the MATLAB terminal to get the documentation of a function.
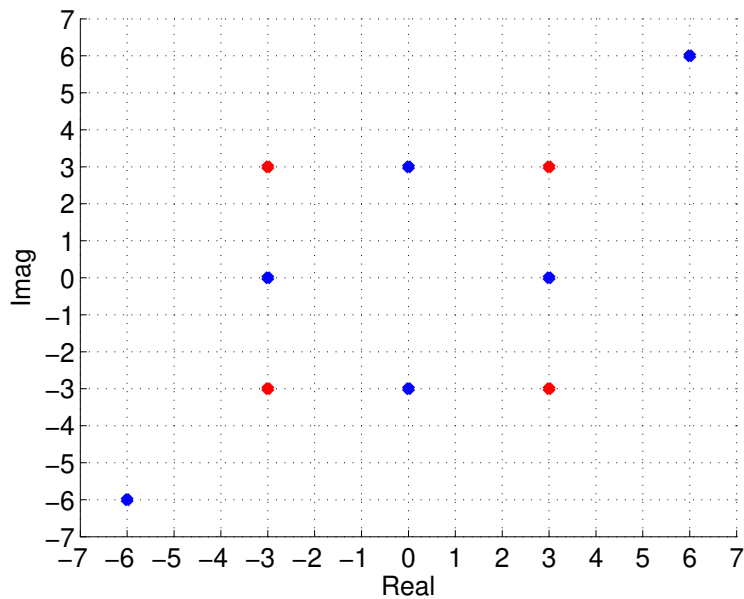
Figure 1.5: Constellation diagram

- Normalize the average transmitted symbol energy to 1 for the two constellations inde- pendently. Plot the energy-normalized constellation points of the two schemes (red and blue) on the same figure. What observations can you make?

A1T3 Now write a function that experimentally reproduces the BER plot from Figure 1.3. Create a random bit sequence, map it onto symbols according to Figure 1.2, and add scaled noise. Demap the noisy symbols to bits and determine the bit error rate. Repeat these steps for different SNR values and then create a BER plot with the function *semilogy* (in order to get a logarithmic scale on the y-axis). Finally, compute the BER plot where you use the symbol mapping as proposed in Figure 1.4.

A1T4 An alternative image decoder based on lossy image compression is implemented by the function *compressed_decoder*. Unfortunately the script was programmed by a lazy PhD student and is therefore full of errors. Your mission is to hunt down all bugs. Hints:

- Keep track of the arrays size under the "Workspace" subwindow in matlab.
- Under the "Run" button in matlab, you can choose to pause the code execution on errors, making the debugging easier.
- Under the "Run" button in matlab, you can set the code to execute when pressing the button, avoiding going back and forth between the main script and the function being debugged.
- A breakpoint placed at the end of a function let you investigate the state of the variables just before they are freed by the function return.
- When founding a mistake, make sure that it was the only location in the code where this error occurs.

1.2.1 Plotting in MATLAB

Guidelines for plotting graphs:

- Plot the BER in a logarithmic scale (*semilogy*).

- Make sure that the ranges of the axis are meaningful.

- If you have multiple curves in one plot, assign different colors and or markers to it.

- Always label the axis (*xlabel* / *ylabel*).

- Assure that each plot has a meaningful legend. (*legend*)

- Use the MATLAB function for export of pictures (*saveas*) **NOT** a screenshot.