Student: Andrea Cavallo

Matricola: 245715

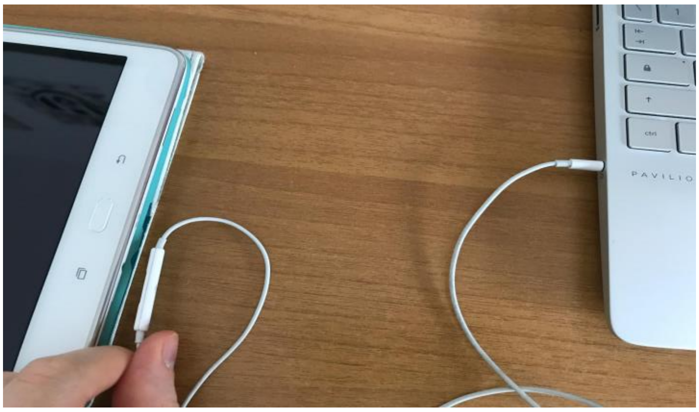
Course: Digital Communication Laboratory

PROJECT #03

1. Image transmission through real channel

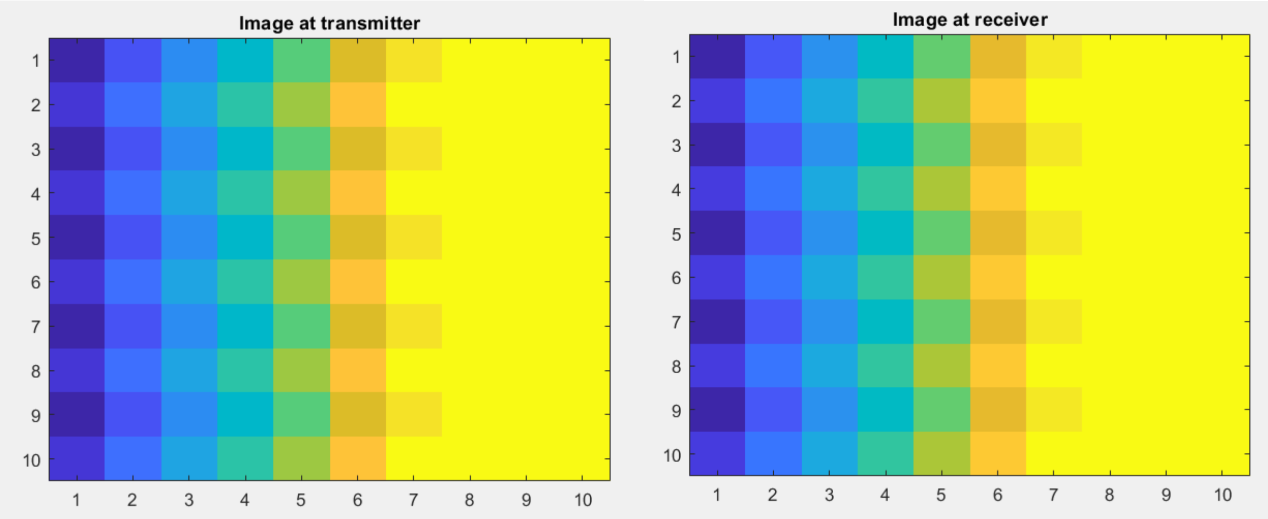
The first part of the project consists in the realization of a real digital transmission system. The final goal is to transmit an image from a transmitter device to a receiver device. The image is converted to an audio signal which is played, recorded and transmitted through a cable to the receiver. Then, the received audio signal is decoded and the image is reconstructed. Several precautions need to be taken in order for the transmission to work properly.

The setup for the transmission is shown in the following picture.



The audio signal is played by the device on the left, it is recorded with a microphone and transmitted via cable in the pc on the right. This setup generates a transmission system which is not as performing as a simple cable between two devices, but it allows discrete results.

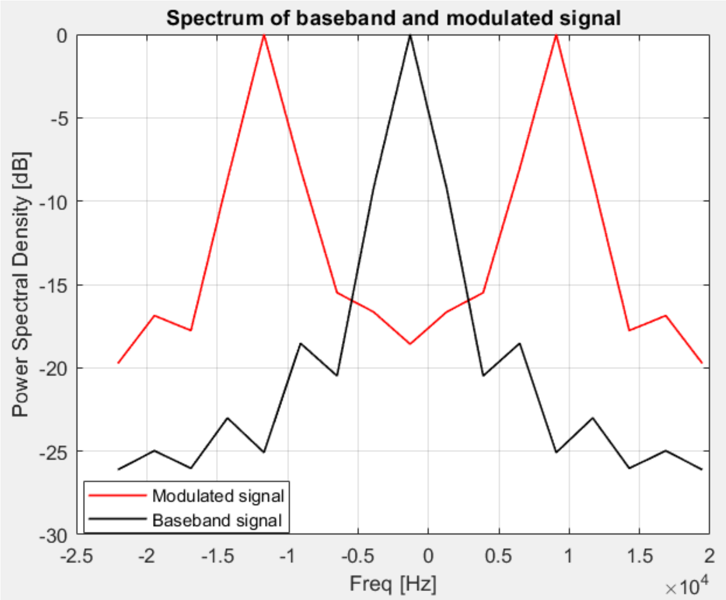
To begin with, in the example reported below, an image is created using Matlab, so that the size can be small (10x10) and the corresponding audio file can last just a few seconds. All the following steps can be analogously applied to a bigger image, with the difference that the audio file can be much longer (even some minutes) and the transmission may suffer some difficulties due to the different clocks in transmitter and receiver. To avoid this problem, a long audio file can be divided into several frames, at the beginning of which synchronization between the two clocks is renewed. Although this method is conceptually the same as the method used in the experience, having a small image allows to transmit just one frame through the cable. The following pictures show the transmitted and the received image, which are identical.



At the transmitter, the image is converted into a string of bits starting from the image matrix (it is possible to obtain a matrix for an image using the Matlab function imread). Then, a Barker sequence is added at the beginning and at the end of the string of bits to be transmitted (the payload), in order to allow synchronization between transmitter and receiver. In the experience reported, a 13 bits Barker sequence is used and each symbol is extended three times. To complete the transmission frame, a dummy sequence of symbols (just zeros) is inserted at the beginning and at the end, to avoid the audio fade-in that happens with some audio cards. After that, the frame is modulated using a 2PAM modulation format. The Barker sequences are modulated using the antipodal configuration in order to exploit their correlation properties as expected, whereas the payload is modulated in the unipolar configuration to avoid the need of a phase recovery algorithm at the receiver. As a matter of fact, the signal can suffer a phase shift during transmission, but the usage of the unipolar configuration allows to simply discern symbols with a threshold detector. The selected simulation frequency is the sampling frequency of the audio card, 44.1 kHz, the chosen number of samples per symbol is 8 and NRZ pulse shape is adopted. In this way, a baseband signal is generated. To avoid distortion in the channel, though, it is necessary to transmit a bandpass signal. To achieve this, the signal is multiplied by a cosine function with a given frequency f_0, which is selected to be 10 kHz. This choice respects the constraints of the system, because the bandwidth of the signal when using NRZ pulse shape is about 1.5 * R_s, so the modulated signal has a spectrum approximately over the frequency range [-1.5 * R_s + f_0 , 1.5 * R_s + f_0]. As requested for the system to work properly, the right end of this interval is smaller than the simulation frequency. In fact,

$$1.5 * R_s + f_0 = 1.5 * 44.1 \text{ kHz} / 8 + 10 \text{ kHz} = 18.269 \text{ kHz} < Fsim / 2 = 22.05 \text{ kHz}$$

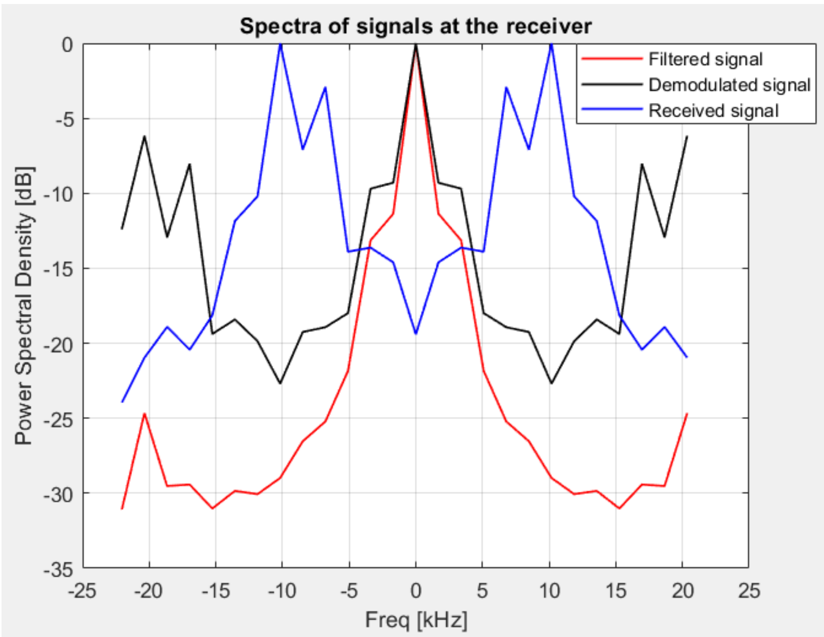
The following picture shows the spectra of the baseband signal and the modulated signal. As expected, the spectrum of the modulated signal is shifted by 10 kHz to the right. To make the spectra smoother, Bartlett periodogram is applied.



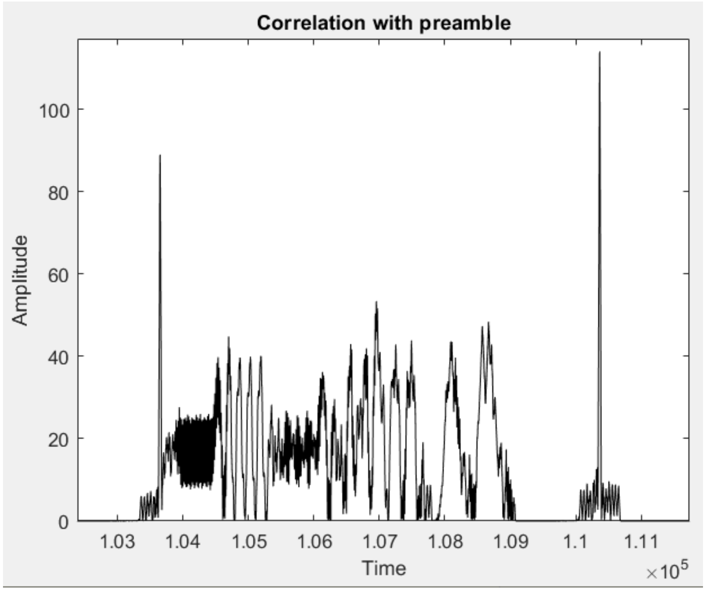
The modulated signal is normalized and an audio file is generated using the Matlab function audiowrite and the simulation frequency (44.1 kHz).

Afterwards, the audio signal is played, recorded and transmitted through a cable from a transmitter device to a receiver PC, where it is sampled using the same sampling frequency used at the transmitter. Even though this is not necessary, it makes it easier to recover data from the signal.

At the receiver, the sampled signal is needs to be demodulated (its frequency spectrum needs to be centered around zero). For this scope, the signal is multiplied by a cosine with the same frequency f_0 of the cosine at the transmitter. The signal obtained has a spectrum centered around zero, but it has a high frequency component that needs to be filtered out. So, a matched filter is applied. The picture below shows the frequency spectra of the received signal (whose spectrum is centered around 10 kHz), of the signal after the multiplication by a cosine (spectrum centered around zero with a high frequency component) and, eventually, of the signal after the application of the matched filter.

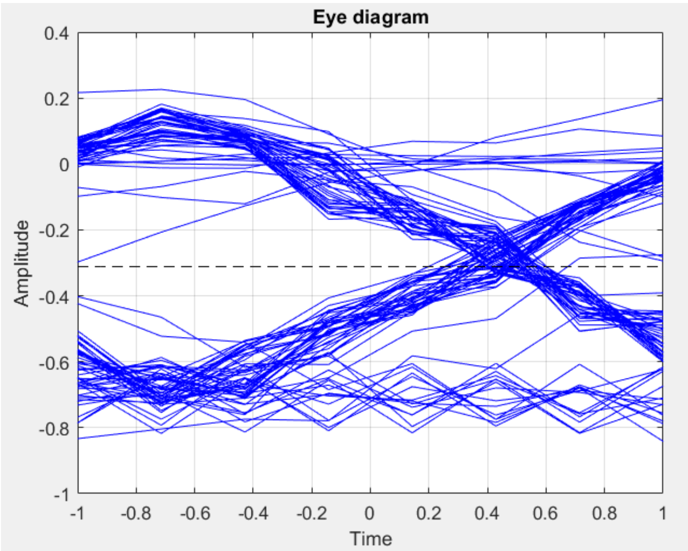


After that, the correlation between the normalized demodulated signal and the extended Barker sequence used at the transmitter is calculated. The result is shown in the following plot.



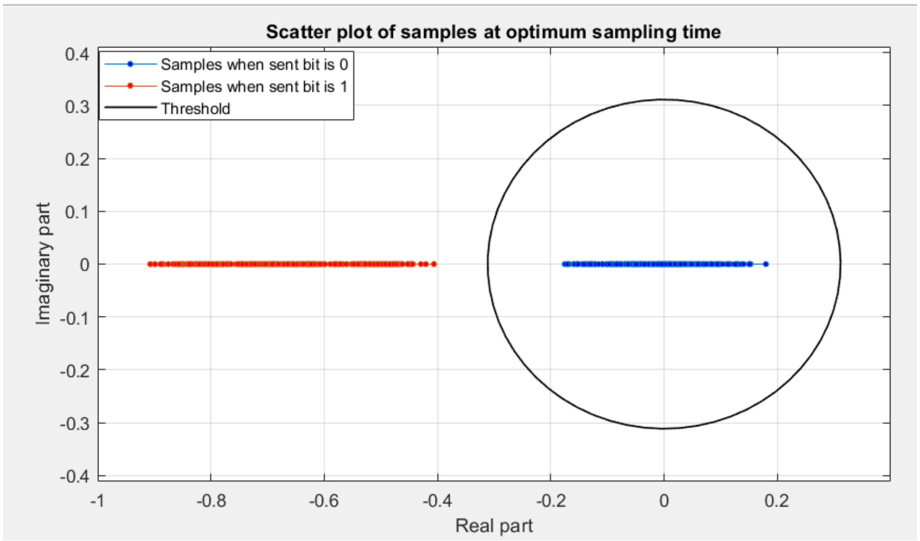
Two peaks can be clearly identified, which indicate the end of the two Barker sequences in the frame and, therefore, allow to recover the payload. It can be observed that in the ideal case these peaks would have the same height, whereas in a real case there is a difference in their height, even though they are still clearly recognizable. Using the position of these peaks, the payload is extracted from the frame and it is normalized.

Then, the eye diagram is plotted, as reported in the picture below.



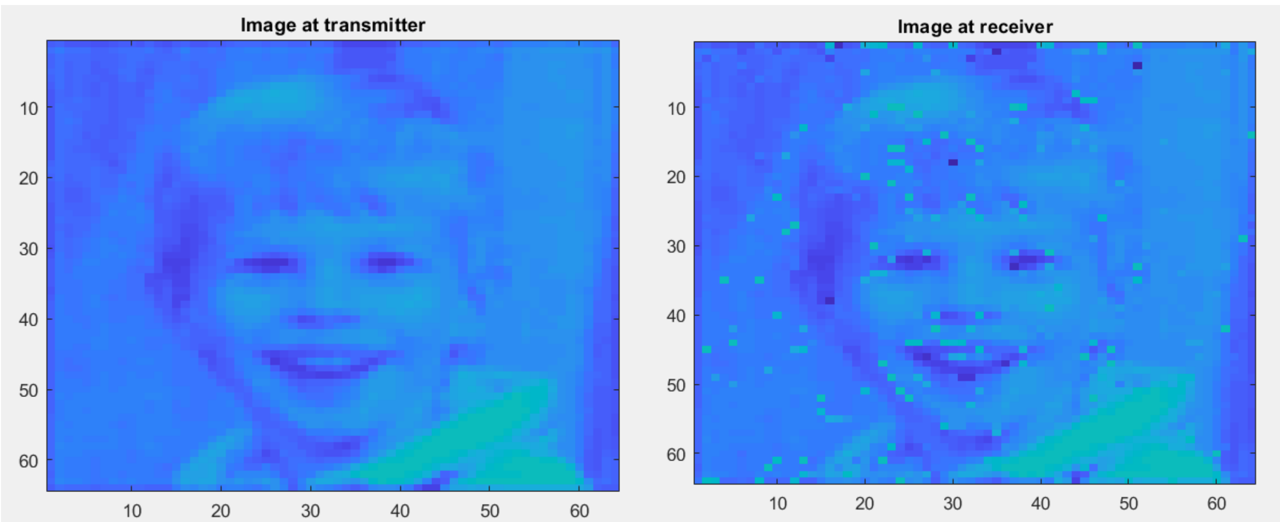
The black dashed line in the picture is the average value of all the lines in the eye diagram. From the plot, it is possible to notice that this value is a good choice as threshold to discern ones and zeros at the optimum sampling instant. This instant is calculated from the eye diagram, by finding the instant corresponding to the maximum eye opening.

Subsequently, the payload is sampled at the optimum sampling instant and the decision is made using the absolute value of the average of the eye diagram as a threshold. The scatter plot below shows the location of the samples extracted from the payload at the optimum sampling time. It is possible to distinguish two different clusters, one with the samples corresponding to a transmitted 0, the other with the samples corresponding to a transmitted 1. As shown in the picture, these two clusters are effectively identified by the chosen threshold, whose value is the radius of the black circle in the picture.

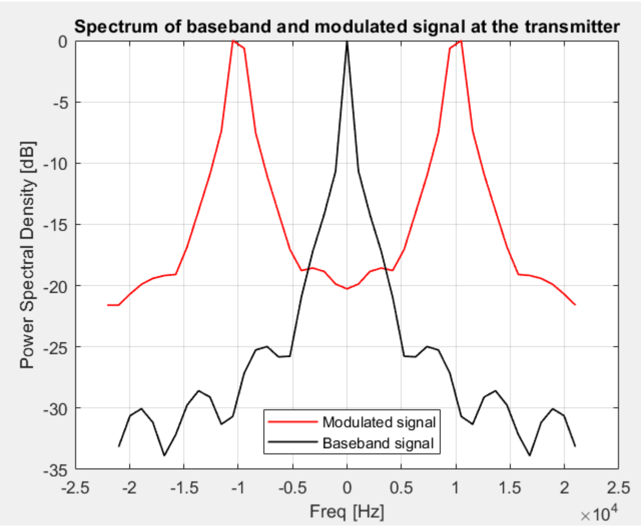


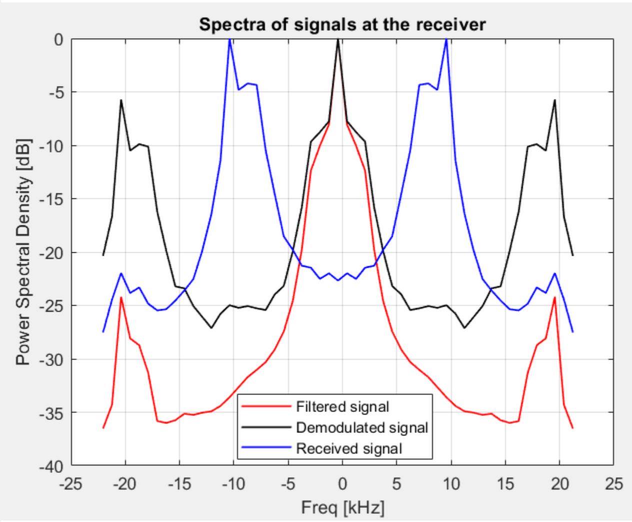
After the decision is made, the BER is evaluated using the transmitted signal and, in this case, no error is detected (BER = 0). The image is then reconstructed by ordering and reshaping the bits.

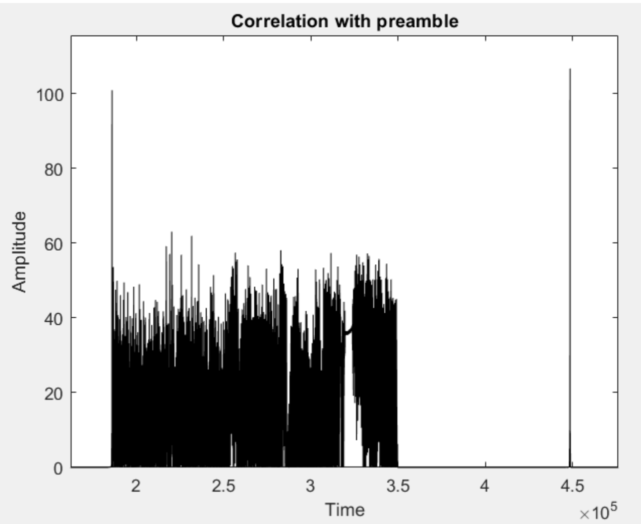
To test the performances of the system created, another image is transmitted using the same approach as before, but this time the image is bigger (64x64) and the transmission shows some weaknesses. As can be seen in the pictures below, the received image is pretty similar to the transmitted one, but some errors occur during the transmission.

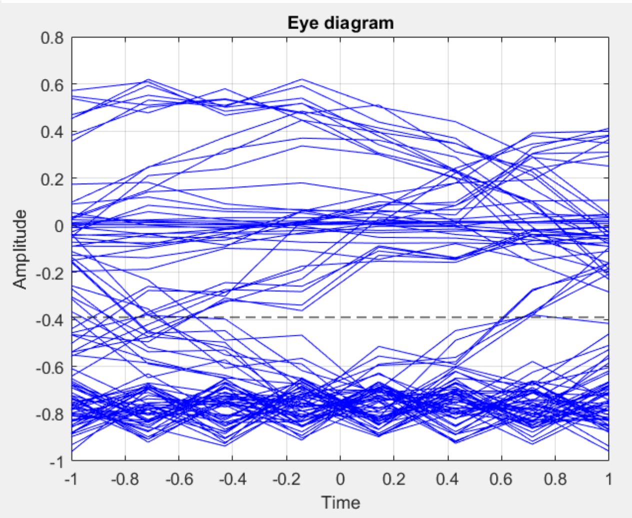


The spectra of the signal at the transmitter and at the receiver and the correlation with the preamble are good, but the eye diagram is very noisy, as shown in the pictures below.







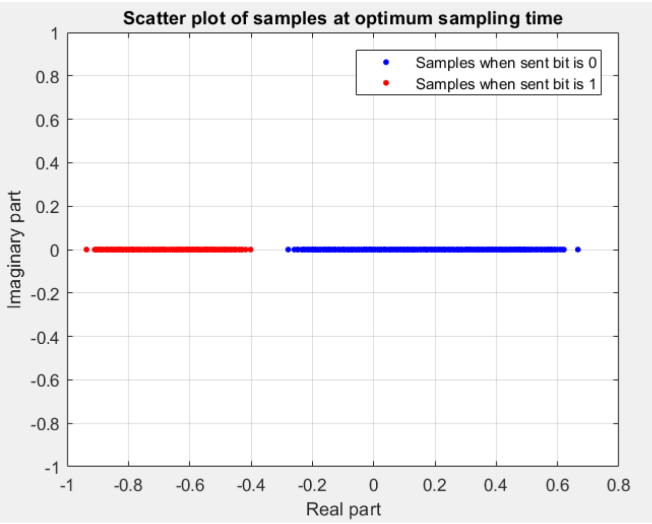


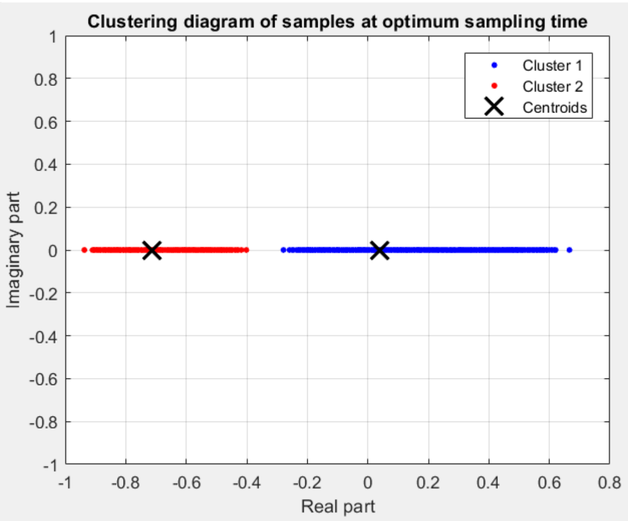
Thus, the decision at optimum sampling instant is not perfect and the final BER is 1.18%. It can therefore be concluded that the transmission system and the recovering algorithm described in this section work well with small images, that are transmitted without errors, but are prone to transmission errors when the bits to be sent increase in number.

To improve the performances of the system, a different recovering algorithm can be tested. The idea is to look at the scattering diagram and identify two different clusters of samples, one of which corresponds to transmitted zeros, the other to transmitted ones. This method can be applied simply by plotting the scattering diagrams for different sampling instants and choosing the best one, dividing the samples into clusters visually. Another possibility is to apply a K-Means clustering algorithm to identify two clusters and then assign each cluster to the corresponding transmitted value.

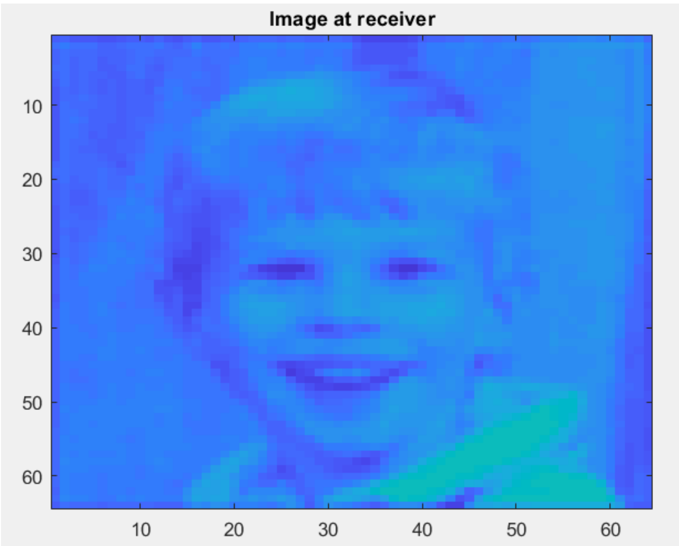
The K-Means clustering algorithm divides a set of point into a given number of clusters (2 for 2PAM modulation), each of which is assigned a centroid. The centroids are first chosen randomly and, then, they are gradually moved towards the optimal configuration, which minimizes the distances between each point and the centroid of its cluster.

The pictures below show that both methods provide the same result: the picture on the left is a scattering diagram of the samples at the optimum sampling time, and the decision on the samples is then made by choosing visually the proper threshold according to the clusters identified. The picture on the right, conversely, shows the two clusters identified by the K-Means clustering algorithm at the optimum sampling instant, which is calculated by applying the clustering algorithm at the samples for each sampling instant and then choosing the configuration where the extreme points of each cluster are more distant from one another. The samples in the cluster whose centroid's absolute value is smaller (closer to 0) are converted to zeros, the other samples are converted to ones. This method does not require to choose a threshold, since the decision is made by dividing the samples into clusters and comparing their centroids.





Both methods provide a BER equal to zero, so the image is properly recovered, as shown in the picture below.



In conclusion, it is possible to deduce that the first algorithm to determine the threshold (average value of the eye diagram) works if the transmission is noiseless enough. Otherwise, the phase shift caused by the channel requires a more general approach to determine the threshold. Two similar methods have been analyzed at this scope. The first one consists in looking at the scattering diagram and determine the proper threshold according to it, whereas the second one splits the samples into two clusters using the K-Means algorithm and assigns each cluster to a transmitted value (0 or 1).

Another limitation of this transmission system is the maximum data rate achievable, which is limited by the sampling frequency of the audio card (44.1 kHz). In particular, the data rate is defined as R_S = Fsim / SpS, where SpS are the samples per symbol. The bandwidth of the signal in bandpass must be smaller than the simulation frequency, so the constraint can be expressed as:

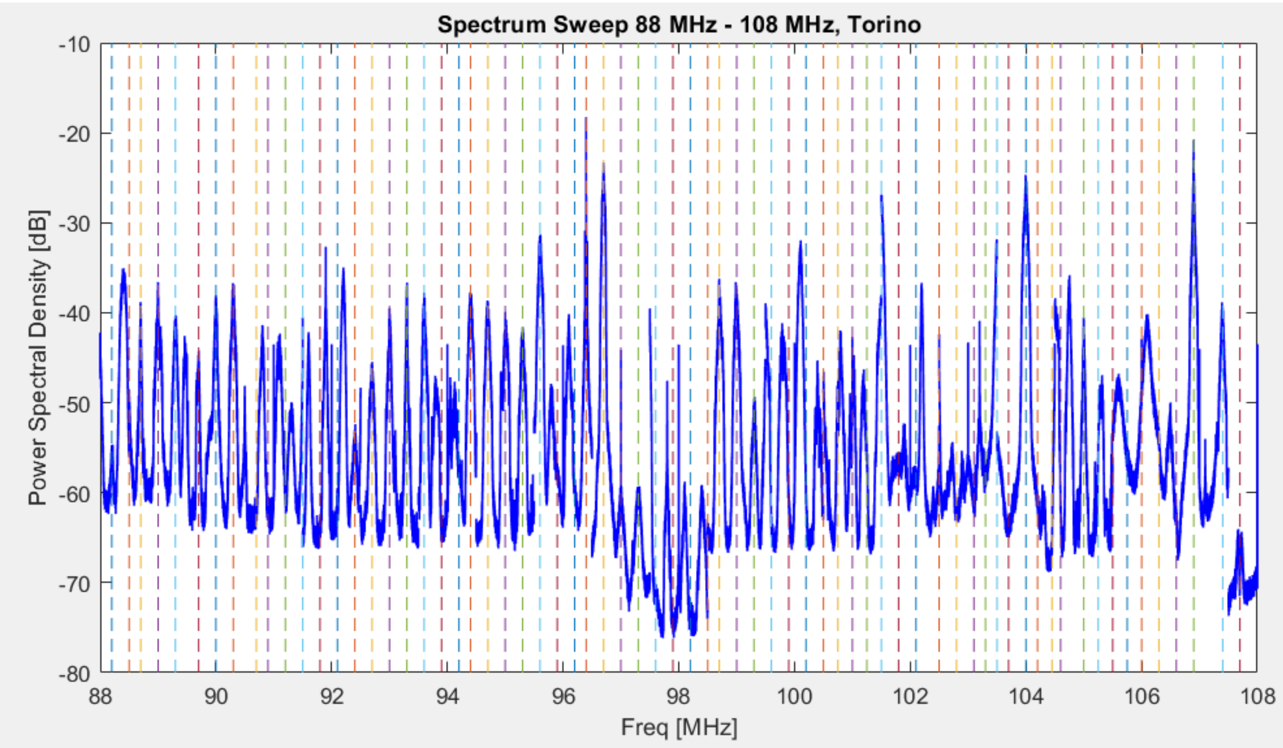
$$1.5 * R_s + f_0 < Fsim / 2 = 22.05 \text{ kHz}$$

This limits the possible values of SpS when f_0 (frequency of the cosine) is fixed. With the number chosen in the experiences reported above, the minimum SpS is 8 and, therefore, the maximum data rate available is R_S = Fsim / SpS = 5.5 kHz. Since these are the values used in the experiences described before, they can be viewed as examples of transmissions at the maximum data rate achievable by this configuration.

2. RF spectrum analyzer using RTL-SDR receiver

The aim of the second part of the project is to use the RTL-SDR receiver to perform a sweep over different radio frequency ranges and analyze the spectrum achieved. To make the spectrum smoother, the Bartlett periodogram is used. Since the tuner has a non-flat bandwidth, precautions need to be taken in order to avoid negative effects on the spectrum. Therefore, spectrum slices are partially overlapped and the overlap of two consecutive windows is then removed, so that the final plot is not affected by the tuner.

The first plot shows FM radio frequencies (88 MHz – 108 MHz) and it was performed in Torino. The dashed lines indicate the position of radio emitters in Torino. It is possible to notice that almost all the peaks in the spectrum correspond to radio stations. Some stations generate higher peaks than others, but almost all the stations are properly received.



The two following plots, instead, show the frequency range of DAB radio (174MHz – 240 MHz).