7 Single-carrier Acoustic Transmission

7.1 Introduction

So far you have learned about all the parts required to build a simple but functional receiver which reverts the effects of the wireless channel and recovers the distorted data. All the channel and timing effects have been generated with MATLAB code. Hence all the observed effects have been kind of a self-fulfilling prophecy.
Now we want to check whether our transmission assumptions have been any good and test the receiver with a real transmission. We will do this by using an acoustic link.

7.2 Acoustic channel

The audible spectrum of humans spans from about 20 Hz to 20 kHz (depending on the age and the number of attended rock concerts). The range that is usable for our acoustic transmissions is further limited by the operating range of the audio hardware. This parameter is called `frequency response`. For cheap equipment usually only a lower and an upper limit are available, for expensive audio equipment the manufacturer provides very detailed information about it.
Acoustic waves propagate with a speed of about $c = 340\,\mathrm{m/s}$ in air. Following the fundamental relation :

$$\lambda = \frac{c}{f}$$

we get wave lengths between $\lambda_{20\,\mathrm{Hz}} = 17\,\mathrm{m}$ and $\lambda_{20\,\mathrm{kHz}} = 1.7\,\mathrm{cm}$. For a typical carrier frequency of 4 kHz we get a wavelength of $\lambda_{4\,\mathrm{kHz}} = 8.5\,\mathrm{cm}$.
Acoustic transmission has several advantages over RF communication. The most important one is that you can hear the transmitted signal and partly debug it with your ears. Nevertheless the similarity to radio waves is large enough to get an insight on the fundamental challenges of wireless communication.

7.3 Hardware

Each group receives a set of hardware from us. It consists of:

- 1x analog microphone
- 1x set of USB speakers

We use low-cost generic computer components for our project. Such devices have usually a frequency range from around 100 Hz to 15 kHz. But be cautious: the performance of these devices over the specified frequency range is not flat. The behaviour can significantly change depending on your carrier frequency.

7.4 Sampling rate

You have to choose a sampling rate for the audio subsystem. The minimal requirement of the sampling frequency for our system is:

$$f_s >= 2 \cdot (f_c + B)$$

where f_c is the carrier frequency and B is the signal bandwidth. But this is a theoretical value that would require perfect filters. Usually the sampling frequency should fulfill:

$$f_s >> 2 \cdot (f_c + B)$$

Not all sampling rates are supported by all soundcards. Typical sampling rates for audio applications are 44100 Hz (CD), 48000 Hz and 96000 Hz. We will use 48000 Hz as default.

7.5 Passband transmission

Because it is not possible to transmit complex baseband signals directly over the real-valued audio-channel, we have to leave the baseband (BB) domain and perform the transmission in passband (PB).
You need to pick a suitable carrier frequency f_c. The range of possible values is limited by the sampling frequency, the signal bandwidth and the frequency response of your hardware. For a sampling rate $f_s = 48kHz$ a carrier frequency of $f_c = 4kHz$ is a good starting point, but you can also try $f_c = 6kHz$ and $f_c = 12kHz$.
The process of transforming the baseband signal into a passband signal is done with a mixer and is called upconversion. We use a `direct conversion` scheme, which means that we mix our signal directly with the carrier frequency.

$$\begin{aligned} s_{PB}[t] &= \Re\{s_{BB}[t] \cdot e^{j2\pi f_c t}\} \\ &= \Re\{s_{BB}[t] \cdot (cos(2\pi f_c t) + jsin(2\pi f_c t)\} \\ &= \Re\{s_{BB}[t]\} \cdot cos(2\pi f_c t) - \Im\{s_{BB}[t]\} \cdot sin(2\pi f_c t) \end{aligned}$$

The received signal $r_{PB}[t]$ must be converted back to the baseband. This is achieved by mixing it again with the carrier frequency.

$$\begin{aligned} r_{dc}[t] &= r_{PB}[t] \cdot e^{-j2\pi f_c t} \\ &= \frac{1}{2} s_{BB} + \frac{1}{2}(\Re\{s_{BB}\} - \Im\{s_{BB}\}) \cdot e^{-4\pi f_c t} \end{aligned}$$

In the noiseless case we get the original baseband signal distorted by some unwanted byproducts at twice the carrier frequency. Those can be removed by means of a low-pass filter.

$$\tilde{r}_{BB}[t] = 2 \cdot \mathrm{LP}\{r_{dc}[t]\}$$

We will provide you a suitable low-pass filter `lowpass.m` on moodle.

7.6 Transmitter specification

Information bits

To ensure that our assumptions regarding the demapping and the decoding are fulfilled, we use uniform distributed random bits as information bits. Choose the number of information bits such that you have a payload of 1000 symbols. For the Monte-Carlo simulations of the BER you have to simulate a sufficient number of payload blocks in order to ensure reliable results.

Modulation

Your transmission system should use BPSK for the preamble and QPSK the payload. Symbol power should be normalized to 1.

Framing

To detect the beginning of the data stream use a pseudo-random sequence in front of your payload. The preamble should be 100 symbols long and BPSK modulated.

TX Pulse

Use a *root raised cosine* pulse with a filterlength of 20 symbol periods and a roll-off factor of 0.22 for the transmit filter.

7.7 Receiver Specification

Build the best **causal** receiver you can. :)

7.8 MATLAB audio functions

There are multiple ways to facilitate the audio I/O with MATLAB. During our test we noticed that the built-in `audioplayer` functions from MATLAB can be unreliable. That's why we provide you a framework file (`audiotrans.m`) that contains 3 different methods to do the audio playback and recording, `matlab`, `native` and `bypass`. `matlab` and `native` have the same functionality but some computer might show issue with one or the other. You can use either od them as long as it manages to play and register audio. `bypass` is practical for debugging as it will bypass the physical channel.

7.9 Tasks

Use the provided `audiotrans.m` audio file as framework. It takes care of audio transmission and recording. Try to stay within the given structure. The data processing should be done in the tx() and the rx() function. The tx function we provide generates a single tone only. It can be used to verify whether transmission and reception works at all.

1. First implement the up and down conversion. Find a baseband signal that generates an unmodulated carrier wave (a continous sine). Check whether the received and down converted signal matches your expectations. (Note: You can verify the correctness of the transmission by ear.)
2. Build a transmission system according to the specifications given above for the acoustic channel. You can reuse code from the exercises. Start with a very slow symbol rate of $f_{sym} = 100$.
3. Plot the BER for a symbol rate beetween 100 and 2000. Explain what you observe regarding the matched-filtered pulses.
4. Introduce a frequency offset at the down-conversion. Describe the resulting problem. How can it be solved? How much frequency offset can your system handle?

7.10 Hints

1. Use enough bits for your BER plots. Usually you should have simulated 10x more bits than the BER you want to justify. this means there should be at least 10 errors to consider the result reliable.
2. Always ensure a fair comparison. Be aware of what you compare regarding energy and bandwidth.
3. If you have too much interference from others groups coordinate your carrier frequency with your neighboring cells.
4. Try to identify which kind of distortion is relevant in your current setup. You don't need to solve problems that are not relevant.
5. We provide you the `lowpass.m` function. This can be used to filter the signal after down-conversion.
6. MATLAB plots:
   - The `plot()` function supports different color and line styles. See `help plot` for the details. You can use this to display multiple datasets in the same graph (see `hold` command).
   - Make sure that you plot complex numbers correctly and don't discard any relevant information. Depending on your application you have to decide whether to plot as $\Re$ and $\Im$, magnitude and phase or scatter graph.
   - Label the axis and lines of your plots correctly (see `legend`, `xlabel`, `ylabel`).
   - Always plot the bit error rate axis logarithmic. A correct `grid` setting can improve the readability.

7.11 Historical Note

The thing you are implementing during this exercise is in fact an acoustic coupler. Such devices were used in the 80ies to connect computers over the public telephone network. They had to be physically attached to a telephone handset, because you were not allowed to modify your telephone or plugin your self-built modem. One such device is displayed at the museum-Bolo in the INF building.

Wikipedia/Rama/CeCILL

The original acoustic coupler used to transmit at a rate of 300 bits per second.