Project in Signals and Transforms

Wireless Audio Communication System

1 Introduction

In this project, you are going to implement a digital wireless communication system for transmitting data between two computers using audio signals. This requires you to derive the theoretical properties of the communication signal (i.e., the physical layer), design key components based on these properties and a set of constraints, and evaluate the performance of the system.

After successful completion of this project, you are able to:

- synthesize simple wireless communication signals and analyze their properties;
- select filter requirements and sampling frequencies based on signal information and extrinsic requirements;
- implement the components of a wireless communication system and the system as a whole.

Important:

- Before starting the project, read through all the instructions to get an idea of the project and tasks, only then start working on the tasks.
- In each task, there are a number of questions that you need to find answers for. Do not advance to the next task before you have answered all the questions. The tasks build on top of each other and require that you have completely and successfully solved previous tasks.

2 Assessment

The project is assessed based on a progress report, a final report as well as a live demonstration.

2.1 Progress report

A first progress report is due during or after the first lab contact session (course week 3). The progress report shall address Question 1 in Section 4. The progress report can be either done orally or written.

Oral: Oral reporting can be done during the scheduled lab session in week 3. Groups who wish to present their results orally must be ready to do so during the scheduled lab sessions. Note that each group is assigned to one specific lab session and reporting must be done during that particular session (the other sessions are reserved for the other groups). Oral reporting after the lab sessions is not possible.

Written: Groups failing to provide an oral report during the scheduled lab session as outlined above must hand in a written report addressing the above-mentioned questions. The written report must be no longer than 2 pages and handed in no later than one week after the scheduled lab session.

2.2 Final Report

The final report must describe your system and answer the points raised in Section 4 in a coherent way. Note that the report should be written as a technical report and *not in the style of an assignment* (also, do not repeat any questions asked below). Thus, it should typically contain the following sections:

- *Summary:* A brief summary of your work, including your approach, key performance measures, and results.
- *Introduction:* A brief introduction to the topic and the problem. The introduction should also outline the requirements and how you are going to measure the performance of your system.
- *Theory:* An overview of the theory. Do not make this too exhaustive; you do not need to repeat the whole theory section below. Instead, focus on the key bits according to the instructions in Section 4. Discuss elements of your system that were provided only on a high level and focus on the explanation and motivation of the design choices (e.g., frequencies, filter designs, etc.) as requested in Section 4.
- *Results and Discussion:* Experimental validation of your work including a clear description of your experimental conditions and set up, performance measures, etc.
- *Conclusions:* A reflection on your work.

The report must be no longer than 6 (six) pages using a font size of at least 11 pt, including all figures, tables, etc. but excluding the title page and references. All reports exceeding the page limit will be returned for supplementation without grading. Note that there is only one chance for supplementation.

Recall to discuss all your chosen parameters and the motivation for your choice. Illustrate important aspects such as key spectra on which you base your decisions or filter frequency responses. The general idea is that someone familiar within the field but unfamiliar with the project should be able to read the report and be able to reproduce your results.

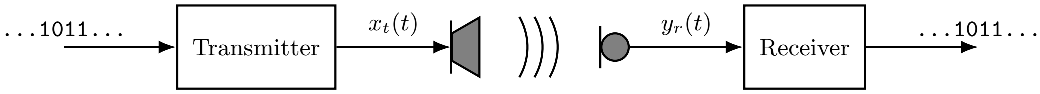

Figure 1. Illustration of a wireless communication system using audio signals.

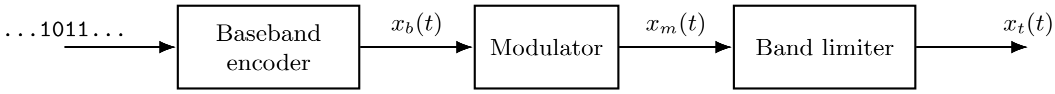

Figure 2. Block diagram of the transmitter.

Also remember that all figures must be referenced in the text, they must be properly scaled such that they are legible, and they must have a caption and be properly labeled (including units, where applicable). Do not use narrative language and be precise. Proof-read your report before submitting.

2.3 Demonstration

During the 15 minutes live demonstration, you will first give a walk-through of your solution (i.e., the code) and demonstrate how it works. You will then be provided with several *binary bit sequences*, that you have to transmit using your system and decode the message at the receiver. Further details and the schedule will be announced on the course homepage.

3 Theory

The general architecture of the wireless communication system is illustrated in Figure 1 and consists of a transmitter and a receiver, both implemented in a computer, and D/A and A/D conversion are performed by the computer's soundcard. Hence, the main task is to implement the transmitter and receiver components and both are described in more detail below.

3.1 Transmitter

The transmitter consists of three steps (Figure 2):

1. Baseband signal encoding,
2. modulation, and
3. band-limitation.

Baseband signal encoding. The input to the system is a binary bit sequence consisting of zeros and ones, for example representing a string or a binary file. The bit sequence has to be encoded into a baseband signal $x_b(t)$. In the baseband signal, each bit is represented by a *symbol*. In our case, it is convenient to encode the bits as pulses of width T_b seconds with signal value -1 for zeros and $+1$ for ones as illustrated in Figure 3a. Hence, a length N message can be described by

$$x_b(t) = \sum_{n=0}^{N-1} b_n \operatorname{rect}\left(\frac{t - \frac{(2n+1)T_b}{2}}{T_b}\right),$$

where b_n is the signal value of the nth bit ($+1$ or -1).

Note that the width of the pulse T_b determines the bit rate, which is given by

$$R = \frac{1}{T_b} \text{ bits/s}.$$

Modulation. Unfortunately, the baseband signal can not be transmitted directly and the signal has to be *modulated*. In the spectrum, this corresponds to a shift of the baseband signal centered around zero to a higher frequency region, centered around the *carrier frequency* ω_c.

There exist many modulation techniques such as amplitude shift keying, frequency shift keying, or phase shift keying. Here, we are going to use binary phase shift keying (BPSK) because it is more robust than the other techniques and the basis for many modern wireless communication systems such as WiFi. The basic idea of BPSK is to encode the two symbols of the baseband signal as two different phases φ_0 (for a zero, symbol -1) and φ_1 (for a one, symbol 1) of a high-frequency *carrier* signal $x_c(t)$. Assume that the carrier signal is

$$x_c(t) = A_c \sin(\omega_c t).$$

where A_c is the amplitude of the carrier. Then, the BPSK modulated signal can be written as

$$x_m(t) = \begin{cases} A_c \sin(\omega_c t + \varphi_0) & \text{for } x_b(t) = -1, \\ A_c \sin(\omega_c t + \varphi_1) & \text{for } x_b(t) = 1. \end{cases}$$

The phase shifts φ_0 and φ_1 should be chosen such that the two resulting components of the modulated signal are as far from each other as possible, which is achieved by maximizing the distance between the two angles. Hence, in BPSK, we use the following phase shifts:

$$\begin{aligned} \varphi_0 &= -\pi, \\ \varphi_1 &= 0. \end{aligned}$$

This yields

$$x_m(t) = \begin{cases} A_c \sin(\omega_c t - \pi) & \text{for } x_b(t) = -1, \\ A_c \sin(\omega_c t) & \text{for } x_b(t) = 1. \end{cases} = \begin{cases} -A_c \sin(\omega_c t) & \text{for } x_b(t) = -1, \\ A_c \sin(\omega_c t) & \text{for } x_b(t) = 1. \end{cases}$$

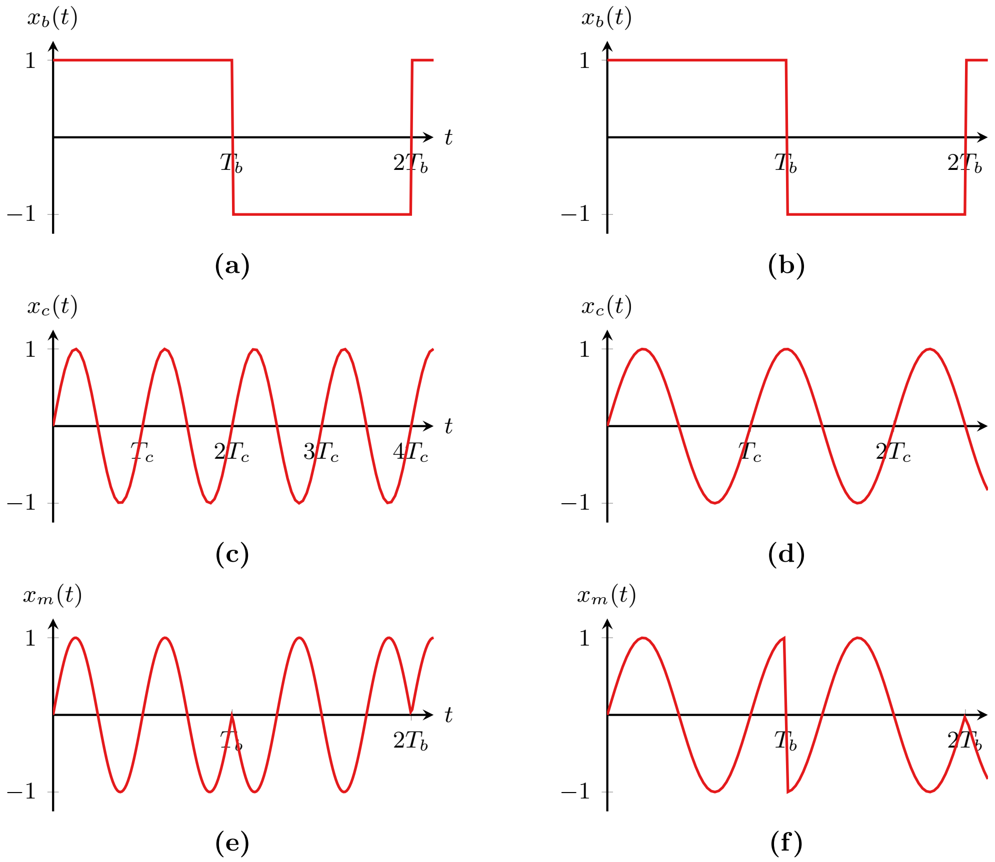

Figure 3. Illustration of BPSK with good (left column) and bad (right column) parameter choices. (a)–(b) Baseband signal, (c)–(d) carrier, and (e)–(f) BPSK-modulated signal.

Thus, the modulated signal can also be written as

$$x_m(t) = x_b(t)x_c(t) = x_b(t)A_c \sin(\omega_c t). \tag{1}$$

and is illustrated in Figure 3.

Note that to avoid sharp phase jumps, the pulse duration should be an integer multiple of the carrier signal's period, that is,

$$T_b = MT_c = \frac{M}{f_c},$$

where M is a natural number (Figure 3). (This is not strictly necessary, but adhering to it improves performance and reduces undesirable effects.)

The choices for f_c and T_b depend on the requirements and limitations of the spectrum imposed onto the communication system, which will be discussed next.

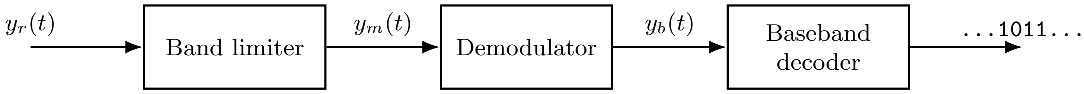

Figure 4. Block diagram of the receiver.

Band-limitation. Before the modulated signal $x_m(t)$ can be sent, we also have to make sure that it adheres to the bandwidth requirements: To avoid interference with other communication systems, each system is only allowed to use a limited frequency range. Hence, the modulated signal has to be band-limited using an appropriate filter to ensure that the frequency range requirements are not violated. This finally yields the transmitter signal $x_t(t)$.

Each group is provided with their own, specific frequency band, which serves as the basis for their system design, see Appendix A.

3.2 Receiver

On the receiver side, the steps of the transmitter have to be reversed to recover the baseband signal $x_b(t)$ and eventually, the bit sequence (Figure 4).

The signal received by the receiver $y_r(t)$ is the transmitted signal $x_t(t)$ which has traveled through the transmission channel (the air) with (unknown) impulse response $h(t)$ plus interference from other audio sources and noise $v(t)$. This can be written as

$$y_r(t) = h(t) * x_t(t) + v(t).$$

Assuming that the channel's frequency response is fairly constant for the frequency band used by the communication system[1] and neglecting the noise, this can be written as

$$y_r(t) \approx |H(\omega_c)| x_t(t - t_0) + v(t) \approx A_r x_t(t - t_0),$$

where $A_r = |H(\omega_c)|$ is the channel's attenuation and $t_0 \approx -\frac{\varphi_r}{\omega_c}$ is the time-delay introduced by the unknown phase shift $\varphi_r = \angle H(\omega_c)$ of the channel.

Band-limitation. First, since the microphone picks up all possible sounds, even sounds outside the band used for transmission of the communication signal, the received signal $y_t(t)$ needs to be band-limited by removing all components outside the desired frequency band to remove (minimize) the effect of the interference $v(t)$, which yields the received modulated signal

$$y_m(t) \approx A_r x_m(t - t_0) = A_r x_b(t - t_0) \sin(\omega_c t + \varphi_r)$$

This is (an approximation of) the modulated signal $x_m(t)$, which has been scaled and phase shifted by the unknown quantities A_r and φ_r.

[1] In practice, this is a quite strong assumption that is easily violated, but it is still practical for our purposes.

Demodulation. After band-limitation, the modulated signal has to be demodulated to recover (an estimate of) the baseband signal $x_b(t)$. In the spectrum, this corresponds to a down-shift in frequency, that is, moving the frequency components that are centered around the carrier frequency ω_c to be centered around $\omega = 0$.

In this project, we use in-phase quadrature (IQ) demodulation. In IQ demodulation, the received high-frequency signal $y_m(t)$ is divided into two components (an in-phase, cosine and a quadrature, sine component) by multiplying the received (and filtered signal) with a sine and cosine of the carrier frequency, that is,

$$y_{I,d}(t) = y_m(t)\cos(\omega_c t) = A_r x_b(t - t_0)\sin(\omega_c t + \varphi_r)\cos(\omega_c t), \tag{2a}$$

$$y_{Q,d}(t) = -y_m(t)\sin(\omega_c t) = -A_r x_b(t - t_0)\sin(\omega_c t + \varphi_r)\sin(\omega_c t). \tag{2b}$$

This does indeed *down-mix* the high-frequency signal to the baseband, but also creates a new high-frequency component at twice the carrier frequency. Hence, to obtain the pure I and Q baseband signals $y_{I,b}(t)$ and $y_{Q,b}(t)$, the signals obtained from (2) need to be lowpass filtered. This yields the complete, *complex-valued* baseband signal

$$y_b(t) = y_{I,b}(t) + \mathrm{j}\, y_{Q,b}(t).$$

The symbol information was encoded in the phase of the signal, which can be recovered from the IQ-signals as

$$y_{\text{phase},b}(t) = \arctan 2(y_{Q,b}(t), y_{I,b}(t)),$$

where $\arctan 2(\cdot)$ is the four-quadrant arc tangent function. Finally, the magnitude

$$y_{\text{mag},b}(t) = \sqrt{y_{I,b}(t)^2 + y_{Q,b}(t)^2}$$

of the I and Q components can be used to detect transmissions (or absence thereof).

Decoding. The final step is to decode the baseband signal into bits and eventually bytes corresponding to the input data. However, since the baseband signal is scaled and time-shifted by A_r and t_0, respectively, this has to be done very carefully.

A crude but pragmatic way of determining the start of the signal is by looking at the magnitude of the IQ-demodulated signal: The magnitude $y_{\text{mag,b}}(t)$ should only be significantly different from zero if there is a -1 or $+1$ in the baseband signal $x_b(t)$. Hence, assuming that there is only one transmission, it is enough to detect the first time instant when $y_{\text{mag},b}(t)$ exceeds a threshold γ. This will yield (a coarse approximation of) the time-shift t_0.

Once, the time shift is determined, the bit values can be read from the phase signal. Since we only have to distinguish two phases, which are either positive or negative and offset by π, we can simply consider the sign of the phase. A simple way to do this is, for example, to determine the sign of the phase signal each T_b samples,

$$b_y(t) = \operatorname{sgn}\left(y_{\text{phase},b}(t_0 + nT_b)\right)$$

for the nth bit in the sequence.

A final complication is that the phase shift introduced by the channel may actually flip the signs of the -1 and $+1$ symbols (e.g., if the shift is $\pm\frac{\pi}{2}$). This can be resolved by always transmitting a known bit sequence first, which can be used to determine whether the signs have been flipped or not.

4 System Design and Implementation

Your main tasks in this project are to design and implement the key components of the communication system. The following steps guide you through this process for you to successfully complete it. In order to make progress, *you need to find answers to the questions raised*[2].

You do not have to implement all the components outlined in Section 3 yourself. In particular, Python methods for encoding and decoding strings, and encoding and decoding the bit sequences are provided on the course homepage. In the same package, you will also find a Python skeleton for your implementation.

1. **Modulator and demodulator analysis and design.** The first step is to analyze how modulation and demodulation works and analyze the result with respect to the requirements.

   a) Use (1) to find an expression for the spectrum of the modulated signal $x_m(t)$. Recall that $x_b(t)$ can be expressed as a sum of rectangular pulses of width T_b and amplitude b_n.

   b) Based on the expression for the spectrum $X_m(\omega)$ determined above and the assigned channel (see Appendix A), determine a suitable carrier frequency ω_c and symbol duration T_b. For reasonable performance, at least the main- and first one or two sidelobes should fall inside of your assigned frequency band. Here, it might help to sketch a qualitative magnitude spectrum for $X(\omega)$.

      What bitrate can you achieve with this choice? Is there a trade-off involving the bitrate?

   c) Determine the maximum amplitude A_c of your carrier signal to avoid violating the energy constraint for your group provided in Appendix A.

   d) Simplify the expressions for the in-phase and quadrature signals in (2), that is, try to eliminate the product between sine and cosine components such that you are only left with sums of sines and cosines. Show that the resulting signals $y_{I,d}(t)$ and $y_{Q,d}(t)$ indeed contain the baseband signal, as well as a component at twice the carrier frequency $2\omega_c$.

[2] You also need to provide the answers to these questions in your report, but please read the reporting instructions in Section 2 carefully.

Based on these expressions, it might be tempting to just use a filtered version of one of the two signals as the baseband signal $y_b(t)$. Why is this a bad idea?

Hint: What is the effect of the unknown phase shift φ_r on the in-phase and quadrature signals? How can it be eliminated?

2. **Filter specification.** Based on the expressions above, you need to design two filters: A band-limiting filter for both the transmitter and receiver, and a lowpass filter for the demodulator.

   a) Determine suitable specifications for your band-limiting filters, taking the frequency band requirements in Appendix A into account. Note that there is a small gap between each frequency band, which may be used as the transition band, but you are not allowed to interfere with your peers.

      If properly designed, you can use the same band-limiting filters for both the transmitter and receiver.

   b) Determine the bandwidth of the demodulated baseband signal. Then, determine specifications for your lowpass filter that removes high-frequency components introduced by the IQ demodulation. Note that the bandwidth of this lowpass filter should match the bandwidth of your baseband signal to avoid noise from corrupting the baseband decoding.

3. **Sampling frequency.** The communication system is to be implemented on a computer. Hence, the signals have to be generated as discrete-time signals. Thus, determine a suitable sampling frequency. In doing this, note that:

   i. To not violate the Nyquist–Shannon sampling theorem, you must give significant extra room in the spectrum as demodulation will give components centered at twice the carrier frequency ($2\omega_c$) as shown above, and

   ii. the sampling frequency should be chosen such that a pure discrete time sine is obtained, that is,
   $$K_c T_s = T_c$$
   where K_c is the period of the discrete time carrier signal. (This is not strictly necessary but improves performance.)

   Hence, your sampling frequency should fulfill the following two conditions:
   $$\omega_s \gg 4\omega_c, \tag{3a}$$
   $$K_c T_s = T_c. \tag{3b}$$

4. **Implementation.** You are now finally ready to start your implementation. On the course homepage, you can find a template as well as a library implementing baseband encoding and decoding to get started. Note that you do not have to make a program that runs continuously in real-time, it is enough to transmit and receive a single binary string of data at a time.

Start by implementing the necessary filters as well as the transmitter and receiver and use the provided channel simulator to simulate a transmission.

a) Implement the design of your band-limiting filters as *digital filters*. To do this, you can, for example, use SciPy's `signal.iirdesign()` function. See the SciPy documentation on how to use the function to design different types of filters (Butterworth, Chebyshev, etc. as well as lowpass, bandpass, etc.).

   To filter a signal later on, you may use `signal.lfilter()` (or any other suitable *discrete time* filtering function, depending on how you designed your filter to begin with).

b) Implement your transmitter. On the course homepage, you will find functions for encoding strings to a binary bit sequences and binary bit sequences to baseband signals (see the skeleton file). Hence, you only have to implement the modulator and band-limiter.

   Note that it might be a good idea to plot your signal at different stages throughout the processing chain for debugging.

c) Implement your band-limiter and demodulator. Once you have retrieved the baseband magnitude and phase signals from the transmission, you can use the functions for decoding the baseband signal into a bit sequence and the bit sequence into clear text provided on the course homepage (see the skeleton file).

Once you have completed the above steps and tested your system in the simulator, make your system run on two different computers. To do this, implement two separate programs, one for the transmitter and one for the receiver. The transmitter should transmit the signal using the speakers, while the receiver should record the signal from the microphone and decode the received signal. You can, for example, use the Python module `sounddevice` for that.

4.1 Evaluation

Evaluate the performance of your system in terms of how well the transmitted message is recovered from the received signal. For example using the simulator, investigate the following aspects (of course you may also come up with your own performance measures):

- Choose a few test sentences of different lengths (short, middle, long) and do several transmissions of the sentences, for example, ten times.
- How many of the messages have been recovered completely? What is the error rate for your messages? Does the error rate depend on the message length?
- When there are errors, is it individual letters that are changed or whole words?
- Compare the bit sequences of the transmitted and received messages. What is the bit error rate, that is, the rate of wrongly decoded bits?

For the real implementation, experiment with different environments, distances between receiver and transmitter, etc. Add disturbances such as light talking or some background music. How robust is your system? Can you successfully communicate at the same time as another group is communicating?

A Communication Channels and Transmission Power

Table 1. List of frequency channels and their maximum average transmission power.

Channel	Group	Frequency Band	Max. Average Power
1	5	900 Hz–1100 Hz	30 dBm
2	8	1150 Hz–1250 Hz	33 dBm
3	12	1300 Hz–1500 Hz	27 dBm
4	21	1550 Hz–1650 Hz	33 dBm
5	18	1725 Hz–1875 Hz	33 dBm
6	3	1950 Hz–2050 Hz	33 dBm
7	17	2100 Hz–2300 Hz	30 dBm
8	19	2400 Hz–2600 Hz	27 dBm
9	4	2700 Hz–2900 Hz	30 dBm
10	13	3050 Hz–3150 Hz	33 dBm
11	14	3200 Hz–3400 Hz	33 dBm
12	10	3475 Hz–3525 Hz	27 dBm
13	9	3550 Hz–3650 Hz	33 dBm
14	16	3750 Hz–3850 Hz	30 dBm
15	7	3900 Hz–4100 Hz	30 dBm
16	11	4150 Hz–4250 Hz	33 dBm
17	15	4300 Hz–4500 Hz	27 dBm
18	6	4550 Hz–4650 Hz	33 dBm
19	2	4750 Hz–4850 Hz	30 dBm
20	20	4900 Hz–5100 Hz	27 dBm
21	1	5150 Hz–5250 Hz	30 dBm

Notes:

- The average power of the sine $x(t) = A\sin(\omega_0 t + \varphi)$ is given by

$$P = \frac{1}{T_0}\int_0^{T_0} |A\sin(\omega_0 t + \varphi)|^2 \, \mathrm{d}t = \frac{A^2}{2} \text{ W}.$$

- The conversions between power and power in decibel-milliwatts (dBm) is

$$P_{\text{dBm}} = 10\log_{10}\left(\frac{P}{1\,\text{mW}}\right) \text{ dBm} \quad \text{and} \quad P = 10^{\frac{P_{\text{dBm}}}{10}} \text{ mW}.$$