Guide Book –
Project in Wireless Communication, Part II and III

Guoda Tian, Dept. of Electrical and Information Technology, Lund University

November 3, 2024

Welcome to the second study period! In this study period, we will continue the journey to the wireless world. Unlike the first part, the project in this study period involves hardware, which requires solid communication theory background, signal processing knowledge, as well as programming skills. Therefore, this project course is time-consuming, please start as early as possible!

1 Part 2

1.1 A review of digital signal processing and OFDM

Before starting the project, it is never a bad idea to recall the basic theory about digital signal processing and OFDM. If it is difficult to answer the following questions, please go back to the textbook.

- What are definitions of analog frequency F_ω, frequency at discrete domain f, and sampling frequency F_s? What are the mathematical relationships between them? What is the range of frequencies used, usually, when plotting a digital spectrum?
- When we plot a digital frequency spectrum, which part represents the negative frequency, and which part the positive frequency?
- Is the digital spectrum symmetric for baseband signals, why or why not?
- Why do we use the DFT for OFDM systems? If there are 1024 subcarriers, how many DFT points do we need? Which subcarriers belong to the positive frequencies and which belong to the negative frequencies? Where are the subcarriers located in the digital spectrum?
- What are the definitions of upsampling and downsampling? If we upsample/downsample a signal by a factor of 2, what changes can we observe in the spectrum?
- The following question is borrowed from course *Digital Signal Processing*, which somewhat resembles our project.

2. A signal s(t) has an analog Fourier Transform that satisfies

$$S(F) \neq 0,\ 4000 \leq |F| \leq 6000 \text{ and 0 otherwise}$$

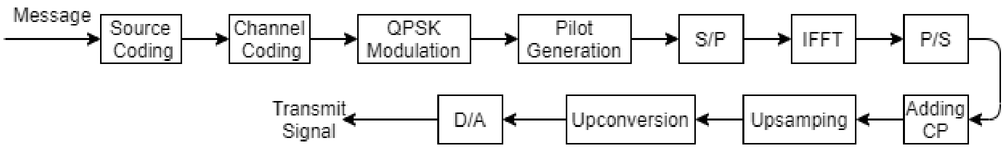
Assume the following processing of the signal s(t):

(I) Sample the signal with sampling frequency $F_s = 20000\ Hz$ to obtain s(n)
(II) Construct $y(n) = s(n)\left[e^{i2\pi n0.1} + e^{-i2\pi n0.1}\right]$
(III) Low pass filter $y(n)$ with an ideal low pass filter with normalized cutoff $f_c = 0.25$.
(IV) Perform D/A conversion with sampling frequency $F_s = 10000\ Hz$.

Plot the analog Fourier transform of the resulting time signal.

1.2 Physical layer transmitter design

It is required in task 2 to program the physical layer of an OFDM system. Since we have already finished the receiver part in Task 1, we only need to program the transmitter part before launching the whole system. The transmitter block diagram is illustrated as follows:



1.2.1 Source coding, channel coding and modulation

In order to transmit a message or an image, the first step is to perform source coding and generate bit streams. Observe that we do not require advanced source coding algorithms, therefore feel free to use built in functions. Note, however, that maybe you need a scrambler.

When it comes to error-correcting coding, you are free to choose any coding scheme you have learned. A recommendation can be a convolutional code, since you already used that in part 1. In addition, built-in functions are also allowed to be used in this step. The stronger code you choose, the less retransmission is required in the next task.

A standard modulation scheme that can be used is QPSK for the audio channel. However, feel free to choose higher-order modulation schemes such as 16 QAM, etc.

1.2.2 OFDM signal generation

The first step is to select a pilot, it is recommended to use the same pilots as task 1. In addition, it is your choice to select the number of subcarriers performing the IDFT. Finally, do not forget that the pilots also need a cyclic prefix (CP) to protect them from inter symbol interference (ISI).

1.2.3 Upsampling, upconversion and D/A

This is one of the most important sections in this project. First, we must figure out why we need upsampling from a signal processing perspective. It is recommended to plot the spectrum and the signal itself at each step, especially before transmitting the signal. Keep in mind that both the microphone and the speaker have limited bandwidth; therefore, it is not possible to transmit an audio signal at whatever frequencies you want. Some hints are listed as follows:

- Find out the available frequency spectrum for your microphone and speaker. For highly motivated students, channel sounding is a great idea!
- Please always take a look at the spectrum and the signal itself before you transmit! It is important to make sure that the transmit spectrum is as expected!
- Suppose that we use 128 subcarriers in total, can you find out the analog frequencies that correspond to subcarrier numbers 5 and 123?

1.3 Some general hints for debugging the physical layer

- Start simple! Try to load the transmit signal directly onto your receiver script at first! If you could successfully decode the signal, then involve the audio channel.
- A solid understanding of signal processing and communication theory plays an important role in the success of this project. Therefore, it is worth reviewing the theory.
- You should check carefully if synchronization works at the receiver side, in other words, you should check if there is a clear sharp pulse.
- Plot the received signal, does it look as expected?
- Much information is embedded in the constellation diagram on the receiver side and thus it is recommended to plot it when debugging the system.

1.4 Implementing the packet based system

After successfully programming the physical layer, it is required to consider a simple higher layer protocol and implement a packet-based system. One recommendation is to consider the stop-and-wait automatic repeat request (ARQ) protocol; however, feel free to apply more advanced network protocols that you learned. To apply ARQ, you should first divide the message into small packets, followed by adding a label to each packet at the beginning if and CRC at the end. In addition, you should also design feedback signals such as ACK and NACK, so that the receiver can give feedback to the transmitter after receiving each packet. Keep in mind that those signals are more important than the payload data itself, therefore, stronger error correction code should be applied to add additional protection. Some hints are given as follows:

- The performance of link layer relies on the reliability of the physical layer. Therefore, before you implement this step, it is important to make sure that the physical layer works as expected.
- Some previous students used OFDM for feedback signals. They chose the ACK signal as 1 and the NACK signal as 0 for all the subcarriers, is it a good idea? Similarly, if a packet has every bit as 1 or 0, how to transmit this packet?
- For synchronization between the transmitter and receiver, you can use the tik and tok commands.

2 Part 3, Radio System Design

2.1 Introduction

The purpose of part 3 is to transmit a message or image from one software defined radio (SDR) device to another device via antennas. Compared to the previous task, the carrier frequencies at both the transmitter and the receiver are much higher, and the hardware typically requires carrier frequency offset compensation. In addition, the up and down conversion carrier signals, which differ from the previous part, are generated by oscillators in the *analog* domain. In this task, you are expected to work more independently; therefore, the help manual will focus more on higher-level instructions, rather than explaining every detail.

2.2 Some hints for this project

- Based on your own understanding, please draw the block diagram for both the transmitter and the receiver. Compared to the block diagrams illustrated in Task 3 and 4, what are the key differences? Do we still need up-sampling and down-sampling? Why or why not?
- When it comes to CFO compensation, you can use the algorithm illustrated in the lecture slides. For highly motivated students, you can also refer to the paper *Timing and Frequency Synchronization in OFDM Systems Using the Cyclic Prefix.* In addition, why it is not necessary to run CFO compensation algorithm for the audio task?
- There are some Matlab examples for Pluto on the website of Mathworks. It is a good idea to run some of them at first in order to make sure that the hardware works. I strongly recommend starting with the *Image Transmission* example. you can find it by searching for *Image Transmission and Reception Using WLAN Toolbox and One PlutoSDR.* To study the entire Matlab code, enter *edit plutoradioWLANTransmitReceiveExample* in the Matlab command window. The example code uses *wlanWaveformGenerator* to generate the full OFDM transmission signal; however, you should definitely program it yourself, the same applies to the receiver side.
- Start simple! Begin with cable and one device first, then extend to antennas and two devices!
- Do not forget the continuous pilot symbols. They are there to help. Study what happens to the amplitude and phase of them within your packets. Does your system behave as expected? If not, find the reason for it. If so, congrats!

Good luck!