ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 13 Modern Digital Communications
Assignment 9 November 19, 2025 (due Nov 25, 2025)

OFDM

Goal: The purpose of this lab exercise is to become acquainted with OFDM (Orthogonal Frequency Division Multiplexing). This is an ingenious and currently quite popular technique used to communicate across channels that do not have a flat frequency response (in the band of interest).

Assignment: The goal of this homework assignment is to create an OFDM transmitter/receiver pair. The transmitter consists of a function called `ofdm_tx_frame()` that produces a sequence of OFDM blocks. Besides some parameters, it takes three input vectors. The first input vector, assumed to be known to the receiver, contains the training symbols used to estimate the channel. The second vector specifies which carriers should be used for transmission. Finally, the third vector contains the message symbols. The parameters specify the number of carriers per OFDM symbol and the prefix-length.

Functions that you may need include `fft()`, `ifft()`, and those already programmed by you for QAM modulation/demodulation.

As in the previous assignments, we provide the solutions for every function that you should write. You can use these solutions to test a whole working system, maybe just substituting one function at a time with your implementation. For MATLAB, the script `function_mapper_ofdm` is used to select for each function whether to use the provided solution or your own implementation. For Python, the functions that you need to write are grouped under `my_utilOFDM.py`.

You can test your implementation by running the script `test_ofdm()`, which implements a whole OFDM transmission system.

Exercise 1. Implement the function `ofdm_tx_frame()` that generates an OFDM frame.

`PSD_MASK` specifies which carriers should be "turned off." This is useful, for example, to avoid transmitting on specific frequencies that interfere with other communications. Note that OFDM offers the possibility to shape the transmitted signal's spectrum by independently adjusting the power on each carrier.

The length of `TRAINING_SYMBOLS` should be equal to the number of useful carriers (i.e., number of ones in `PSD_MASK`), so that there is one training symbol per used carrier.

The length of the output vector `TX_SYMBOLS` will be a multiple of `(NUM_CARRIERS + PREFIX_LENGTH)`.

Exercise 2. Implement the function `ofdm_rx_frame()` that implements the receiver for an OFDM transmitter as the one designed in the first exercise.

Exercise 3. Implement the function `channel_est1()`.

This function determines the channel coefficients in the frequency domain, `LAMBDA`, from the channel impulse response, `h`. (The channel response is assumed to remain constant during the whole frame.) The entries of `LAMBDA` are the strengths of parallel channels created by OFDM.

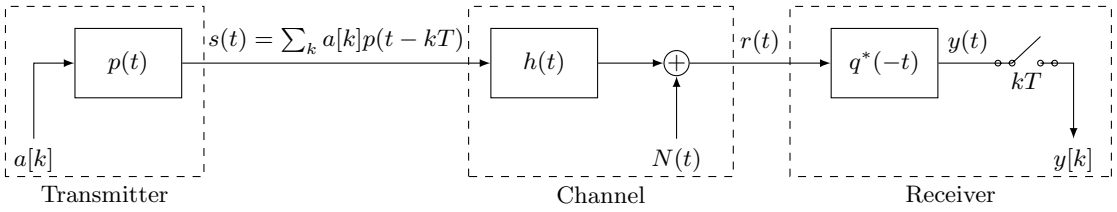
By commenting/uncommenting certain lines of `test_ofdm()`, you can choose different methods to create the channel impulse response `h`, namely[1]

- The line `h = [1 zeros(1,channel_length-1)];` implements a channel with no ISI, just AWGN noise (useful for debugging).
- The line `h = 1/sqrt(2) * (randn(...))` sets randomly an impulse response with the only constraint that its length is shorter than the cyclic prefix. You can use this as second step to check that your implementation is completely generic and that it works correctly with ISI channels.
- Finally, with `create_multipath_channel_filter` you can generate the impulse response `h` (at symbol rate) of a multipath channel.

[1] Using MATLAB syntax, Python is similar.

Exercise 4. *(Symbol-level Channel)*

In this exercise we derive the relation between the symbol-level equivalent discrete-time channel and the physical (continuous-time) channel response. Consider the following communication system:



The transmitter converts the data symbols $a[k]$ to the transmitted signal $s(t) = \sum_k a[k]p(t-kT)$. The channel filters the signal with a linear filter of impulse response $h(t)$ and adds $N(t)$ which is assumed to be complex-valued AWGN with zero mean and variance $\frac{N_0}{2}$ per real-valued dimension. The receiver passes the received signal through a matched filter with the impulse response $q^*(-t)$ and samples its output at times $t = kT$, $k \in \mathbb{Z}$ to form the observables $y[k]$. (Thus, the observables are the projections of the received signal onto the space spanned by $\{q(t-kT)\}_{k\in\mathbb{Z}}$.)

You have already derived the equivalent symbol-level channel for the above communication system in Assignment 3, when we discussed about the ISI. Nevertheless, it is good to repeat the exercise here:

1. Convince yourself that from the receiver's perspective, the communication system is equivalent to the one that uses the pulse-shaping filter $(p \star h)(t)$ and transmits the signal through an ideal AWGN channel.

2. Show that the observables $y[k]$ have the form

$$y[k] = \sum_n a[k-n]h[n] + z[k] \,,$$

   where $z[k]$ is a Gaussian random variable and

$$h[n] = p(t) \star h(t) \star q^*(-t)\Big|_{t=nT}.$$

   Also, compute the mean and covariance of the random process $z[k]$.

3. Show that if $q(t)$ is a Nyquist pulse, i.e., $\langle q(t), q(t-kT)\rangle = \mathbb{1}\{k=0\}$, then $z[k]$ is a white Gaussian noise process.

4. Find the conditions on the pulse shape $q(t)$ under which the sequence $y[k]$ will be a sufficient statistic for deciding on data symbols $a[k]$.

5. For the special case where

$$h(t) = \sum_{l=0}^{M-1} \alpha_l \delta(t-\tau_l)$$

   is the response of an M-path channel with strengths $\{\alpha_l\}$ and propagation delays $\{\tau_l\}$, show that the symbol-level equivalent channel becomes

$$h[n] = \sum_{l=0}^{M-1} \alpha_l f(nT-\tau_l) \,,$$

   where $f(t) = p(t) \star q^*(-t)$.

Note. Recall from Assignment 3 that an offset in the sampling time of the receiver (with respect to the sampling time of the transmitter) is equivalent to additional delay in the channel. In fact, by looking at the sampler's output we cannot tell whether the delay is in the channel or in the sampling time. Therefore, once the OFDM receiver can estimate the channel and equalize it (as we will see in the next lecture) it will automatically be 'robust' against sampling clock offsets.