The goal of this graded exercise is to get an idea of your general understanding of the concepts introduced during the first half of the course. The exercise consists of two parts. The first part is concerned with OFDM channel estimation and the second part is concerned with DSSS/CDMA. Both parts contain short programming tasks as well as theoretical questions. The theoretical questions can be answered independently of the programming tasks.

Part 1: OFDM Channel Estimation

Task 1

In the first exercise you implemented several channel estimation schemes for an OFDM system. What all schemes had in common was that they used a whole OFDM symbol for channel estimation, which was then followed by data OFDM symbols. Another possible way of doing channel estimation, which was presented in class, is by using a "comb"-like structure for the training symbols. This scheme make use of the fact that neighboring subcarriers are correlated. Possibly the simplest implementation of this scheme is to use every M-th subcarrier within the OFDM symbol as training data.
You are given a function `OFDMTask` which simulates OFDM transmission of a single OFDM symbol over a block fading frequency selective channel. The input arguments of the function are only the SNR (in dB) and the combing parameter M. The OFDM system uses the following parameters:

Number of carriers	256
Modulation	4-QAM
Length of cyclic prefix	16
Length of channel impulse response	8

Table 1: OFDM Simulation Framework Parameters

Every M-th subcarrier within the OFDM symbol is used as a training symbol. Unfortunately, the Lazy Teaching Assistants (LTAs) have forgotten to implement the channel estimation part, rendering the whole simulation framework useless.

Your Task

Help the LTAs by implementing comb-like channel estimation in the `ofdm_channel_estimator` function. To this end, we suggest that you first understand exactly how the framework works, as well as how and where the training subcarriers are inserted. It is sufficient if your channel estimation works correctly for $M = 2$. To verify that your solution is correct, draw the actual channel and your estimation on a common plot (i.e., use `doPlot = 1`) for high SNR.

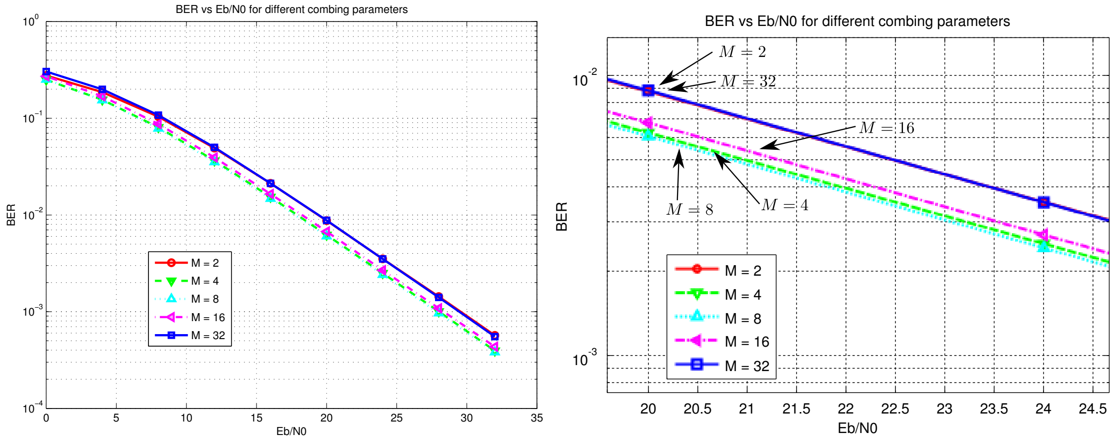

Figure 1: Bit error rate vs. combing parameter M.

Task 2

The LTAs are grateful for your help, channel estimation is now working perfectly! However, your help is once more needed to explain a peculiar behavior which the LTAs observed. Let us first define Eb/N0 (i.e., the energy per bit to noise power spectral density ratio) in the context of channel estimation. Recall that, when using a channel code of rate R, the relation between (linear) SNR and Eb/N0 is Eb/N0 $= R \cdot$ SNR. When using a comb-based channel estimation scheme with parameter M, we can also say that we have a "rate". This rate is the number of information bits transmitted over the number of total transmitted bits (i.e., information bits plus channel estimation bits). For a comb-like channel estimation scheme where every M-th subcarrier is used for channel estimation, the rate is $R = 1 - \frac{1}{M}$.

Your Task

When using the comb-like channel estimation scheme which you implemented in Task 1 and plotting the resulting BER versus Eb/N0, the LTAs observed the behaviour shown in Fig. 1. More precisely, it seems that when moving from $M = 2$ to $M = 4$, there is an improvement in performance (i.e., the BER decreases). When moving from $M = 4$ to $M = 8$, performance is virtually unchanged. However, when moving from $M = 8$ to $M = 16$, performance starts getting worse, and for $M = 32$ we have the same performance as for $M = 2$. Please help the LTAs by providing a qualitative explanation of this behavior.

Task 3

The LTAs do not seem to understand your perfectly valid explanation of Task 2. In order to help them understand the behavior better, they would like to see the curves of Fig. 1 under a different light.

Your Task

Help the LTAs by re-drawing Fig. 1 with SNR instead of Eb/N0 on the horizontal axis. A qualitative plot which demonstrates the relation between the curves is sufficient.

Part 2: DSSS and carrier frequency offset (CFO)

Task 1

In one of the previous exercises you have implemented a RAKE receiver to receive a DSSS modulated signal over a multipath channel. While your receiver works well in your MATLAB simulation it completely fails if you use it for radio transmissions.
After some thinking you notice that no one has thought of the offset between the frequency of the oscillator in the transmitter and in the receiver. You need to compensate for this effect in the receiver.

Modulation	BPSK
Sampling frequency	40 MHz
Carrier Frequency	2.5GHz
Max combined oscillator offset	20ppm

Table 2: DSSS Simulation Framework Parameters

Used frame structure:

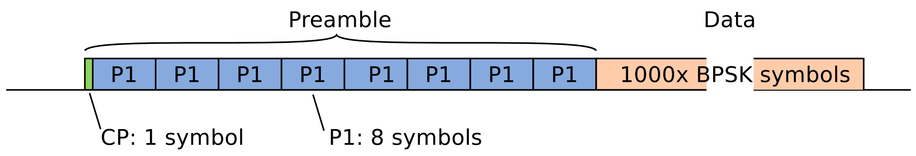

Figure 2: DSSS transmission frame format.

Your Task

Download the MATLAB DSSS simulation framework `simulator.m` (which now also simulates the CFO) and the parameter file `paramcfo.m` from the moodle. The `paramcfo` file defines all necessary parameters for your transmission. (There should be no need to change anything in this file.)
Implement in the `simulator.m` a carrier frequency offset (CFO) estimator which takes advantage of the frame structure. Use this estimation to correct the frequency offset before you decode the data. Plot a comparison of the performance of the system with and without your CFO compensation .
(In this task we neglect synchronization and channel estimation.)

Task 2

Now that your receiver can compensate a frequency offset, you think about saving money by buying cheaper crystals. Cheaper crystals are usually less precise.

Your Task

In order to choose the cheapest one that still works, derive theoretically the maximal oscillator offset your algorithm can deal with (in ppm per oscillator)?
Hint: Remember that we use DSSS.

Hand In Instructions

Please submit all files of your solution online in the course moodle until 4PM. The theory questions can bei either solved (and handed in) on paper or as short textfiles/MATLAB scripts. **Good luck!**