Matlab Simulation of an Orthogonal Frequency-Division Multiplexing (OFDM) System

1 Introduction

This project constitutes the second component of EE810–Communication Theory I. In this project, you will write Matlab scripts to simulate an OFDM system that includes the transmitter, the multipath channel, and the receiver. Whenever feasible, the OFDM system parameters shall be specified as in *upstream* transmission of DOCSIS 3.1 standard. As such, familiarity with DOCSIS 3.1 Physical Layer Specification is expected (available for download at http://www.cablelabs.com/specification/physical-layer-specification).

The project has two parts. In the first part, perfect timing and frequency synchronization is assumed and the focus is on the detection performance of an OFDM system. In the second part, the effects of timing error and frequency offset on the system performance are evaluated. Then two timing and frequency acquisition algorithms are investigated and incorporated in the simulation of the OFDM receiver. The focus of the second part is on the acquisition (estimation) performance of the two investigated algorithms.

2 Part I

Your main Matlab program should look something like this:

```
%% System parameters

N_FFT = 2048; % FFT size in DOCSIS3.1 upstream is 2048 or 4096
N_CP = 96; % cyclic prefix length, either 96,128,160,192,224,256,288,320,384,512,640
N_RP = 0; % number of roll-off samples, either 0,32,64,96,128,160,192,224
Mod_type = 'QPSK';
Sub_chan = [20:230]; % sub-channel mapping
L_CH = 8; % number of channel taps
SNR = 25; % signal-to-noise ratio, in dB
L_bst = 10; % number of OFDM data blocks in one frame
N_frm = 1000; % number of OFDM frames to be simulated

%% Transmiter
[tx,b_orig]=OFDM_Tx(N_FFT,N_CP,N_RP,Sub_chan,Mod_type,L_bst);

%% Channel
[rx,imp_ch]=Channel(tx,L_CH,SNR);

%% Receiver
[b_dec]=OFDM_Rx(rx,N_FFT,N_CP,N_RP,Sub_chan,Mod_type,L_bst,imp_ch,SNR);

% Count errors and plot BER ...
```

2.1 Description of Parameters

The main program call 3 functions that simulate the transmitter (`OFDM_Tx`), the channel (`Channel`) and the receiver (`OFDM_Rx`). The following explains the input and output parameters/variables of these functions.

Transmitter:

- `N_FFT`: This is the size of FFT/IFFT. In DOCSIS3.1 upstream, this value is 2048 and 4096 for the 2-k and 4-k modes, respectively. Note that the IDFT equation specified in the Standard (page 59) is slightly different than the standard IDFT equation given in (4) of the Lecture Notes. You are asked to used the same IDFT/DFT equations as in the Standard. That is:

$$x[n] \triangleq \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} X[k]\mathrm{e}^{j\frac{2\pi n(k-N/2)}{N}}, \quad 0 \le n \le N-1, \tag{1}$$

$$X[k] \triangleq \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-j\frac{2\pi n(k-N/2)}{N}}, \quad 0 \le k \le N-1. \tag{2}$$

  It is pointed out that the above equations can also be efficiently computed by Matlab IFFT and FFT commands. First, the FFT and IFFT in Matlab are defined exactly the same as that in the Lecture Notes, namely:

$$X[k] = \mathrm{FFT}\{x[n]\} = \sum_{n=0}^{N-1} x[n]\mathrm{e}^{-j\frac{2\pi nk}{N}}, \quad 0 \le k \le N-1, \tag{3}$$

$$x[n] = \mathrm{IFFT}\{X[k]\} = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\mathrm{e}^{j\frac{2\pi nk}{N}}, \quad 0 \le n \le N-1. \tag{4}$$

  It follows that (1) and (2) can be computed with the above "standard" IFFT/FFT as:

$$x[n] = \sqrt{N}\cdot \mathrm{IFFT}\{X[k]\}\cdot \mathrm{e}^{-j\pi n}, \quad 0 \le n \le N-1,$$

$$X[k] = \frac{1}{\sqrt{N}}\cdot \mathrm{FFT}\{x[n]\cdot \mathrm{e}^{j\pi n}\}, \quad 0 \le n \le N-1.$$

- `N_CP`: This is the length of cyclic prefix (CP), measured in the number of time samples. The use of cyclic prefix to overcome the effect of ISI is explained in the Lecture Notes. For upstream transmission, there are 11 possible values for the CP length (see Table 7-2 on pages 38–39). The choice depends on the delay spread of the channel and it is required that `N_CP` $\geq$ `L_CH`, where `L_CH` is the length of the equivalent discrete-time channel (explained later).

- `N_RP`: In addition to CP extension, *windowing* is also applied to the time samples before transmission. The purpose of windowing is to maximize channel capacity by sharpening the edges of the spectrum of the OFDMA signal. This windowing operation is specific to DOCSIS 3.1 standard and not explained in the Lecture Notes. You should review Section 7.4.11 of the standard carefully to understand how it is done.

It is pointed out that there is a typo in the equation of the windowing function on page 64. The correct expression for the right half of the window is:

$$w\left(\frac{N+N_{CP}+N_{RP}}{2}+i\right)= \tag{5}$$
$$\begin{cases} 1.0, & i=0,1,\ldots,\frac{N+N_{CP}-N_{RP}}{2}-1 \\ \frac{1}{2}\left[1-\sin\left(\frac{\pi}{N_{RP}}\left(i-\frac{N+N_{CP}}{2}+\frac{1}{2}\right)\right)\right], & i=\frac{N+N_{CP}-N_{RP}}{2},\ldots,\frac{N+N_{CP}+N_{RP}}{2}-1. \end{cases}$$

Also the roll-off parameter α of the windowing function is defined as $\alpha=\frac{N_{RP}}{N+N_{CP}}$. Note that this parameter is frequently referred to in many places of the Standard document.

- `Mod_type`: Type of QAM-modulation, raging from BPSK to 16,384-QAM (see Table A-1 on page 197).
- `Sub_chan`: This is a row vector that specifies the sub-channels to be used. For example, if `N_FFT = 2048`, then `Sub_chan = [20:230]` means that among 2048 sub-channels, only 211 contiguous sub-channels, from 20 to 230, are used.
- `L_bst`: The number of OFDM blocks over which the channel coefficients stay the same (i.e., the coherence time of the channel).
- `N_Sim`: The total number of OFDM blocks to be simulated.
- `b_orig`: Sequence of randomly-generated information bits.
- `tx`: Sequence of time-domain OFDM samples to be transmitted.

<u>Channel</u>:

- `L_CH`: This specifies the length of the equivalent discrete-time multipath channel. Referring to Fig. 6 of Lecture Notes, the value of this parameter is the same as $\mu+1$. Moreover, in this part of the project, the channel is modeled and simulated as multipath *Rayleigh* fading.[1] Specifically, the channel coefficients $h[n]$, $n=0,\ldots,\mu$, are generated as independent complex Gaussian random variables. The real and imaginary parts of $h[n]$ are zero-mean i.i.d. real Gaussian random variables with variance $\frac{\sigma_n^2}{2}$, where the variance of the nth tap has an exponential taper (also known as *channel delay profile*) given by:

$$E\{|h[n]|^2\}=\sigma_n^2=\beta \mathrm{e}^{-n/(\mu+1)}, \quad 0\le n\le\mu \tag{6}$$

  where β is a scaling factor such that $\sum_{n=0}^{\mu}\sigma_n^2=1$.

- `imp_ch`: This vectors contains the channel coefficients as simulated above. The information of this vector is made available at the OFDM receiver for detection (i.e., perfect channel estimation is assumed).

[1] Although multipath Rayleigh fading is a common channel model in mobile wireless communications, it is not applicable for DOCSIS channel. Channel modeling for DOCSIS will be explained at a later date.

- `SNR`: This signal-to-noise parameter sets the variance of complex AWGN component at the output of the equivalent discrete-time channel. Specifically, let $w[n]$ denote the noise component at the output of the channel illustrated in Fig. 6 of the Lecture Notes. Then one has

$$y[n] \quad = \quad \tilde{x}[n] * h[n] + w[n], \tag{7}$$

where $w[n]$ is modeled as a zero-mean complex Gaussian random variable, whose real and imaginary components have variance $\sigma_w^2/2$ each. The SNR is defined as[2]

$$\text{SNR} = 10 \log_{10} \frac{E\{|X[n]|^2\}}{\sigma_w^2} \quad \text{(dB)}, \tag{8}$$

where $E\{|X[n]|^2\}$ is the average symbol energy of the employed QAM constellation.

<u>Receiver</u>:

- `b_dec`: This vector contains the detected bits. It can be compared to `b_orig` to determines the BER.

2.2 Requirements

You should make your Matlab scripts as general as possible, i.e., they should work with any specific parameters as specified in DOCSIS 3.1. To verify that your Matlab scripts work properly, in addition to submitting your Matlab scripts via email, please obtain and submit a figure that plots the BER curves over the SNR range of [0 : 2 : 24] dB and for the following modulation formats: BPSK, QPSK, 8-QAM and 16-QAM (you need to follow page 187 in Appendix A of the standard). The parameters used for obtaining such a figure are as follows:

```
N_FFT = 256; % use this FFT size to speed up the simulation
N_CP = 96;
N_RP = 32;
Sub_chan = [20:230];
L_CH = 8;
L_bst = 1;
N_Sim = 1000;
```

Remarks:

- Interaction with other fellow students on the concepts and expectations of this project are encouraged but you should write your own Matlab scripts and submit them individually. You might want to get together as a group to see if there are any common questions/concerns and then schedule a meeting with me to answer those questions.
- **The due date for Part I is January 24, at 3:00PM**.
- When emailing your Matlab files, please compress them in one folder and name it as `EE810_YourFirstName_YourLastName.zip` (`rar` format is fine as well).

[2]This is only one definition of SNR and it is chosen to facilitate performance comparison in Part II with results reported in [1].

3 Part II

In Section I you simulated an "ideal" OFDM system where no frequency and timing errors are present. This Section examines the important issues of timing and frequency errors in a more practical OFDM system. The effects of timing and frequency errors are first discussed. Then techniques to perform coarse timing and frequency estimations are introduced. The material in this section follows closely the excellent tutorial paper by Morelli et. al [1].

3.1 Theory

In keeping with the notations in [1], define the length of the equivalent complex baseband channel to be L and the length of the cyclic prefix to be N_g. The length-L vector of the channel impulse response (CIR) is denoted as $\mathbf{h} = (h[0], h[1], \dots h[L-1])^\top$. The channel frequency response over the kth subcarrier is defined in [1] as $H[k] = \sum_{l=0}^{L-1} h[l]\mathrm{e}^{-j\frac{2\pi lk}{N}}$. With the modified DFT/IDFT definitions used in DOCSIS 3.1, $H[k]$ can be redefined as

$$H[k] = \sum_{l=0}^{L-1} h[l]\mathrm{e}^{-j\frac{2\pi l(k-N/2)}{N}}, \quad 0 \le k \le N-1. \tag{9}$$

It is simple to show that, with the above definition, $E\{|H[k]|^2\} = \sum_{l=0}^{L-1} E\{|h[l]|^2\}, \forall\, k$.

In addition, the discussion in this section does not include the time-windowing operation specified in DOCSIS 3.1. The phrase "OFDM block" refers to a group of $N + N_g$ time-domain samples after the IFFT and CP extension. To avoid interference between adjacent transmitted OFDM blocks, called inter-block interference (IBI), one needs to ensure that $N_g \ge L-1$.

The ith OFDM block after CP extension has length $N_T = N + N_g$ and can be represented as:

$$\tilde{\mathbf{x}}_i = (\tilde{x}_i[-N_g - iN_T], \tilde{x}_i[-N_g + 1 - iN_T], \dots, \tilde{x}_i[N-1+iN_T])^\top, \quad i = \dots, -1, 0, 1, \dots \tag{10}$$

Considering the transmission of multiple OFDM blocks, the channel's output for a perfectly-synchronized OFDM system is written as:

$$y[n] = \sum_i \sum_{k=0}^{L-1} h[k]\tilde{x}_i[n-k-iN_T] + w[n] \tag{11}$$

In practical situations oscillator instabilities results in a frequency offset, f_d, between the received carrier and the local sinusoids used for signal demodulation. In addition, at the start-up the receiver does not know where the OFDM blocks start and, accordingly, the DFT window may be placed in a wrong position. This results in a *timing error*, denoted by τ_d, which must be properly compensated to avoid severe performance degradation. Since small (fractional) timing errors can be corrected through channel equalization, it suffices to locate the beginning of each received OFDM block within one sampling period. For this reason, it is a common practice to model the timing error as an integer multiple of the sampling period, denoted by θ. The remaining fractional timing error is considered as part of the CIR.

In summary, let $\epsilon = N f_d T_s$ be the frequency offset normalized to the subcarrier spacing of $1/(NT_s)$, the received samples in the presence of synchronization errors has the following form:

$$y[n] = \mathrm{e}^{j2\pi\epsilon n/N} \sum_i \sum_{k=0}^{L-1} h[k]\tilde{x}_i[n-\theta-k-iN_T] + w[n] \tag{12}$$

Fig. 1 shows the block diagram of a practical OFDM receiver. The coarse frequency and timing estimation units employ the received sequence $y[n]$ to compute estimates of ϵ and θ, denoted by $\hat{\epsilon}$ and $\hat{\theta}$, respectively. The estimate $\hat{\epsilon}$ is used to counter-rotate $y[n]$ at an angular speed $2\pi\hat{\epsilon}/N$ (coarse frequency correction), while the time estimate $\hat{\theta}$ is exploited to achieve the correct positioning of the receive DFT window (coarse timing correction). Specifically, the counter-rotated samples with indexes $iN_T + \hat{\theta} \leq n \leq iN_T + \hat{\theta} + N - 1$ are grouped together and passed to the DFT unit.

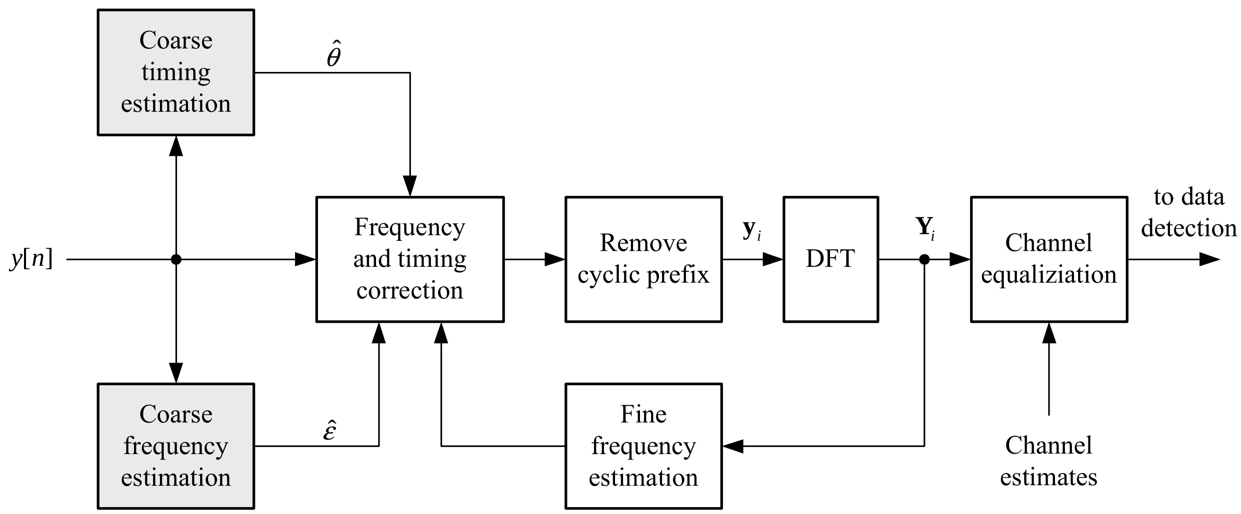

Figure 1: Block diagram of an OFDM receiver with timing and frequency corrections.

3.1.1 Sensitivity to Timing and Frequency Errors

Effect of Timing Offset

To avoid IBI, the DFT window should include samples from only one single block. As shown in Fig. 2, the tail of each received OFDM block extends over the first $L-1$ samples of the successive block as a consequence of multipath dispersion. Since the CP length must be greater than the CIR duration in a well designed system, a certain range of the guard interval is not affected by the previous block at the receiver. As long as the DFT window starts anywhere in this range, no IBI is present at the DFT output. This situation occurs whenever the timing error $\Delta\theta = \hat{\theta} - \theta$ belongs to interval $-N_g + L - 1 \leq \Delta\theta \leq 0$. Such a timing error only results in a cyclic shift of the received OFDM block. Specifically, applying the time-shift property of the Fourier transform and assuming perfect frequency synchronization (i.e., $\epsilon = 0$), the DFT output over the kth subcarrier takes the form

$$Y_i[k] = \exp(j2\pi k\Delta\theta/N) H[k] X_i[k] + W_i[k], \quad 0 \leq k \leq N-1 \tag{13}$$

The above equation indicates that timing error $\Delta\theta$ appears as a linear phase shift across subcarriers. This phase shift cannot be distinguished from the phase shift introduced by the channel. As such, the effect can be compensated for by the channel equalizer.

On the other hand, if the timing error is outside interval $-N_g + L - 1 \leq \Delta\theta \leq 0$, samples at the DFT input will be contributed by two adjacent OFDM blocks. In addition to IBI, this results in a loss of orthogonality among subcarriers which, in turn, generates inter-carrier interference (ICI). In this case, the kth DFT output is given by

$$Y_i[k] = \exp(j2\pi k\Delta\theta/N)\alpha(\Delta\theta)H[k]X_i[k] + I_i(k,\Delta\theta) + W_i[k], \quad 0 \leq k \leq N-1, \tag{14}$$

where $\alpha(\Delta\theta)$ is an attenuation factor, while $I_i(k,\Delta\theta)$ accounts for IBI and ICI. This term and can reasonably be modeled as a zero-mean random variable with power $\sigma_I^2(\Delta\theta)$. Both $\alpha(\Delta\theta)$ and $\sigma_I^2(\Delta\theta)$ depend on the timing error and the channel delay profile as discussed in [2].

A useful indicator to evaluate the effect of timing errors on the system performance is the loss in the SNR. This quantity is defined as

$$\gamma(\Delta\theta) = \frac{\text{SNR}^{(\text{ideal})}}{\text{SNR}^{(\text{real})}} \tag{15}$$

Here, $\text{SNR}^{(\text{ideal})}$ is the SNR of a perfectly synchronized system as defined in (8), namely $\text{SNR}^{(\text{ideal})} = E\{|X[n]|^2\}/\sigma_w^2$. On the other hand, $\text{SNR}^{(\text{real})}$ is the SNR in the presence of a timing offset. Since the three terms in the right-hand-side of (14) are statistically independent, it follows that $\text{SNR}^{(\text{real})} = E\{|X[n]|^2\}\alpha^2(\Delta\theta)/[\sigma_w^2 + \sigma_I^2(\Delta\theta)]$. Substituting these results into (15) yields

$$\gamma(\Delta\theta) = \frac{1}{\alpha^2(\Delta\theta)}\left[1 + \frac{\sigma_I^2(\Delta\theta)}{\sigma_w^2}\right] \tag{16}$$

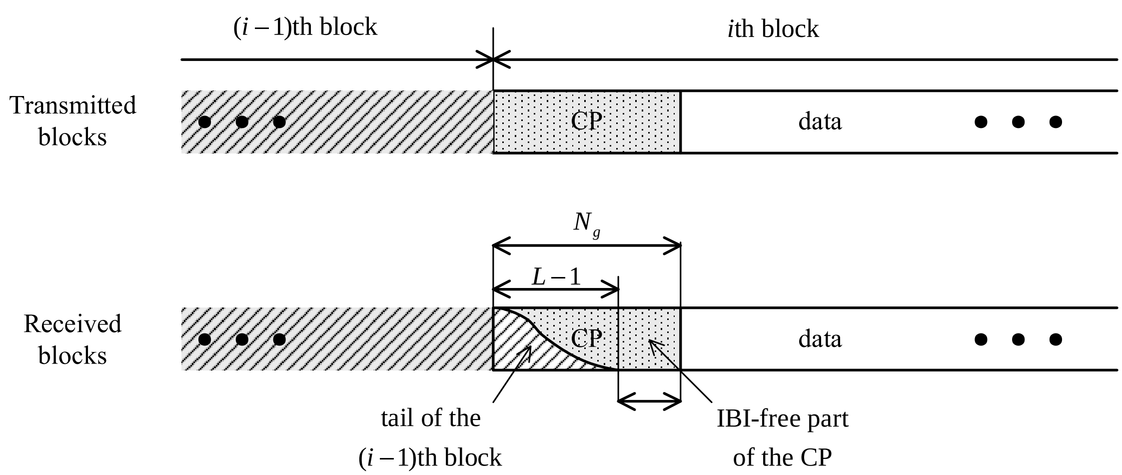

Figure 2: Partial overlapping between received blocks due to multipath dispersion.

Effect of Frequency Offset

A carrier frequency offset produces a shift of the received signal in the frequency domain and may result in a loss of orthogonality among subcarriers. To better explain this problem, let's assume ideal timing synchronization (i.e., $\hat{\theta} = \theta$) and compute the DFT output corresponding to the ith OFDM block in the presence of a frequency error ϵ. Such a computation is given in [3] and yields

$$Y_i[k] = \mathrm{e}^{j\varphi_i} \sum_{p=0}^{N-1} H[k]X_i[k]f_N(\epsilon + p - k) + W_i[k] \tag{17}$$

where $\varphi_i = 2\pi i\epsilon N_T/N$ and $f_N(x)$ is defined as

$$f_N(x) = \frac{\sin(\pi x)}{N\sin(\pi x/N)}\mathrm{e}^{j\pi x(N-1)/N}. \tag{18}$$

Next, it is convenient to distinguish between two distinct situations, depending on whether the frequency error is an integer multiple of the subcarrier spacing $1/NT_s$. In the first case, ϵ is an integer value and (17) reduces to

$$Y_i[k] = \mathrm{e}^{j\varphi_i}H(|k-\epsilon|_N)X_i(|k-\epsilon|_N) + W_i[k] \tag{19}$$

where $|k-\epsilon|_N$ is the value of $k-\epsilon$ reduced to interval $[0, N-1]$. The above equation indicates that an integer frequency offset only results in a shift of subcarriers by ϵ positions. Orthogonality among subcarriers is thus still preserved, even though the received symbols appear at the wrong positions at the DFT output.

The situation is drastically different when ϵ is not an integer. In this case, the subcarriers are no longer orthogonal and (17) can be conveniently rewritten as

$$Y_i[k] = \mathrm{e}^{j\varphi_i}H[k]X_i[k]f_N(\epsilon) + I_i(k,\epsilon) + W_i[k] \tag{20}$$

where $I_i(k,\epsilon)$ is a zero-mean ICI term with power $\sigma_I^2(\epsilon) = E\{|I_i(k,\epsilon)|^2\}$. Subsituting $E\{|H[k]|^2\} = 1$, and after some manipulations one can show that

$$\sigma_I^2(\epsilon) = E\{|X[n]|^2\}[1-|f_N(\epsilon)|^2]. \tag{21}$$

Like the impact of timing error, the impact of the frequency error on the system performance can be assessed in terms of the SNR loss:

$$\gamma(\epsilon) = \frac{\mathrm{SNR}^{(\mathrm{ideal})}}{\mathrm{SNR}^{(\mathrm{real})}}. \tag{22}$$

Here, as before, $\mathrm{SNR}^{(\mathrm{ideal})}$ is the SNR of a perfectly-synchronized system, whereas $\mathrm{SNR}^{(\mathrm{real})} = E\{|X[n]|^2\}|f_N(\epsilon)|^2/[\sigma_w^2+\sigma_I^2(\epsilon)]$ is the SNR in the presence of frequency offset ϵ. Substituting these results into (22) and making use of (21), one obtains

$$\gamma(\epsilon) = \frac{1}{|f_N(\epsilon)|^2}\left\{1 + \frac{E\{|X[n]|^2\}}{\sigma_w^2}[1-|f_N(\epsilon)|^2]\right\}. \tag{23}$$

A simpler expression of $\gamma(\epsilon)$ can be obtained for small values of ϵ by using the Taylor series expansion of $|f_N(\epsilon)|^2$ around $\epsilon = 0$. This produces

$$\gamma(\epsilon) \approx 1 + \frac{1}{3}\frac{E\{|X[n]|^2\}}{\sigma_w^2}(\pi\epsilon)^2. \tag{24}$$

The above equation indicates that the SNR loss is related to the square of the normalized frequency offset.

Equation (23) is plotted in Fig. 3 as a function of ϵ for some values of $\text{SNR}^{(\text{ideal})} = E\{|X[n]|^2\}/\sigma_w^2$. These results indicate that non-negligible performance degradations are incurred when the frequency error exceeds 4%–5% of the subcarrier distance.

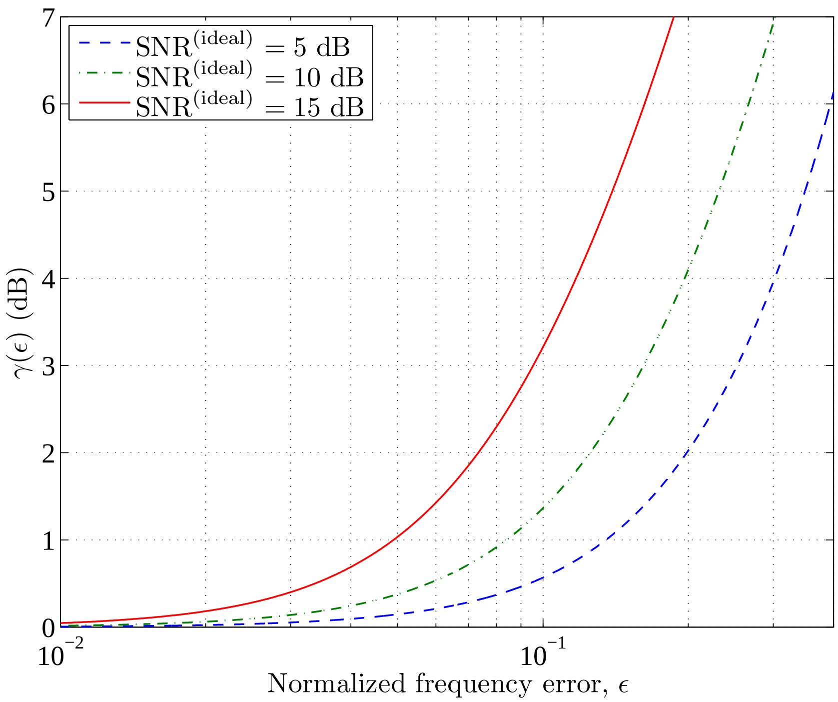

Figure 3: SNR loss due to frequency errors.

3.1.2 Synchronization Algorithms

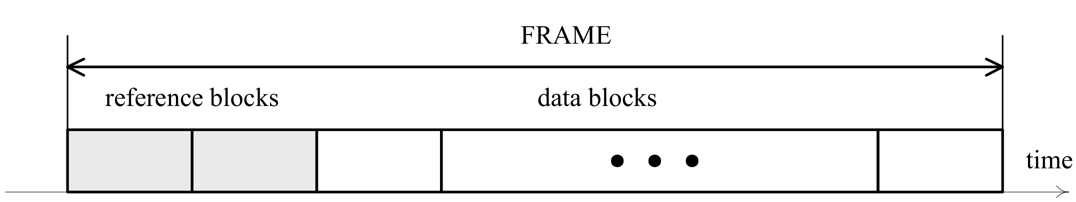

Figure 4: An example of OFDM frame structure.

In OFDM systems, the transmission is normally organized in frames. An example of the frame structure is illustrated in Fig. 4, where some reference blocks with a particular training pattern are appended in front of the payload segment to assist the synchronization process.

This section introduces some popular schemes for timing and frequency estimation. In doing so, it is important to distinguish between an *acquisition step* and a *tracking phase*. During the acquisition, the reference blocks placed at the beginning of the frame are employed to get coarse estimates of the synchronization parameters. These estimates are next refined during the tracking phase to counteract short-term variations produced by oscillator drifts and/or time-varying Doppler shifts.

Timing Acquisition

In most OFDM applications, timing acquisition represents the first step of the synchronization process. This operation has two main objectives. First, it must detect the presence of a new frame in the received data stream. Second, once the frame has been detected, it must provide a coarse estimate of the timing error to find the correct position of the receive DFT window. Since the CFO is usually unknown at this stage, it is desirable that the timing recovery scheme be robust against possibly large frequency offsets.

A popular approach to coarse timing estimate is to use some reference blocks exhibiting a repetitive structure in the time domain. In this case, a robust timing estimator can be designed by searching for the peak of the correlation among repetitive parts. This idea was originally proposed by Schmidl and Cox (S&C) in [4], where a reference block composed by two identical halves of length $N/2$ is transmitted at the beginning of each frame. Note that a block with such a structure can easily be generated in the frequency domain by modulating subcarriers with even indexes by a pseudo-noise (PN) sequence while forcing zero to the remaining subcarriers with odd indexes.

To explain the rationale behind the S&C algorithm, one can verify that, as long as the CP is longer than the CIR duration (i.e., $N_g > L$), the two halves of the reference block will remain identical after passing through the transmission channel, except for a phase shift induced by the CFO. In other words, if we model the received samples corresponding to the first half as

$$y[n] = s^{(R)}[n]\mathrm{e}^{j2\pi\epsilon n/N} + w[n], \quad \theta \le n \le \theta + N/2 - 1 \tag{25}$$

where $s^{(R)}[n]$ denotes the useful signal and $w[n]$ the noise component, then the samples in the second half take the following form:

$$y[n+N/2] = s^{(R)}[n]\mathrm{e}^{j2\pi\epsilon n/N}\mathrm{e}^{j\pi\epsilon} + w[n+N/2], \quad \theta \le n \le \theta + N/2 - 1. \tag{26}$$

In this case, timing acquisition can be performed by feeding the time-domain samples to a sliding-window correlator of lag $N/2$, which is expected to exhibit a peak when the sliding window is perfectly aligned with the received reference block. Thus, the timing estimate can be obtained as [4]

$$\hat{\theta} = \underset{\tilde{\theta}}{\mathrm{argmax}}\left\{|\Gamma(\tilde{\theta})|\right\}, \tag{27}$$

where $\Gamma(\tilde{\theta})$ is the following normalized autocorrelation function:

$$\Gamma(\tilde{\theta}) = \frac{\sum_{q=\tilde{\theta}}^{\tilde{\theta}+N/2-1} y[q+N/2]y^*[q]}{\sum_{q=\tilde{\theta}}^{\tilde{\theta}+N/2-1} |y[q+N/2]|^2}. \tag{28}$$

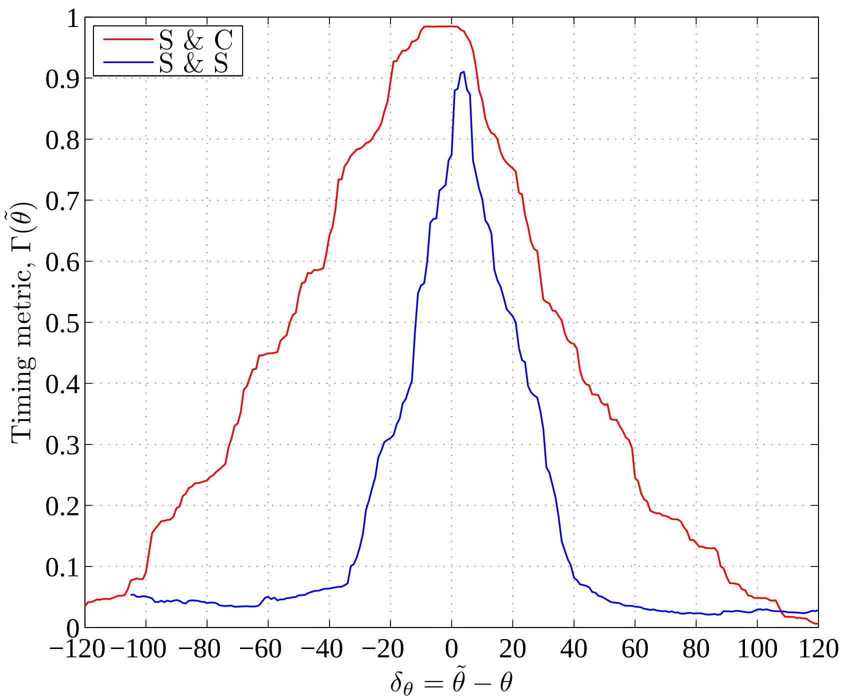

Figure 5: Example of timing metrics for S&C and S&S algorithms.

Fig. 5 shows an example of the timing metric (the red curve), $|\Gamma(\tilde{\theta})|$, as a function of the difference $\delta_\theta = \tilde{\theta} - \theta$. The results are obtained numerically over a Rayleigh multipath channel with $L = 8$ taps. The number of subcarriers is $N = 256$ and the CP has length $N_g = 16$. The SNR over received samples is defined as $\text{SNR} = \sigma_s^2/\sigma_w^2$ with $\sigma_s^2 = E\{|s^{(R)}(k)|^2\}$ and is set to 20 dB.

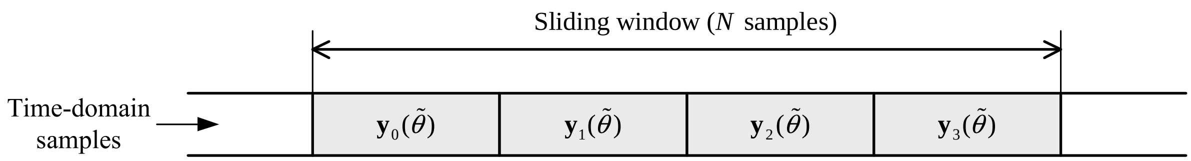

Figure 6: Sliding window used in S&S timing acquisition scheme.

Unfortunately, it is seen from Fig. 5 that the timing metric of the S&C algorithm exhibits a large "plateau" that may greatly reduce the estimation accuracy. Solutions to this problem were proposed in [5, 6], where reference blocks with suitably-designed patterns are exploited to obtain sharper timing metric trajectories [5, 6]. For example, Shi and Serpedin (S&S) used a training block composed of four repetitive parts $[+B, +B, -B, +B]$ with a sign inversion in the third segment [6]. As depicted in Fig. 6, a sliding window of length N spans received time-domain samples with indexes $\tilde{\theta} \leq n \leq \tilde{\theta} + N - 1$ and collects them into four vectors $\mathbf{y}_j(\tilde{\theta}) = \{y[l + jN/4 + \tilde{\theta}]; 0 \leq l \leq N/4 - 1\}$ with $j = 0, 1, 2, 3$. The timing metric is then computed as

$$\Gamma(\tilde{\theta}) = \frac{|\Lambda_1(\tilde{\theta})| + |\Lambda_2(\tilde{\theta})| + |\Lambda_3(\tilde{\theta})|}{\frac{3}{2}\sum_{j=0}^{3} \|\mathbf{y}_j(\tilde{\theta})\|^2} \tag{29}$$

where

$$\begin{aligned}\Lambda_1(\tilde{\theta}) &= \mathbf{y}_0^H(\tilde{\theta})\mathbf{y}_1(\tilde{\theta}) - \mathbf{y}_1^H(\tilde{\theta})\mathbf{y}_2(\tilde{\theta}) - \mathbf{y}_2^H(\tilde{\theta})\mathbf{y}_3(\tilde{\theta}) \\ \Lambda_2(\tilde{\theta}) &= \mathbf{y}_1^H(\tilde{\theta})\mathbf{y}_3(\tilde{\theta}) - \mathbf{y}_0^H(\tilde{\theta})\mathbf{y}_2(\tilde{\theta}) \\ \Lambda_3(\tilde{\theta}) &= \mathbf{y}_0^H(\tilde{\theta})\mathbf{y}_3(\tilde{\theta}).\end{aligned} \tag{30}$$

Fig. 5 also plots $\Gamma(\tilde{\theta})$ for the S&S algorithm in the same operating conditions. Since the plateau region present in the S&C metric is now significantly reduced, more accurate timing estimates are expected. As indicated in [5], reference blocks with more than four repetitive segments can be designed to further increase the sharpness of the timing trajectory.

Frequency Acquisition

After frame detection and timing acquisition, the OFDM receiver must compute a coarse frequency estimate to align its local oscillator to the received carrier frequency. This operation is referred to as frequency acquisition and is normally accomplished at each new received frame by exploiting the same reference blocks used for timing acquisition in addition to possibly other dedicated blocks.

One common approach is to employ a *training pattern* composed by some repetitive parts which remain identical after passing through the transmission channel except for a phase shift produced by the frequency error. The frequency error is then estimated by measuring the induced phase shift. This method was originally employed by Moose in [3], where the phase shift between two successive identical blocks is measured in the frequency domain at the DFT output. More precisely, assume that timing acquisition has already been achieved and let $Y_1[k]$ and $Y_2[k]$ be the kth DFT output corresponding to the two reference blocks. Then, one may write

$$Y_1[k] = S^{(R)}[k] + W_1[k] \tag{31}$$

$$Y_2[k] = S^{(R)}[k]\mathrm{e}^{j2\pi\epsilon N_T/N} + W_2[k] \tag{32}$$

where $S^{(R)}[k]$ is the signal component (the same over each block as long as the channel is static), while $W_1[k]$ and $W_2[k]$ are noise terms. The above equations indicate that an estimate of ϵ can be computed as

$$\hat{\epsilon} = \frac{1}{2\pi(N_T/N)}\arg\left\{\sum_{k=0}^{N-1} Y_2[k]Y_1^*[k]\right\}. \tag{33}$$

The main drawback of the above scheme is the relatively short acquisition range. Since the $\arg\{\cdot\}$ function returns values in the range $[-\pi,\pi)$, it is seen from (33) that $|\hat{\epsilon}| \leq N/(2N_T)$, which is less than one-half of the subcarrier spacing. A viable method to enlarge the acquisition range is proposed by Schmidl & Cox in [4]. They decompose the frequency error into a *fractional part*, less than $1/(NT_s)$ in magnitude, plus an integer part which is a multiple of $2/(NT_s)$. The normalized frequency error is thus rewritten as

$$\epsilon = \nu + 2\eta \tag{34}$$

where $\nu \in (-1,1]$ and η is an integer. The S&C estimator relies on the transmission of two suitably designed reference blocks as depicted in Fig. 7. The first block is the same as that used for timing acquisition. In particular, it is composed by two identical halves of length $N/2$ that are generated by modulating only subcarriers with even indices while setting the others to zero. The second block contains a differentially encoded pseudo-noise sequence, PN1, on even subcarriers and another pseudo-noise sequence, PN2, on odd subcarriers.

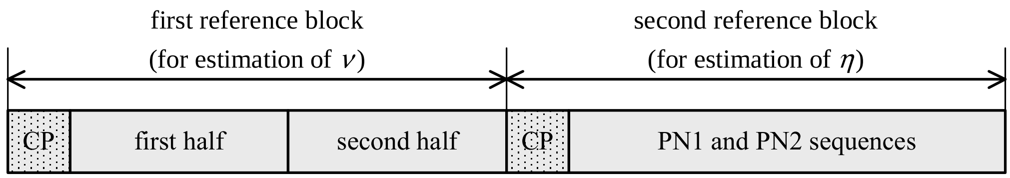

Figure 7: Reference blocks used by the S&C algorithm for frequency acquisition.

The samples in the two halves of the first block are obtained after substituting (34) into (25) and (26). This yields

$$\begin{aligned} y[n] &= s^{'}[n] + w[n], \quad \theta \le n \le \theta + N/2 - 1 && (35)\\ y[n+N/2] &= s^{'}[n]\mathrm{e}^{j\pi\nu} + w[n+N/2], \quad \theta \le n \le \theta + N/2 - 1 && (36)\end{aligned}$$

where, for notational simplicity, we define $s^{'}[n] = s^{(R)}[n]\mathrm{e}^{j2\pi(\nu+2\eta)n/N}$ and the identity $\mathrm{e}^{j2\pi\eta} = 1$ is also used in the derivation. The above equations indicate that, apart from thermal noise, the two halves are identical except for a phase shift of $\pi\nu$. Thus, an estimate of ν is obtained as

$$\hat{\nu} = \frac{1}{\pi}\arg\left\{\sum_{n=0}^{\theta+N/2-1} y[n+N/2]y^*[n]\right\}. \tag{37}$$

As shown above, timing information is necessary to compute $\hat{\nu}$. In practice, θ in (37) is replaced by its corresponding estimate $\hat{\theta}$ given in (27).

The next step is the estimation of the integer frequency offset η. For this purpose, the samples belonging to the two reference blocks of Fig. 7 are first counter-rotated at an angular speed $2\pi\hat{\nu}/N$ to compensate for the fractional offset ν. Next, they are fed to the DFT unit, which produces quantities $Y_1[k]$ and $Y_2[k]$ for $0 \le k \le N-1$. As explained before, once the fractional frequency offset ν is perfectly compensated, the DFT outputs are not affected by ICI. However, they will be shifted from their correct position by a quantity 2η due to the uncompensated integer frequency offset. In fact, from (19) one has

$$Y_1[k] = \mathrm{e}^{j\varphi_1}H(|k-2\eta|_N)X_1(|k-2\eta|_N) + W_1[k] \tag{38}$$

$$Y_2[k] = \mathrm{e}^{j\varphi_2}H(|k-2\eta|_N)X_2(|k-2\eta|_N) + W_2[k] \tag{39}$$

where $|k-2\eta|_N$ is the value $k-2\eta$ reduced to interval $[0, N-1]$. Neglecting for simplicity the noise terms and defining $P[k] = X_2[k]/X_1[k]$ to be the differentially-encoded PN sequence on the even subcarriers of the second block, it is seen from (38) and (39) that

$$Y_2[k] = \mathrm{e}^{j(\varphi_2-\varphi_1)}P[|k-2\eta|_N]Y_1[k], \quad \text{for even } k. \tag{40}$$

Therefore, an estimate of η can be calculated by finding integer $\hat{\eta}$ that maximizes the following metric:

$$B(\tilde{\eta}) = \frac{\sum\limits_{k \text{ even}} Y_2[k]Y_1^*[k]P^*[|k-2\tilde{\eta}|_N]}{\sum\limits_{k \text{ even}} |Y_2[k]|^2} \tag{41}$$

where $\tilde{\eta}$ varies over the range of possible integer frequency offsets. With (34), the estimated CFO is finally obtained in the form $\hat{\epsilon} = \hat{\nu} + 2\hat{\eta}$.

As mentioned before, the main advantage of the S&C method over the Moose scheme is the enlarged frequency acquisition range. An interesting question is whether a similar result can be obtained even with a smaller overhead than that required by S&C. A solution in this sense was proposed by Morelli and Mengali (M&M) in [7]. They consider a single reference block composed by $Q > 2$ identical parts, each containing N/Q samples. The estimated CFO is obtained as

$$\hat{\epsilon} = \frac{1}{2\pi/Q} \sum_{q=1}^{Q/2} \chi(q) \arg\{\Psi(q)\Psi^*(q-1)\} \tag{42}$$

where $\chi(q)$ are suitably designed coefficients, given by[3]

$$\chi(q) = \frac{12(Q-q)(Q-q+1) - 3Q^2}{2Q(Q^2-1)} \tag{43}$$

and $\Psi(q)$ is the following qN/Q-lag autocorrelation:

$$\Psi(q) = \sum_{n=\hat{\theta}}^{\hat{\theta}+N-1-qN/Q} y[n+qN/Q]y^*[n], \quad q = 1, 2, \ldots, Q/2. \tag{44}$$

As shown in [7], the estimation range of this scheme is $|\epsilon| \leq Q/2$. Hence, if Q is designed such that the possible frequency offsets lie in interval $[-Q/2, Q/2]$, the CFO is estimated by means of a single reference block so that the training overhead is reduced by a factor two with respect to the S&C method.

Fig. 8 compares S&C and M&M in terms of the mean squared error (MSE) values of frequency estimates versus SNR $= \sigma_s^2/\sigma_w^2$, where σ_w^2 is the noise power and $\sigma_s^2 = E\{|s^{(R)}[n]|^2\}$. These results were obtained with a total of $N = 256$ subcarriers and a Rayleigh multipath channel with $L = 8$ taps. Parameter Q in the M&M algorithm is fixed to 8. It is seen that the M&M method exhibits improved performance and achieves a gain of approximately 1.0 dB over the S&C method.

[3] There is a typo in Equation (40) of [1].

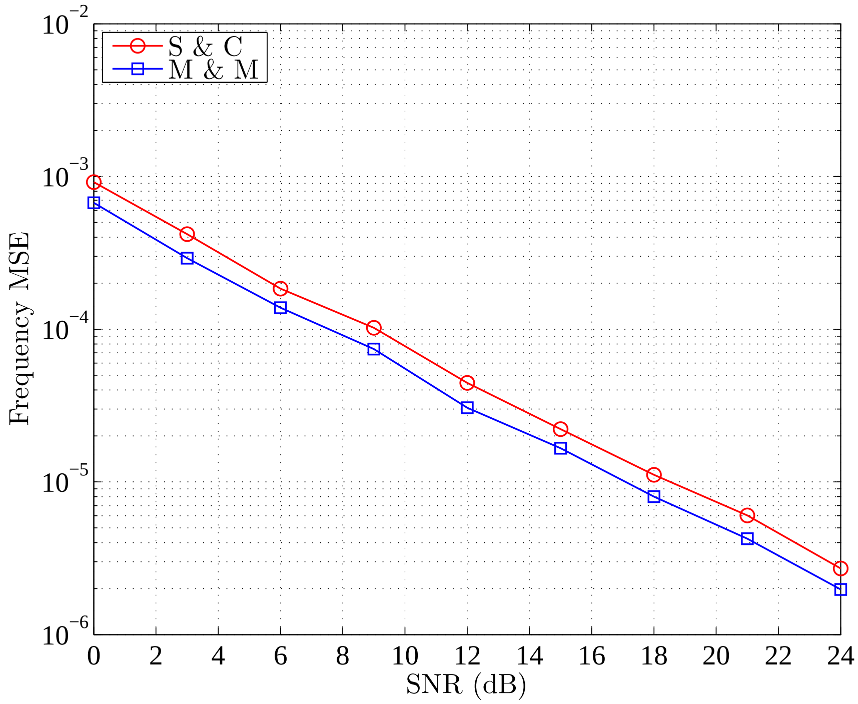

Figure 8: Performance of the S&C and M&M frequency estimation algorithms.

3.2 Matlab Simulation

You shall write Matlab simulation models to investigate coarse timing and frequency offset estimations separately. The template of your main Matlab program might look something like this (this template shall be given to you):

```
%% System parameters
N_FFT = 256; % FFT size
N_CP = 16; % cyclic prefix length
N_RP = 0; % number of roll-off samples
Mod_type = '16-QAM';
Sub_chan = [0:N_FFT-1]; % all sub-channels are used
L_CH = 8; % number of channel taps
SNR = 20; % signal-to-noise ratio, in dB
L_bst = 10; % number of OFDM data blocks in one frame
N_frm = 100; % number of OFDM frames to be simulated
N_Sub = length(Sub_chan); % number of allocated sub-channels.
                          % It should be an even number to work with S&C
                          % It should be a multiple of Q to work with M&M
Prbe_Name = 'S&C'; % use S&C and S&S for timing recovery
                   % use S&C and M&M for cfo estimate
MAP = zeros(N_FFT,N_Sub); % mapping matrix for allocated sub-channels
for ll = 1:N_Sub
    MAP(Sub_chan(ll)+1,ll) = 1 ;
end
```

```
%% Creating reference (preamble) block for coarse timing and fractional
%% frequency estimations
if Prbe_Name == 'S&C'
    S_ref = sqrt(N_FFT/2) * upsample([1+1j +1-1j],N_Sub/2);
    % reference symbols only use QPSK
    % sub-carriers mapping and IFFT to create time-domain reference block
    % with two identical halves in S&C
    s_ref = sqrt(N_FFT)*ifft(MAP * S_ref.').* exp(-j*pi*[0:N_FFT-1]).';
elseif Prbe_Name=='M&M'
    Q=8; M=N_Sub/Q;
    S_QPSK = 1/ sqrt(2) * [1+1j ,1-1j ,-1+1j ,-1-1j];
    S_ref = sqrt(Q)*upsample(S_QPSK(randi([1 4],M,1)),Q); % random QPSK symbols
    s_ref = sqrt(N_FFT)*ifft(MAP * S_ref.').* exp(-j*pi*[0:N_FFT-1]).';
elseif Prbe_Name == 'S&S'
    S_ref = sqrt(N_FFT/4) * upsample([1-1j -1-1j 1-1j -1+1j],N_Sub/4);
    % sub-carriers mapping and IFFT to create time-domain reference block
    % with patterns [+B,+B,-B,+B]
    s_ref = sqrt(N_FFT)*ifft(MAP * S_ref.').*exp(-j*pi*[0:N_FFT-1]).';
    s_ref(N_FFT/2+1:3*N_FFT/4) = -s_ref(N_FFT/2+1:3*N_FFT/4);
else
    error('Preamble type not supported');
end

%% Transmiter
[tx,b_orig] = OFDM_Tx2(N_FFT, N_CP, N_RP, Sub_chan, Mod_type, L_bst, s_ref);
% if the constellation is defined properly, tx should have average power
% very close to unity. If it is not the case, do the following normalization:
% tx=tx/sqrt(sum(abs(tx).^2)/length(tx));

%% Channel
% create random coarse timing offset
timing_offset = randi([N_RP N_FFT]);
perfect_time = timing_offset + N_CP+1;
% create random (fractional) frequency offset
cfo = randi([-299 300])/1000 / N_FFT; % frequency error
[rx, imp_ch] = Channel2(tx, L_CH, SNR, timing_offset, cfo);

%% Coarse timing estimate
[coarse_time_estimate, timing_metric]=CoarseTimeDetect2(rx,N_FFT,N_RP,N_CP,Prbe_Name);
timing_error = coarse_time_estimate - perfect_time

%% (Fractional) frequency offset estimate
cfo_estimate = FrequencyDetect2(rx, perfect_time, N_FFT, Prbe_Name);

frequency_error = cfo_estimate - cfo*N_FFT % frequency estimation error
```

Description of Matlab Parameters/Functions and Requirements

- `L_bst` is the number of OFDM data blocks in one frame. These blocks are preceded by a single reference (preamble) block, which is generated by the provided Matlab template according to different types: `S&C` and `S&S` for timing recovery, and `S&C` and `M&M` for frequency offset estimate. Only one reference block is generated for each OFDM frame since we only focus on coarse timing estimate and fractional frequency offset estimate. Also, as we are only concerned with synchronization in this part, `L_bst` can be set to a small number, say 2, to speed up the simulation.

- The function `OFDM_Tx2` is basically the same as what you wrote in Part I, except that you need to transmit the reference block at the beginning of each OFDM frame.

- The function `Channel2` simulates the effects of multipath fading, timing and (fractional) frequency offsets, and AWGN. This function (together with related Matlab files) shall be given to you. Note that in writing this function it is assumed that the transmitted OFDM samples are generated such that its average power is exactly one when no windowing is applied (i.e., when `N_RP=0`. As such, the input parameter `SNR` is sufficient to set the variance of AWGN.

- You are required to write functions `CoarseTimeDetect2` and `FrequencyDetect2`. Although both of these functions are included in the provided template, you can investigate them separately. If you investigate coarse timing estimation, then set `cfo=0`. On the other hand, if you investigate frequency offset estimation, set `timing_offset=0`.

- To demonstrate your function `CoarseTimeDetect2` works properly, please obtain plots of the timing metrics similar to that of Figure 5. The timing metrics would look different for different runs of the algorithm (one run for one OFDM frame) due to the randomness of the timing offset. Thus execute a few runs to obtain plots similar to Figure 5.

- To demonstrate your function `FrequencyDetect2` works properly, please obtain plots of the mean square error (MSE) of the frequency estimate similar to that of Figure 8. For each SNR value, you need to run your algorithm at least 1000 times to get a vector of 1000 values of `frequency_error` before you can find the MSE at that SNR value.

- Please email your Matlab files (in a compressed folder) and the two figures. The due date for this part is **February 28, 3:00PM**.

Bibliography

[1] M. Morelli, C.-C. Jay Kuo, and Man-On Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proc. IEEE*, vol. 95, no. 7, pp. 1394–1427, 2007.

[2] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broadband systems using OFDM. part i," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1668–1677, 1999.

[3] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2908–2914, 1994.

[4] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, 1997.

[5] H. Minn, V. Bhargava, and K. Letaief, "A robust timing and frequency synchronization for OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 822–839, 2003.

[6] K. Shi and E. Serpedin, "Coarse frame and carrier synchronization of OFDM systems: a new metric and comparison," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1271–1284, 2004.

[7] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," *IEEE Commun. Letters*, vol. 3, no. 3, pp. 75–77, 1999.