Description of ENTS 759B Project

Requirements

1- Documentation and explanations that describe all of your assumptions and explanations about the code, as well as what each code function or section represents in terms of one of the figures of this document.
2- MATLAB simulation with check points that are explained in your Documentation.
   i. Report should describe the output that you intend to demonstrate. The report should contain a screen shot of the final graph with an explanation of how you got it.
   ii. Your MATLAB program should clearly be commented and robust, especially the checkpoint sections.
   iii. Your MATLAB code should be also delivered in a form that can be edited and run by the instructor.

Deliverables

1. Documentation in soft copy (common text editors, MS Word, simple text file, latex, etc.). Documentation should be between 3 to 5 pages.
   a. All cited work must be properly cited, with references, or else the project will not be accepted and the grade will drop 20%. That is to say, if the instructor returns the project with instructions to properly reference or cite, then the grade will automatically be reduced 20%.
   b. Instructions for proper citation are given in the Scholarly Paper Guidelines in http://www.telecom.umd.edu/scholarly-paper
   c. Authors must follow the scholarly paper guidelines in terms of avoiding direct quotes from any source: http://www.telecom.umd.edu/scholarly-paper
2. Hand-signed "Understanding of Terms" form, to be slipped under the instructor's door any time before the due date.
3. Code m-files in a self-contained directory.
4. Collect m-files and documentation into a zipped file. The file should have the last name of the student, with the first initial. For example, Smith_J.zip
5. Email the project to the instructor no later than **WEDNESDAY, MAY 9, 2018 by 11AM.**
   a. Late submissions will receive a reduced grade: 10% per day (For example, submissions received Thursday, May 10th at 1AM will have maximum grade of 90%.). [1]
   b. Submission stamp is marked when the last part of the project is delivered (for example, a team who delivers all files 3 days early, but then resubmits one m-file one day late will count as if all files were delivered one day late).
   c. Authors should receive email confirmation from grader that the submission was received on time.

[1] *This restriction is necessary to be fair to students who delay their other responsibilities to hand in this project on time. Note: every member of the group will receive the same grade on this project, even if only one member was late and the others were not.*

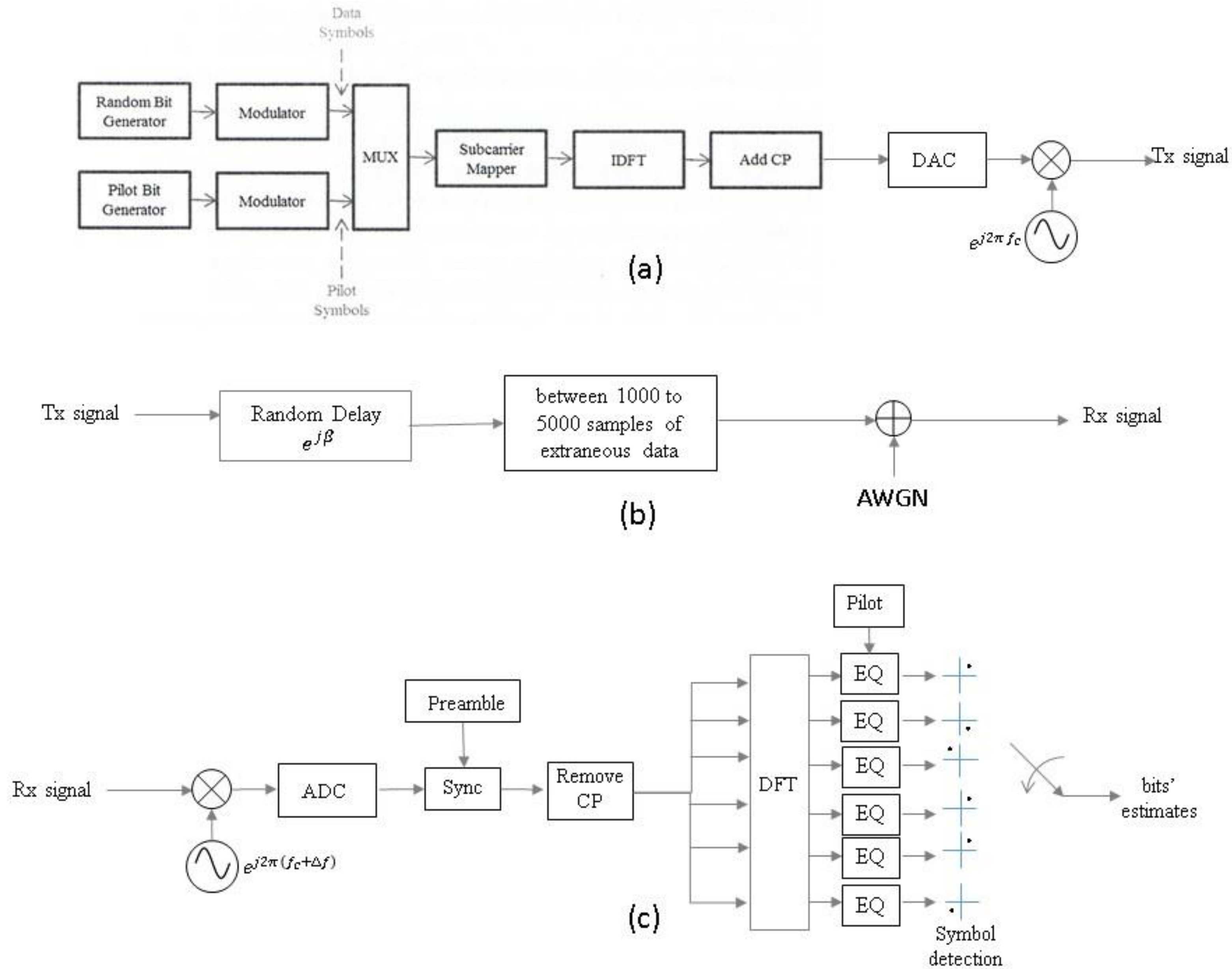

Figure 1. (a) Transmitter, (b) Channel, (c) Receiver.

Project description

Students are required to develop a simulation of an OFDM transmitter/receiver with synchronization, and a tap-delay line channel in Matlab. The simulation should include the transmitter, the channel, and the receiver, as depicted in Figure 1.

Some properties:

- Sampling frequency, $f_s = 3.84$ MHz
- Carrier frequency, $f_c = 3.4$ GHz
- Subcarrier spacing, $\Delta f = 15$ kHz
- Order of IFFT / FFT: N = 256 = OFDM symbol (without CP) number of samples per symbol
- Number of active subcarriers: 179 (note: the DC carrier is not used)
- Transmission bandwidth: BW=2.7MHz
- Cyclic prefix length: 18 samples (fixed)
- Data (information) bits are randomly generated
    - Payload is QPSK modulated (see Section 7.1.2 of [1])
- Preamble should follow the guidelines provided in [1] (Sections 5.7.2), and use the formula for Zadoff-Chu sequence

$$x_u(n),\ n = 0, \dots N_{ZC} - 1)$$

    - Use $N_{ZC} = 839$, and u=2 (note: these values were chosen to provide the maximum discernibility after you pass the Zadoff-Chu samples through the IFFT)
    - Use $C_v = N_{cs} = 0$
    - Only use the first N samples of the Zadoff-Chu sequence to feed through the IFFT
    - The samples are known at the receiver for the correlator
- Pilots are BPSK modulated (see Section 7.1.1 of [1]), with content defined in Sections 6.10 and 6.11 of [1]
    - Every 6th subcarrier is allocated **exclusively** for pilots, **except** for the preamble (See Figure 2)
    - Preamble will be used to detect the beginning of the desired data sequence at the receiver
    - Preamble will be used for correction of L.O. frequency offset at receiver
    - Pilots will also be used for equalization at receiver
- The receiver local oscillator will have a fixed offset (unknown to the receiver), Δf, resulting from a L.O. frequency drift between 7ppm to 10ppm.
- P.S.D. of the AWGN you model should be set so that the signal to noise ratio (SNR) at the receiver (for occupied bandwidth) $\in [15dB, 20dB]$

What you are allowed to use:

- xcorr
- conv

What you are *NOT* allowed to use:

- `lteZadoffChuSeq`
- Filter Visualization Tool
- RayleighChannel nor RicianChannel

The following checkpoints are required to be clearly demonstrated for full credit. The best way of demonstrating your project is through well described graphs. The format of the graphs must be chosen to best describe the data.

Points will be deducted for graphs that are not properly labeled (complete title, abscissa, ordinate, legend). For example, "Magnitude response" is not an acceptable title; "Magnitude of spectrum of signal after equalization" is an adequate title (note that proper grammar is not an issue, as long as the concept is clear).

The checkpoints are:

1. Transmitter chain (see Figure 1 (a)):
   a. Output of your modulator in both time and frequency domains. You should show the I&Q on the same graph with different colors (in other words, use the hold function). *Authors should also add a constellation plot.* You must use two different modulators, and provide plots for both sets:
      i. BPSK for pilots
      ii. QPSK for data
   b. Output of the IFFT for six subcarriers (plot all subcarriers, including the unpopulated ones)
   c. Output of the CP module in time domain, displaying the CP as distinct from the data (you may plot those samples with a different color, or encase them in a box, or some other clear method).
   d. Spectrum of signal after up-conversion

2. Channel (see Figure 1 (b)): As we are trying to isolate the synchronization aspect of OFDM, this will be a special channel called the *"OFDM project and nowhere else"* channel.

   Your channel should do three things:

   i. You must generate random data samples, and concatenate them to the front of your transmission signal... so your receiver cannot immediately begin to receive data, and must rely on synch symbols to begin to detect the data. The number of extraneous samples that you should create should be random, and randomly created every time you run your program. It should be anywhere from 1000 to 5000 samples of extraneous data. The data should be BPSK and QPSK modulated (just random constellation points), but it should carry content that is randomly generated. The receiver should discard the extraneous data.
   ii. Simulate a super-simplified constant phase delay for the whole channel, $e^{-j\beta}$. The phase should be randomly generated from the interval $\beta \in [2,5]$ radians. The phase must remain unknown at the receiver.
   iii. You must create AWGN that complies with the SNR described above. You should use the randn() function in Matlab, setting the variance suitably.

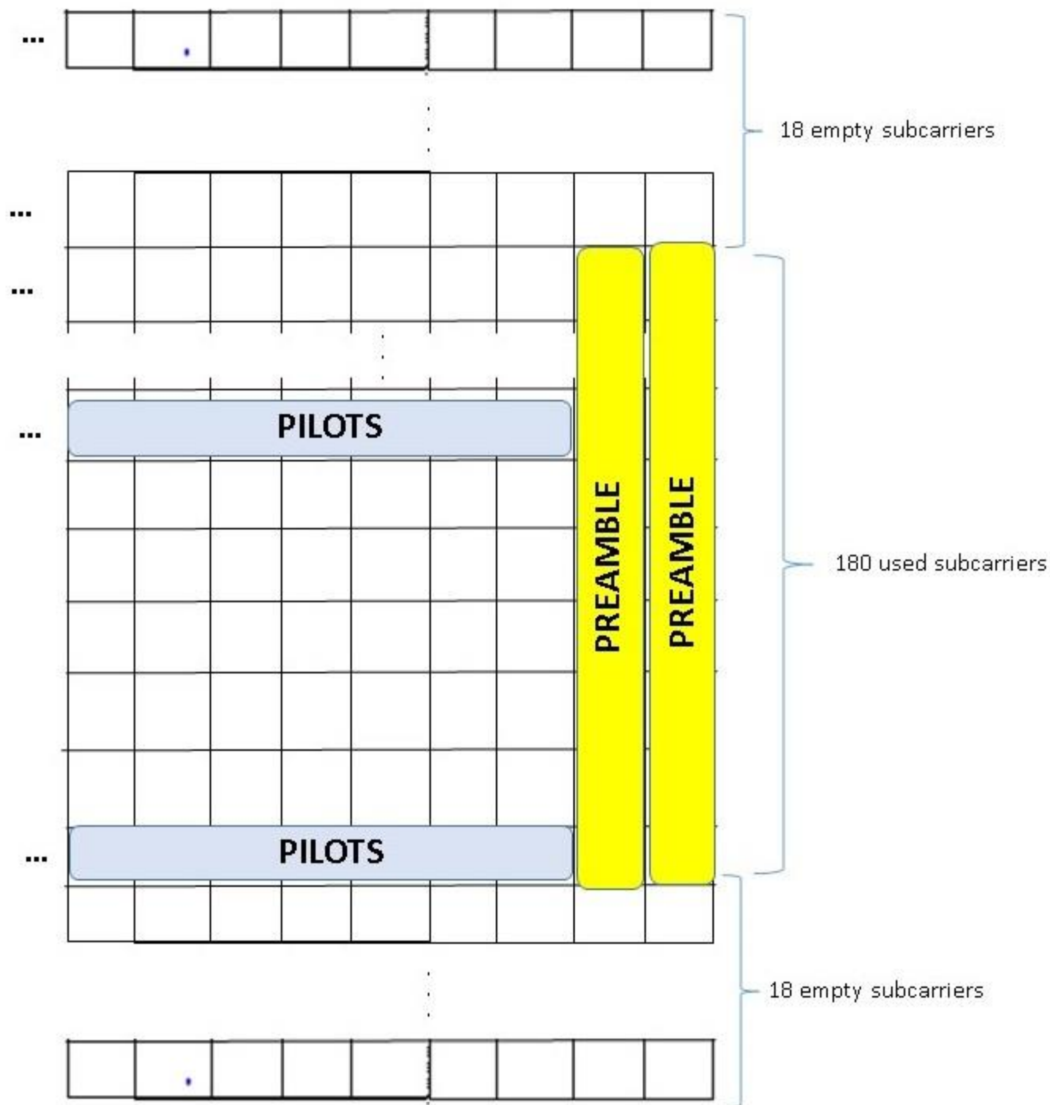

Figure 2. Location of the preamble symbols and pilot symbols in time-frequency organization. The horizontal axis represents time (right-most is first, and left-most follows). The vertical axis represents subcarrier.

3. <u>Receiver chain (see Figure 1 (c)):</u>
    a. Show frequency magnitude and phase plots of the received signal before down-conversion.
    b. The synch block should use a **cross-correlator** to discard the random, extraneous data from in front of the received signal, and find where the desired data begins. Show the output of the cross-correlator for a window of samples where the correlator matches up and yields a large output.
    c. The local oscillator at the receiver should have a frequency offset with respect to the local oscillator of the receiver. So, if the transmitter L.O. has a frequency of f_c, then the L.O. at the receiver should have a frequency of $f_c + \Delta f$. The receiver should estimate this drift, Δf, and correct for it. Your program should show the estimate of Δf in the command line (suitably described).
        i. Use the two in-coming copies of the preamble, and perform frequency domain analysis to estimate Δf.

ii. Through out your code, you may use the code in “Example 15” of the Matlab Tutorial (bottom of page 18 and top of page 19) to do frequency estimation. This example only plots the magnitude; you can plot the phase by using “angle” instead of “abs”

c. Equalizers should be set to a simple phase: $e^{j\alpha}$, where α should be estimated using known pilots. You may assume for this project that all subcarriers use the same equalizer. Use zero-forcing.
d. Provide your estimate for α in the report, so this checkpoint should be included and discussed in your documentation.
e. Graph the constellation points for six of the FFT outputs, before and after equalization on the same graph (use the hold function).
f. Compare the output of the P/S converter with the transmitted signal.
g. Create a stem plot with a sample of bits after the P/S converter. This plot must identify the bits that have error by a different color.
h. The report should include an average BER value for your transceiver for your channel.

Academic Integrity

Authors are advised that they are all responsible for the academic integrity of the entire project. You are reminded that part of your grade is a Hand-signed “Understanding of Terms” form, to be slipped under the instructor’s door any time before the due date.

References

[1] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10); 3GPP TS 36.211 V10.4.0 (2011-12) *Technical Specifications – Available in Canvas (ELMS)*