Lab 4: Wideband System

Report Due: 21:00, 5/26, 2017

1   Goal

In this lab, our goal is to build an OFDM transceiver and demonstrate the performance gain due to its ISI mitigation capability. Note that in this lab, we leave the length of the Cyclic Prefix (CP), L, to your design. We are going to see that how the choice of L affects the performance of the overall system.

2   Archietecture and Experiment Setup

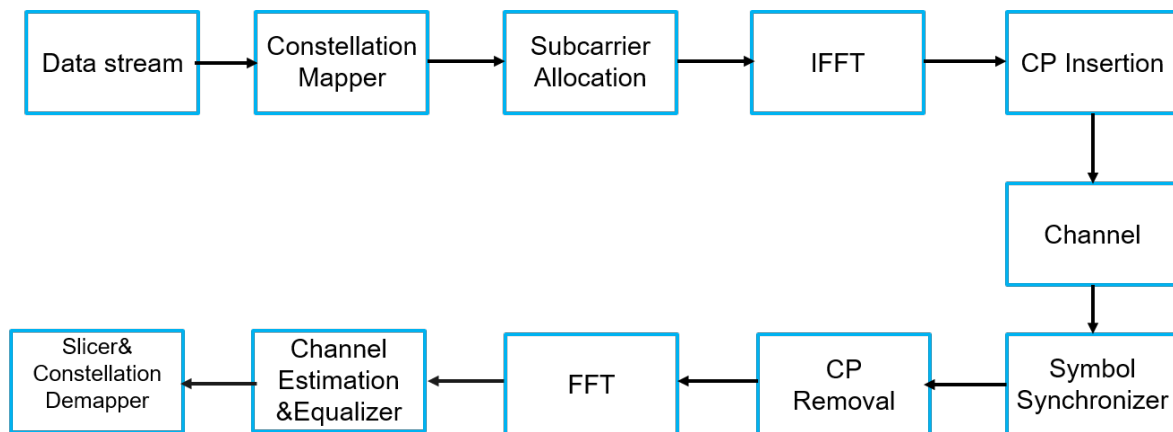The block diagram of the OFDM system is depicted in Figure 1.

Figure 1: OFDM System

At the transmitter, the transmitted bit stream is first mapped onto constellation symbols of a constellation map. Then the stream is allocated onto OFDM subcarriers. The subcarrier allocation block also adds pilot subcarriers. Later, the data is transformed to time-domain by Inverse Fast Fourier Transform(IFFT). At last, the cyclic prefix is added in the CP insertion block.

At the receiver, we first use the symbol synchronizer to estimate the starting time of an OFDM symbol and the carrier frequency offset. Then, we remove CP from the OFDM symbol. This symbol then goes directly to FFT. The channel gain is estimated by pilots. We use linear interpolation to estimate channel gains on data subcarriers. Finally, the receiver makes decision by utilizing the equalized signal, and recovers it by the constellation de-mapper.

3   Implementation

In this lab, we are going to implement an OFDM system including both the transmitter and the receiver. There are 7 files for you to complete in this lab (see Figure 3). Each of these files is a component of either the transmitter or the receiver. In the rest of this section, we will explain how to implement and build an OFDM system with them.

- **Transmitter:**
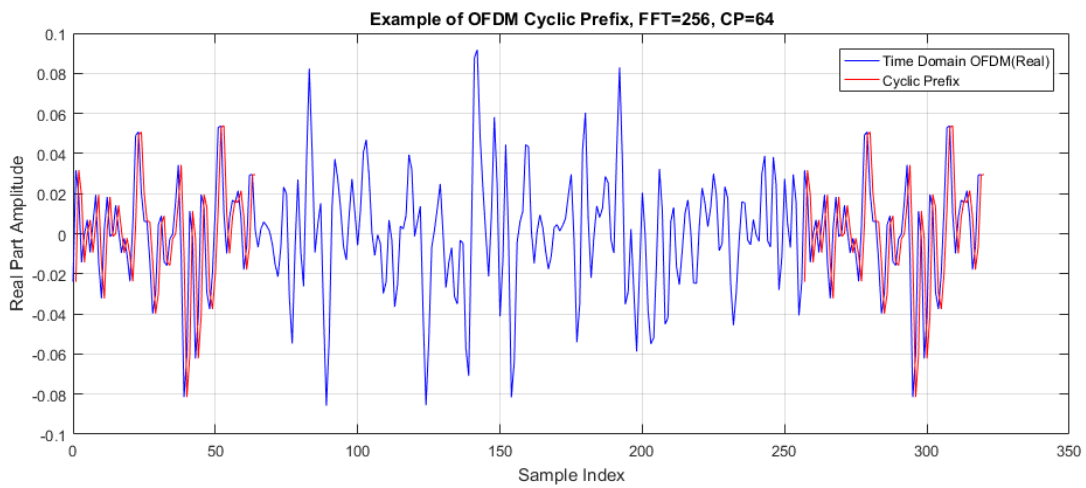   1. `sub_cyclic_prefixer.vi`
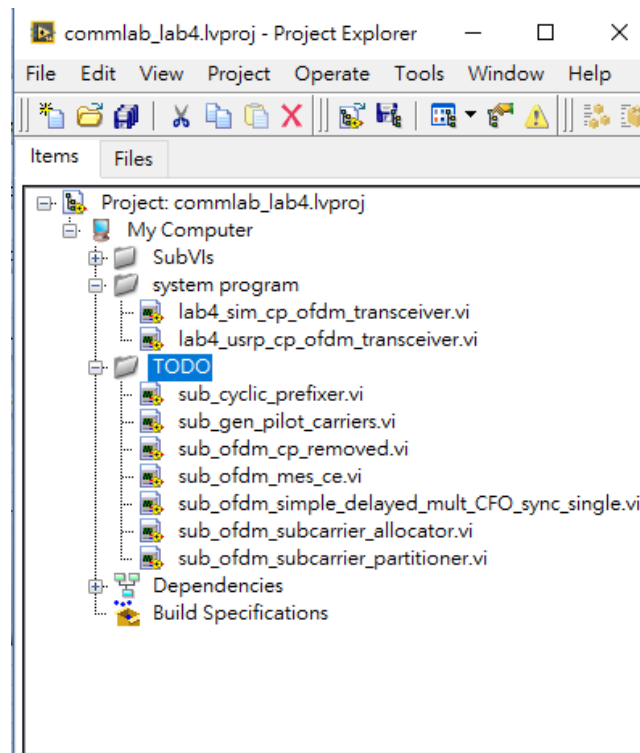
Figure 2: Example of Cyclic Prefix

Figure 3: Labview Project of Lab4

    2. `sub_ofdm_subcarrier_allocator.vi`

    3. `sub_gen_pilot_carriers.vi`

- **Receiver:**

    1. `sub_ofdm_cp_removed.vi`

    2. `sub_ofdm_simple_delayed_mult_CFO_sync.vi`

    3. `sub_ofdm_mes_ce.vi`

    4. `sub_ofdm_subcarrier_partitioner.vi`

3.1   Building OFDM transmitter components

The components for OFDM transmitter are: 1) cyclic prefixer, 2) subcarrier allocator and 3) pilot symbol generator.

3.1.1   Cyclic prefixer

Suppose the length of cyclic prefix is L. This block simply copies L time domain samples from the end, and insert them to the begin of whole signal. Figure 2 is an example for illustration. The red-colored signal is where the cyclic prefix is.

The first input of this block is time domain signal, an one-dimensional complex array. The second input is the length of the cyclic prefix (CP length). Output of this block should be signal with cyclic prefix. The data type is also the one-dimensional complex array.

Note: A simple debugging tips is that the length of the output should be the sum of the length of input plus cyclic prefix length.

3.1.2   Subcarrier allocator

An OFDM symbol can be divided into several subcarriers. The number of subcarriers should be equal to the FFT size. In practice, not all subcarriers are used to transmit message. There are three types of subcarriers. The first is *pilot subcarrier*, used for receiver to do synchronization tasks such as channel estimation and equalization. The second is *zero subcarrier*, which is the subcarrier turned of due to higher risk to encounter bad channel conditions such as deep fade or DC noise. The last type is *message subcarrier* where modulated data symbols are placed in an OFDM symbol.

The next natural question is, how to allocate these three kinds of subcarriers? There are literatures comparing the performance of OFDM systems under different kinds of subcarrier allocation schemes. However, this is out of the scope of this course, and we will implement one of them for simplicity.

The scheme we are going to implement is a frequency-domain-based pilot allocation. The spacing between pilot carriers are **5**. In other words, 4 message subcarriers are placed between two pilot subcarriers. The DC subcarrier will always be zero, and the other zero subcarriers are on the left and right side of message subcarriers. Fig. 4 shows an example of this allocation scheme. The FFT size used here is 256. The total number of message subcarriers is 120 and that of pilot carriers is 32. We can label each subcarrier with an index, and let the DC subcarrier be index zero. Then the other subcarriers are allocated symmetrically to DC subcarrier. Consider the right-hand side to the DC subcarrier, the indices of pilot subcarriers as a result are: $\{1, 6, 11, ..., 81\}$. You can easily find out the indices of pilot subcarriers on the left-hand side of DC similarly.
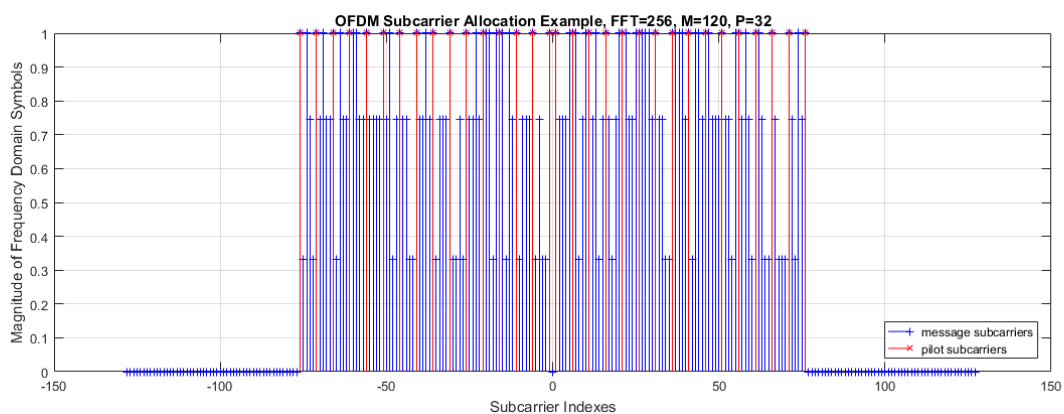

Figure 4: Subcarrier Allocation Example

In summary, your have to implement the followings functions:

1. Initialize an complex array with length equal to FFT size.

2. Interleave message subcarriers and pilot subcarriers. (Note that two additional pilot symbols should be placed beside DC subcarrier)

3. Replace the symbols in Step 1 with proper subcarrier indices.

4. Output the frequency domain OFDM symbol.

3.1.3 Pilot symbol generator

The objective of this module is to generate correct number of pilot subcarriers so that the OFDM symbol is valid. The pilot symbols from this block are used by both the transmitter and the receiver. For the transmitter, pilot symbols are used in subcarrier allocator. For the receiver, the pilot symbols are used at channel estimator and equalizer. In our OFDM system, the pilot symbols are all QPSK symbols for simplicity.

In program `sub_gen_pilot_carriers.vi`, the first input is the number of message subcarriers and the second one is the FFT size. Recall that the spacing between pilot carriers are 5. Therefore, the number of pilot subcarriers required can be calculated accordingly. For example, suppose we use 256-point FFT and the number of message subcarriers is 120. Then, the number of required pilot subcarriers is:

$$\textbf{\# of Pilots} = \frac{\text{\# of message subcarriers}}{\text{Pilot spacing}} + 2 \text{ pilots besides DC} = \frac{120}{4} + 2 = 32$$

You can use "Array subset" in LabVIEW to extract 32 elements from the begin and connect to the output port. Steps to build this block are outlined below:

1. Calculate the required number of pilot subcarriers with length of message subcarriers and FFT size.

2. Sort the subcarriers according to their indices (ascending order)

3. Use "array subset" module and Output the proper number of pilot subcarriers.

3.2 Building OFDM receiver components

There are four components of the OFDM receiver. The first one is a symbol synchronizer. The second is a module to remove cyclic prefix. The third is a subcarrier partitioner, and the last one is the channel estimator.

3.2.1 Symbol synchronizer

The symbol synchronizer will utilize the structure of an OFDM symbol (the cyclic prefix) to locate an OFDM symbol and jointly estimate the carrier frequency offset of it. The idea is explained as follow: Suppose the FFT and cyclic prefix length are N and L respectively, and the CFO is f_d. Then the received signal y will rotate at a rate proportional to f_d. This can be written as:

$$y[n] = e^{j2\pi f_d n} x[n].$$

The estimator computed in this block is the autocorrelation function:

$$R(k) = \frac{1}{L} \sum_{i=0}^{L-1} y[k+i] y^*[k+i+N].$$

You can easily check that f_d can be estimated by:

$$\hat{f}_d = -\frac{1}{2\pi N} \arg R\left(\hat{k}_{max}\right).$$

where

$$\hat{k}_{max} = \arg\max_{\hat{k}} |R(k)|$$

Note that the maximum of $R(k)$ is also detected as the begin of OFDM symbol. The following summarize the things you need to do to implement this block:

- Compute the autocorrelation $R(k)$ of received signal given FFT size N and CP length L.

- Find the index of the maximum of the amplitude of $R(k)$

- Estimate the CFO with the estimator mentioned above.

- Correct the CFO with your estimate. (Tip: use ramp function)

- Output synchronized OFDM symbol

As a final notice, the input signal consists of successive OFDM symbols. Therefore, the output type should be two dimensinoal array to contain multiple synchronized OFDM symbols.

3.2.2   Removing cyclic prefix

This block simply does the opposite of cyclic prefixer. Suppose the cyclic prefix length is L, then L samples from the beginning of the input signal are removed.

The first input of this module is time domain signals, an one dimensional complex array. The second input is the length of the cyclic prefix. The output should also be a time domain signal with cyclic prefix removed. You can also check the length as in cyclic prefixer to verify your implementation.

3.2.3   Subcarrier partitioner

This block does the opposite of subcarrier allocator. Furthermore, it collects all message subcarriers and pilot subcarriers to do channel estimation and equalization. Steps to build this block are as follows:

1. Remove zero subcarriers.

2. Collect message subcarriers (ascending order of subcarrier indices).

3. Collect pilot subcarriers (ascending order of subcarrier indices).

4. Output message subcarriers and pilot subcarriers.

3.2.4   Channel estimator

Since the pilot subcarriers in this lab are all allocated in frequency domain with equal spacing, the channel estimation process can be done by interpolation. The idea is explained in Figure 5. Given channel coefficients of pilot subcarriers and the subcarrier spacing, the channel coefficients of message subcarriers can be estimated by linear interpolation. Note that this scheme assume slow fading channel. For fast fading channel, more complex interpolation scheme should be used.

The input are channel coefficients of pilot subcarriers. The data type is the one-dimensional complex array. Note that the DC subcarrier are always zero. So the collected pilot subcarriers are across two sides of the DC subcarrier. The output should be channel coefficients of message subcarriers. The coefficients are used by the equalizer.
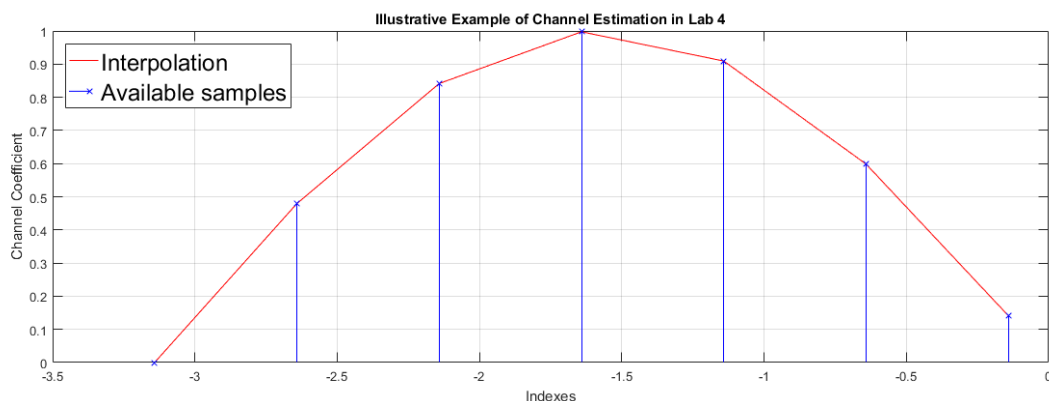

Figure 5: Illustrative Example of Interpolation

In summary, the steps to finish this block are:

1. Divide the channel coefficients symmetrically about the DC subcarrier.

2. For each side of channel coefficients, do linear interpolation between two channel coefficients of pilot subcarriers to approximate the channel coefficients of message subcarriers (4 points between 2 pilot coefficients)

3. Sort the estimated channel coefficients according to the subcarrier indices (ascending order).

4. Output channel coefficients of message subcarriers.

3.3 LabVIEW **Simulation**

The simulator is an OFDM system composed of the modules we implemented in the previous section. Although the simulator is already provided for you, it is important to understand the system structure so that you can learn how to connect the components yourselves. Therefore, we will first explain the structure of the basic OFDM system, and then we will explain how to use this simulator.

3.3.1 **Basic OFDM System Structure**

Figure 6 and 7.are the block diagrams of a basic OFDM system, including both the transmitter and the receiver. At the transmitter side, the message bits are first modulated to symbols, and then the symbols combined with pilot symbols are passed to the subcarrier allocator. Next, inverse FFT is used to transform frequency-domain symbols to time domain, and the cyclic prefix is added to form a time-domain OFDM symbol. This completes the transmitter side.

On the other hand, OFDM receiver will first receive time-domain OFDM symbol, and then the symbol is passed to symbol synchronizer to locate and correct the frequency offset of it. The next step is to remove cyclic prefix and pass the signal to FFT to transform time-domain signal to frequency domain. Afterwards, the subcarrier partitioner will extract message subcarriers and pilot subcarriers. With the knowledge of pilot subcarriers, the channel coefficients of message subcarriers can be estimated by interpolation. The last step is to use estimated channel coefficients to equalize message subcarriers and demodulate symbol to bits. This completes the receiver part of an OFDM system. An OFDM system can therefore, be built with the components we implemented in the beginning of this section.
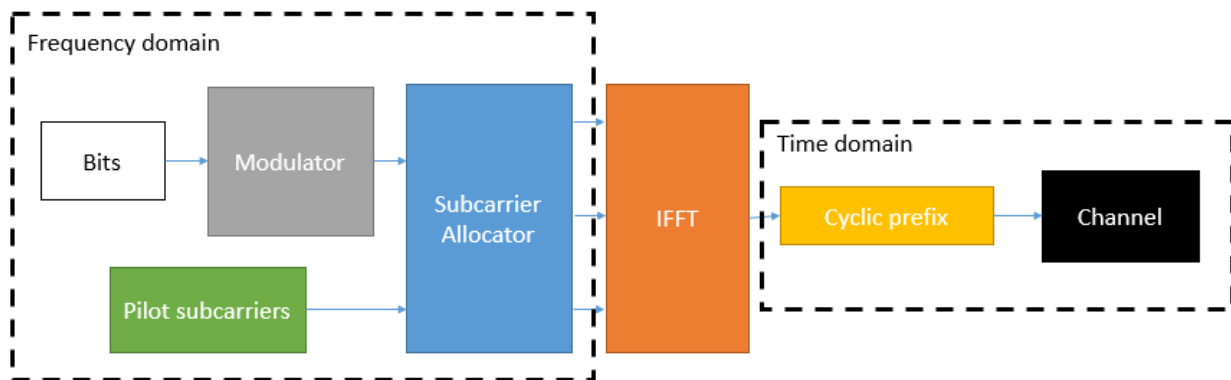

Figure 6: Block Diagram of the OFDM Transmitter

3.3.2 **Usage of Simulator**

The control panel of the program `lab4_sim_cp_ofdm_transceiver.vi` has many parameters, and most of them are similar to those in previous labs. We will only explain the parameters related to OFDM system for simplicity.

FFT size The heart of OFDM is the transformation of signal between time domain and frequency domain. FFT is an efficient algorithm to complete this task. The computation complexity of FFT depend on the points of FFT. Therefore, FFT size will affect the performance of OFDM system. The supported FFT size of this simulator are powers of 2, for example, 64, 128, 256, etc.
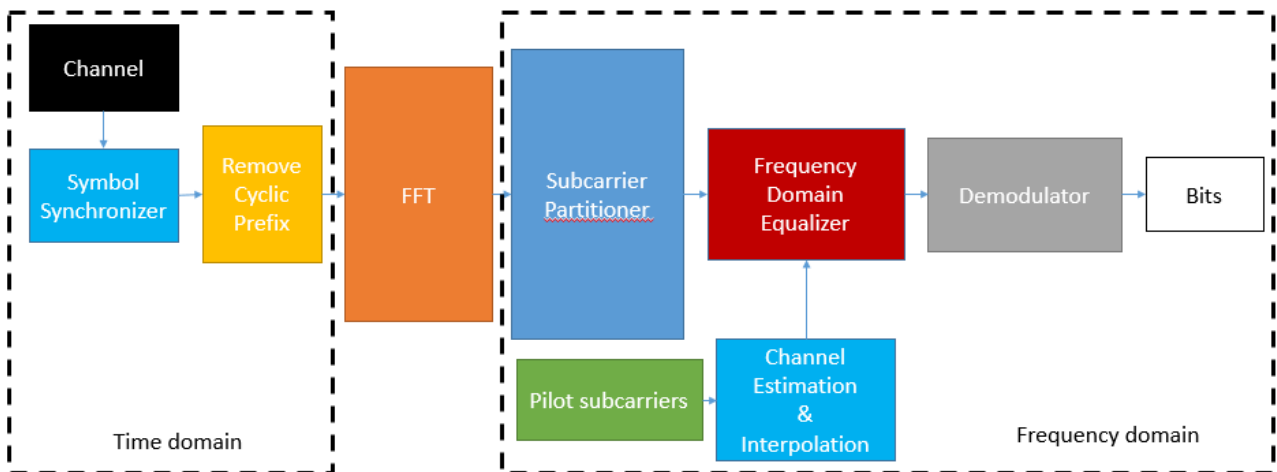
Figure 7: Block Diagram of the OFDM Receiver

Message subcarriers   Message subcarriers are the number of subcarriers where modulated symbol are transmitted. Since the spacing of pilots subcarriers of this OFDM system is 5, the valid number of message subcarriers should be multiples of 4. Note that the value should not exceed the FFT size.

Pilot subcarriers   Pilot subcarriers are related to the message subcarriers and the FFT size. This is also due to the fixed spacing of pilot subcarriers of our OFDM system. A simple rule is to divide the number of message subcarriers by 4 and plus 2. The additional two subcarriers account for the 2 pilots subcarriers beside the DC subcarrier.
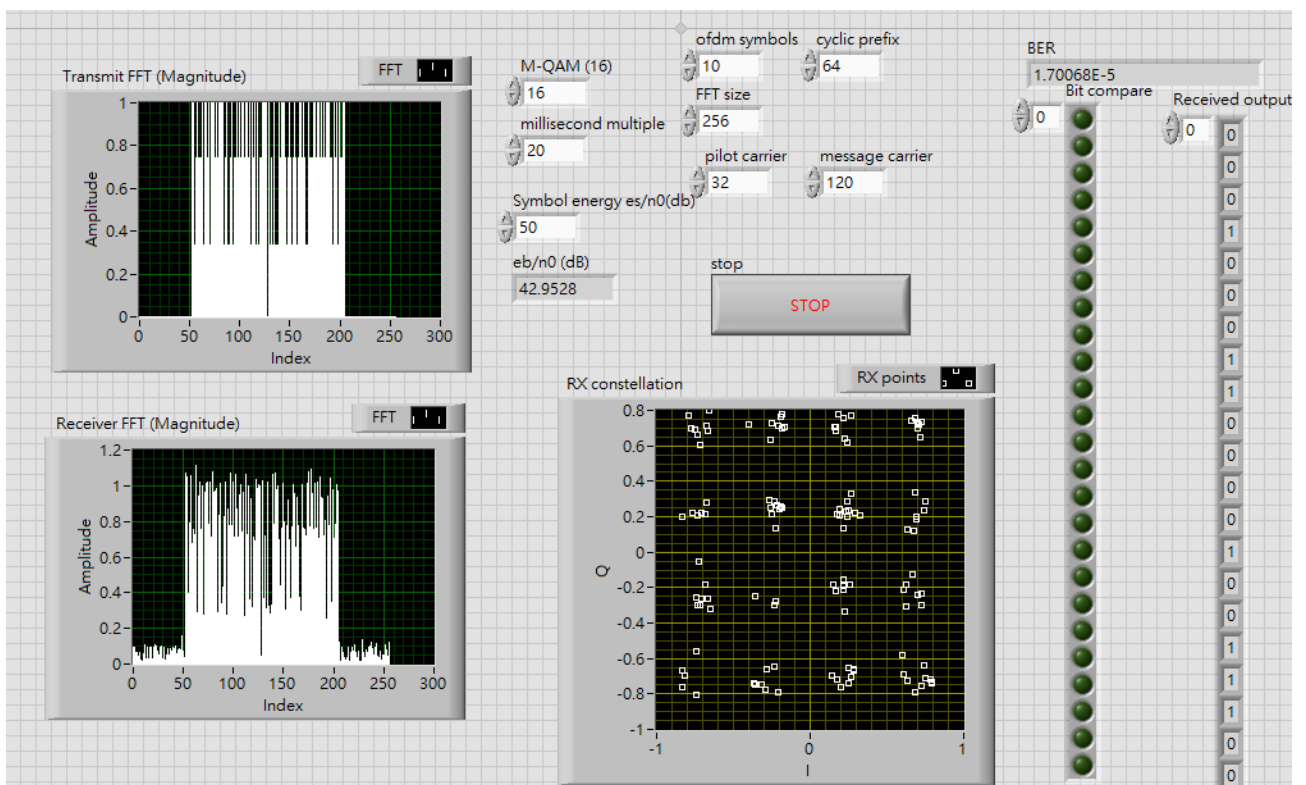

Figure 8: Interface of the Simulator

3.4 USRP **Implementation**

After simulation, we can extend our implementation to USRP. `lab4_usrp_cp_ofdm_transceiver.vi` is the one we implemented for you. The parameters and system structure are similar to that in the previous part, while there are some additional parameters you should notice. In the following we will explain the additional controls so that you can use this system to do experiments. After that, some verified parameter sets are provided for you as reference.

3.4.1 **Additional controls**

The first one is "Threshold". This is the variable to change the threshold of energy detector embedded in the system. The second is "Min length". This will force receiver to compute signal with length greater than this value to save computation time. The third one is "Power scale". This parameter will scale the signal at transmitter before sending it to USRP. It is relative to the original value. The purpose of it is to scale the signal to low SNR regime and the unit is represented in dB. The value of it should below zero. The last one is "TX on". It is a button which controls the transmission of signal.

3.4.2 **Reference parameters**

Due to hardware constraint, the IQ rate of the USRP should not exceed 2 MS/s. Table 1 lists some verified parameter sets. You are recommended to use one of them to do experiment on your implementation.

Case	IQ rate	gain	number of samples	Threshold	min length	FFT size	CP	Message	Pilot	bits per packet
1	1M	0	20000	-70	200	256	64	120	32	1600
2	2M	0	20000	-80	100	128	32	80	22	1600
3	1M	0	20000	-84	200	512	128	280	72	1600

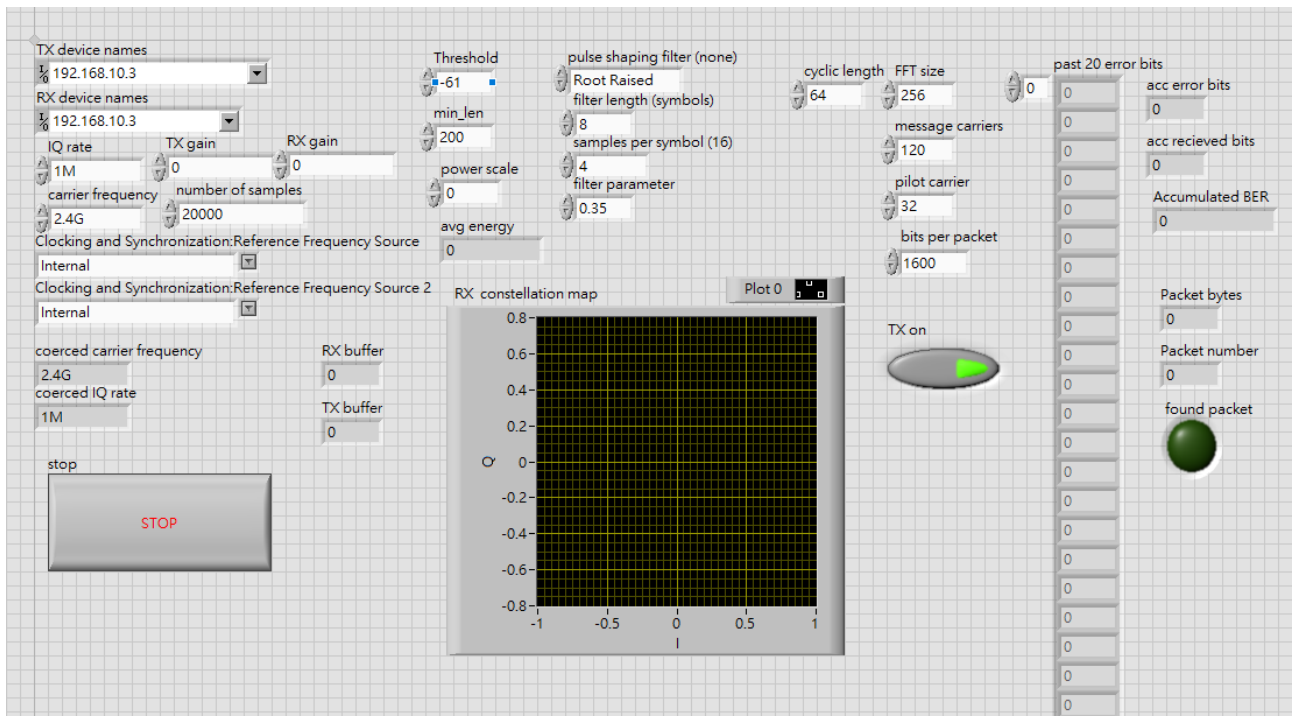Table 1: Verified Parameter Sets

Figure 9: Interface of USRP OFDM System

4 Exercises

4.1 LabVIEW

1. Simulate your OFDM system with CP length $L = 16, 32, 64, 128$. Plot in the same figure of the overall BER versus $\frac{E_b}{N_0}$ curves for the OFDM system built above. $\frac{E_b}{N_0}$ should be from 0 dB all the way to the $\frac{E_b}{N_0}$ that generates the BER of 10^{-4}.

2. Calculate the peak-to-average power ratio (PAPR) in the above simulation.

3. Repeat 1. Plot channel gain on data subcarriers on one realizaton.

4.2 USRP

1. Simulate your OFDM system with CP length $L = 32, 64, 128$. Plot in the same figure of the overall BER versus $\frac{E_b}{N_0}$ curves for the OFDM system built above. $\frac{E_b}{N_0}$ should be from 0 dB all the way to the Eb N0 that generates the BER of 10^{-3}.

2. Try to observe received signal on time domain, and record Image part Amplitude of one OFDM symbol from $L = 32, L = 64$. Note the SNR of your diagram.

3. Repeat 1. Plot channel gain on data subcarriers on one realizaton.

5 Lab Report

There is no format requirements for your lab report. In the report, you should address the results of the exercises mentioned above. You should also include your simulation program in the appendix of the report. Include whatever discussions about the new findings during the lab exercise, or the problems encountered and how are those solved. Do not limit yourself to the exercises specified here. You are highly encouraged to play around with your simulation program on self-initiated extra lab exercises/discussions.