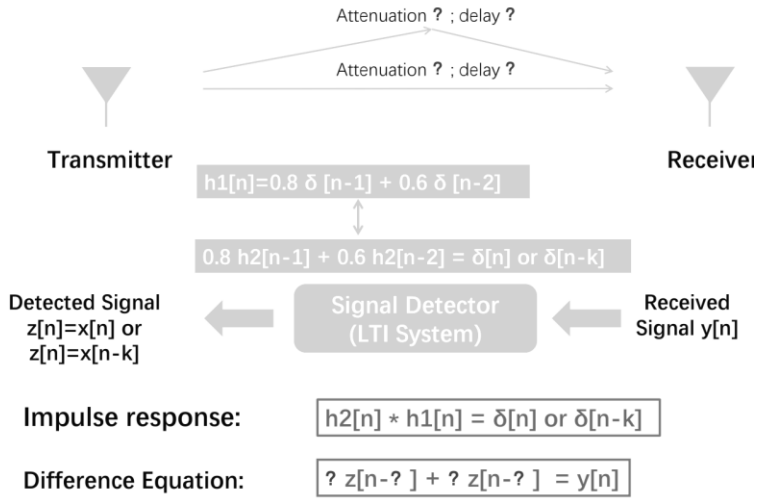
Lab 5：Channel estimation and time domain equalization

Author	Name：宋宜航　　张皓东 Student ID:12112717　　12113010

Introduction

1.1 Channel Estimation

Channel estimation involves determining the characteristics of the communication channel during the signal transmission process. In digital communication, signals may encounter various interferences such as noise, interference, and fading. Accurate channel estimation is essential for the receiver to faithfully reconstruct the transmitted signal.



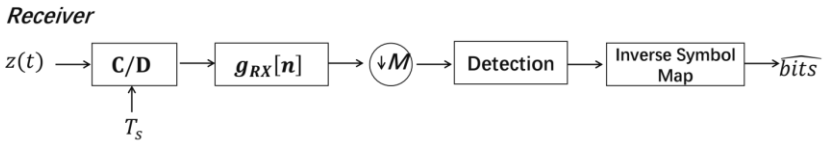
There are various channel estimation algorithms, with a common approach being the use of pilot symbols embedded in the transmitted signal. These pilot symbols are known to both the transmitter and the receiver, allowing the receiver to estimate the channel response by comparing the received pilot symbols with the known transmitted ones.

Understanding the complexity of channel estimation is crucial for optimizing communication systems, especially in situations where channel characteristics may change over time.

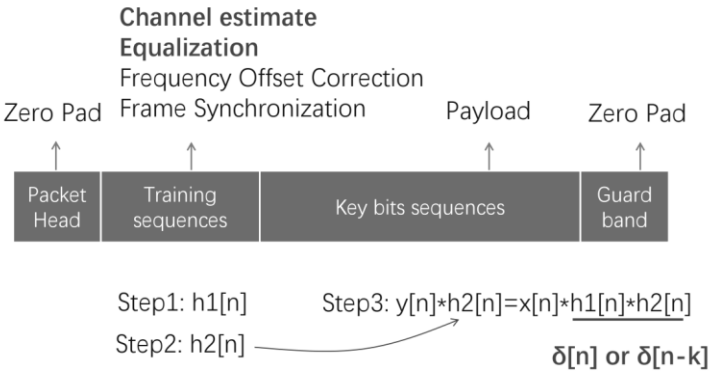
1.2 Time-Domain Equalization

Time-domain equalization is employed to mitigate the impact of channel distortion, especially in scenarios where the channel introduces inter-symbol interference (ISI). ISI occurs when multiple symbols overlap in time, making it challenging to distinguish between them at the receiver.

$$z(t) = \alpha_0 e^{j\varphi_0} x(t-\tau_0) + \alpha_1 e^{j\varphi_1} x(t-\tau_1) + v(t)$$



Algorithms like Maximum Likelihood Sequence Estimation (MLSE) and Zero-Forcing Equalization are commonly used for time-domain equalization. MLSE attempts to find the most likely sequence of transmitted symbols given the received signal, while Zero-Forcing Equalization aims to eliminate ISI by inversely filtering the received signal.



1.3 Least Mean Squares Algorithm (LMS)

The main mathematical technique that will be used in this lab is known as linear least squares, which will be used both for estimating the channel and for computing the equalizer.

Let **A** denote a $N \times M$ matrix. The number of rows is given by N and the number of columns by M. If $N = M$ we say that the matrix is square. If $N>M$ we call the matrix tall and if $M>N$ we say that the matrix is fat. We use the notation $\boldsymbol{A^T}$ to denote the transpose of a matrix, $\boldsymbol{A^*}$ to denote the Hermitian or conjugate transpose, and $\boldsymbol{A^c}$ to denote the conjugate of the entries of **A**. Let **b** be a $N \times 1$ dimension vector. The 2-norm of the vector is given ||a|| = $\sqrt{\sum_{n=1}^{N} |b_n|^2}$.

Consider a system of linear equations written in matrix form:

$$\boldsymbol{Ax = b}$$

where A is the known matrix of coefficients sometimes called the data, x is a vector of unknowns, and b is a known vector often called the observation vector. We suppose that A is full rank. First consider the case where N = M. Then the solution to Eq. (6) is $\mathbf{x} = \mathbf{A^{-1}b}$.

Now suppose that N>M. In this case A is tall and the system overdetermined in general. This means there are N equations but M unknowns, thus it is unlikely (except in special cases) that an exact solution exists. In this case we pursue an approximate solution known as least squares. Instead of solving Eq. (6) directly we instead propose to find the solution to the squared error:

$$\min \ ||\boldsymbol{Ax} - \boldsymbol{b}||^2$$

Using matrix calculus, it can be shown that the solution to this problem assuming that A is full-rank is:

$$\boldsymbol{x_{LS}} = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*\boldsymbol{b}$$

Note that $\boldsymbol{A}^*\boldsymbol{A}$ is a square matrix and invertible because of the full-rank assumption. We refer to $\boldsymbol{x_{LS}}$ as the linear least square error (LLSE) solution. For $\boldsymbol{Ax} = \boldsymbol{b}$.

We use the squared error achieved by $\boldsymbol{x_{LS}}$ to measure the quality of the solution. With some calculus and simplifying, it can be shown that the least squares error achieved is given by:

$$\boldsymbol{J}(\boldsymbol{x_{LS}}) = \left|\left|\boldsymbol{Ax_{LS}} - \boldsymbol{b}\right|\right|^2 = \boldsymbol{x}^*\boldsymbol{A}^*(\boldsymbol{Ax} - \boldsymbol{b}) - \boldsymbol{b}^*(\boldsymbol{Ax} - \boldsymbol{b})$$
$$= \boldsymbol{b}^*\boldsymbol{b} - \boldsymbol{b}^*\boldsymbol{Ax}$$

Next, Applying the algorithm to channel estimation.

Suppose that the known training symbols. are inserted into the transmitted sequence such that s[n] = t[n] for n = 0, 1, ..., N_t. To solve for the estimate of the channel using the LLSE solution, it is necessary to form a system of linear equations. Consider:

$$\boldsymbol{y}[\boldsymbol{n}] = \sum_{\boldsymbol{l}=\boldsymbol{0}}^{\boldsymbol{L}} \boldsymbol{h}[\boldsymbol{l}]\boldsymbol{s}[\boldsymbol{n} - \boldsymbol{l}] + \boldsymbol{v}[\boldsymbol{n}]$$

where s[n] = t[n] for n = 0, 1, ..., N_t . Since s[n] is unknown for n ≥N_t (that is the unknown data), we need to write the squared error only in terms of the unknown data.

With this in mind the least squares problem is to find the channel coefficients that minimize the squared error:

$$\{\hat{h}[0], \hat{h}[1], \dots, \hat{h}[L]\} = arg \min_{a[0],a[1]\dots} \sum_{n=L}^{N_t-1} \left| y[n] - \sum_{l=0}^{L} a[l]t[n-l] \right|^2$$

The summation starts with n = L to ensure unknown data is not included. The least squares estimator is simply a generalization of the narrowband estimator.

First write the observed data as a function of unknown in matrix form:

$$\underbrace{\begin{bmatrix} y[L] \\ y[L+1] \\ \vdots \\ y[N_t-1] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} t[L] & \cdots & t[0] \\ t[L+1] & \ddots & \vdots \\ \vdots & & \vdots \\ t[N_t-1] & \cdots & t[N_t-1-L] \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} a[0] \\ a[1] \\ \vdots \\ a[L] \end{bmatrix}}_{\mathbf{a}},$$

where we refer to T as the training matrix. If T is square or tall and full rank, then $\boldsymbol{A^*A}$ is an invertible square matrix, The tall assumption (square with equality) requires that:

$$\boldsymbol{N_t - L \geq L + 1}$$

Or, equivalently

$$\boldsymbol{N_t \geq 2L + 1}$$

Generally choosing N_t much larger than L+1 (the length of the channel) gives better performance. The full rank condition can be guaranteed by ensuring that the training sequence is persistently exciting. Basically, this means that it looks random enough. Random training sequences perform well while the all constant training sequence fails. Training sequences with good correlation properties generally satisfy this requirement. Finally, we can get a[0] by the Least Mean Square Algorithm.

1.4 Matrix Representation in Channel Estimation

As an example, the object of this experiment is to convert the equation below:

$$y[n] = a[n] * t[n] = \sum_{l=0}^{L} a[l]t[n-l]$$

into a matrix multiplication form to facilitate computer calculations and program block diagram construction.

When the n = L to get the $y[L]$,

$$y[L] = a[0]t[L] + a[1]t[L-1] + \cdots + a[L]t[0]$$

Similarly, when the n=L+1 to get $y[1]$,

$$y[L+1] = a[0]t[L+1] + a[1]t[L] + \cdots + a[L]t[1]$$

When the n = $\boldsymbol{N_t} - 1$,

$$y[\boldsymbol{N_t} - 1] = a[0]t[\boldsymbol{N_t} - 1] + a[1]t[\boldsymbol{N_t} - 2] + \cdots + a[L]t[\boldsymbol{N_t} - 1 - L]$$

Transforming the summation into the form of a vector product, for example:

$$y[L] = \begin{bmatrix} t[L] & t[L-1] & \ldots & t[0] \end{bmatrix} \begin{bmatrix} a[0] \\ \vdots \\ a[L] \end{bmatrix}$$

Finally, the row vectors are stitched together to obtain the matrix form:

$$\underbrace{\begin{bmatrix} y[L] \\ y[L+1] \\ \vdots \\ y[N_t - 1] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} t[L] & \cdots & t[0] \\ t[L+1] & \ddots & \vdots \\ \vdots & & \vdots \\ t[N_t - 1] & \cdots & t[N_t - 1 - L] \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} a[0] \\ a[1] \\ \vdots \\ a[L] \end{bmatrix}}_{\mathbf{a}},$$

1.5 Construction of Toeplitz Matrix

Assuming we have a channel impulse response of length N, denoted as h = [h0, h1, …, hN−1] where **h** is a column vector. If we transmit a signal of length M, represented by X = [x0, x1, …, xM−1], the received signal **y** can be expressed as a convolution operation:

$$\mathbf{y} = \mathbf{H} * \mathbf{x}$$

Here, **H** is a Toeplitz matrix, constructed from the channel impulse response **h**

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \ldots & 0 \\ h_1 & h_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \ldots & h_0 \end{bmatrix}$$

This Toeplitz matrix represents the convolution relationship between the channel impulse response and the transmitted signal.

Matrix Representation in Equalization: In equalization problems, we aim to design an equalizer W to make the received signal as close as possible to the transmitted signal. Assuming the received signal is y and the transmitted signal is x, the equalization operation can be represented as:

$$\mathbf{x}_{\text{eq}} = \mathbf{W} * \mathbf{y}$$

Similarly, **W** is also a Toeplitz matrix, and its weights are related to the equalizer design

$$\mathbf{W} = \begin{bmatrix} w_0 & 0 & \dots & 0 \\ w_1 & w_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ w_{M-1} & w_{M-2} & \dots & w_0 \end{bmatrix}$$

Here, the weights in **W** can be obtained through optimization algorithms such as the Least Mean Squares (LMS) algorithm to minimize the error between the equalized signal and the transmitted signal

$$\underbrace{\begin{bmatrix} \hat{h}[0] & 0 & \cdots & \cdots \\ \hat{h}[1] & \hat{h}[0] & 0 & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ \hat{h}[L] & & & \\ 0 & \hat{h}[L] & \cdots & \cdots \\ \vdots & & & \end{bmatrix}}_{\hat{\boldsymbol{H}}} \underbrace{\begin{bmatrix} f[0] \\ f[1] \\ \vdots \\ \\ \\ f[L_f] \end{bmatrix}}_{\boldsymbol{f}} = \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}}_{\boldsymbol{e_{n_d}}}$$

$$\hat{\boldsymbol{f}}\boldsymbol{n_d} = \left(\hat{\boldsymbol{H}}^{*}\hat{\boldsymbol{H}}\right)^{-1}\hat{\boldsymbol{H}}^{*}\boldsymbol{e_{n_d}}$$

$$J_f[n_d] = \| \hat{\boldsymbol{H}}\hat{\boldsymbol{f}} - \boldsymbol{e_{n_d}} \|^2$$

1.6 Indirect equalization algorithm

The LMS algorithm is one of the indirect equalization algorithms

When channel equalization is involved, the LMS algorithm (Minimum Mean Square Algorithm) is a common adaptive filtering algorithm that adjusts the weight of the equalizer to minimize the mean square error between the output signal and the desired signal。

Least Mean Squares Algorithm (LMS): The LMS algorithm is an adaptive filtering technique used to adjust filter weights to minimize the mean square error of the error signal. In the realms of channel estimation and equalization, the LMS algorithm is commonly employed to adaptively update the weights of filters, accommodating changes and time variations in the channel.

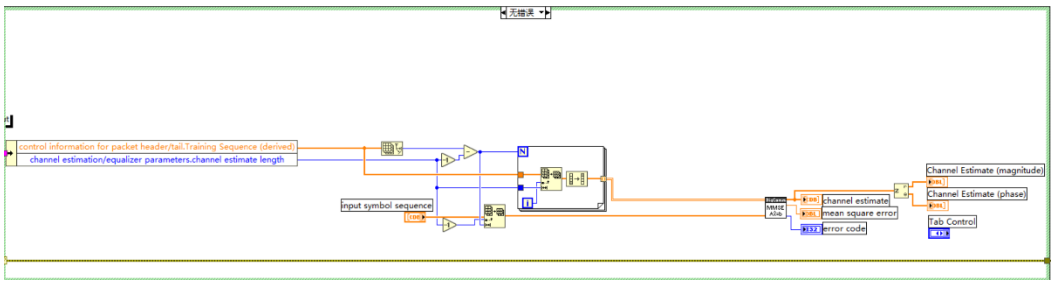
In channel estimation, the LMS algorithm is utilized to adaptively adjust the estimated channel response, providing a more accurate reflection of the actual channel characteristics. By monitoring the error between the received signal and the estimated channel response, the LMS algorithm dynamically updates the channel estimation to adapt to changing channel conditions.

In equalization, the LMS algorithm is applied to adaptively adjust the weights of the equalizer to reduce interference and distortion in the received signal. The equalizer aims to eliminate or minimize multipath interference and temporal distortion in the received signal, making the signal more easily demodulated and decoded at the receiver

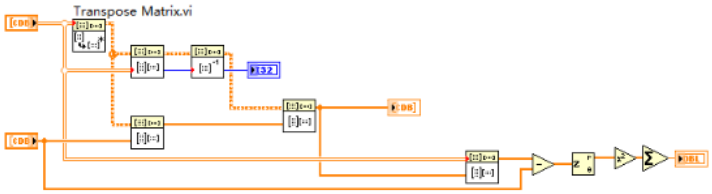
Lab results & Analysis：

Channel Estimation with MMSE Algorithm:

We can use the MMSE algorithm introduced earlier to implement channel estimation. The general process for channel estimation is outlined in the following diagram. Initially, the input bit stream is transformed into a matrix format representing the training sequence, facilitating subsequent calculations. The obtained results are then passed to the MMSE module for channel estimation.

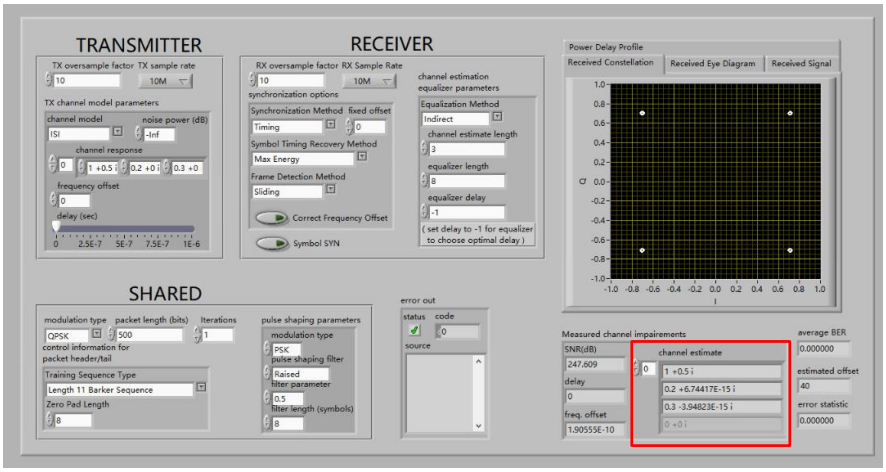


The program of the MMSE module is shown as bellow:



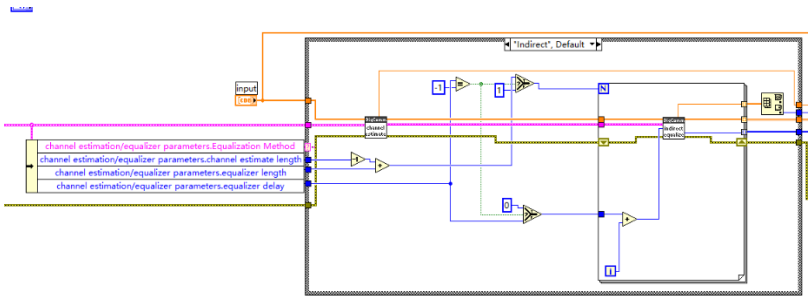
The basic process involves using the least squares method to manipulate matrices, obtaining the previously derived least squares matrix. This matrix is multiplied by the input signal sequence, and the output is the estimated channel sequence. By subtracting, taking the absolute value, and summing, the error is computed.

The simulation results are as follows, we set the channel response is 1+0.5i, 0.2 and 0.3, then we can find that the channel estimate sequence is exceedingly closed to the true channel response which indicate that the effect is excellent and the algorithm and program are nearly correct.

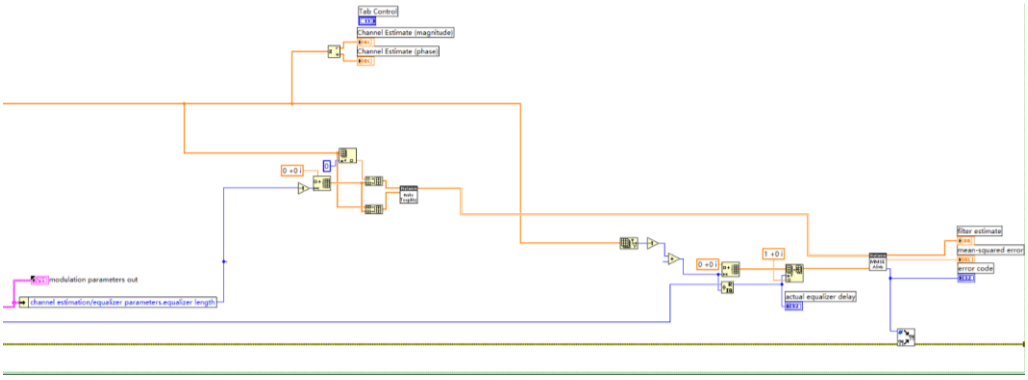


Indirect Equalization Algorithm and Construction of Toeplitz Matrix

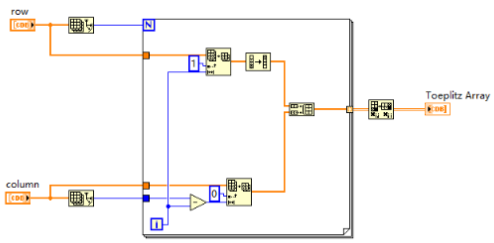
The whole program of the indirect equalization is as follows:



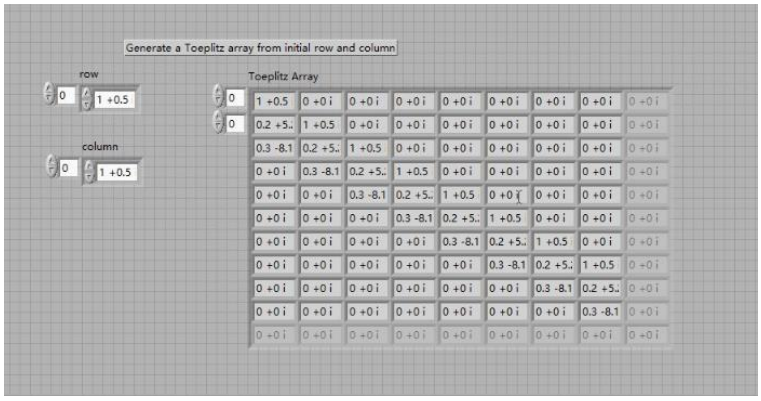
This process involves first using the channel estimation module to estimate the channel, and then importing the indirect equalization module for equalization operations. The detailed procedure for the indirect equalization module is as follows:



Its first step is to construct the estimated channel sequence obtained earlier into a Toeplitz matrix, and then use the least squares method mentioned earlier to achieve equalization. Therefore, a crucial step is to construct the Toeplitz matrix. The detailed procedure for this module is as follows. Its construction method is the same as the method introduced earlier, which is to arrange the sequence into a Toeplitz matrix according to the specified row-column structure.



The result of Toeplitz matrix is as shown below:

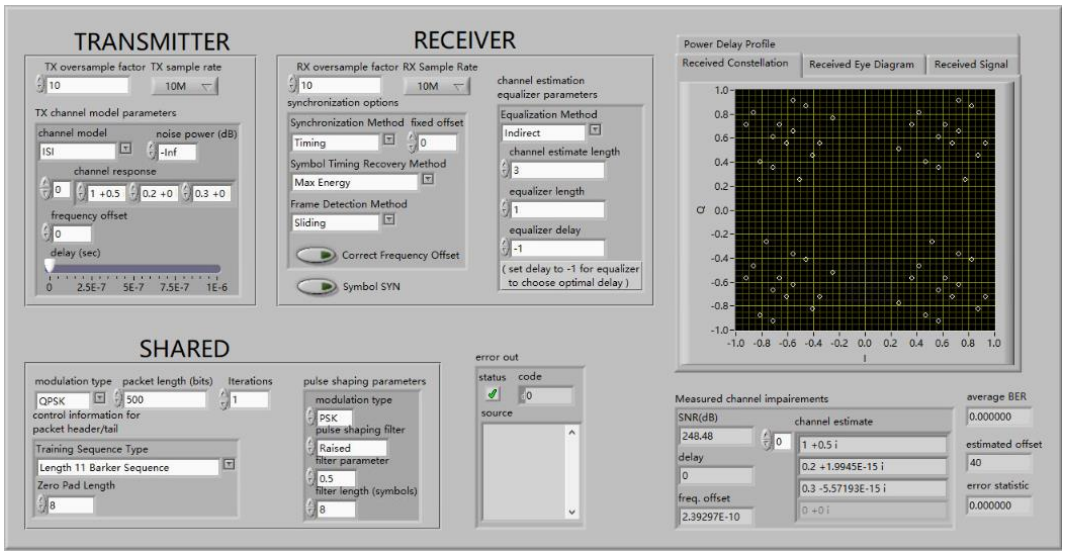


Its form conforms to the structure of a Toeplitz Matrix as introduced earlier.

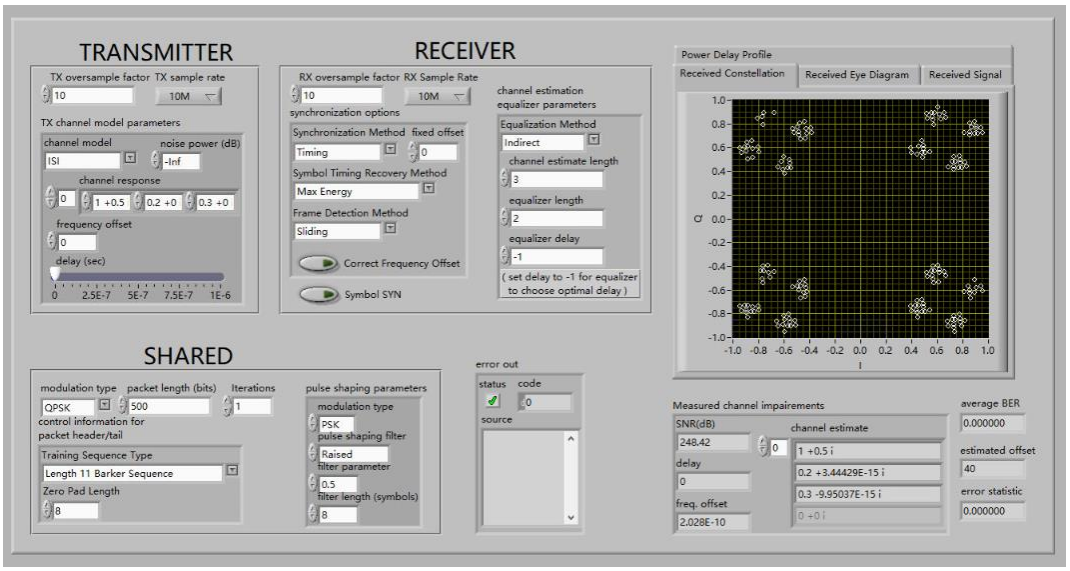
The Impact of Equalizer Length on Equalization Performance

In theory, The length of the equalizer has a significant impact on its performance, as it directly determines its ability to adjust to the channel. When there is frequency-selective fading in the channel, a shorter equalizer may not effectively compensate for the frequency-selective fading. In such cases, a longer equalizer is needed to cover a broader frequency range. Delay spread in the channel causes different parts of the signal to arrive at the receiver at different times. A longer equalizer can better handle delay spread, ensuring that the received signal is aligned in time, thereby improving equalization performance. There is a trade-off between equalizer length and computational complexity. Longer equalizers typically require more computational resources, so the choice of equalizer length needs to balance computational complexity with equalization performance. Longer equalizers may be more susceptible to noise, leading to instability. Therefore, when choosing the length of the equalizer, a careful consideration is needed, taking into account channel conditions, system requirements, and computational resources.

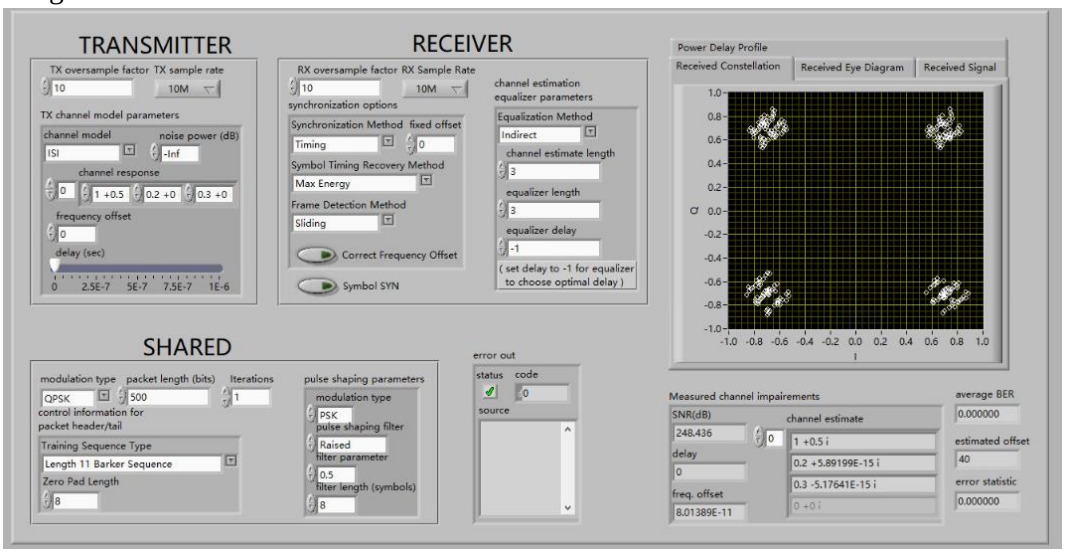
Through the simulation, the results are as follows:
Length = 1:



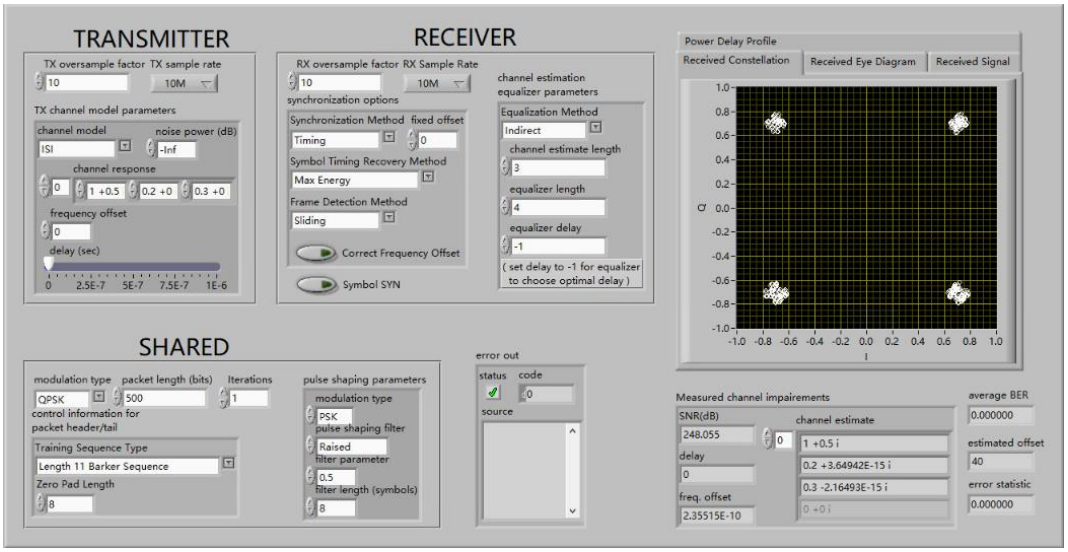
Length = 2:



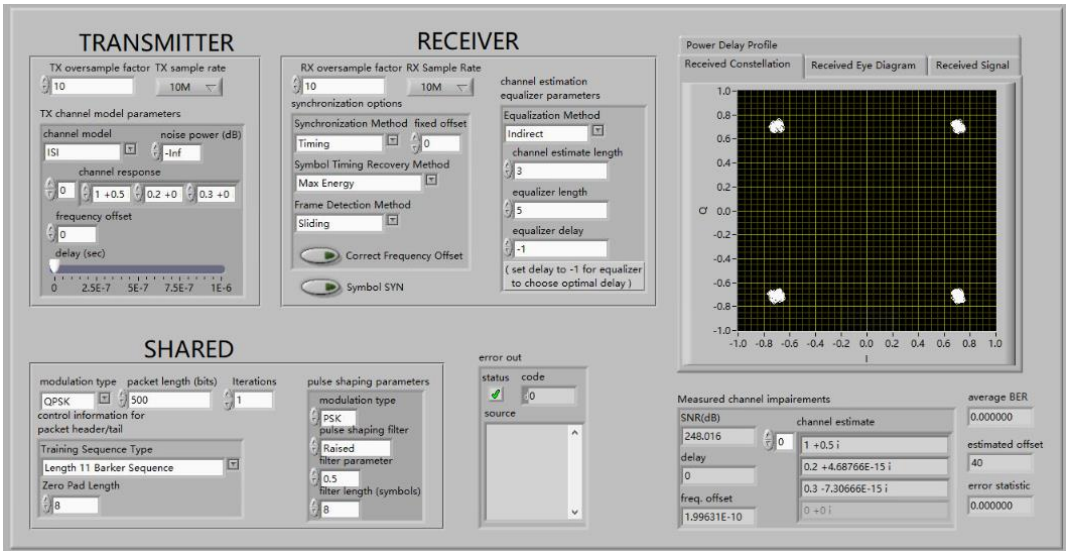
Length = 3:



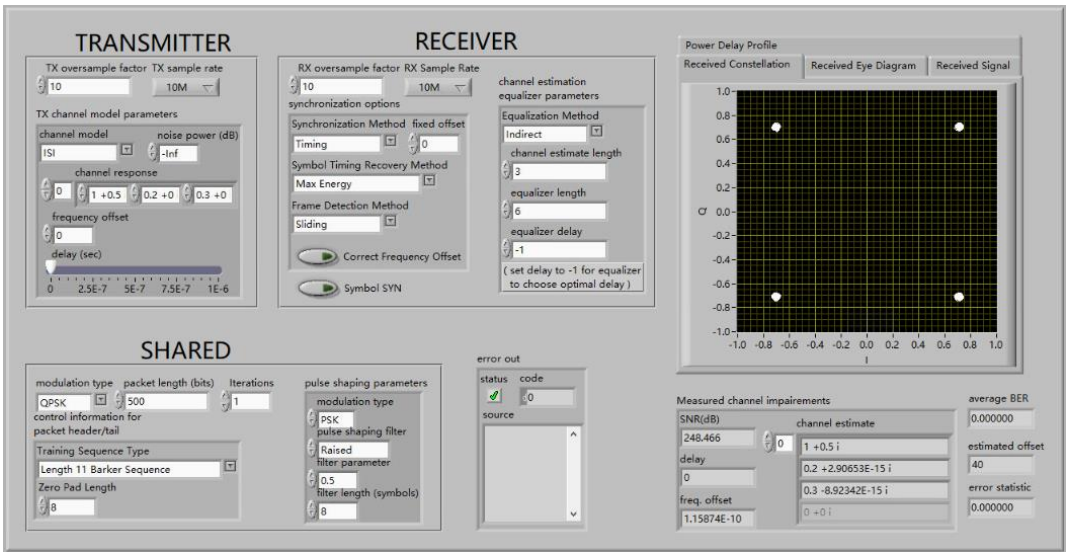
Length = 4:



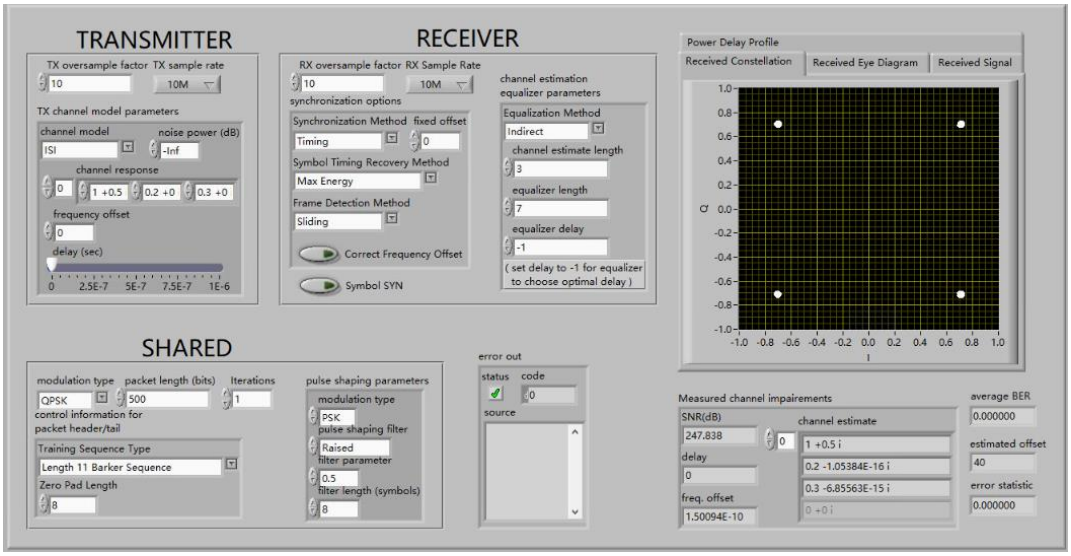
Length = 5:



Length = 6:



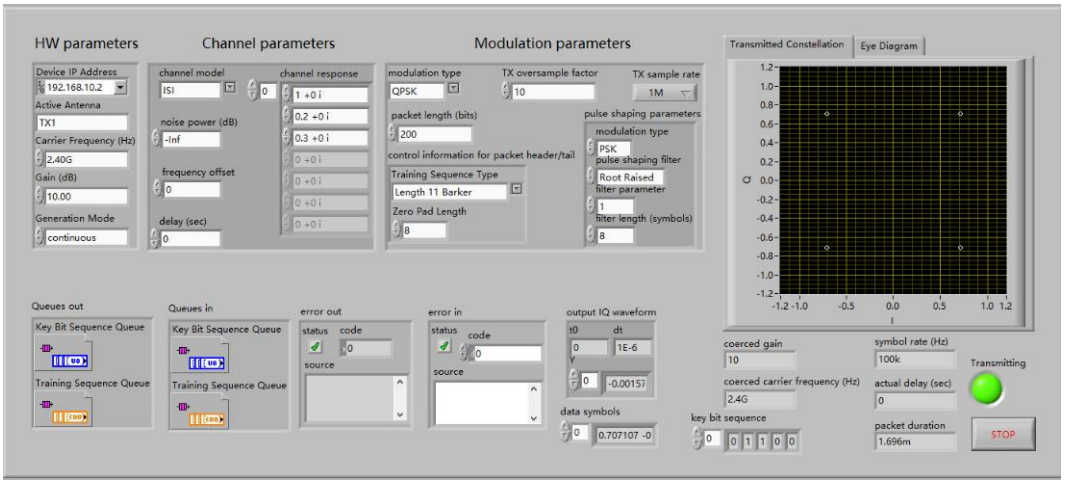
Length = 7:



From the results in the above figure, it can be observed that as the equalizer length increases, the constellation points become more convergent rather than divergent, and the results of channel estimation become more stable. Therefore, the longer the equalizer, the better the performance. Moreover, after the equalizer length exceeds 7, the performance becomes quite good, with little noticeable difference.

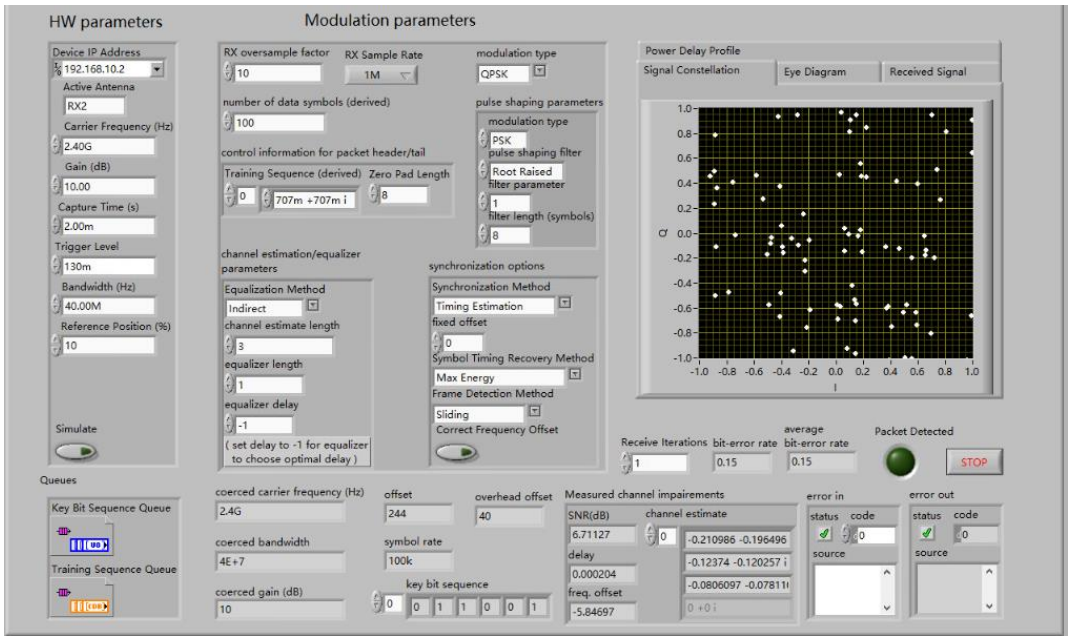
USRP Verification of Indirect Equalization

The TX:

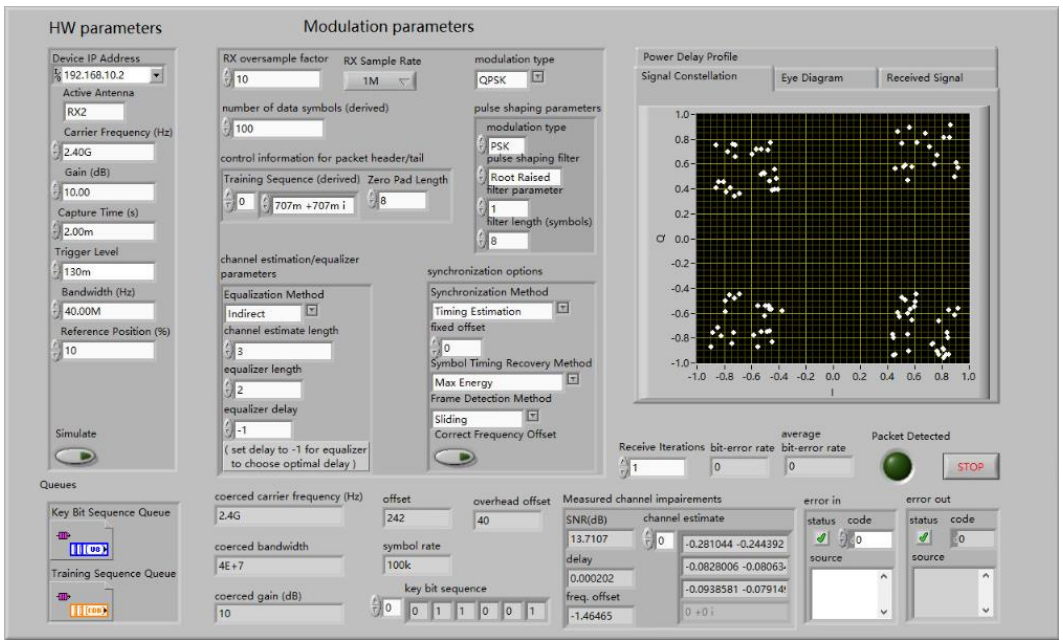


The RX:

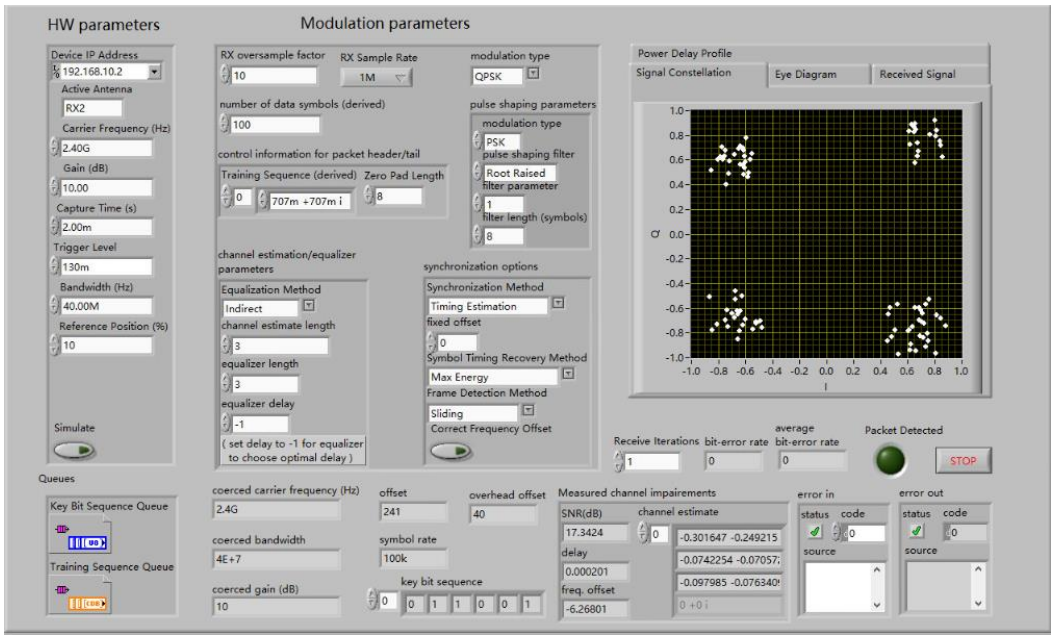
Length = 1:



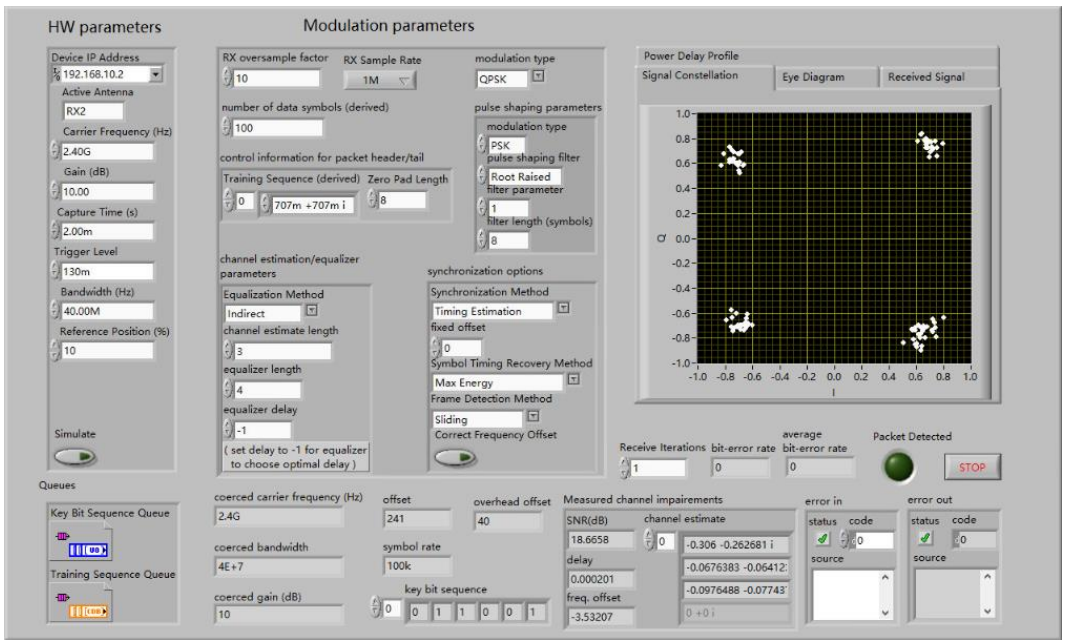
Length = 2:



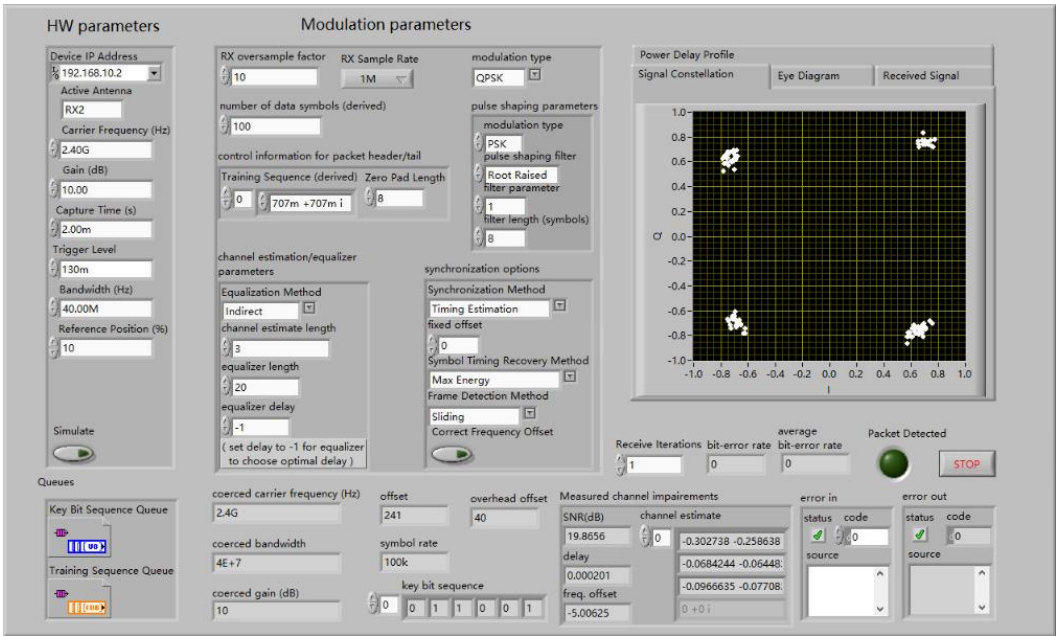
Length = 3:



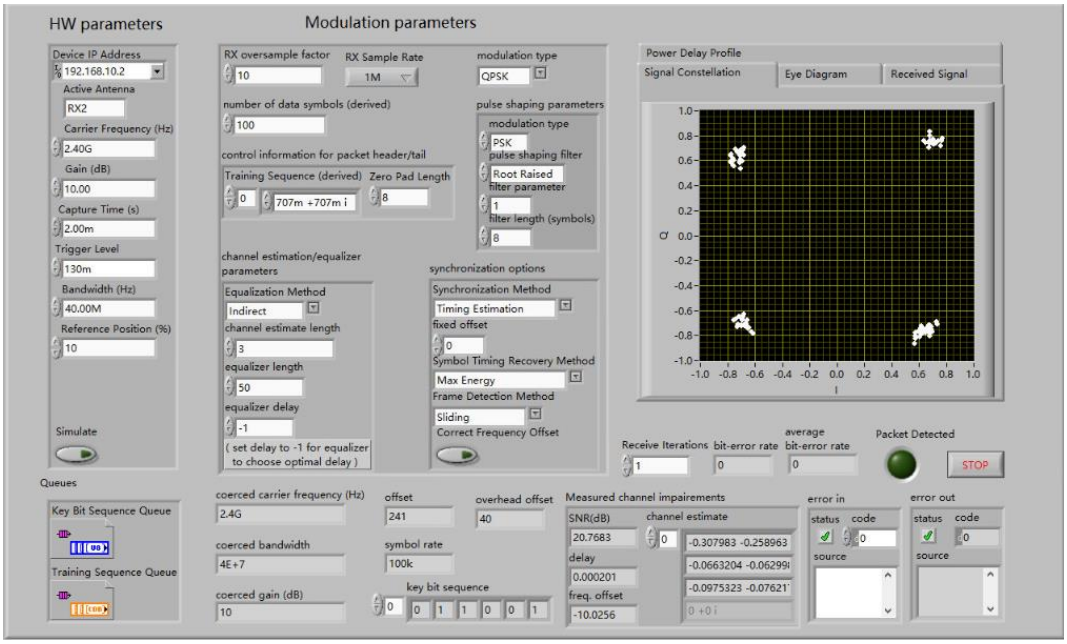
Length = 4:



Length = 20:



Length = 50:



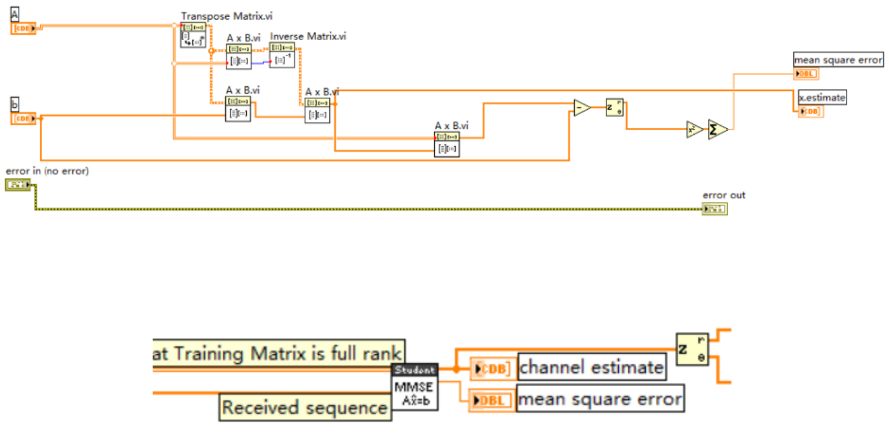
From the above figure, it can be seen that in the USRP implementation, as the length of the equalizer increases, its equalization performance gradually improves. Due to real-world environmental influences, even when the equalizer length reaches 50, the constellation points do not converge perfectly as in the simulation. However, it can be observed that there is not much difference in performance from a length of 20 to 50, indicating that further increasing the equalizer length does not bring additional gains. Therefore, an equalizer length of around 10 is sufficient to meet the requirements.

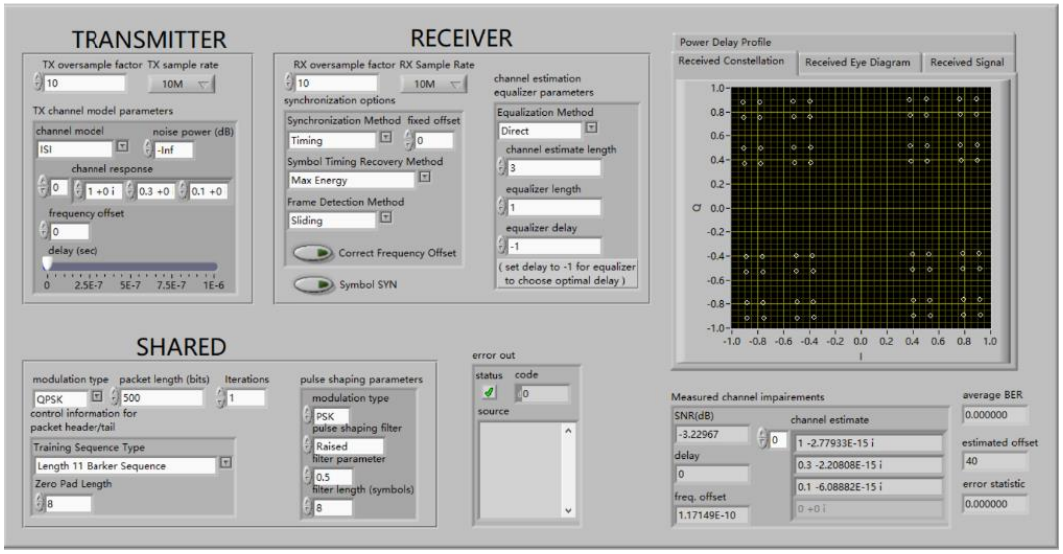
Another related observation is that as the equalizer length increases, the SNR value also increases, indicating that increasing the equalizer length can enhance SNR and improve communication quality. However, the effect is not too pronounced, and the overall SNR

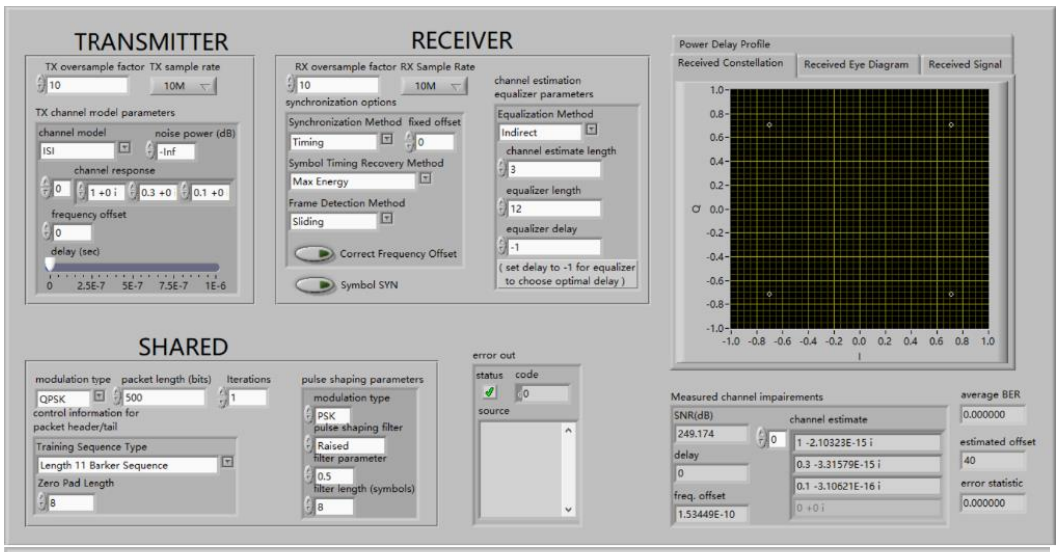
does not increase significantly. Moreover, with a sufficiently long equalizer length, the improvement in SNR becomes almost negligible.

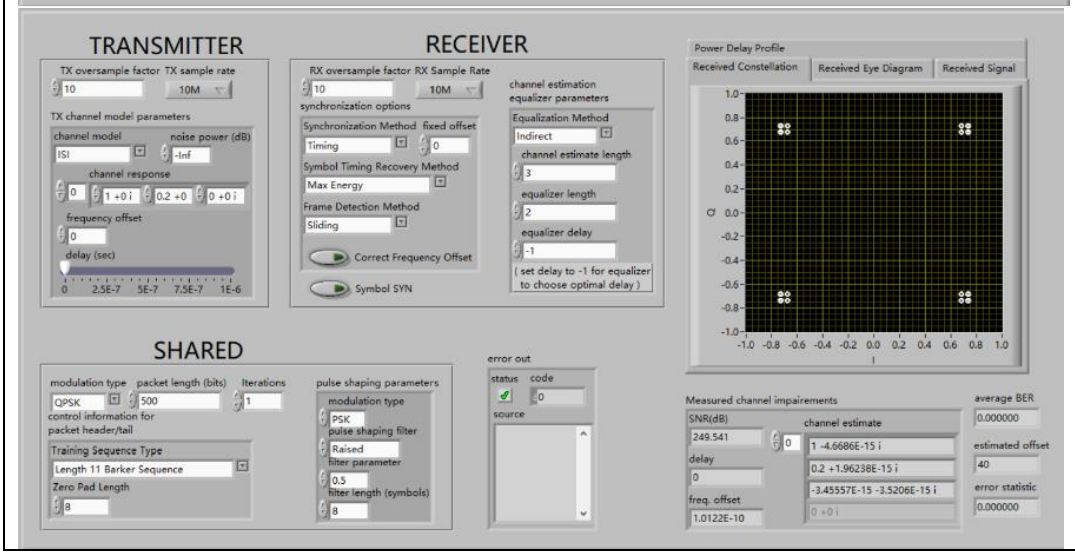
Experience

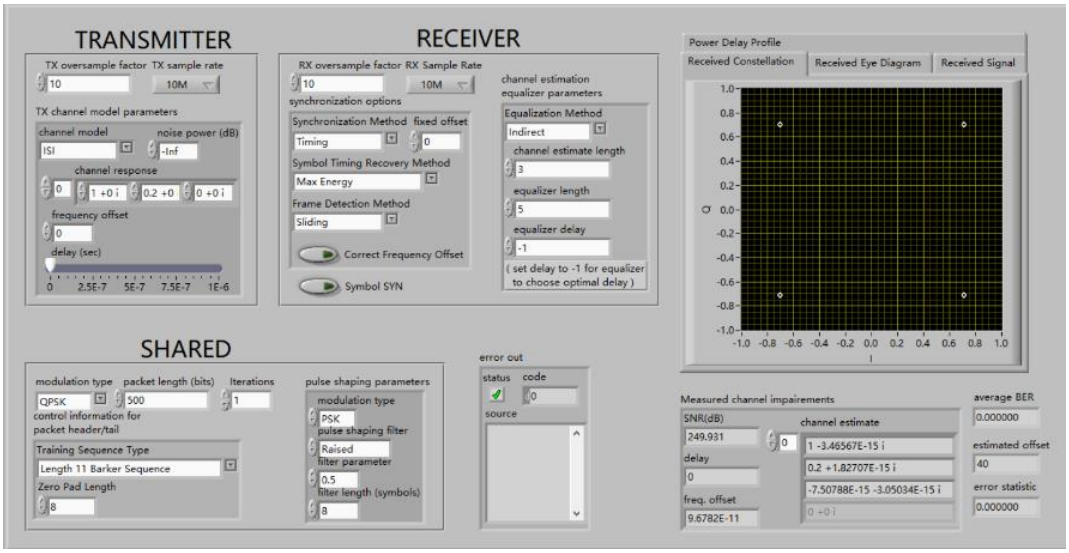
1. In-class lab screenshot




TRANSMITTER
TX oversample factor
10
TX sample rate
10M
TX channel model parameters
channel model
ISI
noise power (dB)
-Inf
channel response
0
1 +0 i
0.3 +0
0.1 +0
frequency offset
0
delay (sec)
0
2.5E-7
5E-7
7.5E-7
1E-6
RECEIVER
RX oversample factor
10
RX Sample Rate
10M
synchronization options
Synchronization Method
Timing
fixed offset
0
Symbol Timing Recovery Method
Max Energy
Frame Detection Method
Sliding
Correct Frequency Offset
Symbol SYN
channel estimation equalizer parameters
Equalization Method
Direct
channel estimate length
3
equalizer length
1
equalizer delay
-1
(set delay to -1 for equalizer to choose optimal delay)
SHARED
modulation type
QPSK
packet length (bits)
500
Iterations
1
control information for packet header/tail
Training Sequence Type
Length 11 Barker Sequence
Zero Pad Length
8
pulse shaping parameters
modulation type
PSK
pulse shaping filter
Raised
filter parameter
0.5
filter length (symbols)
8
error out
status
code
0
source
Power Delay Profile
Received Constellation
Received Eye Diagram
Received Signal
Measured channel impairements
SNR(dB)
-3.22967
channel estimate
0
1 -2.77933E-15 i
0.3 -2.20808E-15 i
0.1 -6.08882E-15 i
0 +0 i
delay
0
freq. offset
1.17149E-10
average BER
0.000000
estimated offset
40
error statistic
0.000000

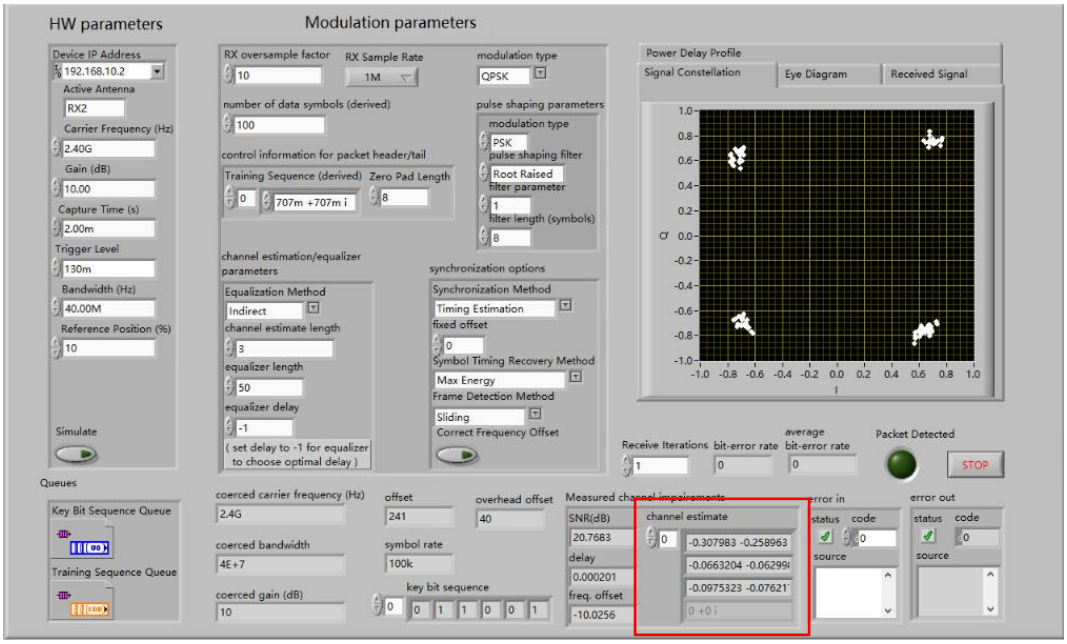

TRANSMITTER
TX oversample factor
10
TX sample rate
10M
TX channel model parameters
channel model
ISI
noise power (dB)
-Inf
channel response
0
1 +0 i
0.3 +0
0.1 +0
frequency offset
0
delay (sec)
0
2.5E-7
5E-7
7.5E-7
1E-6
RECEIVER
RX oversample factor
10
RX Sample Rate
10M
synchronization options
Synchronization Method
Timing
fixed offset
0
Symbol Timing Recovery Method
Max Energy
Frame Detection Method
Sliding
Correct Frequency Offset
Symbol SYN
channel estimation equalizer parameters
Equalization Method
Indirect
channel estimate length
3
equalizer length
12
equalizer delay
-1
(set delay to -1 for equalizer to choose optimal delay)
SHARED
modulation type
QPSK
packet length (bits)
500
Iterations
1
control information for packet header/tail
Training Sequence Type
Length 11 Barker Sequence
Zero Pad Length
8
pulse shaping parameters
modulation type
PSK
pulse shaping filter
Raised
filter parameter
0.5
filter length (symbols)
8
error out
status
code
0
source
Power Delay Profile
Received Constellation
Received Eye Diagram
Received Signal
Measured channel impairements
SNR(dB)
249.174
channel estimate
0
1 -2.10323E-15 i
0.3 -3.31579E-15 i
0.1 -3.10621E-16 i
0 +0 i
delay
0
freq. offset
1.53449E-10
average BER
0.000000
estimated offset
40
error statistic
0.000000


TRANSMITTER
TX oversample factor
10
TX sample rate
10M
TX channel model parameters
channel model
ISI
noise power (dB)
-Inf
channel response
0
1 +0 i
0.2 +0
0 +0 i
frequency offset
0
delay (sec)
0
2.5E-7
5E-7
7.5E-7
1E-6
RECEIVER
RX oversample factor
10
RX Sample Rate
10M
synchronization options
Synchronization Method
Timing
fixed offset
0
Symbol Timing Recovery Method
Max Energy
Frame Detection Method
Sliding
Correct Frequency Offset
Symbol SYN
channel estimation equalizer parameters
Equalization Method
Indirect
channel estimate length
3
equalizer length
2
equalizer delay
-1
(set delay to -1 for equalizer to choose optimal delay)
SHARED
modulation type
QPSK
packet length (bits)
500
Iterations
1
control information for packet header/tail
Training Sequence Type
Length 11 Barker Sequence
Zero Pad Length
8
pulse shaping parameters
modulation type
PSK
pulse shaping filter
Raised
filter parameter
0.5
filter length (symbols)
8
error out
status
code
0
source
Power Delay Profile
Received Constellation
Received Eye Diagram
Received Signal
Measured channel impairements
SNR(dB)
249.541
channel estimate
0
1 -4.6686E-15 i
0.2 +1.96238E-15 i
-3.45557E-15 -3.5206E-15 i
0 +0 i
delay
0
freq. offset
1.0122E-10
average BER
0.000000
estimated offset
40
error statistic
0.000000



2. Problem we meet and experience

1. When conducting experiments with USRP, it's common to encounter situations where the received signal is mostly noise. This occurs because the trigger level is set too low, causing the noise to overwhelm the desired signal. To address this issue, one can increase the trigger level to some extent.
2. We observed that during simulations, setting certain values for the channel response can render the algorithm ineffective, preventing it from producing accurate channel estimates and stable, correct constellation points. This might be due to specific values causing the H matrix to become singular, leading to a significant difference between the results obtained through the least squares method and the ideal results.
3. When implementing on the USRP, we observed that the channel response set in TX is significantly different from the estimated channel, and the disparity is considerable. Initially, we thought there might be an issue with the program; however, it turns out that this is not the case. Since the USRP operates in a real signal propagation environment, its estimation reflects the actual channel, naturally differing from the manually set channel response. This is both correct and reasonable.



3. Respective contributions

We two finish the whole LabVIEW program together. Zhang Haodong complete procedure design including channel estimation, MMSE design and indirect equalizer algorithm. And then verify the indirect equalizer algorithm with USRP and analyze the performance. The introduction of basic principle in channel estimation, MMSE, indirect equalizer algorithm and construction of Toeplitz Matrix were elaborated by Song Yihang.

Score	98