Software-Defined Radio Final Project

PHY implementation of 802.11a using USRP SDRs & Labview Communications System Design Suite

Background - WiFi

- Wi-Fi does not stand for Wireless Fidelity - It is a misconception.
- Wi-Fi is simply a trademarked phrase that refers to *IEEE 802.11x*.
- The IEEE 802.11 standard consists of a series of technological advances that have been developed over many years.
- Each new advancement is defined by an amendment to the standard that is identified by a one or two letter suffix to "802.11".

IEEE 802.11

- IEEE 802.11 is part of the IEEE 802 set of LAN protocols, and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) Wi-Fi computer communication in various frequencies, including but not limited to 2.4, 5, and 60 GHz frequency bands.

Development

- The original 802.11 standard allowed up to 2 Mbps on only the 2.4 GHz band.
- 802.11b added new coding schemes to increase throughput to 54 Mbps.
- 802.11g brought OFDM from 802.11a to the 2.4 GHz band.
- 802.11n added an assortment of high throughput advances to increase throughput roughly 10 times, such that high-end enterprise access points achieve signalling throughputs of 450 Mbps.

Development

- The current 802.11ac standard exceeds 1 Gbps of throughput.
- The popular standards in use now are 802.11a, 802.11b, 802.11g and 802.11n, 802.11ac
- The newest standard, 802.11ax, is the latest and fastest standard.

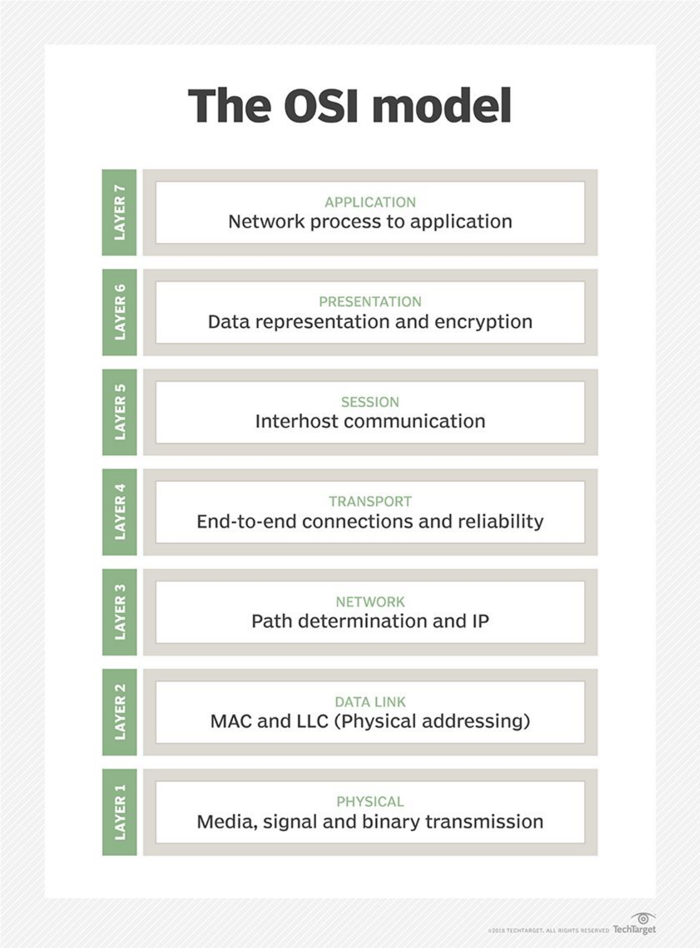
IEEE 802.11a

- 802.11a is an extension of the original 802.11 specification developed by the IEEE for wireless LAN (WLAN) technology.

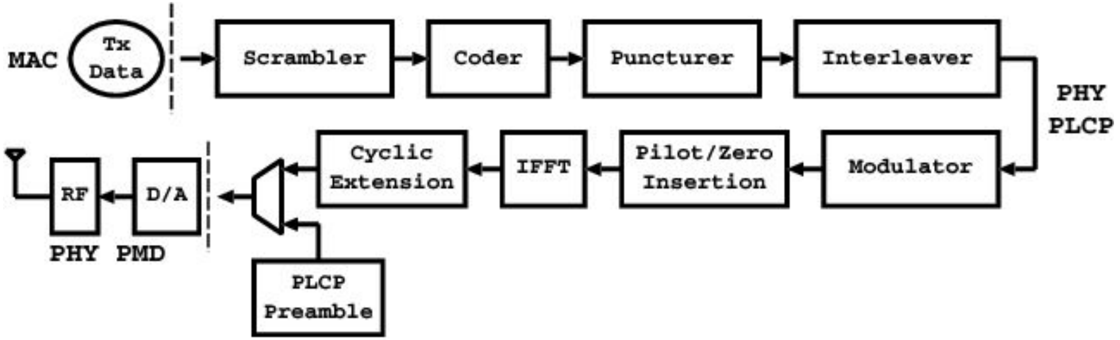
- The 802.11a specification uses an Orthogonal Frequency Division Multiplexing encoding scheme as opposed to FHSS or DSSS used by the original 802.11, and provides up to 54 Mbps in the 5GHz band.

The OSI Model

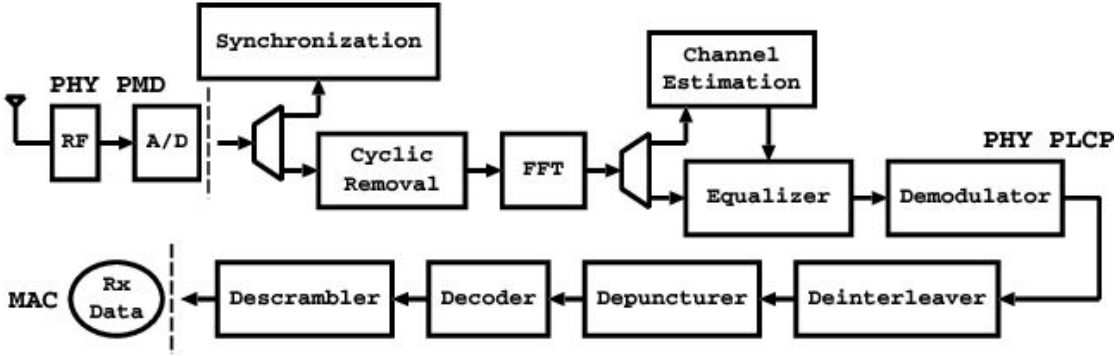
OSI: Open Systems Interconnection



The Physical Layer (802.11a)

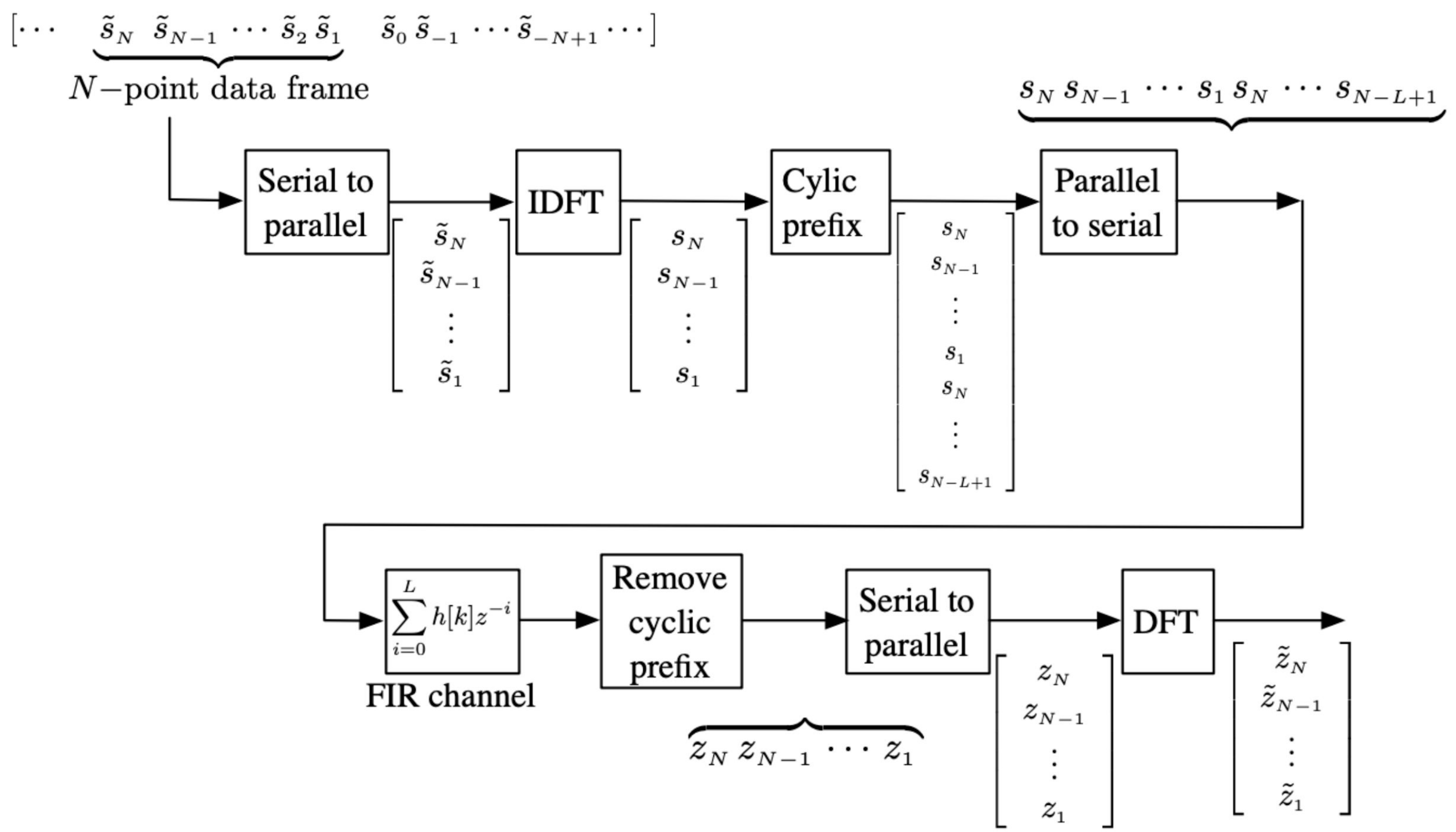


(a) IEEE 802.11a PHY transmitter



(b) IEEE 802.11a PHY receiver

OFDM



$[\cdots \quad \underbrace{\tilde{s}_N \ \tilde{s}_{N-1} \cdots \tilde{s}_2 \ \tilde{s}_1}_{N\text{-point data frame}} \quad \tilde{s}_0 \ \tilde{s}_{-1} \cdots \tilde{s}_{-N+1} \cdots]$
Serial to parallel
IDFT
Cylic prefix
Parallel to serial
$\overbrace{s_N \ s_{N-1} \cdots s_1 \ s_N \cdots s_{N-L+1}}$
$\sum_{i=0}^{L} h[k] z^{-i}$
FIR channel
Remove cyclic prefix
$\underbrace{z_N \ z_{N-1} \cdots z_1}$
Serial to parallel
DFT

Implementation

Symbol Timing Recovery

Time delay exists:

- When td = dT (0<d<1), there can be sample timing error.

  (ISI is introduced because the Nyquist pulse shape is not sampled at nT)

- When td = nT, mismatch will be introduced between the indices of transmitted and received symbols. Frame synchronization will be needed to solve this problem.

Max Energy & Early-Late gate

Output energy function defined as the following:

$$J(\tau) = \mathbb{E}\left|y\left(nT+\tau\right)\right|^2 = E_x \sum_m \left|g\left(mT+\tau-\tau_d\right)\right|^2 + \sigma_v^2$$
$$\leq E_x |g(0)|^2 + \sigma_v^2.$$

We want to find an optimum solution such that the output energy function is maximized. The maximum of the function J(T) occurs when t-td is an integer multiple of the symbol rate.

Max Energy Method

- Here, we try to find the sample point that maximizes the average received energy from the output of the matched receiver filter that is given by:

$$r[n] = \sum_{m} z[m] g_{\mathrm{tx}}[n-m]$$

- The discrete-time output energy can be calculated as follows:

$$J[k] = \mathbf{E} \left| r\left[nMT + k\right] \right|^2$$

Max Energy Method

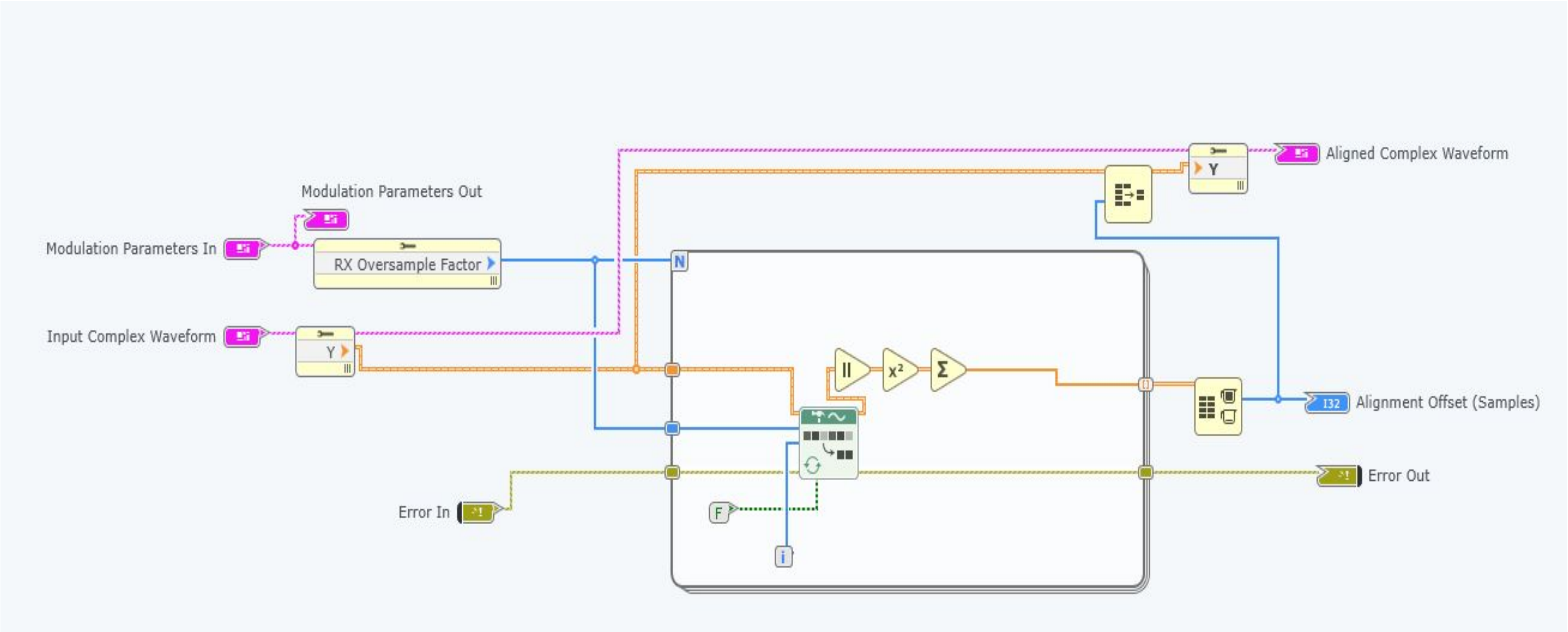
- For practical implementation, we can replace the expectation with a time average over ‘P’ symbols to create the function:

$$J_{\text{approx}}[k] = \frac{1}{P} \sum_{p=0}^{P-1} \left| r\left[pM + k\right] \right|^2 .$$

- This is a fairly reasonable approximation. Larger values of ‘P’ generally give better performance.

Max Energy implementation

$$J_{\text{approx}}[k] = \frac{1}{P} \sum_{p=0}^{P-1} |r\,[pM + k]|^2 \,.$$



Early-Late Gate

Late-early symbol timing recovery

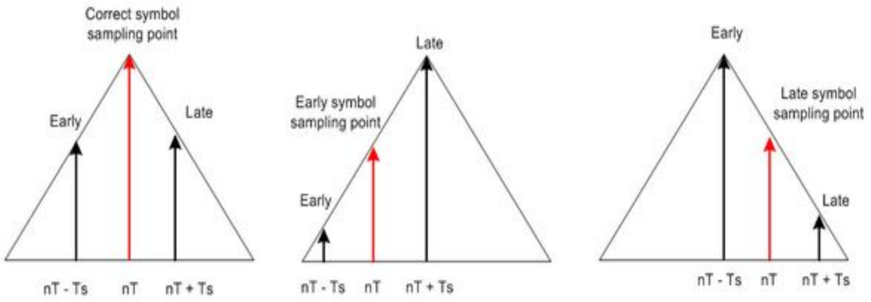


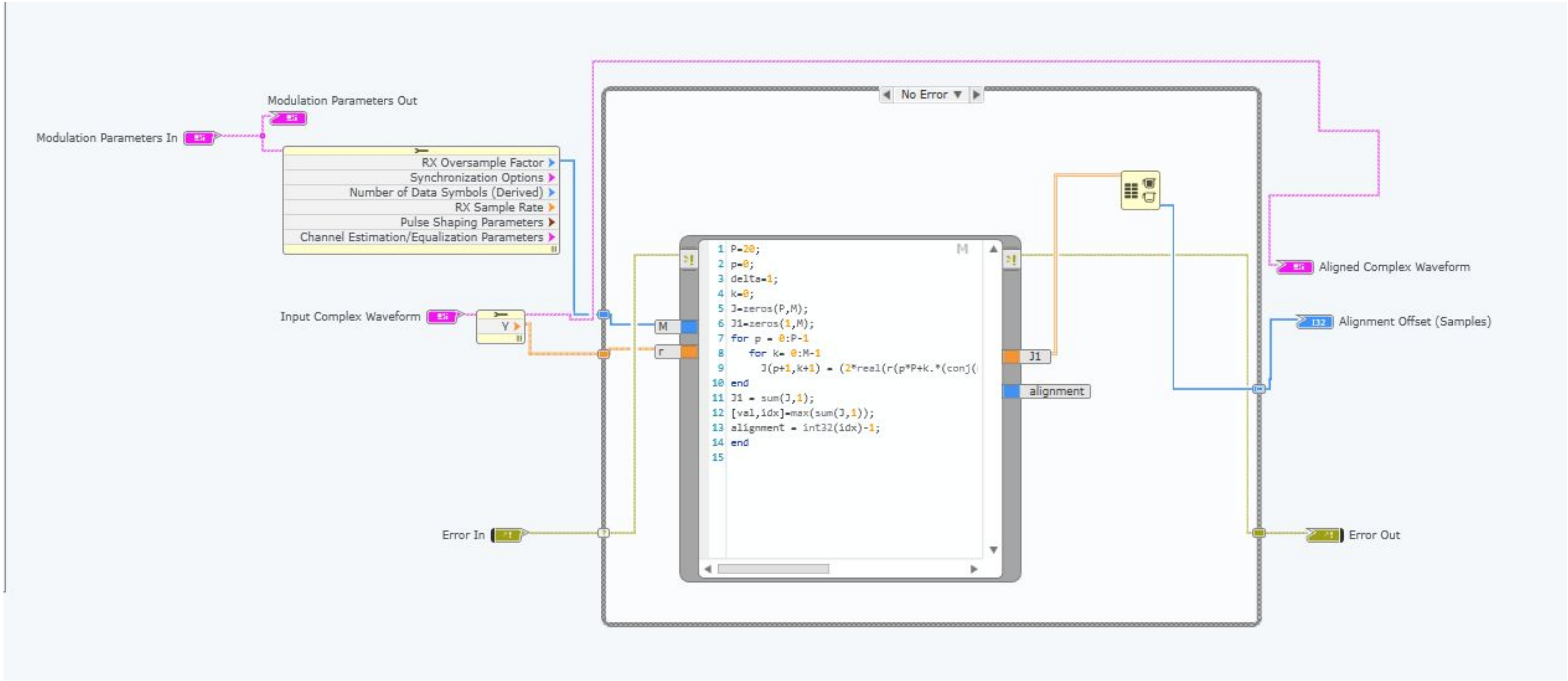
Figure 1: Late-early timing error computation

When the signal is early, it will be sampled at nT-Ts. When the signal is late, it will be sampled at nT+Ts. The difference between them is the timing error. Mathematically, it is as follows:

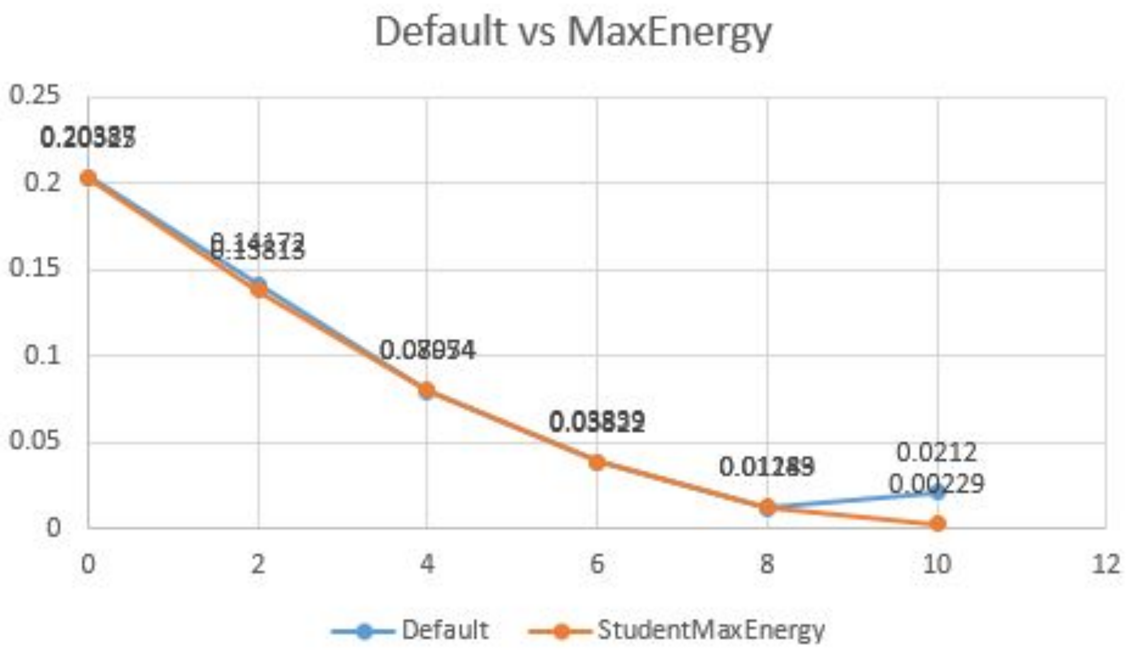
$$J_\delta[k] = \sum_{n=0}^{P-1} 2\mathrm{Re}\left\{r\left[nP+k\right]\left(r^*\left[nP+k+\delta\right] - r^*\left[nP+k-\delta\right]\right)\right\}. \quad (4)$$

The goal is to minimize this.

Early-Late gate implementation

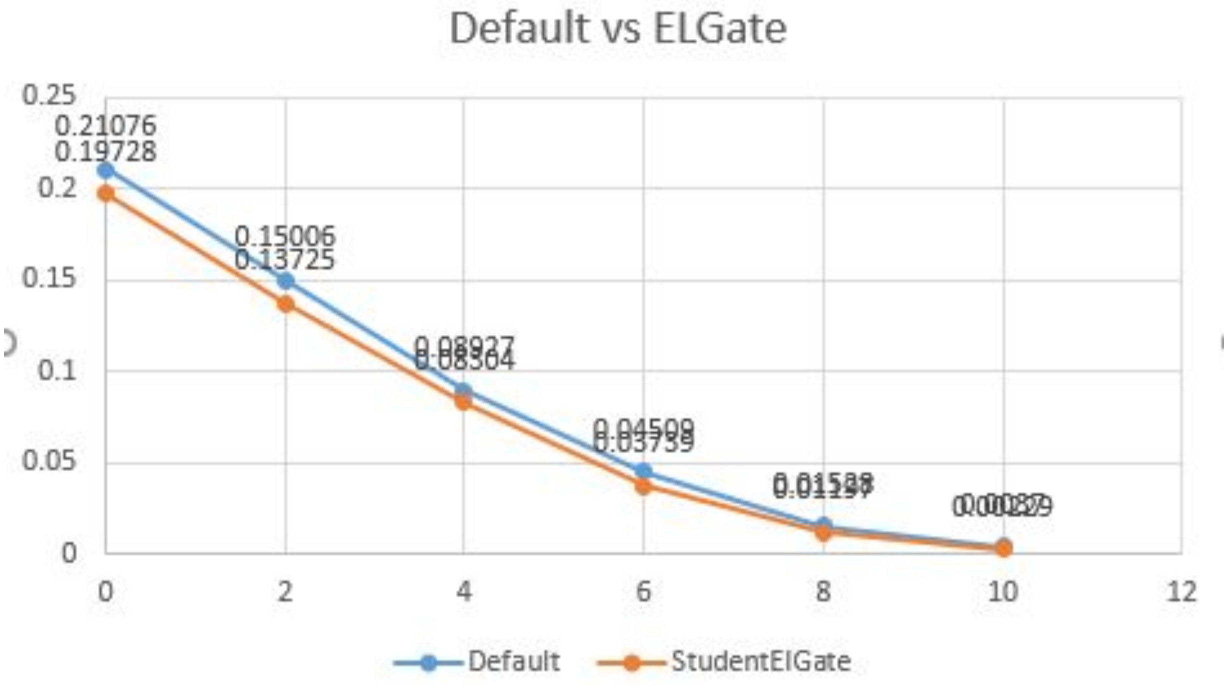


Results

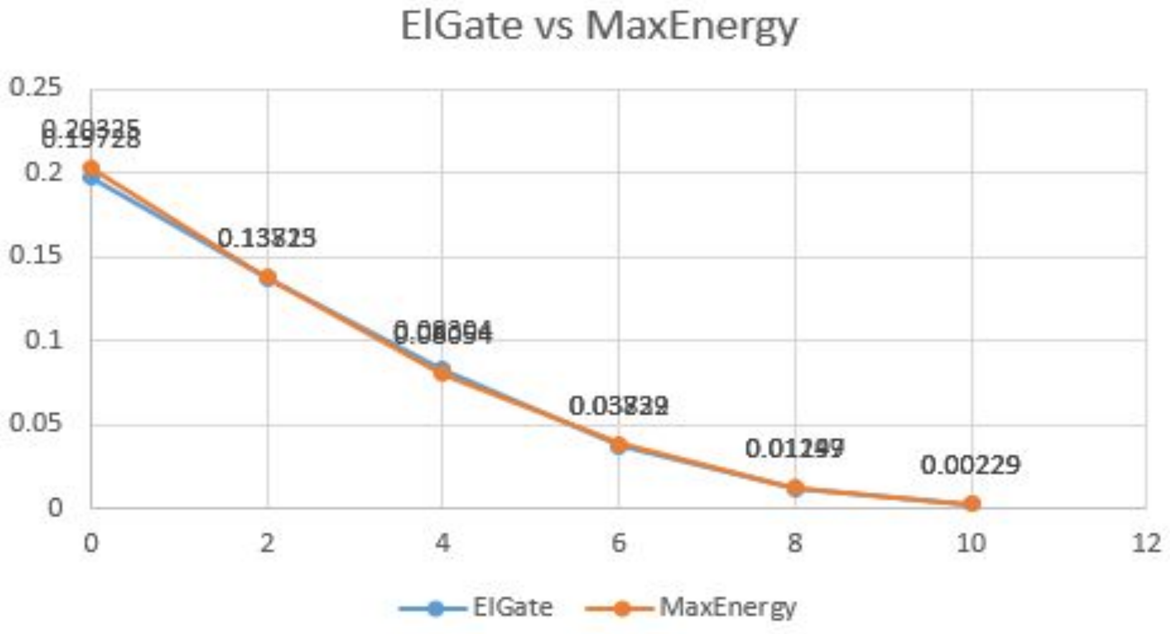


Default vs MaxEnergy
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
12
0.0212
0.00229
Default
StudentMaxEnergy

Results


Default vs ELGate
0.25
0.2
0.15
0.1
0.05
0
0.21076
0.19728
0.15006
0.13725
0
2
4
6
8
10
12
Default
StudentElGate

Results - Comparison



Verdict: ELGate is the winner!

Frame detection & Frequency offset correction

We introduce the training sequence in this case to help with synchronization and channel estimation.

Frame synchronization is used to resolve the multiple symbol delays. Under this topic, we built two blocks: Sliding correlator and Moose. Sliding correlator is used for frame detection, and Moose helps to detect the frequency offset and then correct it.

The Moose subroutine is invoked in the Sliding Correlator VI.

Sliding Correlator

- Training sequence - Barker codes (aperiodic):

  $\{a_k\}_{k=1}^{N_t}$ -> sequence of values ±1, such that $\left| \sum_{i=1}^{N_t-k} a_i a_{i+k} \right| \leq 1$

- We use four length-11 Barker sequences, concatenated, as the training sequence.

- Therefore, the training sequence is 44 symbols long.

Code Length	Barker Sequence
2	$[-+, --]$
3	$[- - +]$
4	$[- + --, - + ++]$
5	$[- - - + -]$
7	$[- - - + + - +]$
11	$[- - - + + + - + + - +]$
13	$[- - - - - + + - - + - + -]$

Sliding Correlator

- Since the training sequence chosen has excellent correlation properties, it is useful in frame detection

$$R[n] = \left| \sum_{k=0}^{N_t-1} t^*[k] y[n+k] \right|^2$$

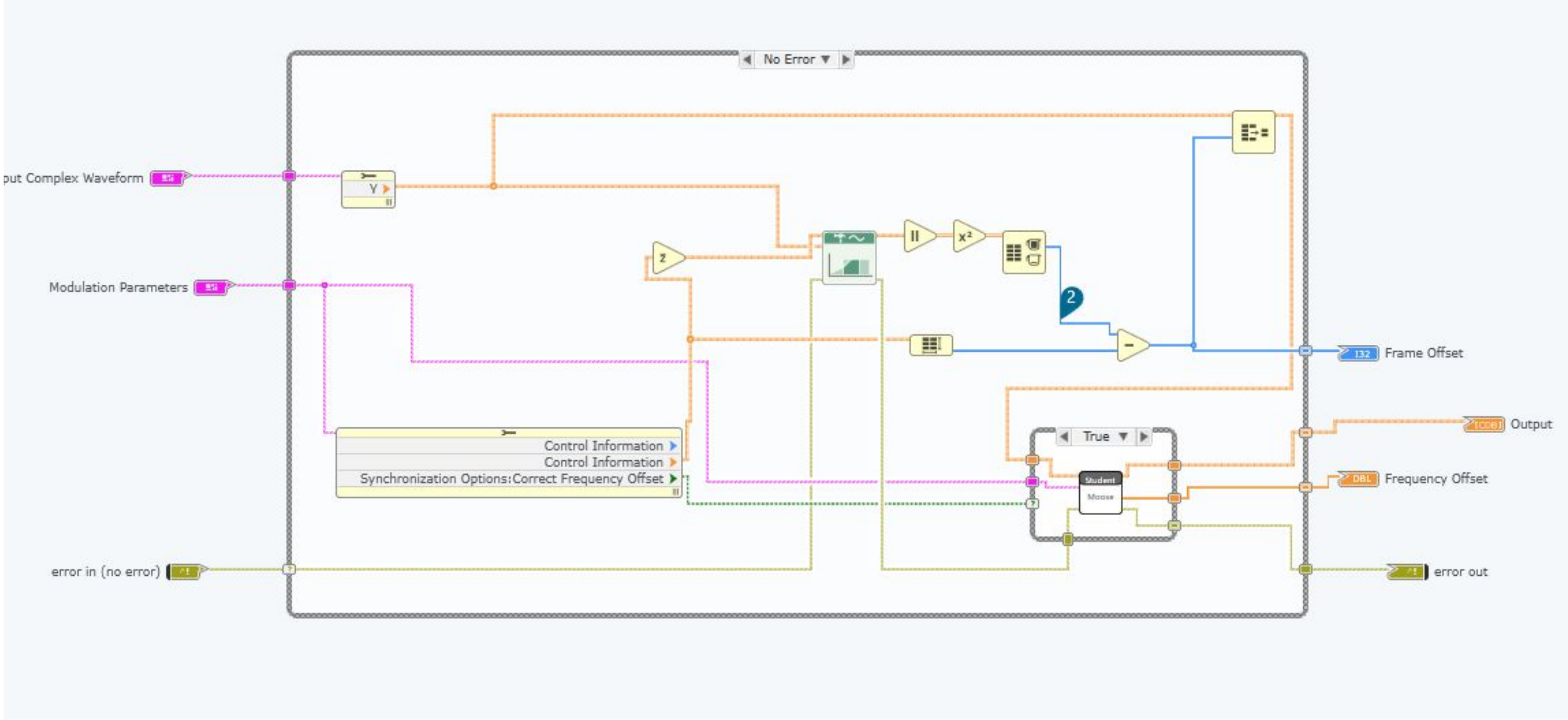
- We correlate the received sequence with the training sequence

- The index of the peak of the correlation function is nothing but the frame offset

$$\hat{d} = \operatorname*{argmax}_n R[n].$$

Sliding Correlator

$$R[n] = \left| \sum_{k=0}^{N_t - 1} t^*[k] y[n+k] \right|^2$$



Moose

The received signal after matched filtering and downsampling:

$$y[n] = e^{j2\pi\epsilon n}\sum_{l=0}^{L} h[l]s[n-l] + v[n],$$

or

$$\hat{\epsilon} = \frac{\text{phase}\sum_{l=L}^{N_t-1} y[l+N_t]y^*[l]}{2\pi N_t} \quad (8)$$

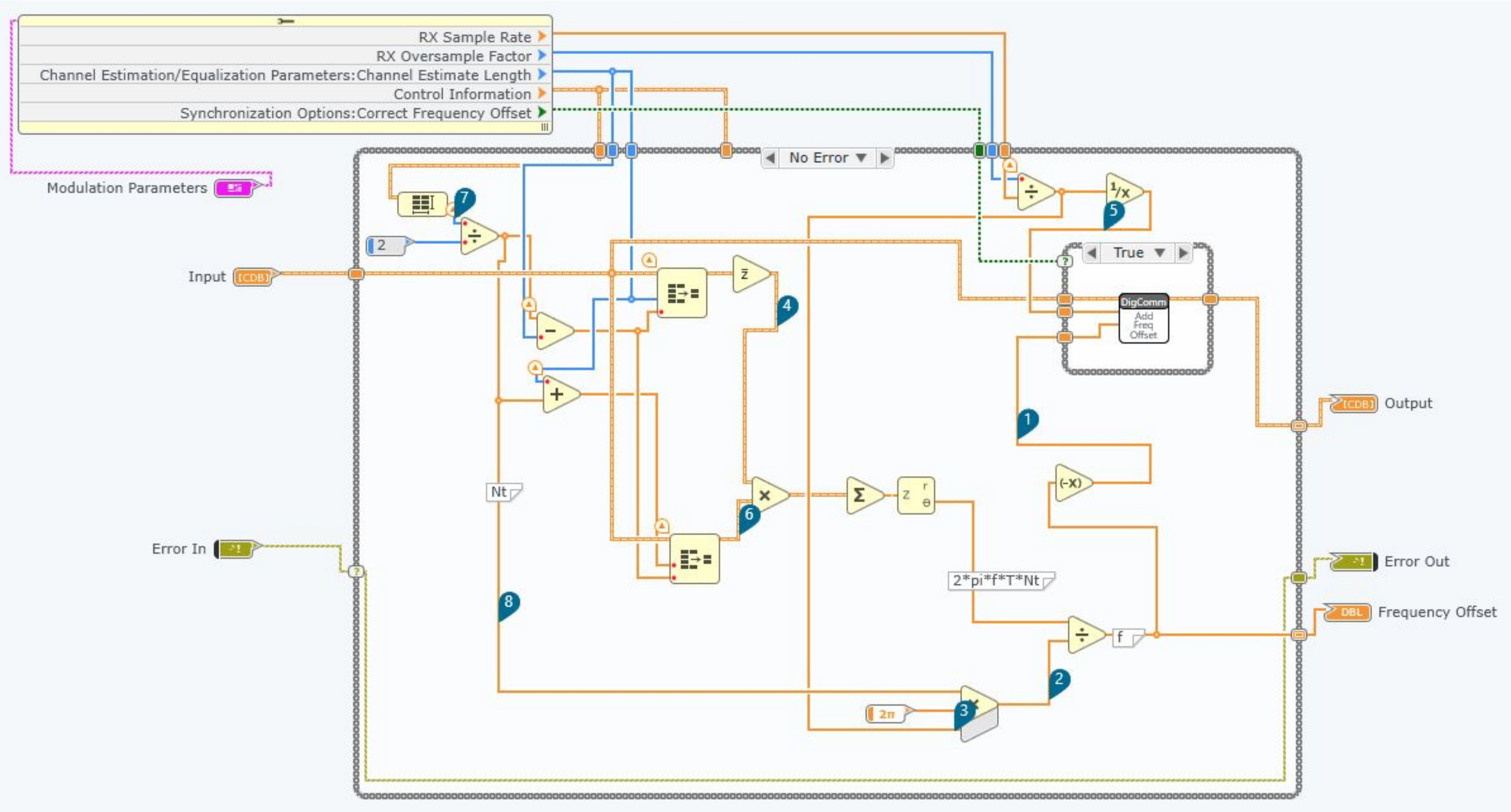
$$\hat{f}_e = \frac{\text{phase}\sum_{l=L}^{N_t-1} y[l+N_t]y^*[l]}{2\pi T N_t}, \quad (9)$$

To detect the frequency offset, the least square linear approximation method is used here. By doing that, we compute an estimate for the frequency offset.
We use the periodic training data here (IEEE 801.11a) and obtain the following formula:

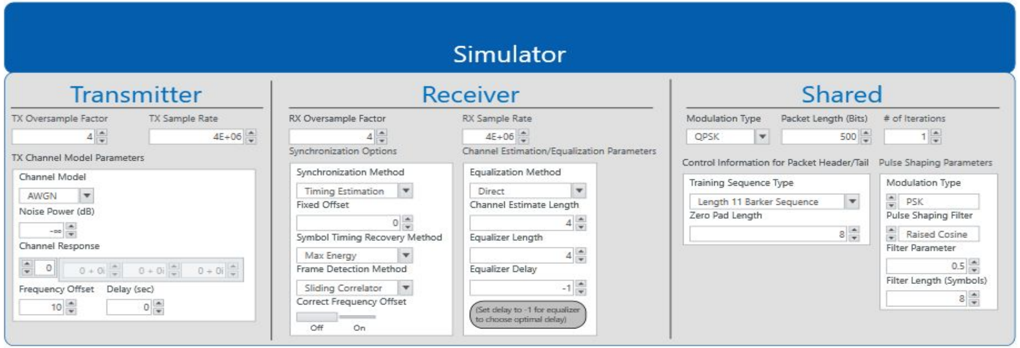
$$\begin{aligned} y[n+N_t] &= e^{j2\pi\epsilon N_t} e^{j2\pi\epsilon n}\sum_{l=0}^{L} h[l]t[n-l] + v[n+N_t] \\ &\approx e^{j2\pi\epsilon N_t} y[n]. \end{aligned}$$

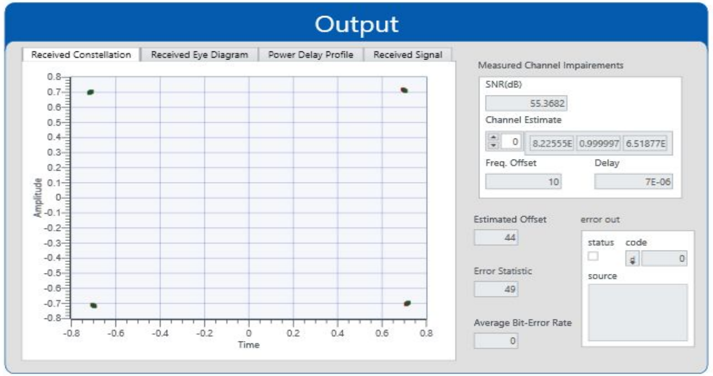
In order to correct the frequency offset, we replace the epsilon in the formula with the computed estimate.

Moose Implementation



Results for frame detection and offset correction

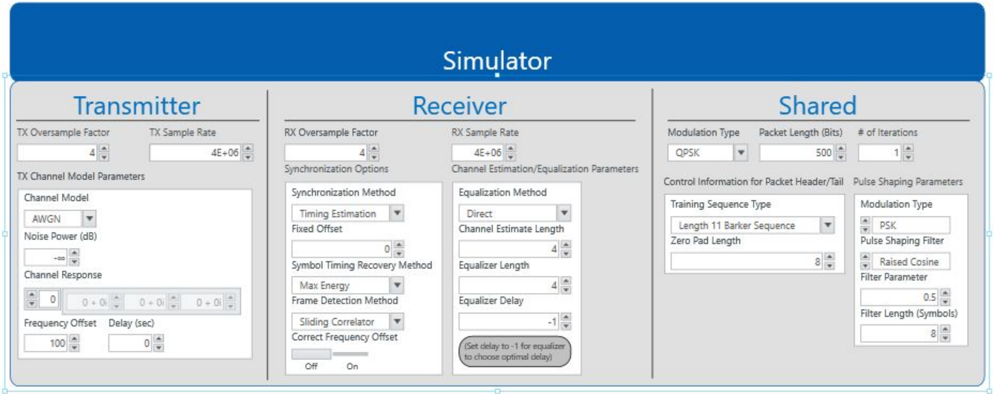


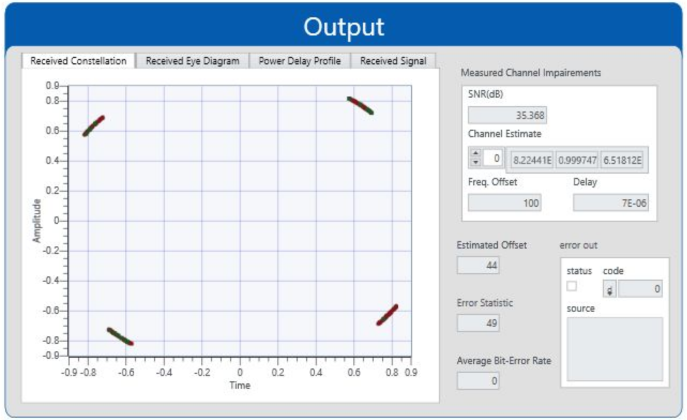


Offset 10

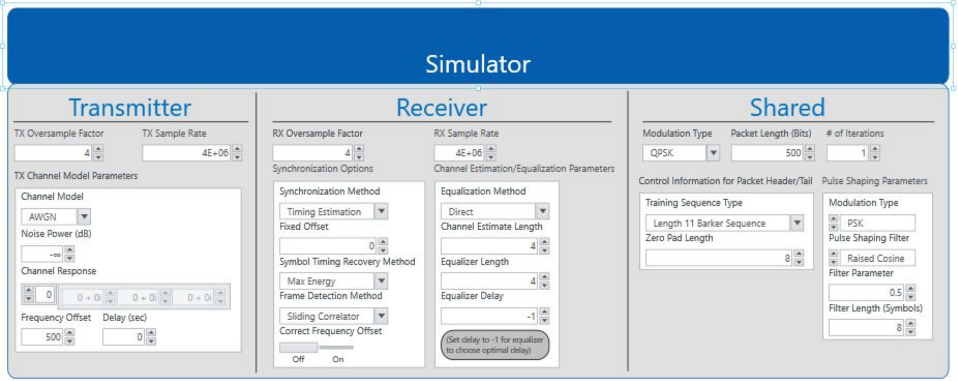
BER 0

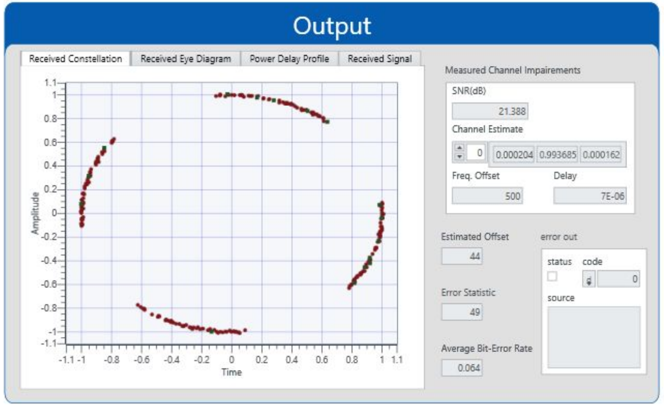
Offset 100; BER 0



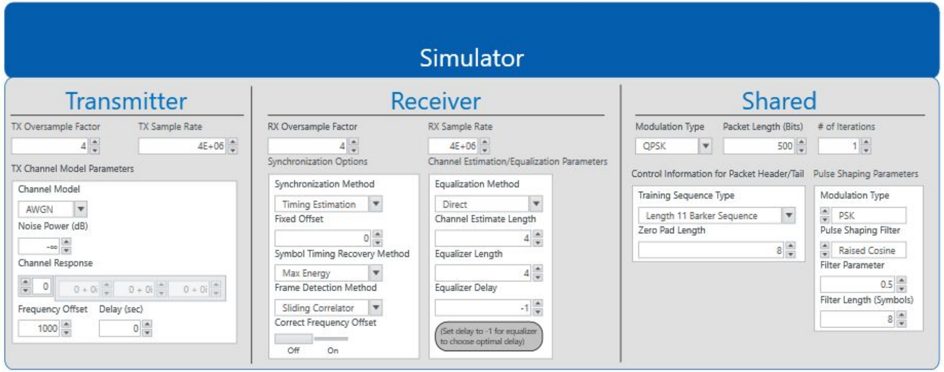


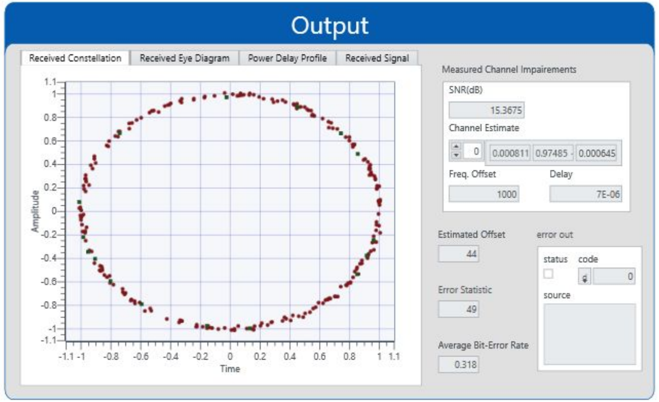
Offset 500; BER 0.064



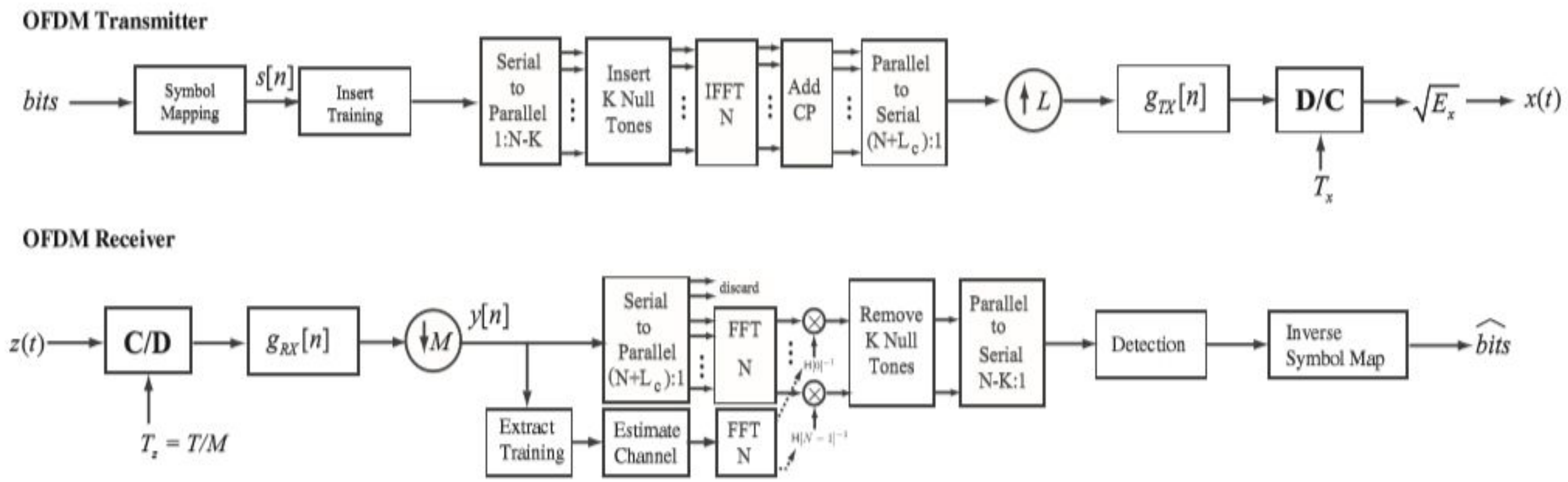


Offset 1000; BER 0.3

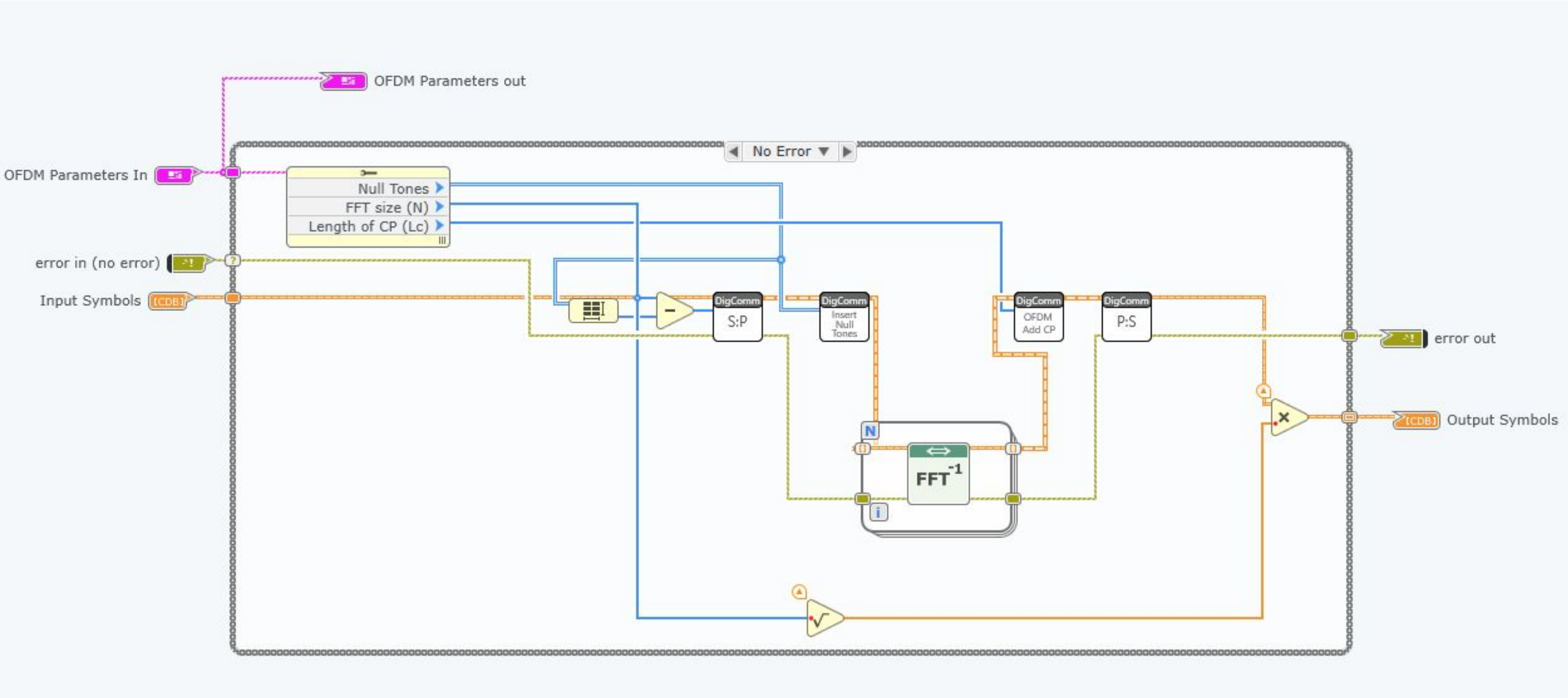




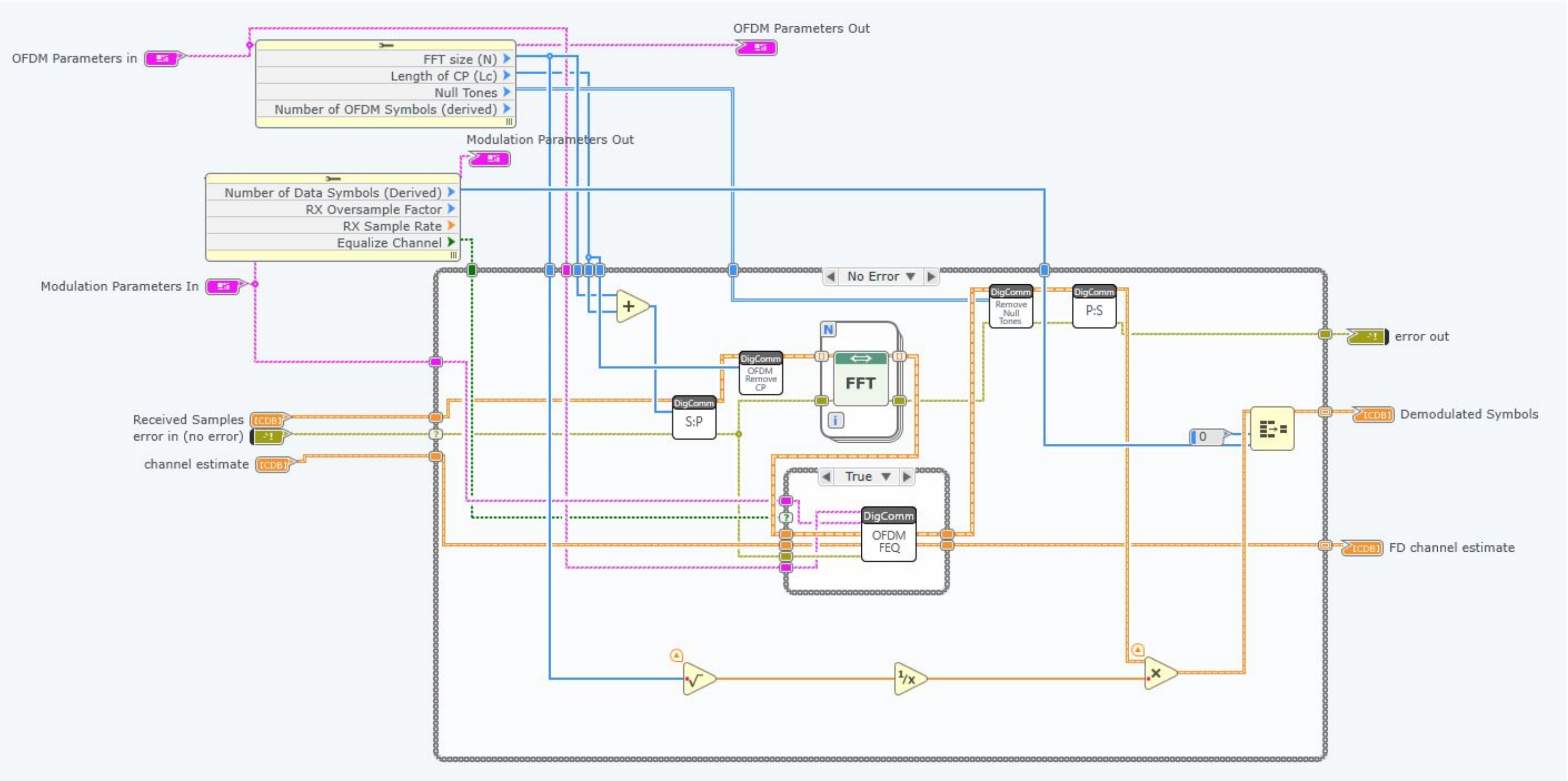
OFDM Modulator



OFDM modulator implementation



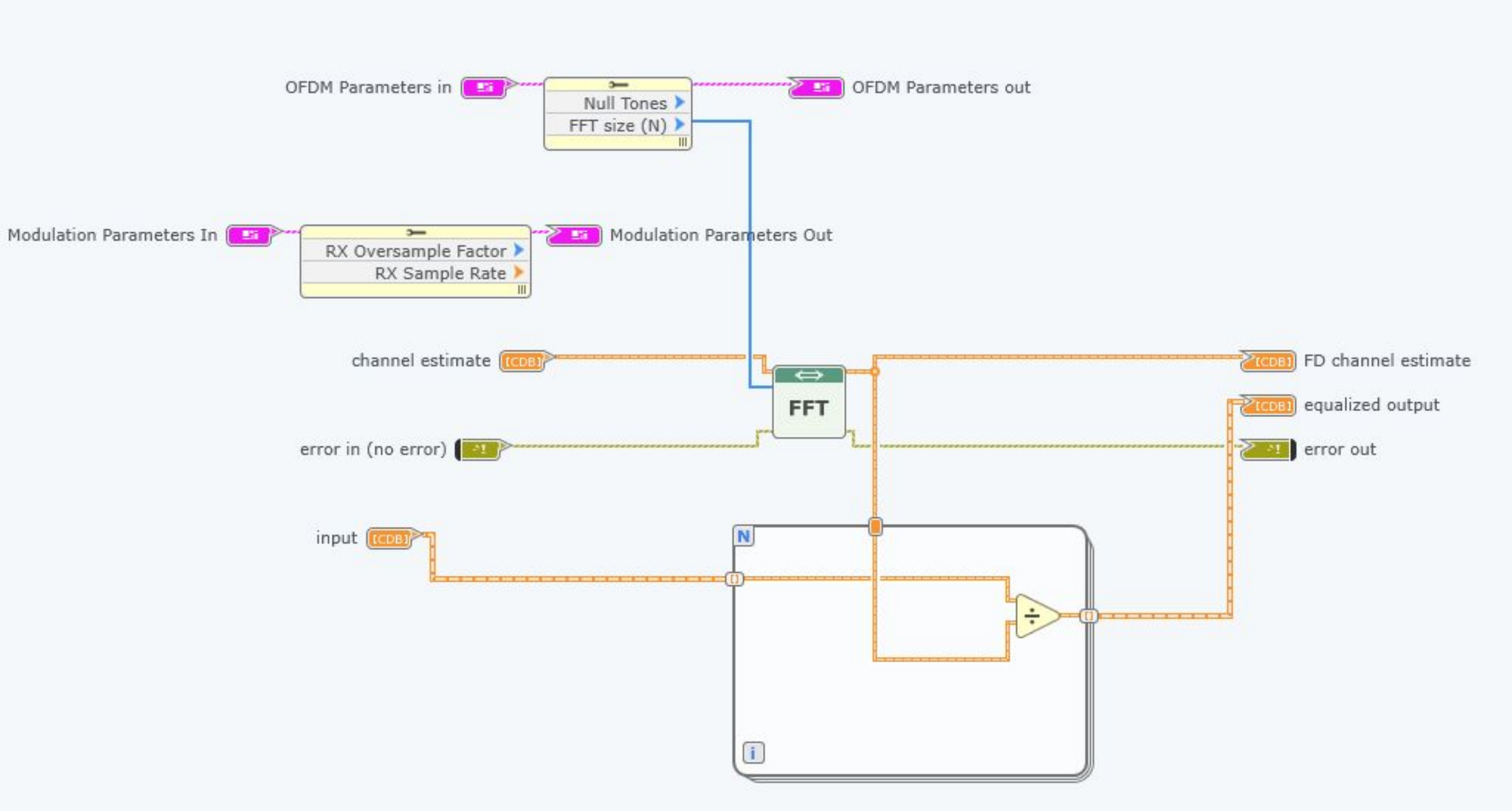
OFDM Demodulator implementation



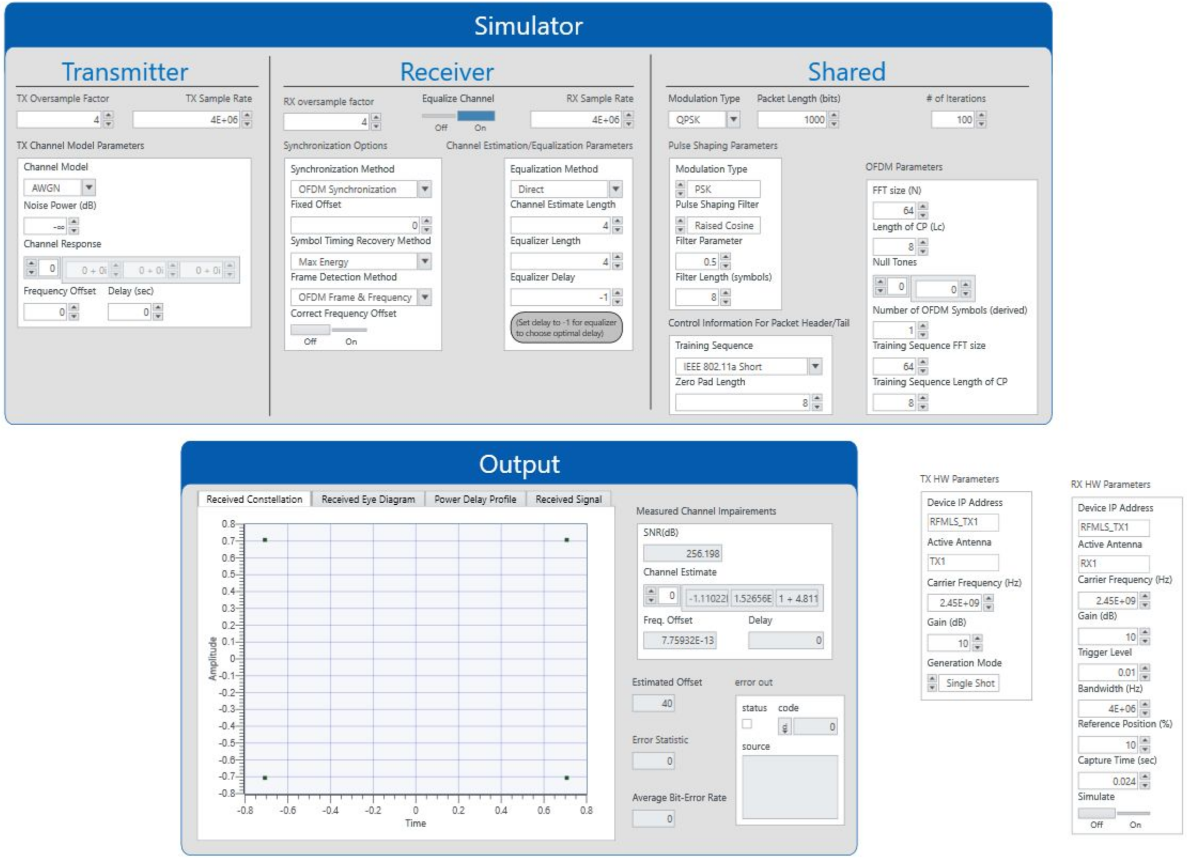
$$\widehat{H}[k] = DFT\{h[l]\})$$

OFDM FEQ implementation

$\widetilde{X}[k]\}_{k=0}^{N-1}$, where $\widetilde{X}[k] = Y[k]/\widehat{H}[k])$



OFDM result


Simulator
Transmitter
Receiver
Shared
Output
Received Constellation
Received Eye Diagram
Power Delay Profile
Received Signal
Measured Channel Impairements
SNR(dB)
256.198
Estimated Offset
40
Error Statistic
0
Average Bit-Error Rate
0
TX HW Parameters
RX HW Parameters

References

gate:https://www.nutaq.com/blog/symbol-timing-recovery-methods-digital-iq-demodulator

Heath Lab manual

Modern Digital and Analog Communication Systems Textbook