EEC 165: Final Project Report

Fundamental PHY Implementation of IEEE 802.11a
Using USRP Software Defined Radios
& LabVIEW Communications System Design Suite

Under the guidance of instructor Carlos Feres and Professor Zhi Ding

Abhinav Kamath
Junyi Tu

University of California, Davis

Introduction:

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". 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 the throughput. 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. The current 802.11ac standard exceeds 1 Gbps of throughput. The individual standards in use now are 802.11a, 802.11b, 802.11g and 802.11n (which uses a more advanced technology than the others). The newest standard, 802.11ax, is the latest and fastest standard.

IEEE 802.11a introduced the Orthogonal Frequency Division Multiplexing (OFDM) scheme. In this project, we implement a basic version of the physical layer of the 802.11a protocol and demonstrate the significance of OFDM, with real-time simulations and results.

Implementation of the physical layer of 802.11a:

Symbol timing recovery:

When transmitting a signal using hardware, there exists a time delay, ‘td’, between the transmitter and the receiver. When this ‘td’ is a fraction of the sampling period, that is, when td = dT where 0< d <1, there will be a sample timing error. This means that ISI will be introduced because the Nyquist pulse shape is not sampled at nT. Therefore, we implement symbol timing recovery.

There are two methods to overcome this: the Max-Energy method and the Early-Late Gate method.

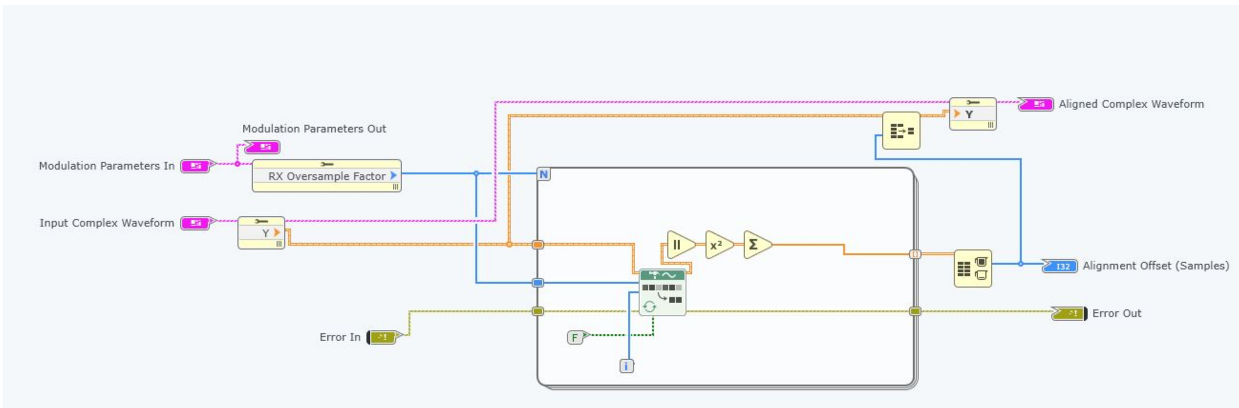
Max Energy:

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

Above is the formula for energy of the output signal of the matched filter at the receiver. In order to find the maximum energy, we approximate it as follows:

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

This is a fairly reasonable approximation. Larger values of ‘P’ generally give better performance. The implementation of the above function is as follows:



Early-Late Gate Method:

Late-early symbol timing recovery

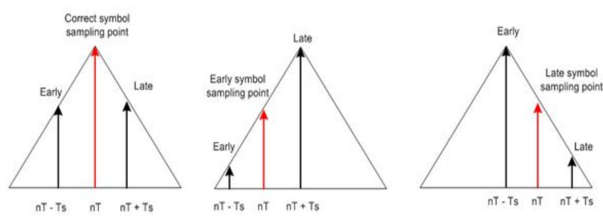
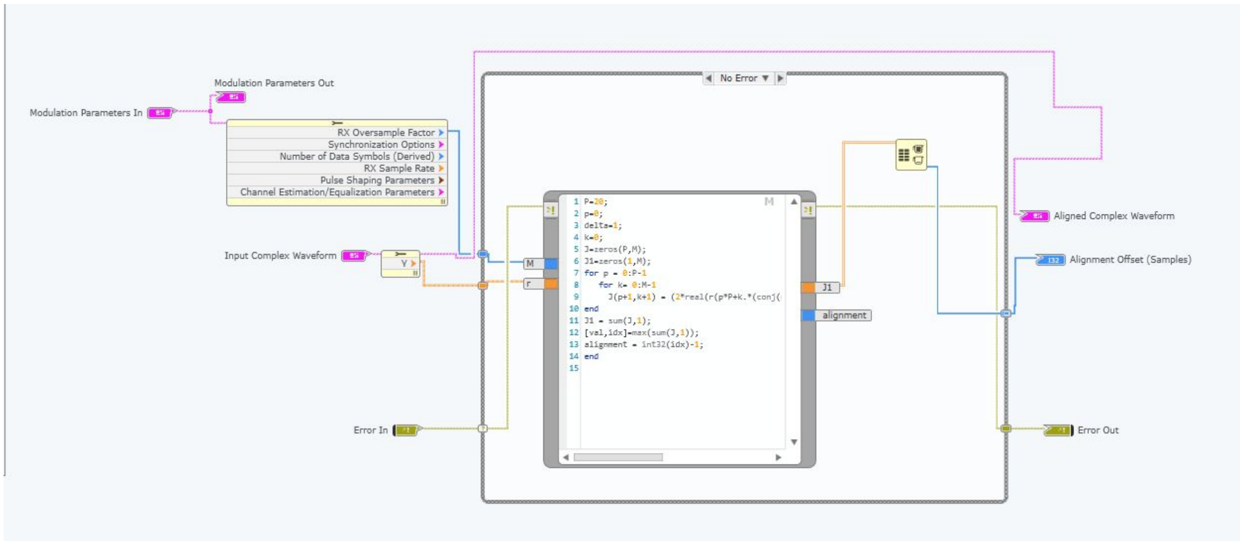


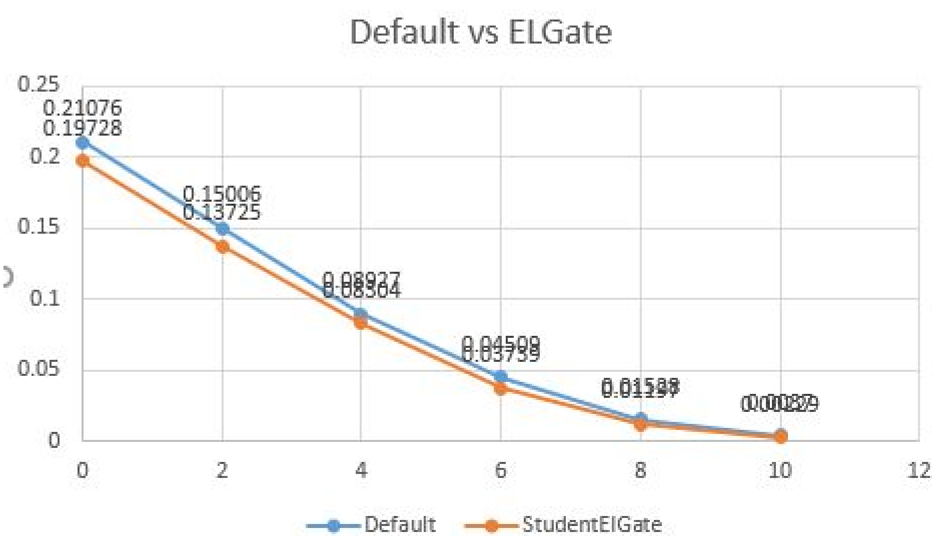
Figure 1: Late-early timing error computation

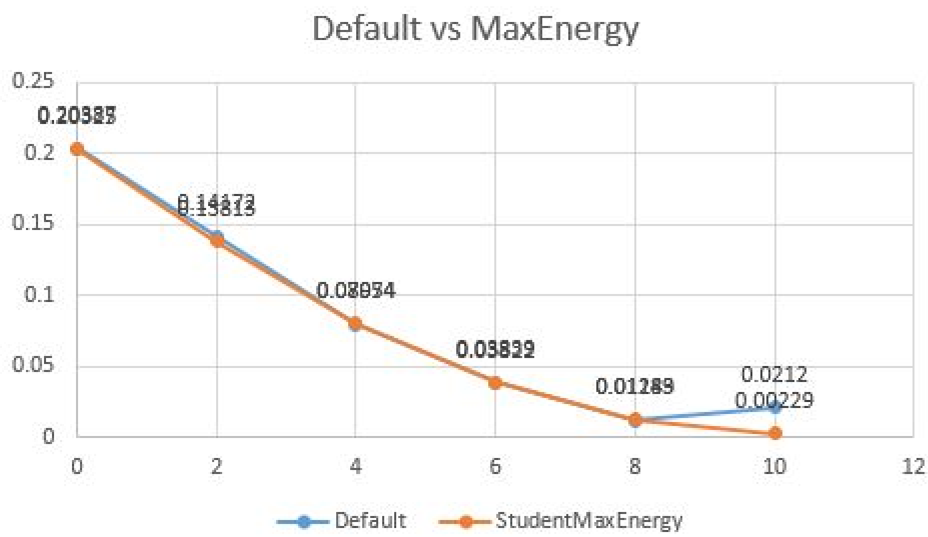
$$J_\delta[k] = \sum_{n=0}^{P-1} 2\mathrm{Re}\left\{r\,[nP+k]\,(r^*\,[nP+k+\delta] - r^*\,[nP+k-\delta])\right\}. \quad (4)$$

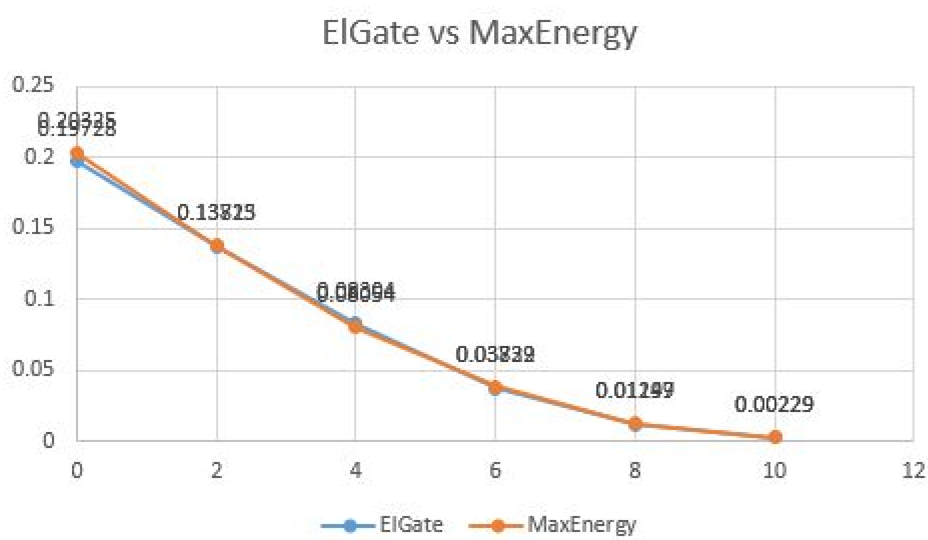
In the equation above, delta represent a small time offset value. It is equal to Ts shown in the first figure. r*[nP+k+delta] describes the case when the signal is being sampled at a delay of delta. r*[nP+k-delta] describes the case when the signal is being sampled early by a value of delta. So the difference between them represents the error function and our goal is to minimize that. Therefore, we implement the following algorithm using the mathscript block in LabView, and then take the minimum value of it to minimize the error.



Following are the BER v Noise Power plots and comparisons for the two symbol-timing recovery approaches:







Based on these results, our verdict is that the EL Gate method performs slightly better than the Max-Energy method.

Frame detection and frequency offset correction:

When td = nT, there is a mismatch between the indices of transmitted and received symbols. Therefore, in this case, we need frame synchronization to solve the problem. In order to achieve this, we built two blocks: Sliding Correlator and Moose. Sliding Correlator is used for frame detection, and Moose detects the frequency offset and then corrects it.

Moose:

We introduce a periodic training sequence here: IEEE 801.11a.

To find the offset, we use the least linear square approximation method. The approximation is as follows:

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

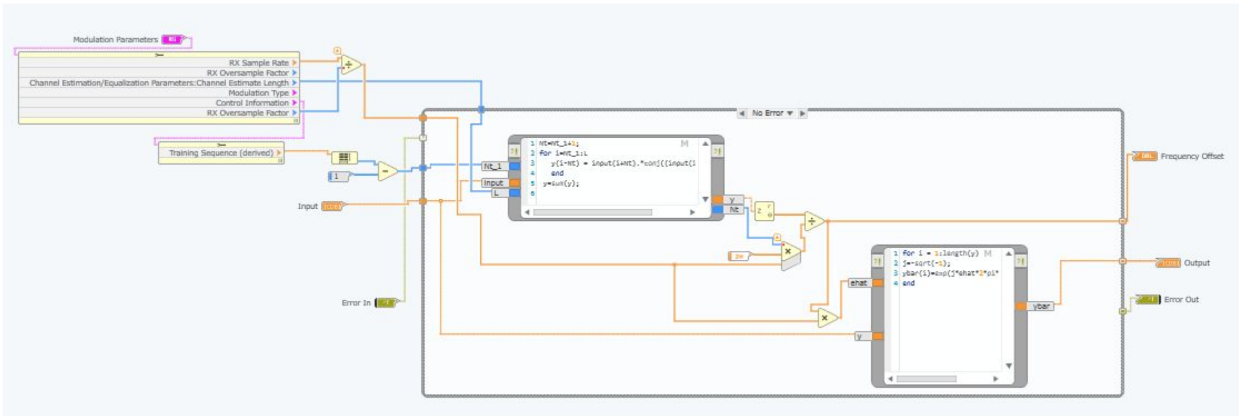
Then, we use the periodic property of the training data. By adding Nt (the length of the training sequence) and using the property of periodicity, we obtain the following:

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

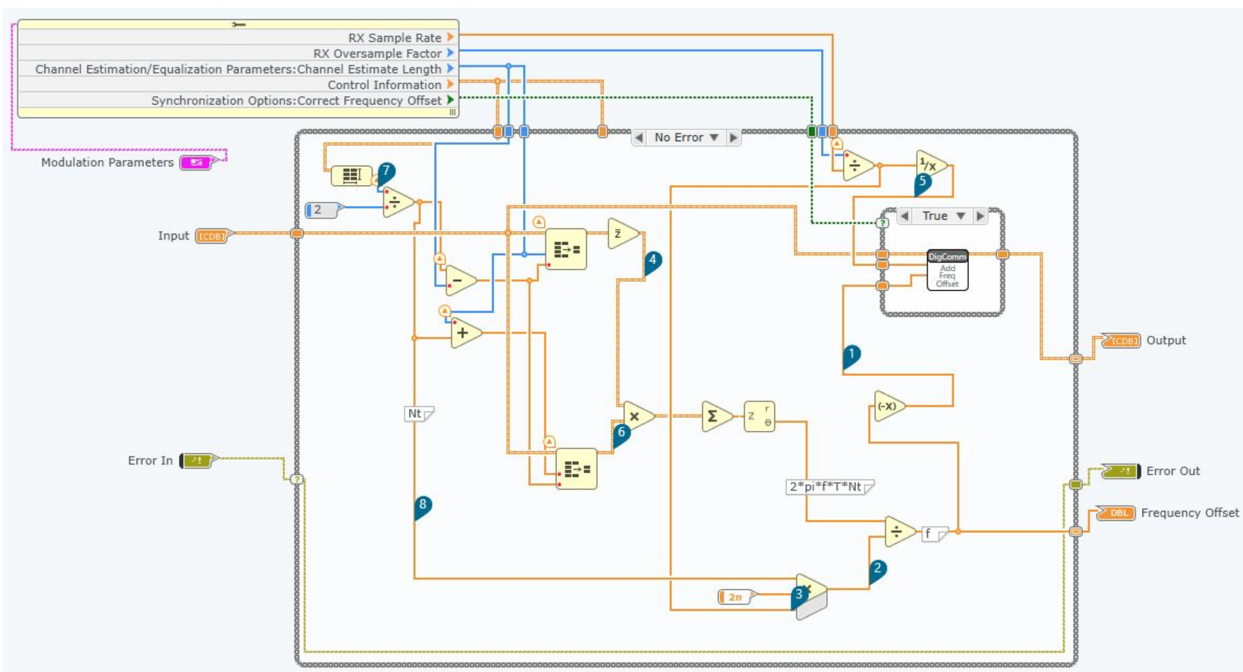
$$y[n] \;=\; e^{j2\pi\epsilon n} \sum_{l=0}^{L} h[l] s[n-l] + v[n],$$

The algorithm is implemented as follows in LabView:

This is the attempt with the Mathscript:



We also built the Moose subVI using blocks:



Sliding Correlator:

We consider a flat-fading channel, where symbol synchronization has already been employed to give y[n], after the matched filter and down-sampling at the receiver.

Considering a training sequence t[n] that is known at the receiver, we correlate the received signal with it since it has good correlation properties:

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

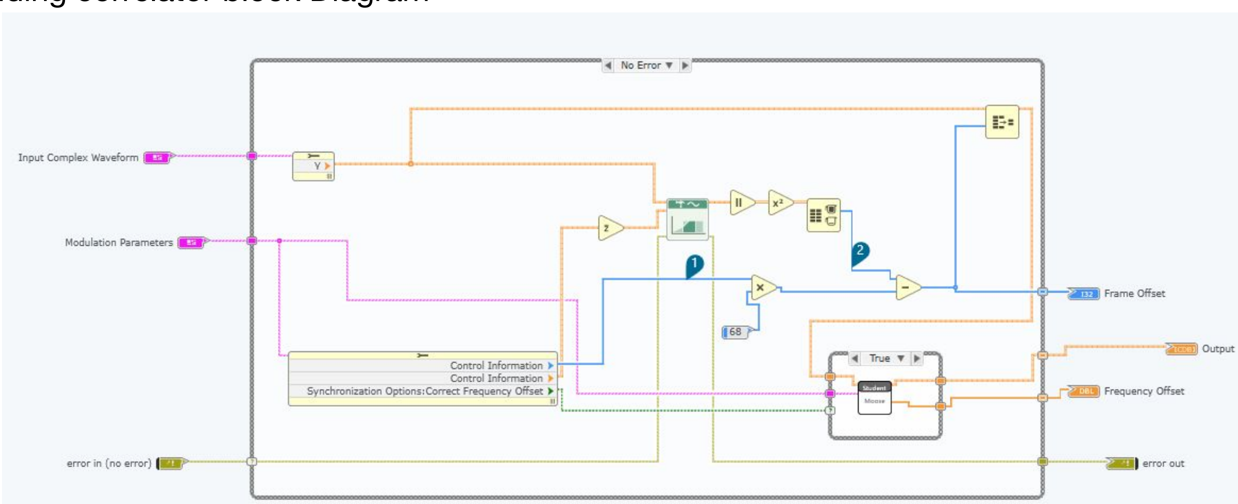
We then solve for the frame offset as:

$$\hat{d} = \underset{n}{\operatorname{argmax}}\, R[n].$$

We call the Moose sub-routine in this VI. For the IQ waveform cluster (input complex waveform) and the boolean for whether the frequency offset is correct or not as inputs, the outputs of the sliding correlator VI are: 1. Received sequence after correcting for estimated delay and frequency, 2. The frame offset, and 3. The frequency estimate computed using the Moose sub-routine.

After implementation, we observed that the subVI was able to detect artificially introduced offsets.

Sliding correlator block Diagram

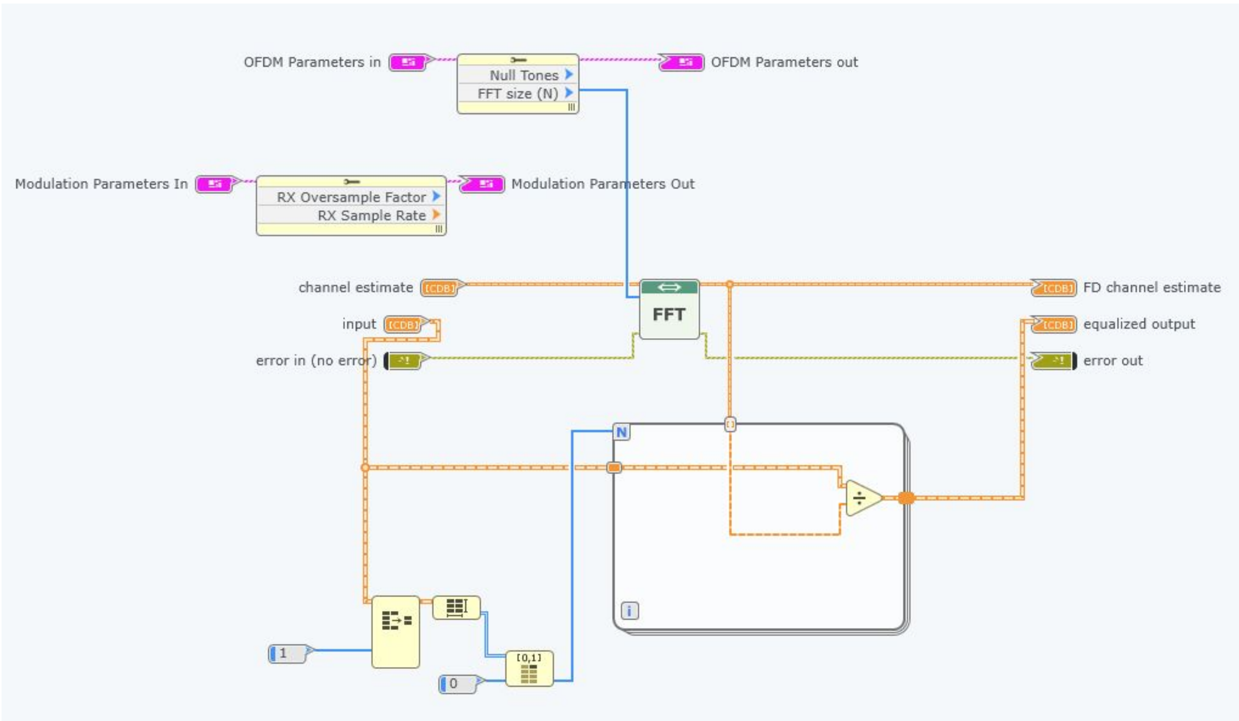


Frequency Domain Equalization (FEQ):

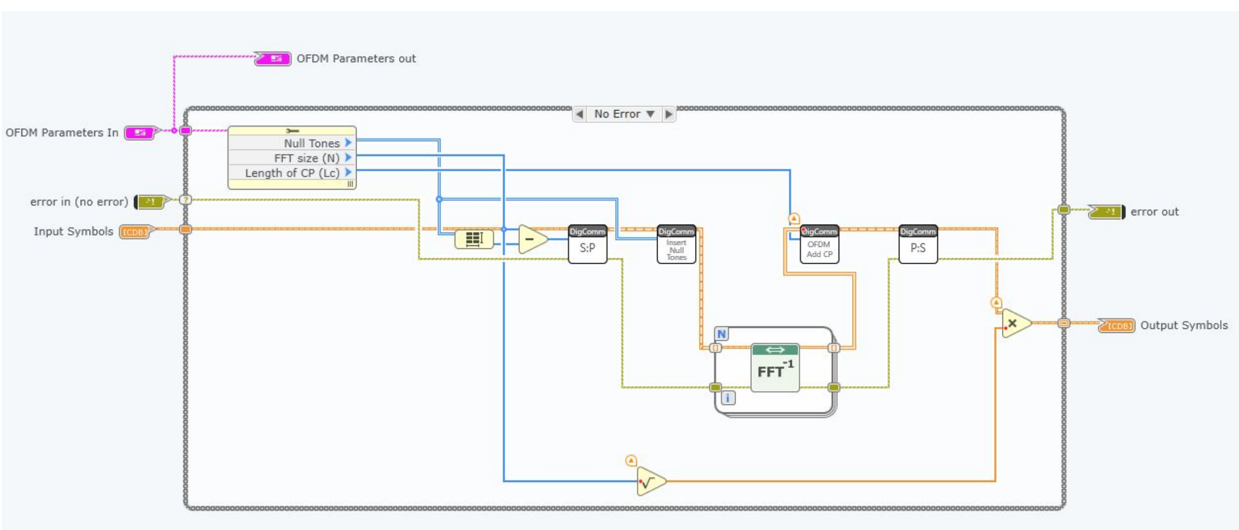
The FEQ sub-VI takes in 2 inputs and gives out two outputs. The inputs are as follows: 1. A parallel block structure of symbols that constitute the DFT output, where each row corresponds to N symbols of Y[k] and 2. The channel estimate computed prior to demodulation.

The outputs are as follows: 1. The equalized symbols, where each row corresponds to a block of N symbols, of the sequence corresponding to Y[k]/DFT{h[l]} and 2. The frequency domain response of the channel estimate computed by taking the DFT of the channel estimate (DFT{h[l]}).

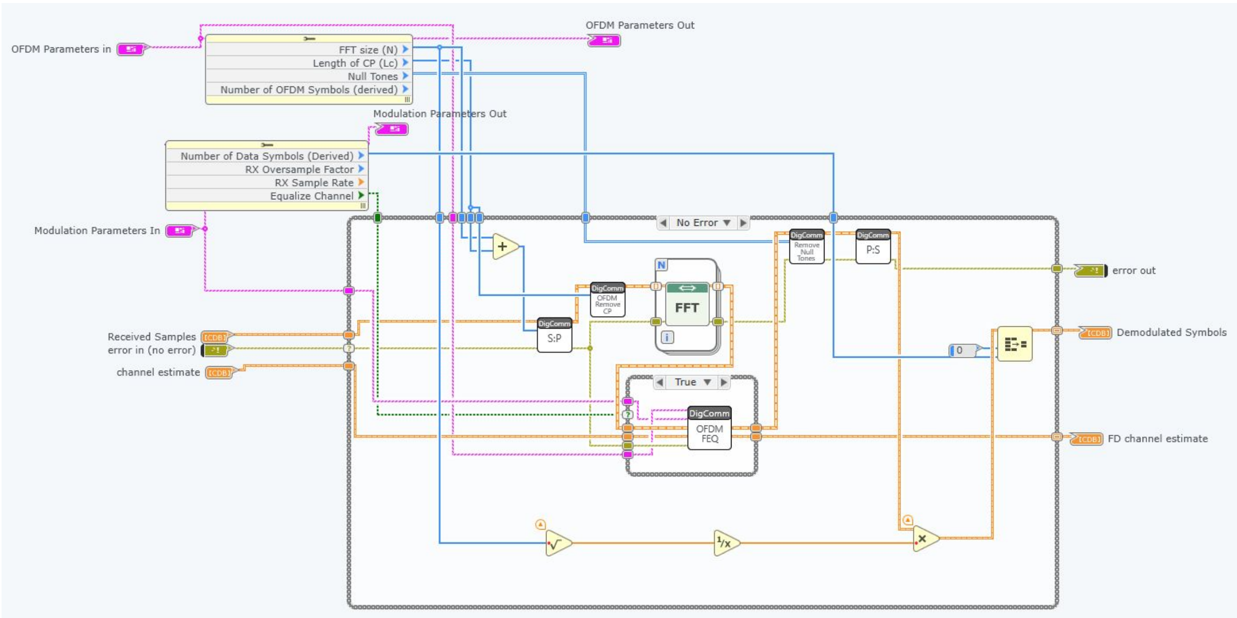
Block Diagram:



OFDM Modulator:



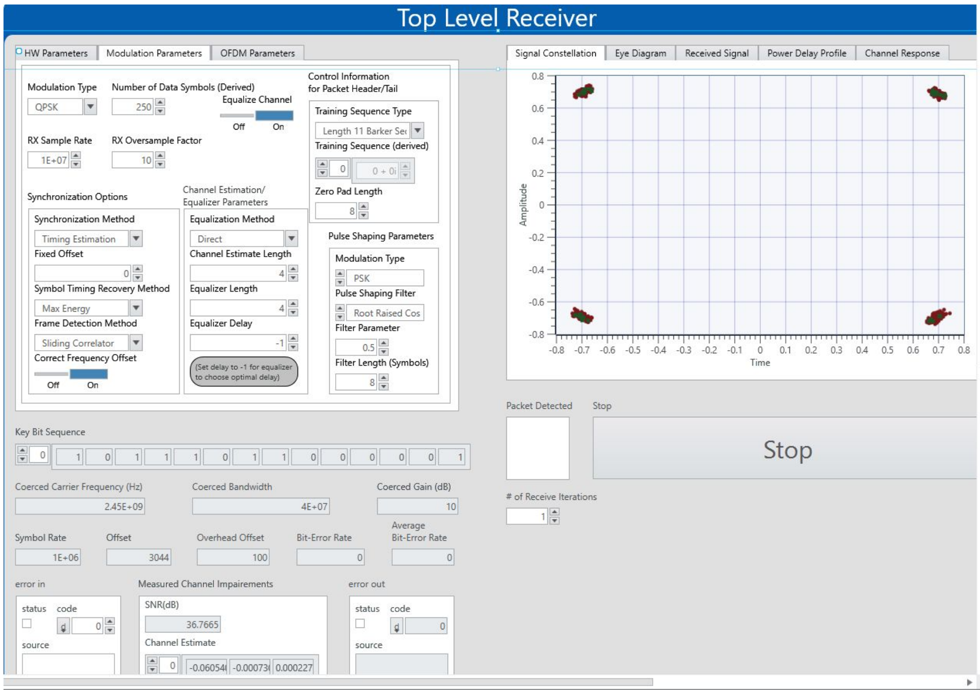
In the OFDM modulator, the OFDM symbols are passed through the serial to parallel block. Then we add null tones. After that, we perform the inverse fourier transform to each row of the matrix. Finally, we add a cyclic prefix with length Lc. The block diagram is shown above.

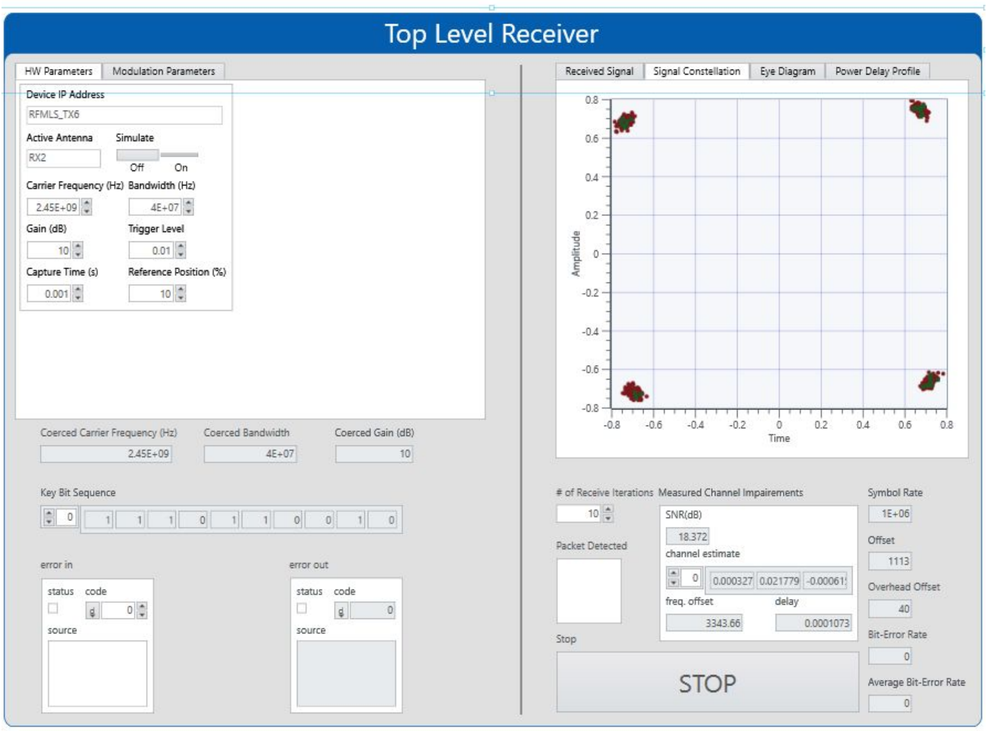


Testing:

All of our blocks worked nominally both with the simulator and in the wired USRP scenario. We observed that it was important to keep the symbol rate consistent between the transmitter and the receiver in order to ensure that the eye-diagram wasn't distorted. We also tested our blocks in a wireless scenario. While testing OFDM, we kept a textbook in between the antennas of the two USRPs under test, and we were able to see distortions in the constellation. When testing using a metal board, we weren't able to see any signal at all.

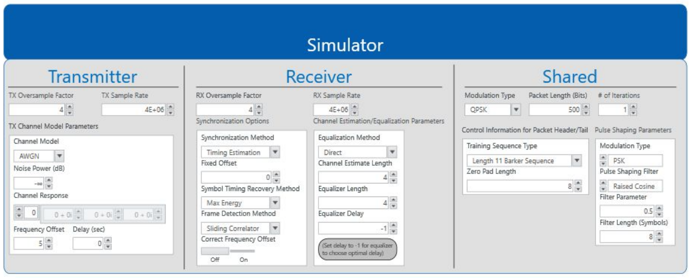
The following are the results for wireless transmission:

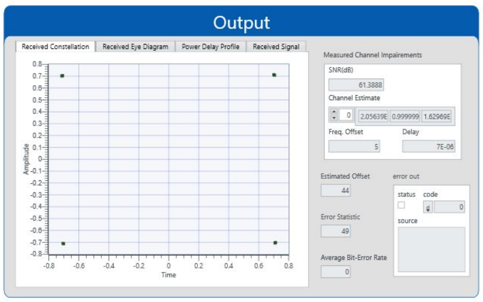


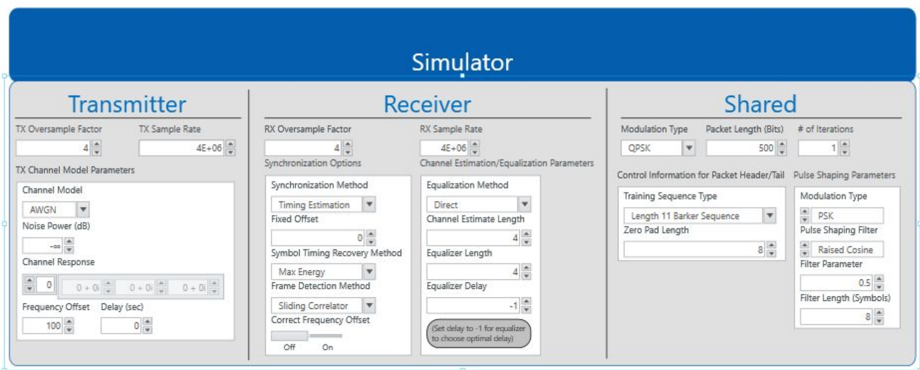


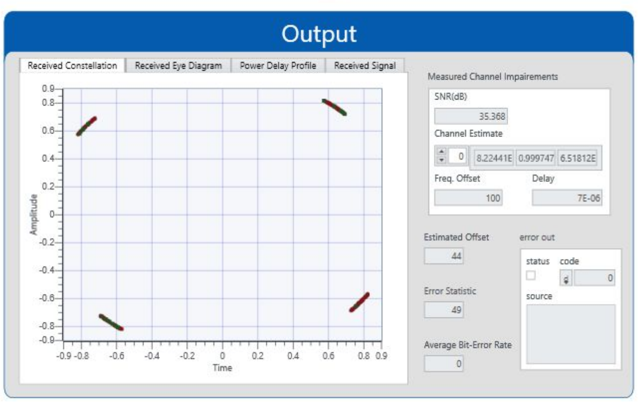
The BER is zero.

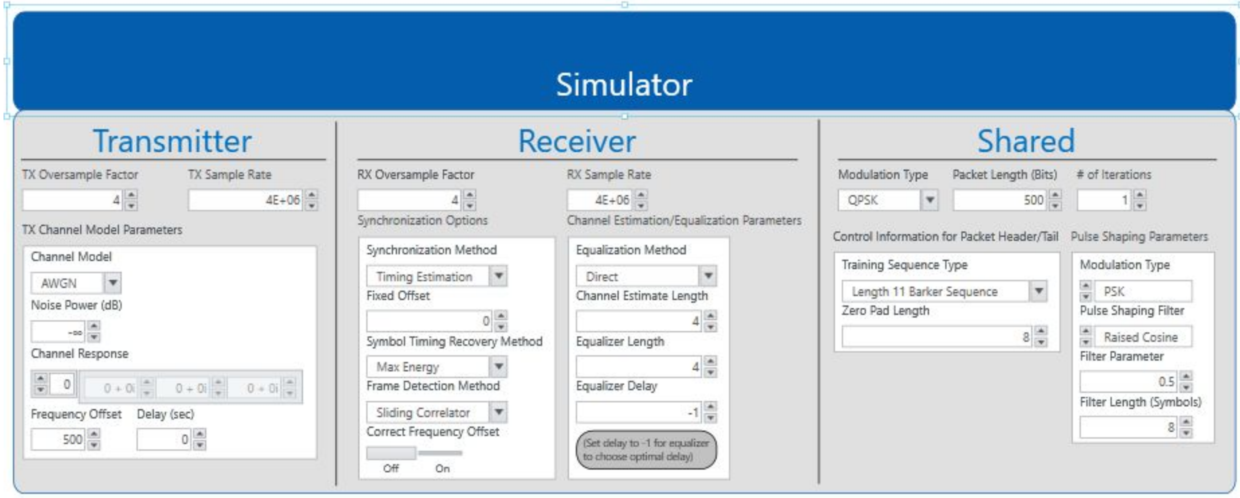
We also tested how much frequency offset the system could tolerate using our system:

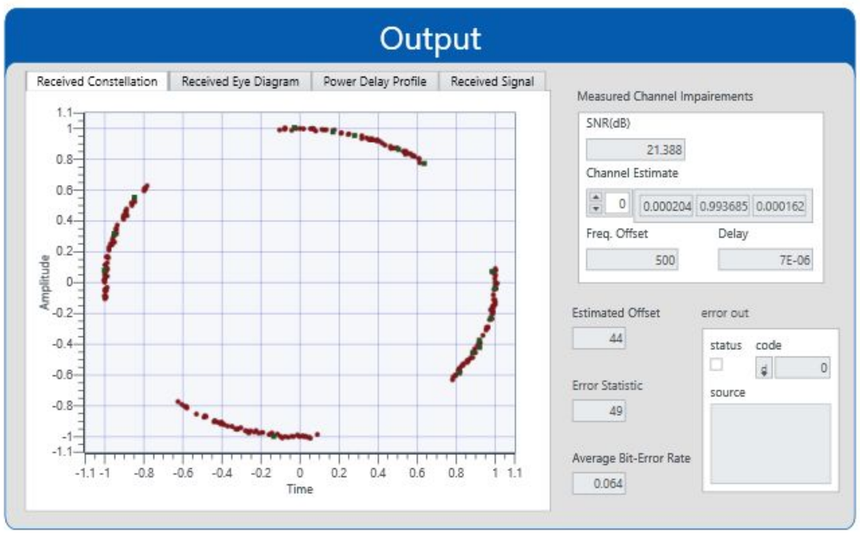

Simulator
Transmitter
TX Oversample Factor
TX Sample Rate
TX Channel Model Parameters
Channel Model
AWGN
Noise Power (dB)
Channel Response
Frequency Offset
Delay (sec)
Receiver
RX Oversample Factor
RX Sample Rate
Synchronization Options
Channel Estimation/Equalization Parameters
Synchronization Method
Timing Estimation
Fixed Offset
Symbol Timing Recovery Method
Max Energy
Frame Detection Method
Sliding Correlator
Correct Frequency Offset
Off
On
Equalization Method
Direct
Channel Estimate Length
Equalizer Length
Equalizer Delay
(Set delay to -1 for equalizer to choose optimal delay)
Shared
Modulation Type
Packet Length (Bits)
# of Iterations
QPSK
500
Control Information for Packet Header/Tail
Pulse Shaping Parameters
Training Sequence Type
Length 11 Barker Sequence
Zero Pad Length
Modulation Type
PSK
Pulse Shaping Filter
Raised Cosine
Filter Parameter
0.5
Filter Length (Symbols)

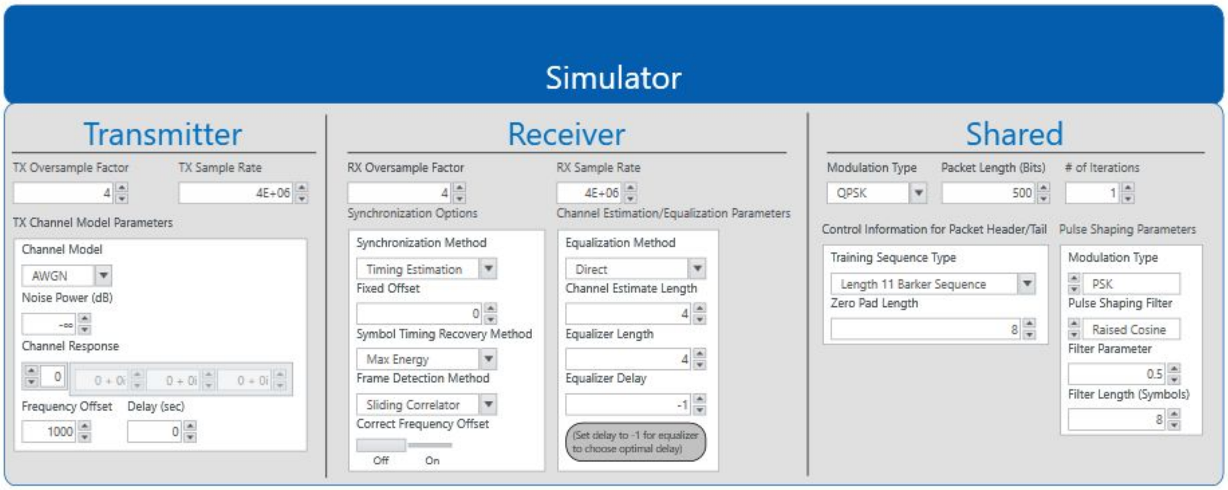

Output
Received Constellation
Received Eye Diagram
Power Delay Profile
Received Signal
Amplitude
Time
Measured Channel Impairements
SNR(dB)
61.3888
Channel Estimate
Freq. Offset
Delay
7E-06
Estimated Offset
44
Error Statistic
49
Average Bit-Error Rate
error out
status
code
source

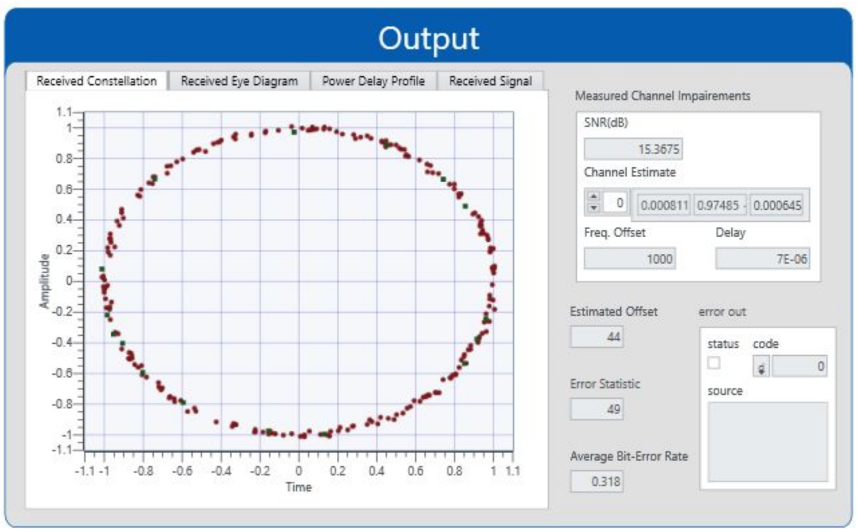

Simulator
Transmitter
TX Oversample Factor
TX Sample Rate
4E+06
TX Channel Model Parameters
Channel Model
AWGN
Noise Power (dB)
Channel Response
Frequency Offset
Delay (sec)
100
Receiver
RX Oversample Factor
RX Sample Rate
Synchronization Options
Channel Estimation/Equalization Parameters
Synchronization Method
Timing Estimation
Fixed Offset
Symbol Timing Recovery Method
Max Energy
Frame Detection Method
Sliding Correlator
Correct Frequency Offset
Off
On
Equalization Method
Direct
Channel Estimate Length
Equalizer Length
Equalizer Delay
(Set delay to -1 for equalizer to choose optimal delay)
Shared
Modulation Type
Packet Length (Bits)
# of Iterations
QPSK
500
Control Information for Packet Header/Tail
Pulse Shaping Parameters
Training Sequence Type
Length 11 Barker Sequence
Zero Pad Length
Modulation Type
PSK
Pulse Shaping Filter
Raised Cosine
Filter Parameter
0.5
Filter Length (Symbols)


Output
Received Constellation
Received Eye Diagram
Power Delay Profile
Received Signal
Amplitude
Time
Measured Channel Impairements
SNR(dB)
35.368
Channel Estimate
8.22441E
0.999747
6.51812E
Freq. Offset
100
Delay
7E-06
Estimated Offset
44
Error Statistic
49
Average Bit-Error Rate
error out
status
code
source


Simulator
Transmitter
TX Oversample Factor
4
TX Sample Rate
4E+06
TX Channel Model Parameters
Channel Model
AWGN
Noise Power (dB)
-∞
Channel Response
0
0 + 0i
0 + 0i
0 + 0i
Frequency Offset
500
Delay (sec)
0
Receiver
RX Oversample Factor
4
RX Sample Rate
4E+06
Synchronization Options
Synchronization Method
Timing Estimation
Fixed Offset
0
Symbol Timing Recovery Method
Max Energy
Frame Detection Method
Sliding Correlator
Correct Frequency Offset
Off
On
Channel Estimation/Equalization Parameters
Equalization Method
Direct
Channel Estimate Length
4
Equalizer Length
4
Equalizer Delay
-1
(Set delay to -1 for equalizer to choose optimal delay)
Shared
Modulation Type
QPSK
Packet Length (Bits)
500
# of 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 Cosine
Filter Parameter
0.5
Filter Length (Symbols)
8


Output
Received Constellation
Received Eye Diagram
Power Delay Profile
Received Signal
Amplitude
Time
Measured Channel Impairements
SNR(dB)
21.388
Channel Estimate
0
0.000204
0.993685
0.000162
Freq. Offset
500
Delay
7E-06
Estimated Offset
44
Error Statistic
49
Average Bit-Error Rate
0.064
error out
status
code
0
source





We increase the offset from 10 to 1000. The signal starts to get distorted when the offset is 500. The constellation continues to rotate. At the point when it rotates 180 degrees, the machine is not able to recognize the direction of rotation or how many cycles it has rotated, and therefore, we get an erroneous result.