

## II. LAB

**Software Required:** NI LabVIEW 2012, NI LabVIEW 4.3 Modulation Toolkit.

**Functions and VI (Virtual Instrument) from the LabVIEW software to be used in this lab:** For Loop (Function), Unbundle By Name (Function), Bundle By Name (Function), Array Subset (Function), Decimate (single shot) (VI), Complex to Polar (Function), Square (Function), Add Array Elements (Function), Greater? (Function), Select (Function), Numeric Constant, DBL Numeric Constant, and Shift Register.

### PART-1: MAXIMUM ENERGY ALIGNMENT

**INPUTS:** Input complex waveform (cluster of 3 elements: t0 (double [64-bit real (~ 15 digits precision)]), dt (double [64-bit real (~ 15 digits precision)]), and Y (1-D array of double complex [~15 digit precision])).

**OUTPUTS:** Aligned complex waveform (cluster of 3 elements: t0 (double [64-bit real (~ 15 digits precision)]), dt (double [64-bit real (~ 15 digits precision)]), and Y (1-D array of double complex [~15 digit precision])), alignment offset (samples) (long [32-bit integer (-2147483648 to -2147483647)]).

A virtual instrument template (“student\_align\_MaxEnergy.vit”) is provided to you with these inputs and outputs. The template should be populated according to instructions. Then, newly created VIs are plugged in the receiver USRP control file.

### PROCEDURE

In this part, we develop a LabVIEW virtual instrument as a timing synchronization unit which uses maximum energy alignment algorithm for symbol timing recovery operation that can use the received complex waveforms through the USRP N210 receiving hardware as input and applies time synchronization algorithm on them.

We now detail the construction of the VI at hand in Lab View:

#### a) Setup

\* Open the LabVIEW program (Start > All Programs > National Instruments > LabVIEW 2012 > LabVIEW 2012).

\* Open “student\_align\_MaxEnergy.vit” file (Open Existing (OR Ctrl+O) > Select “student\_align\_MaxEnergy.vit” > OK). A basic graphic user interface (GUI) opens. (NOTE: If a warning window pops up, click on “Ignore” option).

\* Open Block Diagram (Window > Show Block Diagram (OR Ctrl+E)).

- \* Delete “error in (no error)” and “error out” clusters.
- \* Enable Functions Palette (View > Functions Palette or right click on anywhere of the block diagram window).
- \* Place a For Loop (Functions Palette > Programming > Structures > select and drag For Loop).

## **b) Define number of iterations**

- \* Connect “RX oversample factor” to the *count (N)* terminal of the *For Loop*.
- \* Connect this *count (N)* terminal of the *For Loop* to *decimating factor* input of “Decimate (single shot).vi” function.

## **c) Decimation (filtering)**

- \* Place an *Array Subset* inside the *For Loop* (Functions Palette > Programming > Array > Array Subset). This function returns a portion of the input array starting at <index> and containing <length> elements.
- \* Place an *Unbundle By Name* outside the *For Loop* (Functions Palette > Programming > Cluster, Class, & Variant > Unbundle By Name). This function returns the elements of the input cluster whose names we specify.
- \* Connect the “input complex waveform” to *input cluster* port of one of the *Unbundle By Name* functions. This input specifies the continuous-time samples of the modulated complex baseband waveform data.
- \* The box reads “t0”. Change it to “Y” (Right click menu > Select Item > Y). Y will give us the array of complex samples.
- \* Connect *element* output of this *Unbundle By Name* to *Array* input of the *Array Subset* function.
- \* There will be an error mark on this new wire. In order to remove this error, right click on the loop tunnel (connection box on the border of the *For Loop*) and select “Disable Indexing” option.
- \* Place a *Decimate (single shot).vi* inside the *For Loop* (Functions Palette > Signal Processing > Signal Operation > Decimate (single shot).vi). This function decimates (reduces the sampling rate of ) the input sequence <X> by the <decimating factor> and the <averaging> Boolean control. Since we do not use the averaging feature of this function, it will give an output

sequence  $y_i = x_{i*m}$  where  $i=0,1,2,\dots,(size-1)$  and  $size = \left\lfloor \frac{n}{m} \right\rfloor$ ,  $n$  is the number of elements in  $\langle X \rangle$  and  $m$  is the  $\langle$ decimating factor $\rangle$ .

\* Connect *Subarray* output of *Array Subset* function to  $X$  input of “Decimate (single shot).vi” function.

#### **d) Energy calculation**

\* Place a *Complex To Polar* inside the *For Loop* (Functions Palette > Mathematics > Numeric > Complex > Complex To Polar). This function breaks a complex input into its polar components (i.e., magnitude and phase).

\* Connect *Decimated Array* output of “Decimate (single shot).vi” function to  $Z$  input of *Complex To Polar* function.

\* Place a *Square* inside the *For Loop* (Functions Palette > Mathematics > Numeric > Square). This function computes the square of the input value.

\* Connect  $r$  (magnitude) output of *Complex To Polar* function to the input of *Square* function.

\* Place an *Add Array Elements* inside the *For Loop* (Functions Palette > Mathematics > Numeric > Add Array Elements). This function returns the sum of all elements in the numeric input array.

\* Connect output of *Square* function to the input of *Add Array Elements* function.

#### **e) Maximum energy detection**

\* Place a *Select* function inside the *For Loop* (Functions Palette > Programming > Comparison > Select).

\* Add a shift register to the *For Loop* (Right click on the border of the *For Loop* > Right click menu > Add Shift Register). This function passes values from previous iterations through a loop to the next iteration.

\* Place a *DBL Numeric Constant* outside the *For Loop* (Functions Palette > Mathematics > Numeric > DBL Numeric Constant). Leave its value as “0”.

\* Place a *Greater?* function inside the *For Loop* (Functions Palette > Programming > Comparison > Greater?). This function compares two inputs and gives a Boolean response at its output.

\* Connect output of *Add Array Elements* function to  $x$  input of the *Greater?* function.

\* Connect the *DBL Numeric Constant* to shift register’s downward arrow.

- \* Connect this shift register arrow to  $y$  input of the *Greater?* function.
- \* Connect the wire between *DBL Numeric Constant* shift register arrow and the *Greater?* function to  $f$  input of *Select* function.
- \* Connect  $x > y?$  output of the *Greater?* function to  $s$  input of this *Select* function.
- \* Connect the output of this *Select* function to the upward arrow of its own shift register.

#### **f) End of iteration?**

- \* Place a *Numeric Constant* outside the *For Loop* (Functions Palette > Mathematics > Numeric > Numeric Constant). Leave its value as "0".
- \* Place another *Select* function inside the *For Loop* (Functions Palette > Programming > Comparison > Select).
- \* Add another shift register to the *For Loop* (follow the same steps as above).
- \* Connect the *Numeric Constant* to this shift register's downward arrow.
- \* Connect this shift register arrow to  $f$  input of this new *Select* function.
- \* Connect  $s$  inputs of both *Select* functions to each other.
- \* Connect the iteration terminal ( $i$ ) of the *For Loop* to *index* input of the *Array Subset* function.
- \* Connect  $t$  input of the second *Select* function (connected to the *Numerical Constant*) to this wire (between  $i$  and *index* input).
- \* Connect the output of this second *Select* function to the upward arrow of its own shift register.
- \* Connect this second shift register upward arrow to "alignment offset (samples)".

#### **g) Form aligned complex waveform**

- \* Place an *Array Subset* outside the *For Loop* (Functions Palette > Programming > Array > Array Subset).
- \* Connect *index* input of the outside *Array Subset* function to the wire between second shift register upward arrow and "alignment offset (samples)".

- \* Place another *Unbundle By Name* outside the *For Loop* (Functions Palette > Programming > Cluster, Class, & Variant > Unbundle By Name).
- \* Connect the wire between “input complex waveform” and the first *Unbundle By Name* to the input of this second *Unbundle By Name* function.
- \* The box reads “t0”. Change it to “Y” (Right click menu > Select Item > Y).
- \* Connect *element* output of this *Unbundle By Name* to *Array* input of the outside *Array Subset* function.
- \* Place a *Bundle By Name* outside the *For Loop* (Functions Palette > Programming > Cluster, Class, & Variant > Bundle By Name).
- \* Connect *Subarray* output of this *Array Subset* function to *element* input of *Bundle By Name* function.
- \* Connect the wire between “input complex waveform” and *Unbundle By Name* functions to *input cluster* input of *Bundle By Name* function.
- \* The box reads “t0”. Change it to “Y” (Right click menu > Select Item > Y).
- \* Connect *output cluster* output of *Bundle By Name* function to the “aligned complex waveform” element.

At the end, your block diagram should look like the one below in Fig 1 and your front panel should look like the one in Fig 2:

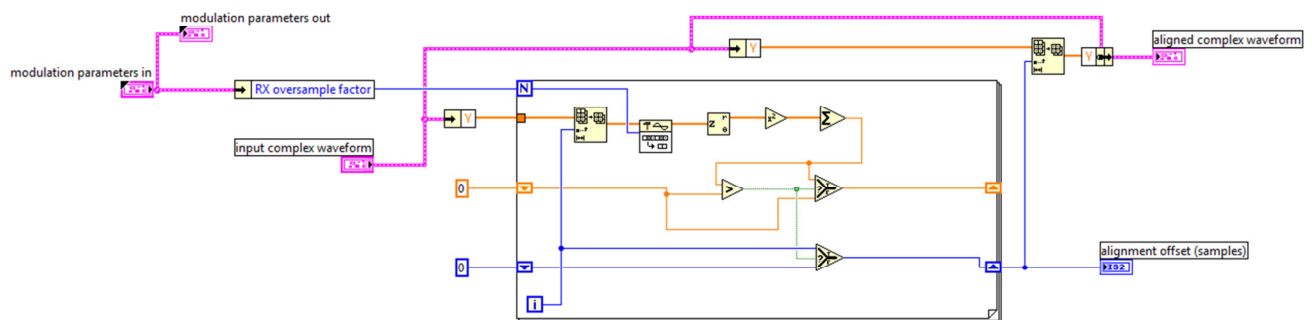


Fig 1 – Complete block diagram

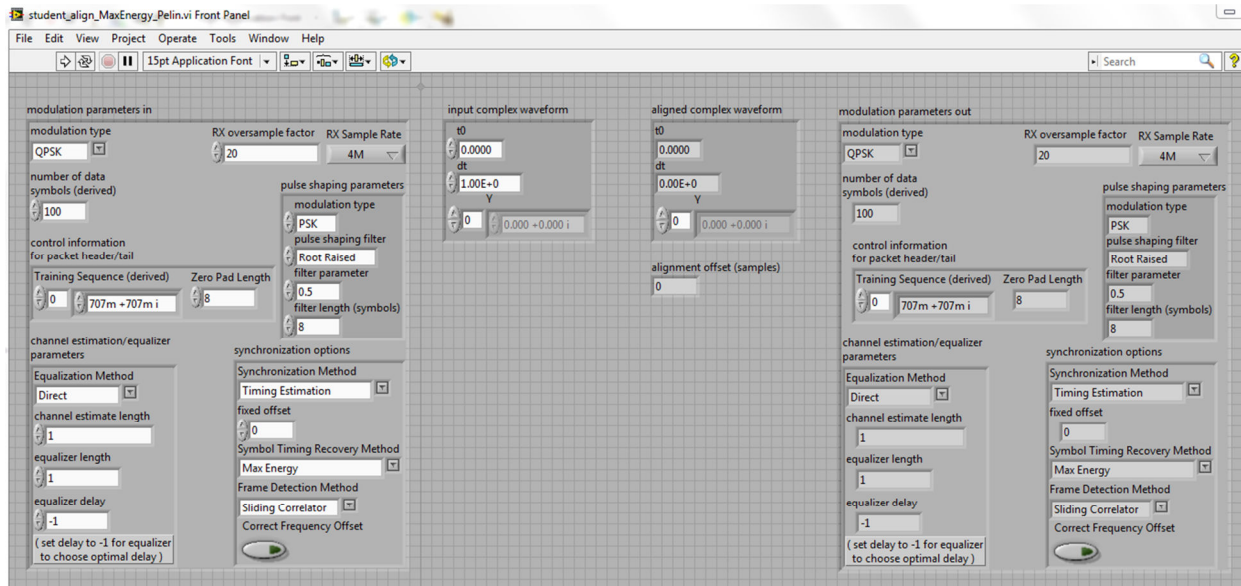


Fig 2 – Complete front panel

For visual reference, check this video out (note to Gil: **please add your video here**): [See the video: How to build Lab5 align MaxEnergy student.vi](#)

\* In order to check if your file has errors click on Operate > Run (OR Ctrl+R). If no error window is generated, then your file is good.

\* SAVE it AS a new file by adding your name at the end (example: if your name is John, then the new file name is “student\_align\_MaxEnergy\_John.vi”).

## **PART-2: SIMULATION**

In this part we use the provided transmitter and receiver templates to plug in our VI files and test their operation before using them with the USRP boards.

In order to simulate your maximum energy and early late gate timing synchronization designs, please follow these steps:

\* Open the LabVIEW program (Start > All Programs > National Instruments > LabVIEW 2012 > LabVIEW 2012)

### **a) Receiver setup**

\* Open “receiver.vi” file (Open Existing (OR Ctrl+O) > Select “receiver.vi” > OK). A basic graphic user interface (GUI) opens. (NOTE: If a warning window pops up, click on “Ignore” option.)

\* Open Block Diagram (Window > Show Block Diagram (OR Ctrl+E)).

\* Locate the “synchronize.vi” (synch) in the block diagram and double click on it to open its front panel and block diagram.

\* On “synchronize.vi” block diagram, go to case selector and select the “timing estimation” case.

\* Locate the “symbol\_timing.vi” (symbol timing) in the block diagram and double click on it to open its front panel and block diagram.

\* On “symbol\_timing.vi” block diagram, go to case selector and select the “Max Energy” case.

\* Locate the “align\_MaxEnergy.vi” (max energy align) in the block diagram and replace it with your newly created “student\_align\_MaxEnergy\_<name>.vi” file (Right click on the “max energy align” box > Replace > All Palettes > Select a VI...>insert your “student\_align\_MaxEnergy\_<name>.vi” file).

## **b) Simulator setup**

\* Open “simulator.vi” file (Open Existing (OR Ctrl+O) > Select “simulator.vi” > OK). A basic graphic user interface (GUI) opens. (NOTE: If a warning window pops up, click on “Ignore” option.)

\* Click on “Run Continuously” button (two looping arrows).

## **c) Data collection**

Bring up the simulator front panel and perform the following measurements shown on Fig 5, 6, and 7. Detailed information on front panel objects can be found in Appendix-A of this document.

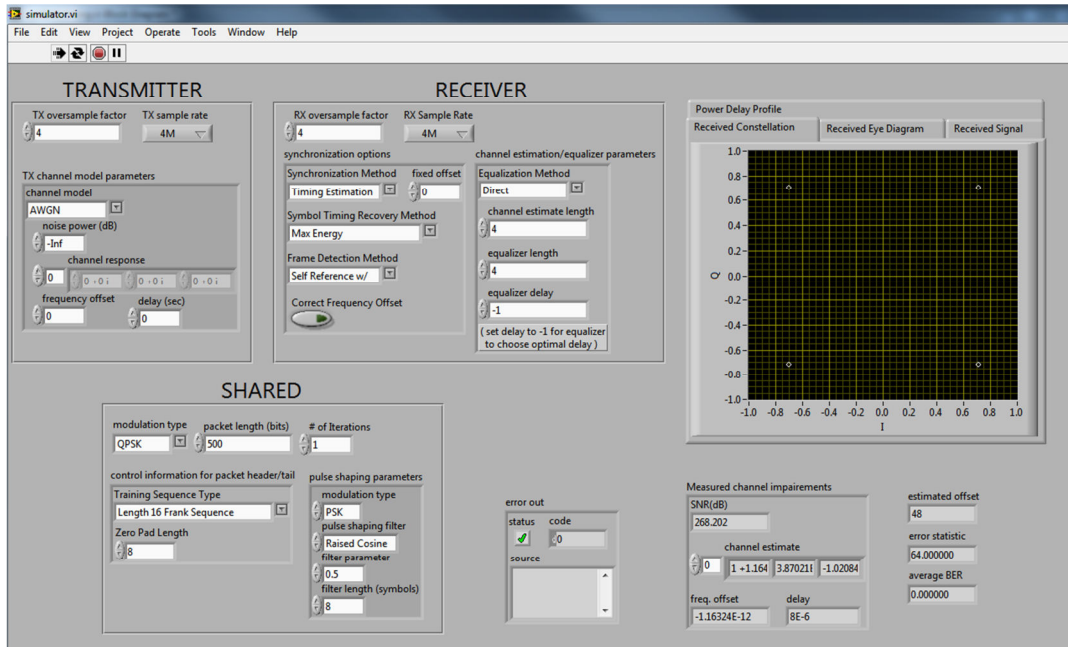


Fig 5 – Simulator front panel – received constellation view

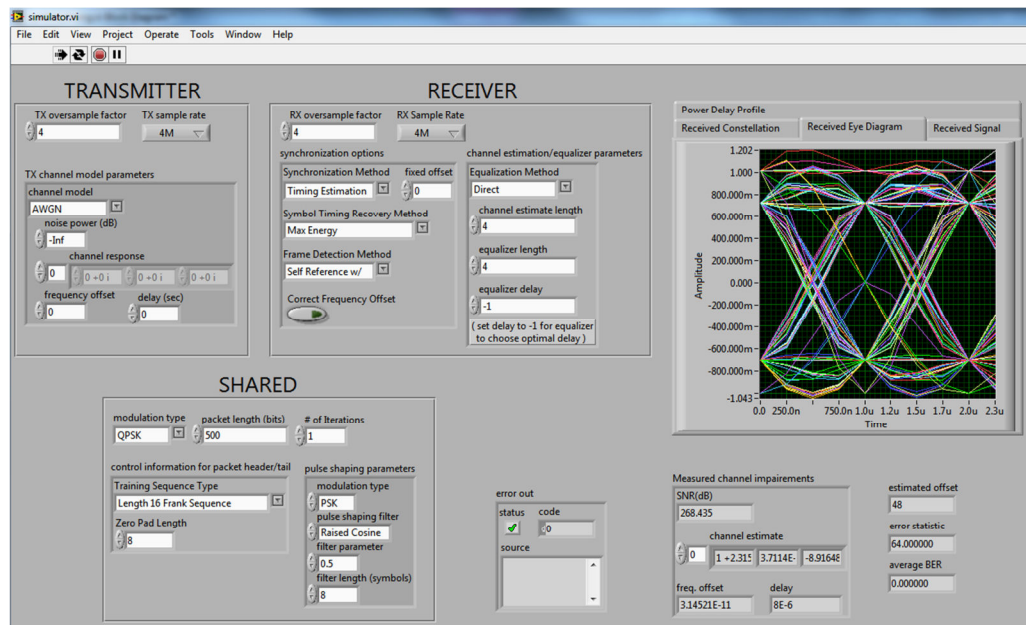


Fig 6 – Simulator front panel – eye diagram view with ideal noise power setting (-infinity)



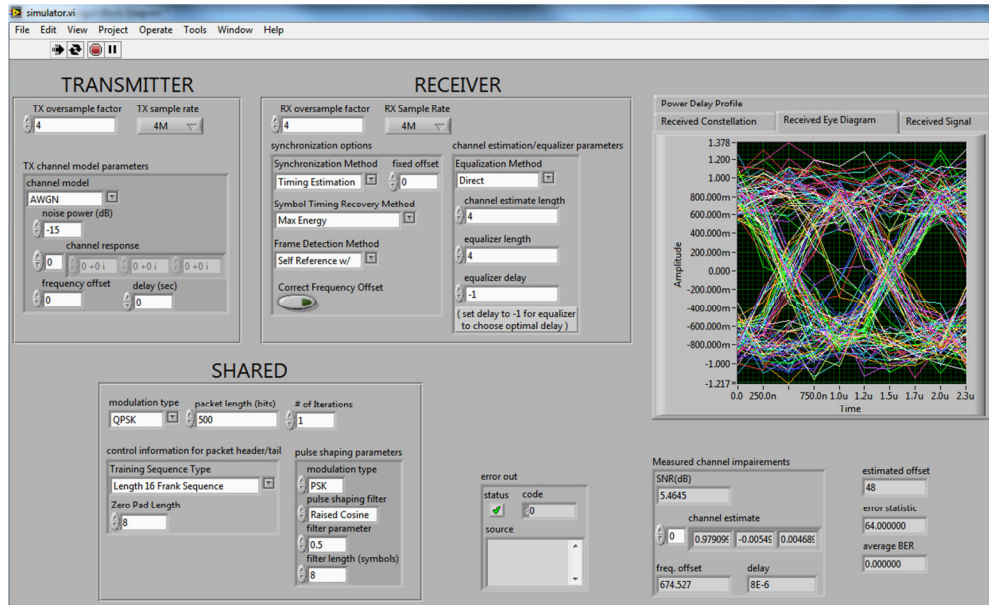


Fig 7 – Simulator front panel – eye diagram view with more realistic noise power setting (-15 dB)

Repeat the simulation using following *TX oversample factor* and *RX oversample factor* settings: from “4” to “20” by steps of “4” for both BPSK and QPSK and collect screenshots of both constellation and eye diagram views.

Also, change the *delay (sec)* value to “8.2E-10” for both BPSK and QPSK and collect screenshots of both constellation and eye diagram views. All collected screenshots should be used for report preparation.

### **PART-3: USRP OPERATION**

In this part we use the provided top\_tx and top\_rx templates to plug in our VI files and use them with the USRP boards.

In order to operate using USRPs, please follow these steps:

\* Open the LabVIEW program (Start > All Programs > National Instruments > LabVIEW 2012 > LabVIEW 2012).

#### **a) Transmitter setup**

\* Open “top\_tx.vi” file (Open Existing (OR Ctrl+O) > Select “top\_tx.vi” > OK). A basic graphic user interface (GUI) opens. (NOTE: If a warning window pops up, click on “Ignore” option.)

\* Open Block Diagram (Window > Show Block Diagram (OR Ctrl+E)).

\* Set the following values in Table 1 on top\_tx front panel, HW parameters tab:

LABEL	VALUE
USRP IP Address	<IP> from NI-USRP configuration utility
Active antenna	J1
Carrier frequency (in HZ)	2.47G
Gain (in dB)	10
Generation Mode	Continuous

Table 1 – Top\_tx HW parameters settings

\* Set the following values in Table 2 on top\_tx front panel, modulation parameters tab:

LABEL	VALUE
TX oversample factor	20
TX Sample Rate	20M
Packet length (bits)	500
Pulse shaping filter	Root Raised Cos
Training Sequence Type	Length 16 Frank Sequence

Table 2 – Top\_tx modulation parameters settings

\* Click on “Run Continuously” button (two looping arrows).

## **b) Receiver setup**

On another computer with a connected USRP:

\* Open “receiver.vi” file (Open Existing (OR Ctrl+O) > Select “receiver.vi” > OK). A basic graphic user interface (GUI) opens. (NOTE: If a warning window pops up, click on “Ignore” option.)

\* Open Block Diagram (Window > Show Block Diagram (OR Ctrl+E)).

\* Locate the “synchronize.vi” (synch) in the block diagram and double click on it to open its front panel and block diagram.

\* On “synchronize.vi” block diagram, go to case selector and select the “timing estimation” case.

\* Locate the “symbol\_timing.vi” (symbol timing) in the block diagram and double click on it to open its front panel and block diagram.

\* On “symbol\_timing.vi” block diagram, go to case selector and select the “Max Energy” case.

\* Locate the “align\_MaxEnergy.vi” (max energy align) in the block diagram and replace it with your newly created “student\_align\_MaxEnergy\_<name>.vi” file (Right click on the “max energy align” box > Replace > All Palettes > Select a VI...>insert your “student\_align\_MaxEnergy\_<name>.vi” file).

\* Open “top\_rx.vi” file (Open Existing (OR Ctrl+O) > Select “top\_rx.vi” > OK). A basic graphic user interface (GUI) opens. (NOTE: If a warning window pops up, click on “Ignore” option.)

\* Set the following values in Table 3 on top\_rx front panel, HW parameters tab:

LABEL	VALUE
USRP IP Address	<IP> from NI-USRP configuration utility
Active antenna	J1
Carrier frequency (in HZ)	2.47G
Gain (in dB)	10
Capture Time (s)	400u

Table 3 – Top\_rx HW parameters settings

\* Set the following values in Table 4 on top\_rx front panel, modulation parameters tab:

LABEL	VALUE
RX oversample factor	2
RX Sample Rate	2M
Pulse shaping filter	Root Raised Cos
Training sequence	Frank
Frame Detection Method	Self Reference w/ Power Normalization

Table 4 – Top\_rx modulation parameters settings

\* Click on “Correct Frequency Offset” button.

\* Click on “Run Continuously” button (two looping arrows).

For visual reference, check this video out: [See the video: How to build USRP operation](#)

### c) Data Collection

Your top\_tx (which we use to control the transmitting USRP) window should look like the ones in Fig 8 and Fig 9 below:

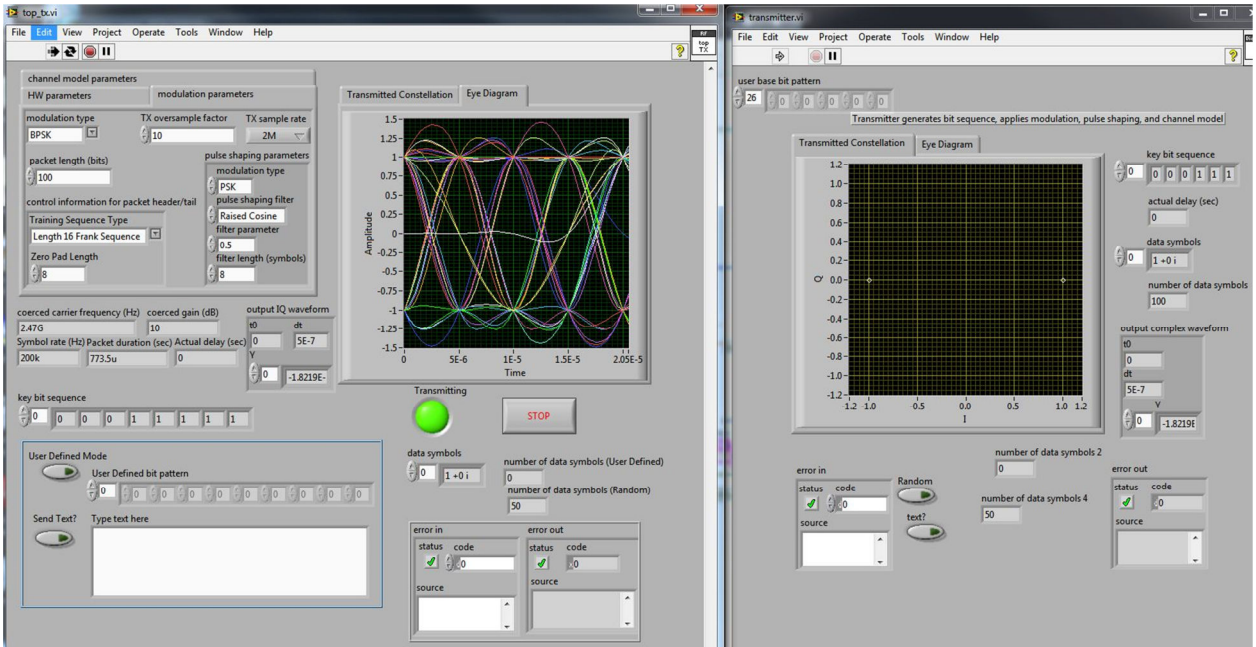


Fig 8 – Top\_tx front panel when BPSK is selected as modulation type

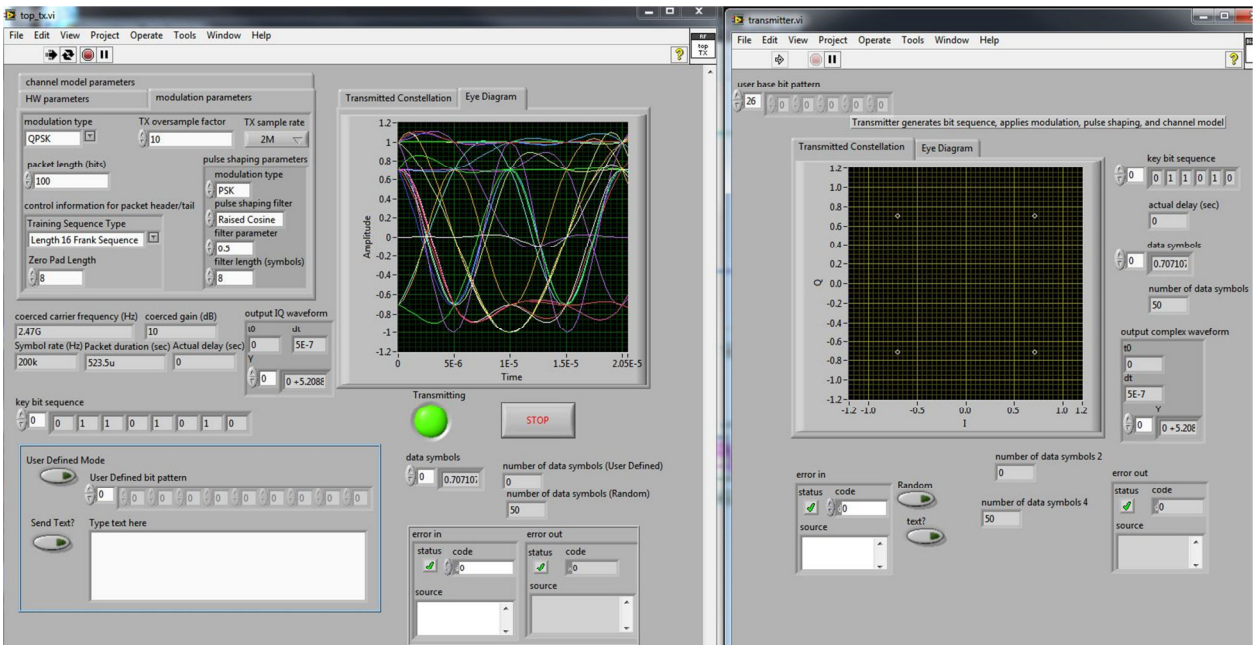


Fig 9 – Top\_tx front panel when QPSK is selected as modulation type

Your top\_rx (which we use to control the receiving USRP) window should look like the ones in Fig 10 and Fig 11 below:



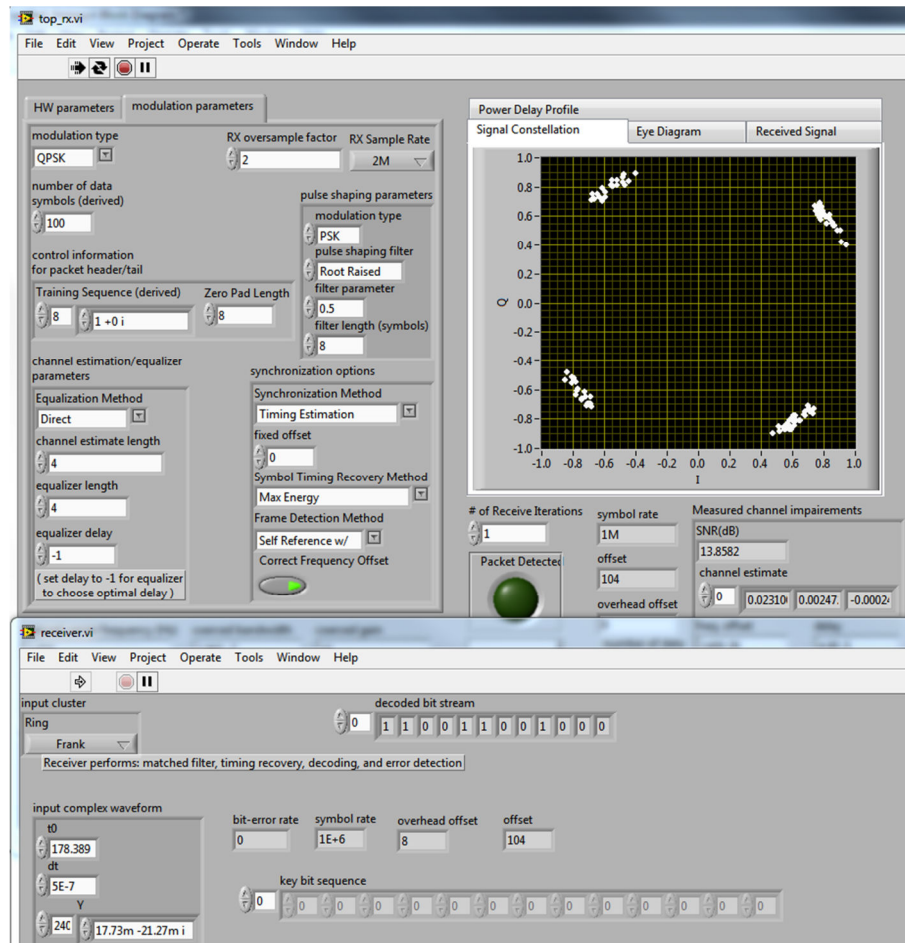


Fig 11 – Top\_rx front panel when QPSK is selected as modulation type

Detailed information on front panel objects can be found in Appendix-A of this document.

Bring up the top\_tx and top\_rx front panels and perform the measurements shown on Fig 8, 9, 10, 11 above.

Repeat the USRP operation using following *TX oversample factor* and *RX oversample factor* settings: from “4” to “20” by steps of “4” for both BPSK and QPSK and collect screenshots of both constellation and eye diagram views of both top\_tx and top\_rx front panels. They should be used for report preparation.

NOTE: Do not forget to keep the symbol rate constant by fixing the rate between oversample factors and sample rates. For example, if you increase oversample factor from 2 to 4, increase sample rate from 2M to 4M as well.

#### **PART-4: HOMEWORK QUESTIONS**

**#1:** Write a lab report describing the experiments and results you observed. Include in the report the images you collected in Part-3 and Part-4.

**#2:** Answer the following questions:

- What is the effect of TX and RX oversample factor and added delay in simulation on both constellation and eye diagram?
- What is the difference between simulation and USRP operation?

## APPENDIX A

### PART-1: SIMULATION

On “simulator.vi” front panel you can manipulate the following values:

#### TRANSMITTER SECTION

- TX oversample factor:** Data type is long [32-bit integer (-2147483648 to 2147483647)]. This specifies the number of desired samples per symbol for the pulse shaping filter. It must be an even number larger than 2.
- TX sample rate:** Data type is double [64-bit real (~15 digit precision)]. This specifies the transmitter sampling rate in Hertz (or samples per second). It must be specified between 195.312k and 100M.
- Noise power:** Data type is double [64-bit real (~15 digit precision)]. This specifies the level of noise power in dB. It can be any value between -Infinity (Inf) and +Inf.

#### RECEIVER SECTION

- RX oversample factor:** Data type is long [32-bit integer (-2147483648 to 2147483647)]. This specifies the number of desired samples per symbol for the pulse shaping filter. It must be an even number larger than 2. It should be same as TX oversample factor.
- RX sample rate:** Data type is double [64-bit real (~15 digit precision)]. This specifies the transmitter sampling rate in Hertz (or samples per second). It must be specified between 195.312k and 100M.
- Correct frequency offset:** Data type is Boolean [TRUE or FALSE]. This compensates the effect of the frequency offset between transmitted and received signals. It can be set as on or off.
- Symbol timing recovery method:** Data type is string. This chooses the method to be used for symbol timing recovery. It can be any one of the following five options: “MT Align to Ideal Symbols”, “Max Energy”, “Early-Late Gate”, “Gardner”, or “Fixed Offset”.

#### SHARED SECTION

- h) Modulation type:** Data type is string. This chooses the modulation scheme to be used in both transmitter and receiver. It can be either BPSK or QPSK.
- i) Packet length:** Data type is long [32-bit integer (-2147483648 to 2147483647)]. This specifies the packet length to be transmitted and received in bits.
- j) Iterations:** Data type is long [32-bit integer (-2147483648 to 2147483647)]. This changes how many times the simulation is performed. Higher number of iterations increases the accuracy, but decreases the processing speed.
- k) Training sequence type:** Data type is string. This chooses the training sequence to be used in both transmitter and receiver. It can be *Length 11 Barker Sequence*, *Length 16 Frank Sequence*, or *Neumann Hoffman Binary Code*.
- l) Pulse shaping filter:** Data type is unsigned word [16-bit integer (0 to 65535)]. This selects the pulse shaping and matched filtering pulses to be used in both transmitter and receiver. It can be *Raised Cosine*, *Root Raised Cosine*, or *Gaussian*.
- m) Filter parameter:** Data type is double [64-bit real (~15 digit precision)]. It sets the roll-off (or excess bandwidth) factor for Raised Cosine and Root Raised Cosine filters. It can be any real number between 0 and 1.

In the graph field you see four different tabs:

- 1) Received constellation:** This graph shows continuously changing I (in-phase or real) and Q (quadrature or imaginary) values of the constellation points as symbols received. These constellation points are the ones detected by the ML detector we constructed in Part-2.
- 2) Received eye diagram:** An “eye diagram” is an oscilloscope display in which a digital data signal from a receiver is repetitively sampled and applied to the vertical input, while the data rate is used to trigger the horizontal sweep. It is called so because its shape resembles an eye. An open eye pattern corresponds to minimal signal distortion (due to intersymbol interference and noise).
- 3) Received signal:** These two graphs show the Amplitude (in Volts) change of the received signal I and Q components in time domain.
- 4) Power delay profile:** This graph gives the amplitude of the signal received through a multipath channel as a function of time delay (sec). The time delay is the difference in travel time between multipath arrivals.

## **PART-2: USRP OPERATION**

On “Top\_tx.vi” and “Top\_rx.vi” front panels you can manipulate the following common values:

### **HW PARAMETERS TAB**



- a) **Device name:** Data type is IVI session of class niUSRP. This area is to enter the IP address of the connected USRP.
- b) **Active antenna:** Data type is string. This area is to enter the antenna name to be used.
- c) **Carrier frequency:** Data type is double [64-bit real (~15 digit precision)]. This area is to enter the frequency of RF signal in Hertz (Hz).
- d) **Gain:** Data type is double [64-bit real (~15 digit precision)]. This area is to enter the aggregate gain applied to the RF signal in decibel (dB).

#### MODULATION PARAMETERS TAB

- e) **Modulation type:** Data type is string. This chooses the modulation scheme to be used in both transmitter and receiver. It can be either BPSK or QPSK.
- f) **TX (or RX) oversample factor:** Data type is long [32-bit integer (-2147483648 to 2147483647)]. This specifies the number of desired samples per symbol for the pulse shaping filter. It must be an even number larger than 2.
- g) **TX (or RX) sample rate:** Data type is double [64-bit real (~15 digit precision)]. This specifies the transmitter sampling rate in Hertz (or samples per second). It must be specified between 195.312k and 100M.
- h) **Pulse shaping filter:** Data type is unsigned word [16-bit integer (0 to 65535)]. This selects the pulse shaping and matched filtering pulses to be used in both transmitter and receiver. It can be *Raised Cosine*, *Root Raised Cosine*, or *Gaussian*.
- i) **Filter parameter:** Data type is double [64-bit real (~15 digit precision)]. It sets the roll-off (or excess bandwidth) factor for Raised Cosine and Root Raised Cosine filters. It can be any real number between 0 and 1.
- j) **Training sequence type:** Data type is string. This chooses the training sequence to be used in both transmitter and receiver. It can be *Length 11 Barker Sequence*, *Length 16 Frank Sequence*, or *Neumann Hoffman Binary Code*.

The following items are only on top\_rx.vi:

- k) **Correct frequency offset:** Data type is Boolean [TRUE or FALSE]. This compensates the effect of the frequency offset between transmitted and received signals. It can be set as on or off.
- l) **Symbol timing recovery method:** Data type is string. This chooses the method to be used for symbol timing recovery. It can be any one of the following five options: “MT Align to Ideal Symbols”, “Max Energy”, “Early-Late Gate”, “Gardner”, or “Fixed Offset”.
- m) **Training sequence:** Data type is unsigned word [16-bit integer (0 to 65535)]. This chooses the training sequence to be used in the receiver. It can be *Barker*, *Frank*, or *Neumann H*.

In the graph fields of “top\_tx.vi” and “top\_rx.vi” you see the following graphs:

- 1) **(top\_tx) Transmitted constellation:** This graph shows the theoretical/ideal constellation points for the selected modulation scheme (two constellation points one at  $(I,Q)=(+1,0)$  and the other one at  $(-1,0)$  for BPSK; four constellation points one at  $(I,Q)=(+0.707,+0.707)$ , second at  $(+0.707,-0.707)$ , third at  $(-0.707,+0.707)$ , and fourth at  $(-0.707,-0.707)$  for QPSK).
- 2) **(top\_rx) Signal constellation:** This graph shows continuously changing I (in-phase or real) and Q (quadrature or imaginary) values of the constellation points as symbols received.
- 3) **(top\_tx and top\_rx) Eye diagram:** It is shown on both transmitting and receiving window. The transmitting window diagram is static, while receiving window diagram is dynamic.
- 4) **Received signal:** These two graphs show the Amplitude (in Volts) change of the received signal I and Q components in time domain.

In order to export the values of a graph or array on the simulation window, follow this path and pick the output format you need: Right click on the graph/array > Export > Export (clipboard/excel/simplified image).

In order to copy the current value of an indicator, right click on the indicator while simulation is running and select "Copy Data". This copies the current value as an image.

Also, if you need to run the simulator only for one loop, you can click on the single arrow ("Run").