BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 438 – Wireless Communication Laboratory

Experiment 01

Basic Digital Modulation Techniques

PART I: Modulation of digital data

Line codes map arbitrary binary bit streams to physical signal levels. However, as it is evident from the PSD of these line codes, power is mostly concentrated in low frequency bandwidth. These signals are not appropriate for communicating over long distances. Long distance communication typically requires signals to be sent at high frequencies. Binary data is hence sent using high frequency carriers. The act of sending binary data using high frequency carriers is called modulation. On the other hand, the act of retrieving a message signal from the modulated signal at the receiving side is called demodulation.

The most basic modulation techniques include-

- Binary Amplitude Shift Keying (BASK)
- Binary Phase Shift Keying (BPSK)
- Binary Frequency Shift Keying (BFSK)

The following table summarizes the modulation techniques for the aforementioned modulation schemes.

Modulation Technique	**Bit value**	**Representation**
BASK	0	$A_0 \sin(2\pi ft)$
	1	$A_1 \sin(2\pi ft)$
BPSK	0	$-A \sin(2\pi ft)$
	1	$A \sin(2\pi ft)$
BFSK	0	$A \sin(2\pi f_1 t)$
	1	$A \sin(2\pi f_2 t)$

In BASK we typically expect that $A_1 > A_0$. It should be noted in the case of BASK that if $A_0 = 0$, the modulation scheme becomes a special case of BASK called On-Off keying (OOK). It is so named because the transmitting circuit is sporadically turned on and off depending on the bit value.

The following image illustrates the modulated signals for BFSK, BPSK and BASK modulation techniques.

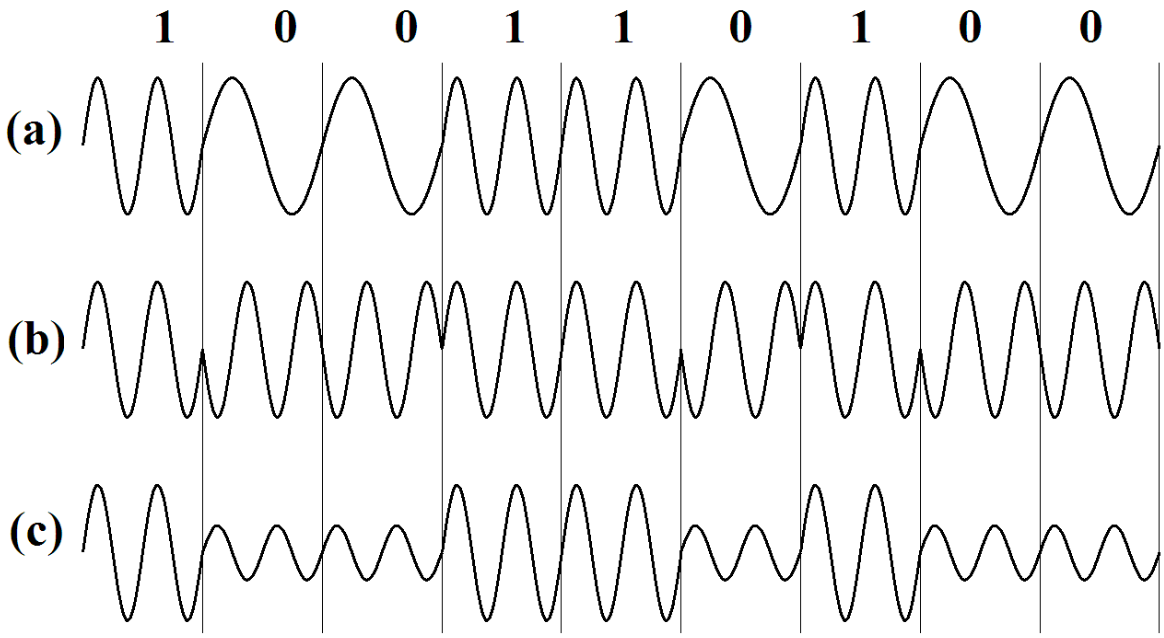


Figure 1: Modulated signal for (a) BFSK, (b) BPSK and (c) BASK

Lab Task 1

(a) Write MATLAB functions to generate the BASK, BPSK and BFSK modulated signals for a given bit stream. In each case, assume that the carrier frequency is an integer multiple of the bitrate.

(b) Take a large number of bits to obtain the PSD of BASK, BPSK and BFSK signals.

(c) In case of BFSK, vary f_1 and f_2 so that $|f_1 - f_2| = \frac{k}{T_b}$ for k $= \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2$. Observe the PSD in each case and comment on your observations.

PART II: Demodulation Techniques and Noise Performance

We can represent the BASK signal as

$$s_i(t) = m(b_i)\,\sin(2\pi f t)$$

where $m(b_i) = A_0$ or A_1 depending on the bit value. Typically, this signal passes through a noisy channel. We assume that the channel is an AWGN channel. So the received signal is given by

$$r_i(t) = s_i(t) + n_i(t)$$

At the receiving end, we mix the signal with the same carrier (phase matched). The resulting signal is given by

$$\begin{aligned} r_i(t)\,\sin(2\pi f t) &= m(b_i)\,\sin^2(2\pi f t) + n_i(t)\,\sin(2\pi f t) \\ &= \tfrac{1}{2} m(b_i) - \tfrac{1}{2} cos(4\pi f t) + n_i(t)\,\sin(2\pi f t) \end{aligned}$$

We integrate this mixed signal over one bit period to eliminate the high frequency terms to take the average value (this is equivalent to making the signal pass through an ideal low pass filter). Then we obtain-

$$\begin{aligned} R_i = \frac{1}{T_b} \int_{(i-1)T_b}^{iT_b} r_i(t) sin(2\pi f t) dt \;=\; \frac{1}{2} m(b_i) + \varepsilon_n \;&=\; \frac{1}{2} A_0 + \varepsilon_n \text{ if } b_i = 0 \\ &=\; \frac{1}{2} A_1 + \varepsilon_n \text{ if } b_i = 1 \end{aligned}$$

The term ε_n appears because of the noise term, it is a random variable with a mean value of 0 and standard deviation governed by the noise power.

Now we need to decide on the transmitted bit value. We compare the value R_i with a threshold of $\frac{1}{4}(A_0 + A_1)$ and decide in favor of bit 0 if $R_i < \frac{1}{4}(A_0 + A_1)$ and vice versa.

The following block diagram summarizes the demodulation and decision process for BASK modulated signals.

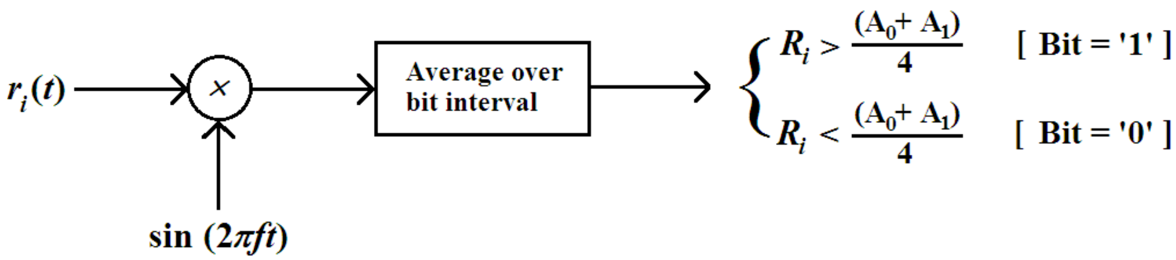


Figure 2: Demodulation and Decision Process for BASK modulated signal

Similar block diagrams for BPSK and BFSK modulated signals are given in the following figures.

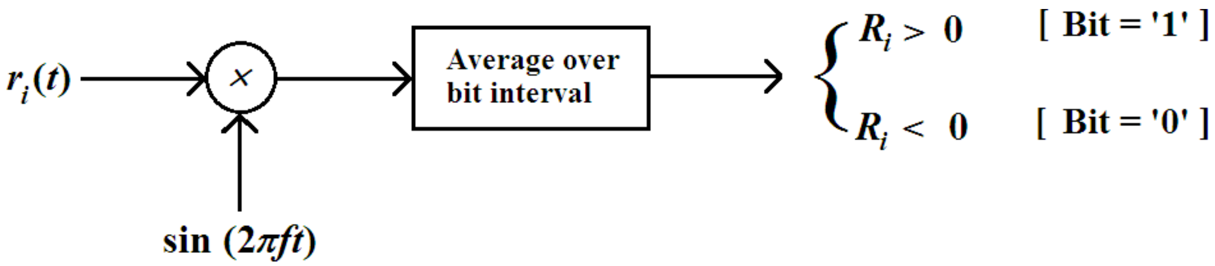


Figure 3: Demodulation and Decision Process for BPSK modulated signal

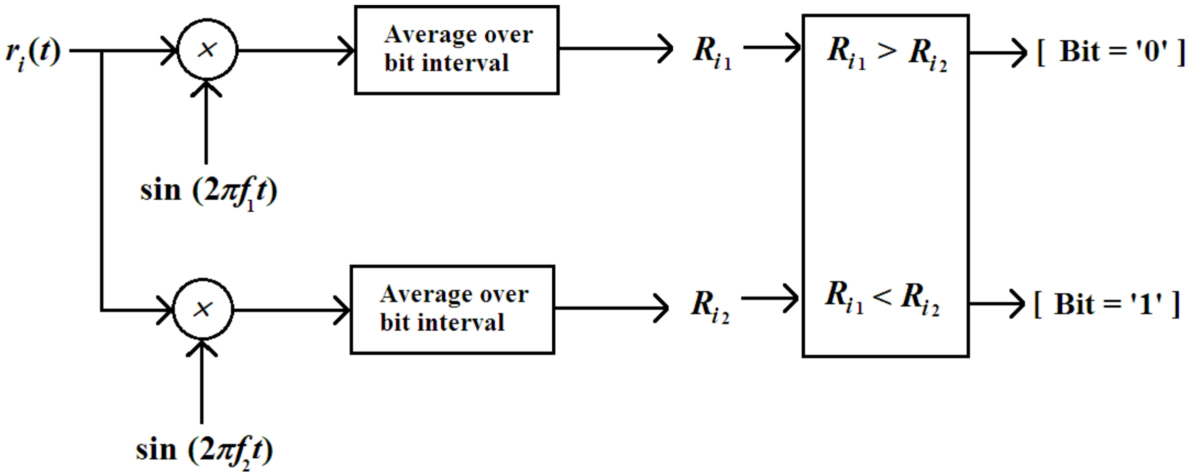


Figure 4: Demodulation and Decision Process for BFSK modulated signal

Lab Task 2

(a) Write MATLAB functions to demodulate BASK, BPSK and BFSK modulated signals.

(b) Obtain BER vs SNR curves for BASK, BPSK and BFSK signals and comment on their performance.

Prepared by: **Avik Roy**
Lecturer, Department of EEE, BUET

Supervised by: **Dr. Md. Saifur Rahman**
Professor, Department of EEE, BUET

PART III: Hardware Implementation in NI USRP 2901 with NI LabVIEW

Objective:

We will use NI LabVIEW and USRP devices to test basic digital modulation techniques over a real wireless channel and compare the results with earlier MATLAB simulation. This experiment helps students recognize practical implementation challenges, and relate theoretical knowledge to hardware-based communication systems.

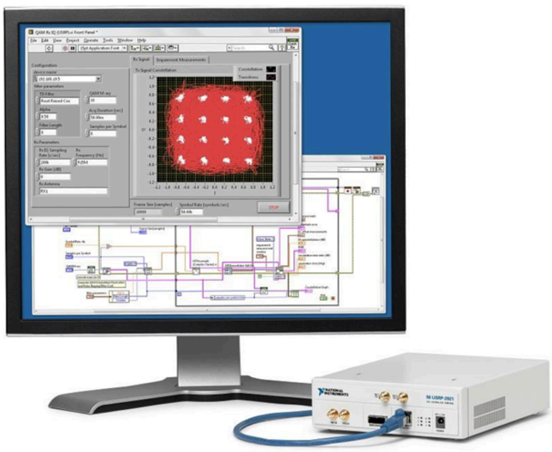
Brief Overview of NI USRP 2901

Software-Defined Radio (SDR) is a radio communication architecture in which many of the traditional radio functions—such as filtering, modulation, demodulation, and mixing—are implemented in software rather than being fixed in hardware. By changing only the software, the same hardware platform can support different modulation schemes, bandwidths, and communication standards.

The NI USRP family, including the USRP-2901 used in this laboratory, forms a flexible SDR platform. A typical NI USRP contains:

1. An analog RF transceiver front end
2. High-speed ADC and DAC for digitizing and generating RF signals
3. An FPGA for real-time digital signal processing
4. A high-speed host interface (USB/Ethernet) to a PC

At the transmitter, the USRP generates baseband I/Q samples, which are interpolated, upconverted to the RF carrier, and then sent to the RF output ports. At the receiver, incoming RF signals at the SMA antenna ports are mixed down to baseband I/Q, filtered and decimated, and then sent digitally to the host PC.



Hardware Setup in Wireless Communication Lab

See "**Exp 00 Introduction to LabVIEW and NI USRP**" for further info about these USRP devices.

Hardware Setup for Exp 01

For this experiment, we will use two NI USRP-2901 devices: one configured as the transmitter (lets say it's Device_A) and the other as the receiver (lets say it's Device_B). We will place them side by side or at a short distance. Both USRP are connected to the host PC through separate high-speed USB 3.0 cables.

The RF connections are as follows:

For the transmitter, we can use either the RF0 channel or RF1 channel of the transmitter USRP (Device_A). Make sure an antenna is connected to the TX1/RX1 port of the channel which you want to use.

For the receiver, we can use either the RF0 channel or RF1 channel of the receiver USRP (Device_B). Make sure an antenna is connected to either the TX1/RX1 or RX2 port of the channel which you want to use.

The antennas are positioned to have a clear line-of-sight path between them.

An example:
Tx = Device_A > RF0 > TX1/RX1
Rx = Device_B > RF0 > TX1/RX1

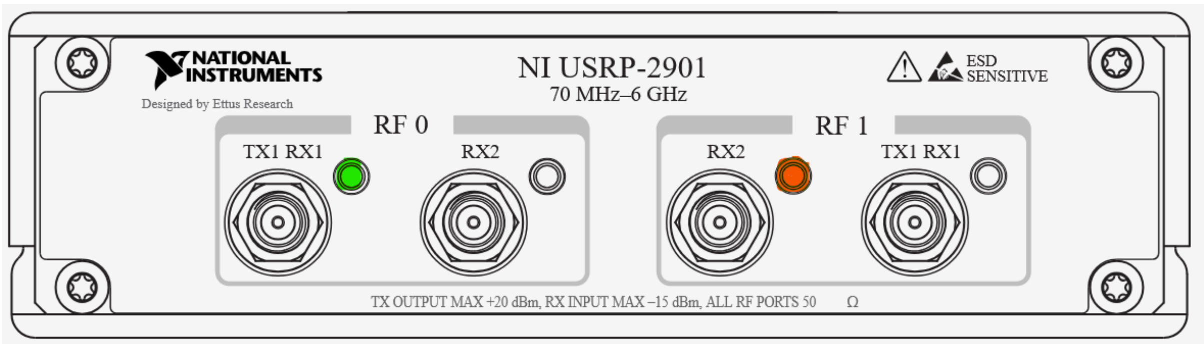
Or,

Tx = Device_A > RF0 > TX1/RX1
Rx = Device_B > RF0 > RX2

Or,
Tx = Device_A > RF0 > TX1/RX1
Rx = Device_B > RF1 > TX1/RX1

Or,

Tx = Device_A > RF0 > TX1/RX1
Rx = Device_B > RF1 > RX2



Note that, we can also use a single USRP device for this experiment. In that case, we may use one channel for transmitter and the other channel as receiver.
An example:

Tx = Device_A > RF0 > TX1/RX1

Rx = Device_A > RF1 > TX1/RX1

Or,

Tx = Device_A > RF0 > TX1/RX1
Rx = Device_A > RF1 > RX2

Software Description (LabVIEW)

The USRP hardware is controlled using NI LabVIEW Software. In this environment, the behavior of the transmitter and receiver is implemented using VI files.

So, we will use one VI file as transmitter and another VI file as receiver.

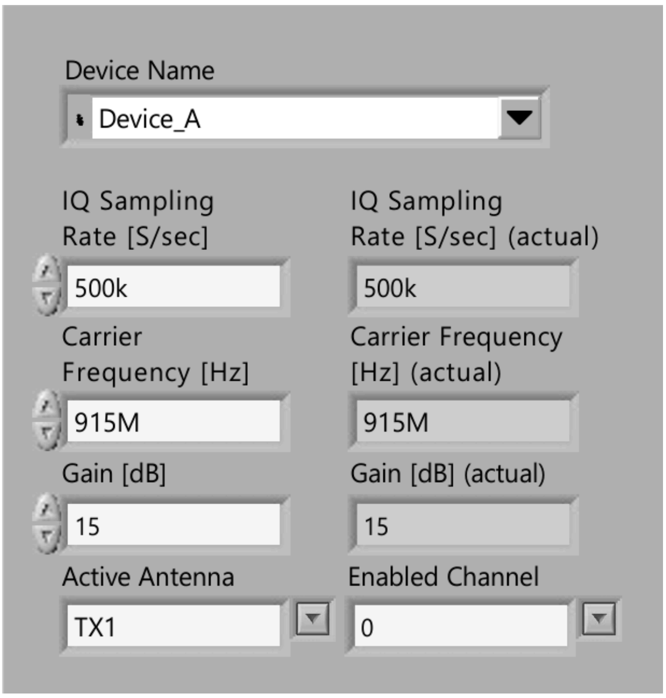
Each VI file has two main parts:

1. Front Panel – the graphical user interface containing controls, indicators, and graphs.
2. Block Diagram – the dataflow implementation connecting functional blocks and USRP interfaces.

Remember, the block diagram of the transmitter and receiver VIs is the actual programming that does processing of data to implement communication systems.

The front panels of the transmitter and receiver VIs contain several important controls and indicators that the student can adjust to observe their effects on the communication link.

Description of Front Panel Controls and Indicators (Transmitter)



1. **Device Name:** Write the USRP device name (e.g., Device_A or Device_B) you want to use as a transmitter. Make sure the device is connected to the PC. It must be the same as the name set in the USRP device (in this case “Device_A”).

2. **IQ Rate:** Baseband sampling frequency for In-Phase and Quadrature components used for transmission and reception. USRP devices can support upto 15 Msample/s for streaming data. A typical value for IQ Rate can be 500 ksample/s. Higher IQ rate increases the real-time sampling throughput, allowing wider bandwidth signals to be processed, but it also produces more samples per second, which requires more CPU and memory bandwidth to handle the data buffers without overflow.

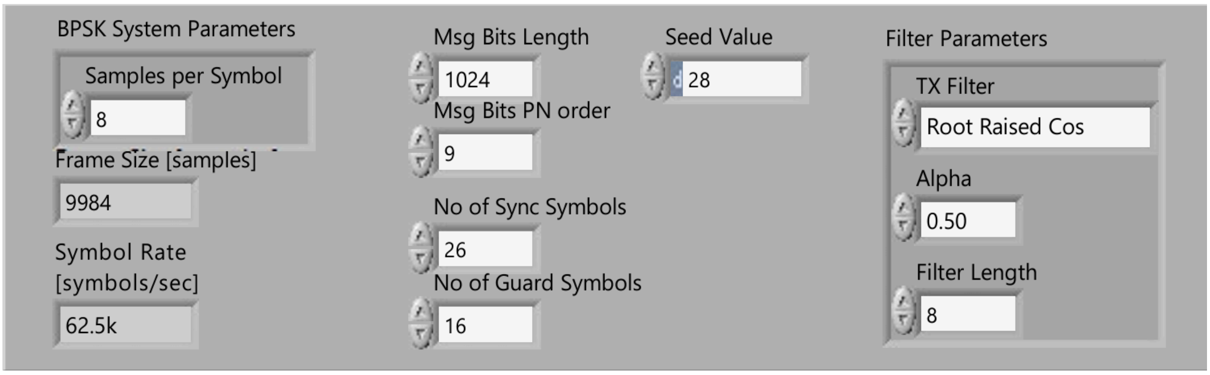
3. **Carrier frequency:** The RF center frequency, set by the local oscillator, at which the USRP transmits and receives. The NI USRP-2901 supports approximately 70 MHz to 6 GHz. In this lab, a carrier of 915 MHz is used, which lies in the ISM band and typically experiences relatively low background interference in indoor environments.

4. **Gain (dB):** Amplification factor of RF output power of the transmitter. Gain can vary from 0-50 dB. Higher values mean a stronger transmitted signal but with a higher risk of distortion and interference.

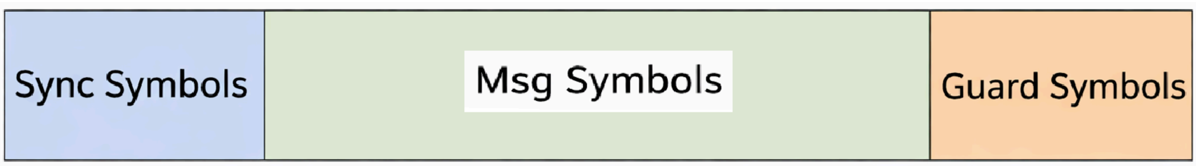
5. **Enabled Channel:** Selects which RF channel you want to use. It can be either 0 (RF0 channel) or 1 (RF1 Channel).

6. **Active Antenna:** Selects which RF port (TX1 or TX2) you want to use. Choose TX1 since USRP 2901 only has one transmitter antenna port (TX1) per RF channel.

Note that, for IQ Rate, Carrier Frequency and Gain [dB], if your chosen value is not supported by the USRP, the device will automatically select the nearest supported value and show it in the corresponding “IQ Rate [actual]”, “Carrier Frequency [actual]” and “Gain (dB) [actual]” field respectively.



We have constructed our transmitted signal frame as this-



7. **No of Sync Symbols:** Number of Symbols used for frame synchronization (to find the start of a frame) in receiver.

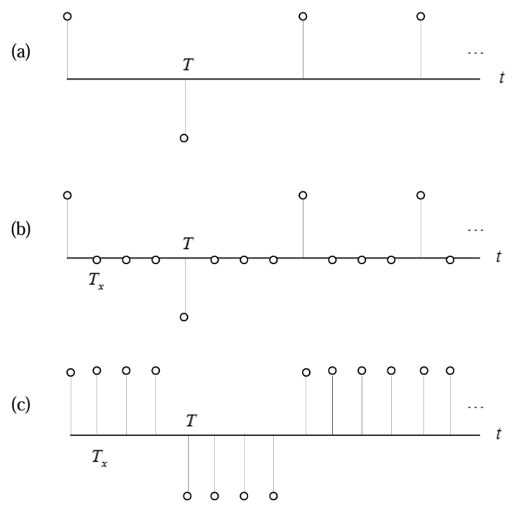
8. **Msg Bits Length:** Number of bits in actual message. Number of Msg Symbols = Number of Msg Bits / $\log_2 M$, where M is the modulation order.

9. **Msg Bits PN order:** Number of shift registers used to generate the random PN sequence as msg bits. The period of the msg bits sequence will be 2^N-1, where N is the Msg Bits PN

order.

10. **Seed Value:** Each seed value will generate a new PN sequence as msg bits.

11. **No of Guard Symbols:** Number of Symbols inserted at the end of a frame in order to separate two consecutive frames in the receiver.

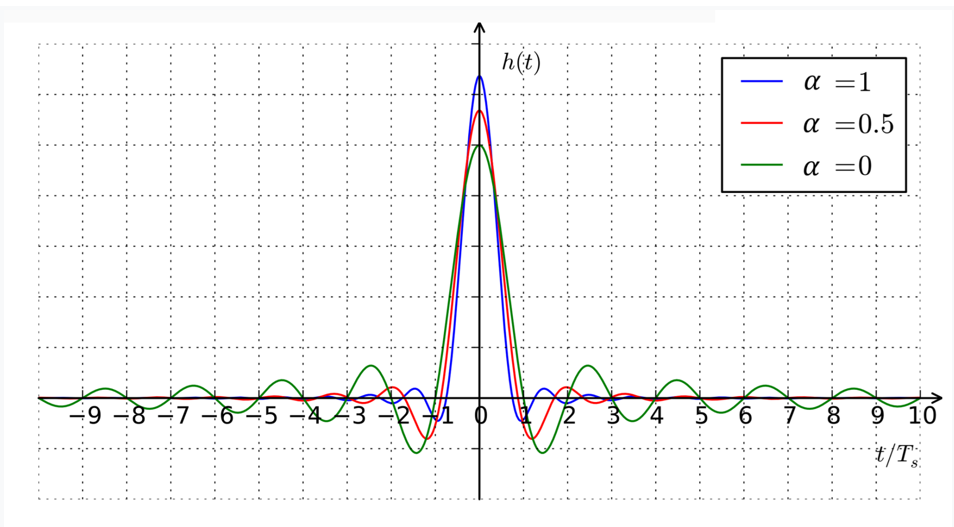


(a) Symbol sequence, (b) Upsampled symbol sequence, (c) Filtered upsampled symbol sequence.

12. **Sample per symbol:** Number of samples used to represent each symbol in the digital baseband signal. Typical values are 4, 6, 8, 12, etc. This parameter is considered as the upsampling factor. Higher values improve pulse-shaping accuracy and timing resolution but increase computational load.

13. Pulse-Shaping Filter Parameters:

Filter Name: There are 4 pulse shaping filters- none (rectangular), Raised cosine, Root Raised Cos and Gaussian. Root Raised Cos is a popular choice to eliminate ISI at the sampling points in the receiver. In that case, we need to use a Matched Filter RRC in the receiver to yield an overall raise cosine response.



The impulse response of a root-raised cosine filter multiplied by T_s, for three values of α : 1.0 (blue), 0.5 (red) and 0 (green).

Alpha (roll-off Factor): Controls the excess bandwidth, defining how much the spectrum spreads beyond the minimum theoretical bandwidth. Range: $0 \le \alpha \le 1$, $\alpha = 0$ gives a narrow, sinusoidal-like pulse, while $\alpha = 1$ gives a wider, flatter pulse. A typical value can be $\alpha = 0.5$

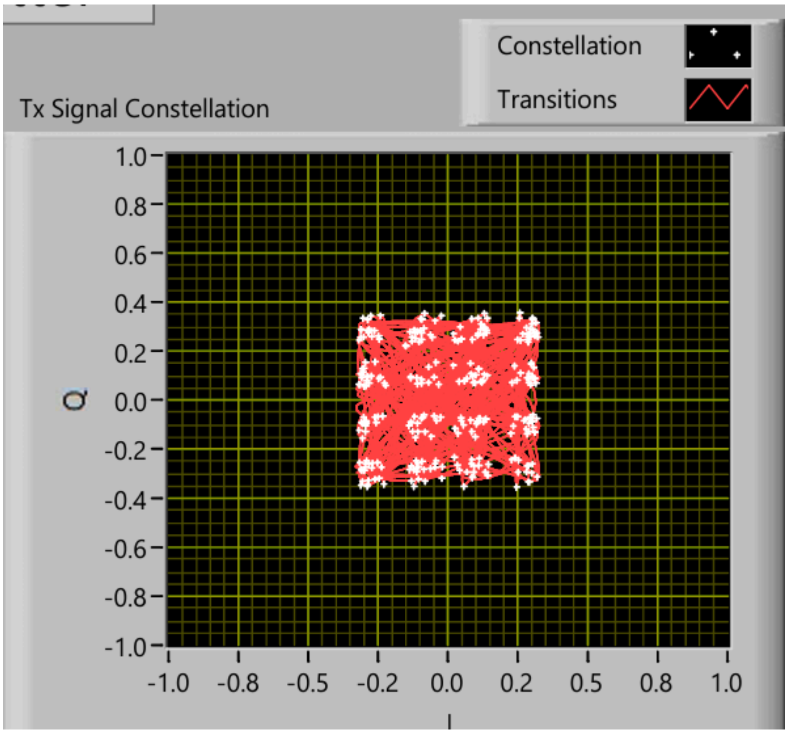
Filter Length (symbols): Number of symbol periods spanned by the filter impulse response; larger length improves approximation of the ideal response and ISI suppression, but increases computation and processing delay. A typical value can be 4, 6 or 8.

14. **Frame Size [samples]:** Transmitted signal frame size is calculated as this-

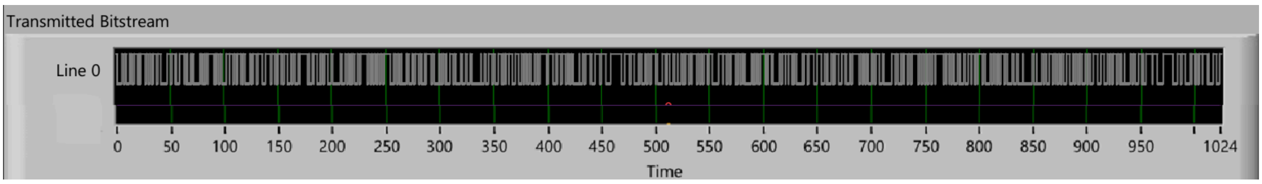
Frame size [samples] = Frame size [symbols] x Samples per symbol

15. **Symbol Rate:** Symbol rate is calculated as this-

$$\text{Symbol Rate [Symbols per second]} = \frac{IQ\ Rate\ (Samples\ per\ second)}{Samples\ per\ symbol}$$



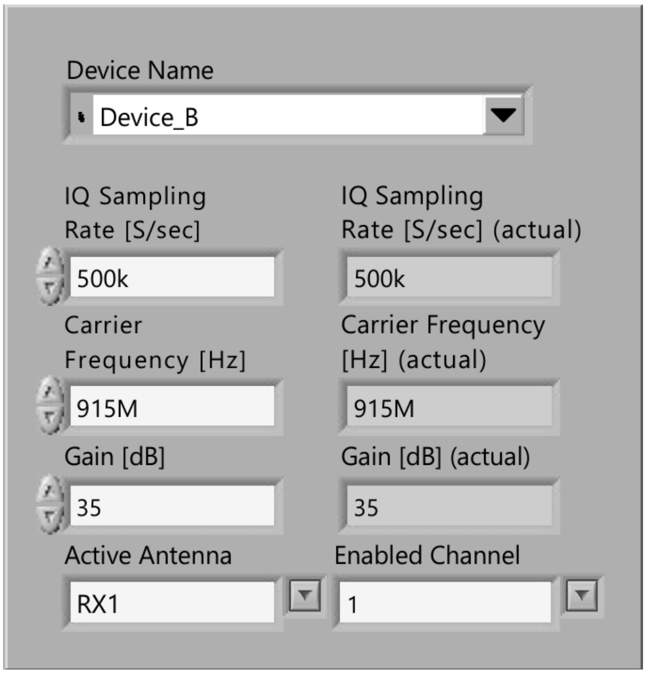
16. **Tx Constellation Diagram:** Displays transmitted symbols in the complex IQ plane.Here white dots represent the complex symbols and red lines represent the transition from one symbol to another.



17. **Transmitted Bitstream:** Shows the msg bits of the frame.

18. **Error Out:** Shows the error status and error code in case of any error. Also points out the source of that error.
19. **Stop:** Click this to stop the transmission.

Description of Front Panel Controls and Indicators (Receiver)



1. **Device Name:** Write the USRP device name (e.g., Device_A or Device_B) you want to use as a receiver. Make sure the device is connected to the PC. It must be the same as the name set in the USRP device (in this case "Device_B").

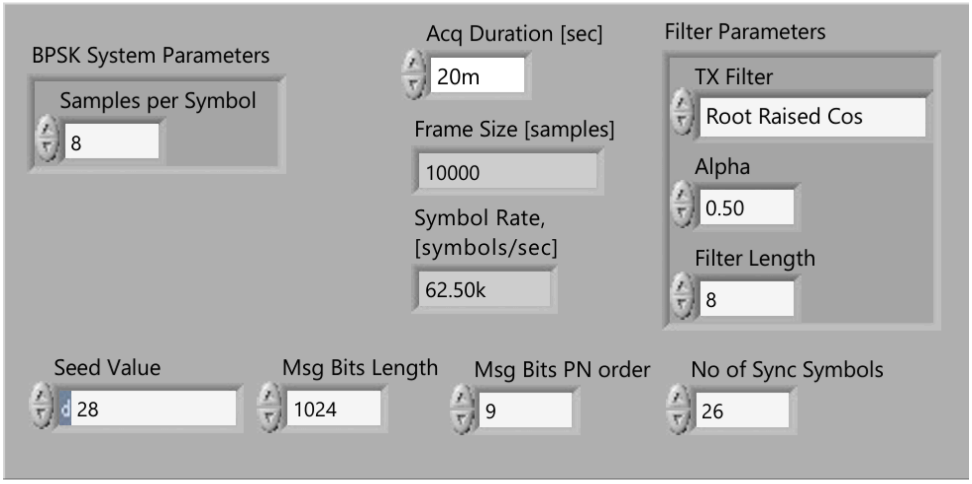
2. **IQ Rate:** Must use the same value as in the transmitter.

3. **Carrier frequency:** Must use the same value as in the transmitter.

4. **Gain (dB):** Amplification factor in the receiver front end. Can vary from 0-75 dB. Higher values make weak signals easier to detect but increase noise and risk of clipping.

5. **Enabled Channel:** Selects which RF channel you want to use. It can be either 0 (RF0 channel) or 1 (RF1 Channel)

6. **Active Antenna:** Selects which RF port (RX1 or RX2) you want to use to receive the signals.



7. **No of Sync Symbols:** Must use the same value as in the transmitter. This sync symbol sequence is known at the receiver which is used for frame synchronization.

8. **Msg Bits Length:** Must use the same value as in the transmitter.

9. **Msg Bits PN Order:** Must use the same value as in the transmitter.

10. **Seed Value:** Must use the same value as in the transmitter.

    Note that 8,9,10 are used to make a copy of the transmitted signal in the receiver file to calculate the bit error rate.

11. **Samples per symbol:** Must use the same value as in the transmitter.

12. **Matched Filter Parameters:** Must use the same values as in the transmitter.

13. **Symbol Rate:** Calculated in the same way as in the transmitter.

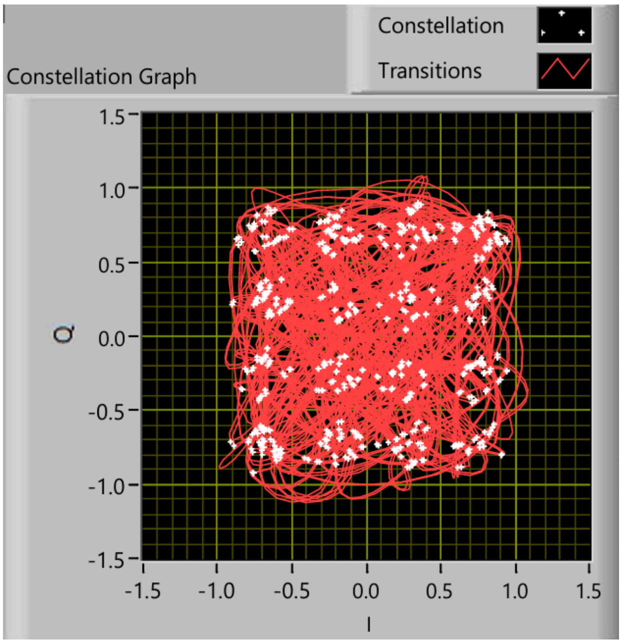
    Symbol Rate [Symbols per second] = $\dfrac{IQ\ Rate\ (Samples\ per\ second)}{Samples\ per\ symbol}$

14. **Acq Duration [sec]:** Duration of time for which the receiver USRP captures and stores IQ samples. Higher values can help capture longer segments of signals but increases buffer size and processing time. Lower values can be used to capture shorter segments but parts of the frame can be lost. A typical value can be 20 ms.

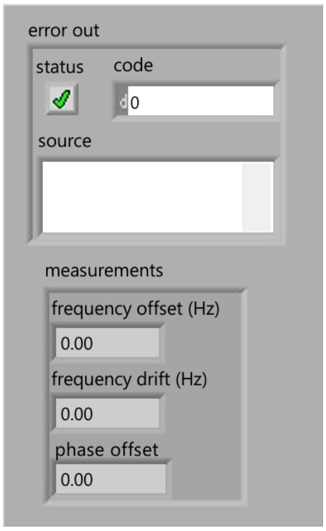
15. **Frame Size [samples]:** Frame size [samples] is calculated as this-

    Frame Size [samples] = Acq Duration [sec] x IQ Rate [samples per sec]

    Note that this frame size is not the same size as the transmitted signal frame size in the transmitter. Rather, this frame size must be equal or greater than twice of the transmitted signal frame size. So, adjust the Acq duration accordingly.

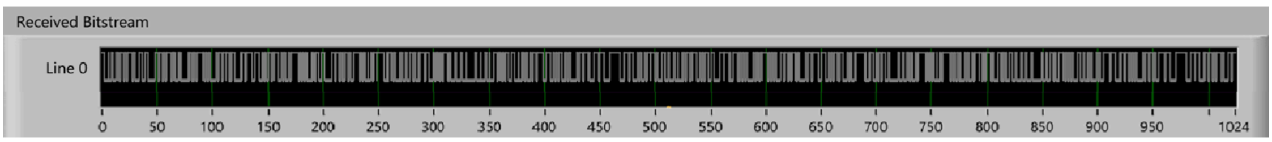


16. **Rx Constellation Diagram:** Displays received symbols in the complex IQ plane after demodulation.

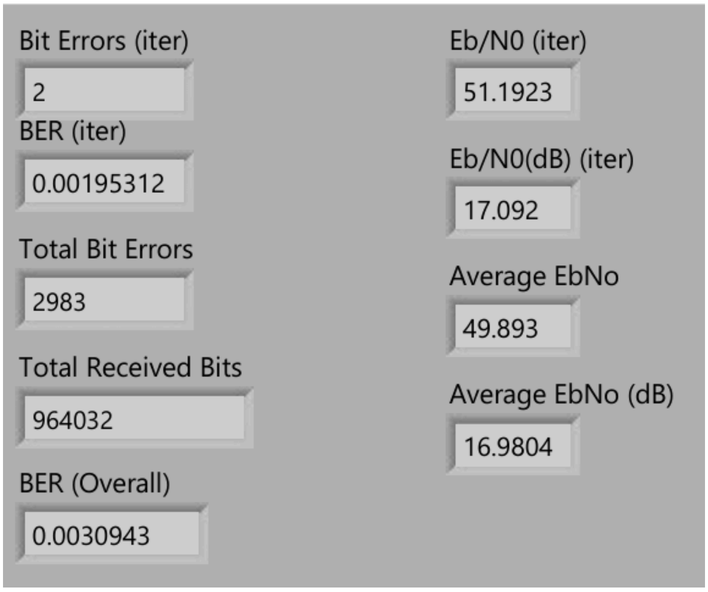


17. **Error Out:** Shows the error status and error code in case of any error. Also points out the source of that error.

18. **Measurements:** Shows some parameters which represent the quality of the received signal.



19. **Received Bitstream:** Shows the received msg bits.



20. **Bit Errors (iteration):** Shows number of bit errors in each iteration of receiver code. Suppose, transmitter is continuously sending 1024 bits long msg. In a certain iteration of receiver USRP, 1010 bits are correctly received and 14 bits are wrong. In that case, the bit error of this iteration is 14. This value will change in each iteration.

21. **BER (iteration):** Shows the BER of each iteration. In the previous example, BER (iter) will be 14/1024 = 0.0137. This value will also change in each iteration.

22. **Eb/N0 (iter):** Shows the estimated energy per bit to noise power spectral density ratio in each iteration.

23. **Total Bit Errors:** Shows total number of bit errors from the moment of starting the receiver (Summation of Bit Errors (iter) across all iterations)

24. **Total Received Bits:** Shows total number of received bits from the moment of starting the receiver (Summation of Received Bits (iter) across all iterations)

25. **BER (Overall):** Shows the overall BER (Total Bit Errors / Total Received Bits)

26. **Average Eb/N0:** Shows the average value of Eb/N0 (iter) across all iterations.

Working Steps for EXP 01

1. Place two USRP devices side by side. Ensure antennas are connected to the USRP devices. Connect both USRP devices with the PC through USB 3.0 cables.

2. Open NI USRP Configuration Utility on the PC and check whether the USRP devices showed up in the list. If not, check the USB connection and click “Refresh Device List”.

3. Open two VI files with LabVIEW-

   For BPSK, the BPSK Transmitter VI and the BPSK Receiver VI files.
   For BASK, the BASK Transmitter VI and the BASK Receiver VI files.
   For BFSK, the BFSK Transmitter VI and the BFSK Receiver VI files.

4. Write/edit the parameters in both files as instructed above.
5. Run the transmitter VI file first, then run the receiver VI file. You will be able to see the transmitted bit stream, Tx constellation window in the transmitted VI file as well as the received bit stream, Rx constellation window, BER, Eb/N0 in the receiver VI file.
6. Stop the receiver VI file first, then stop the transmitter VI file.
7. Change some parameters in the VI files and run them again.

Lab Task 3

(a) Data Collection

BPSK:

Set the transmitter gain to 0, 5, 10, 15, 20, 25 dB.
Keep the receiver gain fixed at 30 dB for all experiments.
For each transmitter gain value:

1. Run both the Transmitter VI and Receiver VI.
2. Take a screenshot of the PSD of the transmitted signal in the transmitter VI.
3. Take a screenshot of the Rx Constellation Diagram in the receiver VI.
4. Record the average BER and corresponding Eb/N0 (dB) observed at the receiver VI (Wait for at least 10-15 seconds for each gain setting before recording values to ensure sufficient data has been transmitted for reliable averaging)

Tx Gain [dB]	Rx Average Eb/No [dB]	Rx Average BER
0		
5		
10		
15		
20		
25		

BASK: Do the same for BASK.
BFSK: Do the same for BFSK.

(b) Using MATLAB:

- Plot BER vs. Eb/N0 (dB) for BASK, BPSK, BFSK on the same figure.
- Plot BER vs. Transmitter Gain (dB) for BASK, BPSK, BFSK on the same figure.

Use a logarithmic scale for the BER axis and clearly label all axes and legends.
Attach these figures to your report.

(c) Are there any discrepancies between the PSD, constellation diagram, and BER vs. Eb/N0 results obtained from the **LabVIEW-USRP experiments** and the **MATLAB simulation**? If yes, explain the possible reasons.

Acknowledgement

“PART III: Hardware Implementation in NI USRP 2901 with NI LabVIEW” is prepared by Kazi Ahmed Akbar Munim and S. M. Azmain Awsaf under the supervision of Dr. Lutfa Akter and Dr. Md. Forkan Uddin at the Department of EEE, BUET, on 28/12/2025.