Alexandria University

Faculty of Engineering

Electrical Department

Image transfer and Software Defined Radio using USRP and GNU Radio

ID	Name
209	Mostafa Mahmoud Khalil
104	Safaa Mohamed Gamal
107	Abd El-Rahman Ibrahim Rezk
228	Hagar Mohamed Gamal
232	Hesham Gaber Ahmed

Submitted to: Dr. Ahmed Saied

Content:

- **Overview**
- **BSPK modulation**
- **QPSK modulation**
- **File transfer Gnuradio (test)**
- **Image transfer Gnuradio (main)**
- **QPSK Wolf- Mathematica implementation**

Overview:

Our goal in this project is to perform wireless transfer of image data, in JPEG format, using the SDR (Software Defined Radio) paradigm. Figure 1 shows a block diagram of a typical SDR setup.

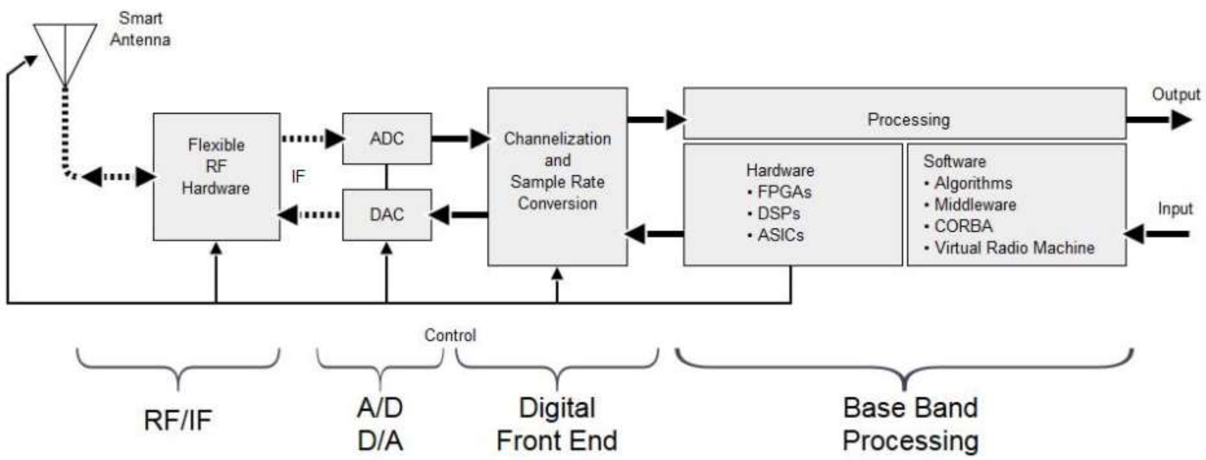

Figure 1

We modulate the in-phase and quadrature components of a sinusoid to represent the information (an image) which we wish to send. We will utilize a technique called Phase Shift Keying (PSK). In PSK, the transmitter does not vary the amplitude or the frequency of the sinusoid being transmitted. Instead, it varies the phase according to the input bit sequence.

BPSK (Binary Phase Shift Keying) modulation:

BPSK represents each singular bit as a symbol, i.e. it transmits one of two possible phases corresponding to the bit value. For example, binary 0 is mapped to a phase offset of π=2 , i.e. 1j in complex notation. Binary 1 is mapped to a phase offset of 3π=2 , i.e. -1j in complex notation. This symbol mapping is shown in figure 2. Mathematically, we represent the modulated signal for one bit as:

$$s_b(t) = \cos(2\pi f_m t + \varphi_b)$$

where φ_b is the phase shift which represents $b = 0$ or 1 and f_m is the frequency of the baseband modulation.

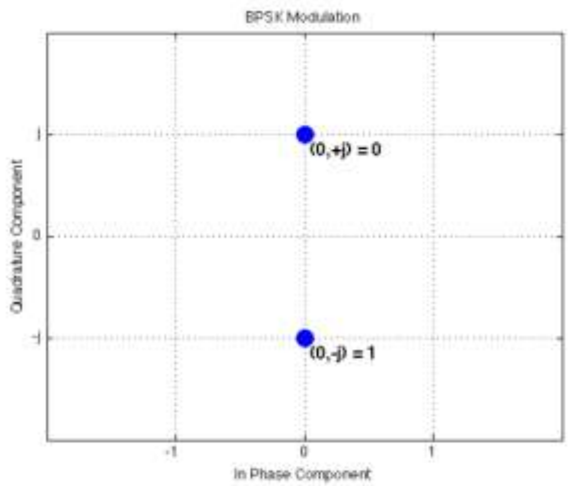

Fig.2 BPSK symbol map

QPSK (Quadrature Phase Shift Keying) modulation:

Quadrature phase shift keying (QPSK) is another modulation technique, and it's a particularly interesting one because it actually transmits two bits per symbol. A QPSK symbol doesn't represent 0 or 1. It represents 00, 01, 10, or 11.

We have 360° of phase to work with and four phase states, and thus the separation should be 360°/4 = 90°. So our four QPSK phase shifts are 45°, 135°, 225°, and 315°.

There's another reason why it makes sense to choose 45°, 135°, 225°, and 315°: they are easily generated using I/Q modulation techniques because summing I and Q signals that are either inverted or non-inverted results in these four phase shifts. The following table should clarify this:

I	Q	Phase shift of I+Q
Non-inverted (0)	Non-inverted (0)	45
Inverted (1)	Non-inverted (0)	135
Inverted (1)	Inverted (1)	225
Non-inverted (0)	Inverted (1)	315

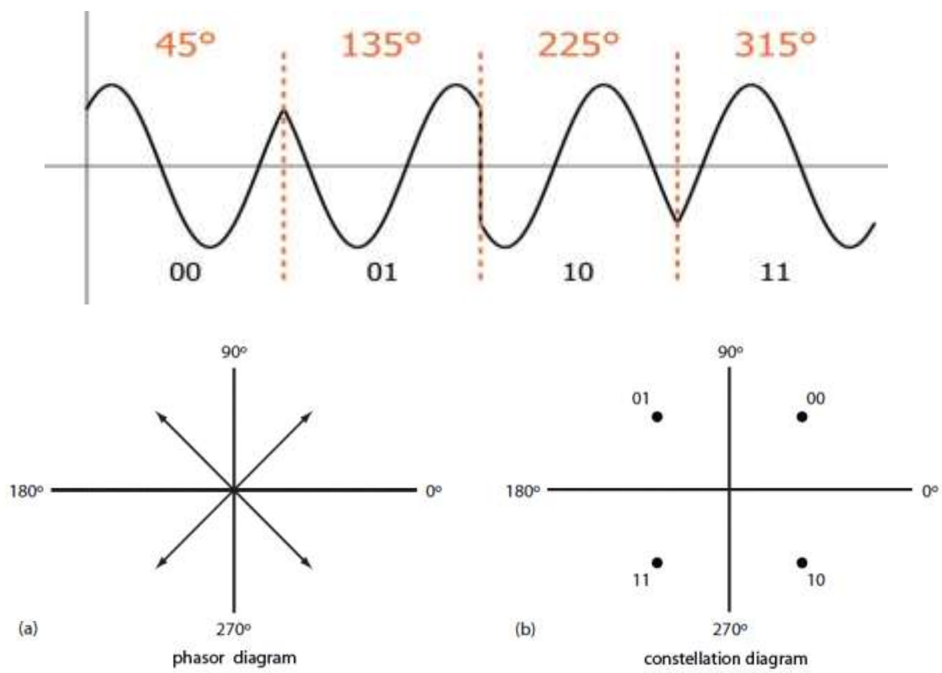

Fig. 3 shows QPSK with a phasor diagram and a constellation diagram.

Why using **Gray Coding?** According to the plot showing the QPSK symbol map in figure 3. We notice that the symbols which are diagonal from each other are further apart than the symbols which are directly above/below or next to each other. Over communication channels with noise, sooner or later the receiver will make a mistake deciding which symbol was sent. For QPSK modulation, it's possible that we could choose a symbol in which both bits are guessed incorrectly, which is twice as bad as guessing only 1 bit incorrectly. In Gray coding, we map the bits such that the nearest neighboring symbols differ by only one bit. Thus, for the more likely incorrect guesses of the transmitted symbol, we make a mistake in only **one bit**. For this assignment, our gray code mapping is as follows:

- 00 goes to 0.707 + 0.707i
- 01 goes to 0.707 – 0.707i
- 10 goes to -0.707 + 0.707i
- 11 goes to -0.707 – 0.707i

File transfer Gnuradio (test):

We generate .txt file to test the PSK modulation. Vector source is used to generate ABC in ASCII code. We assume 0 noise. The output will be in the text file like this (ABC ABC ABC …etc.) As we notice there is no error due to no noise, but if we assume a high value for noise we will not receive the text file as we wanted.

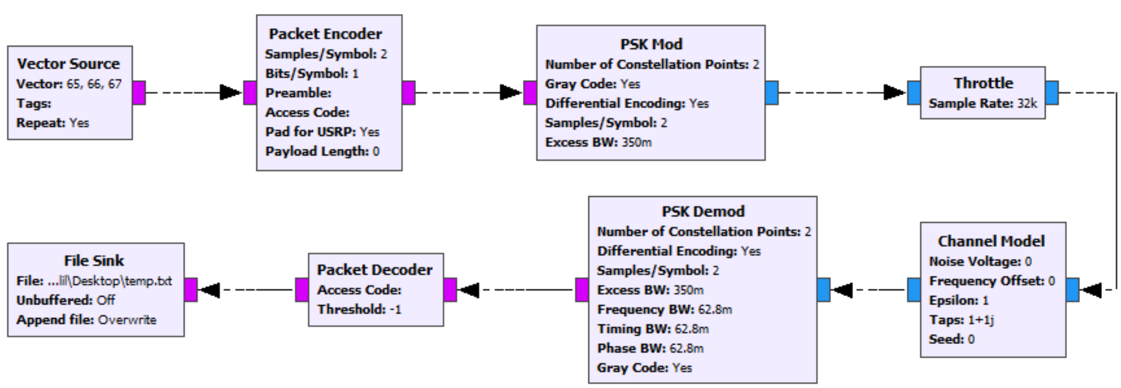

Fig. 4 gnuradio .txt file generation (test)

Image transfer Gnuradio (main):

Now we will transfer an image using PSK modulation and USRP sink to transfer the image. The image will be converted to bits 0s and 1s. Then it will be passed through PSK modulation then USRP sink. A multiply constant is used to avoid error in USRP.

The block diagram will be as follow:

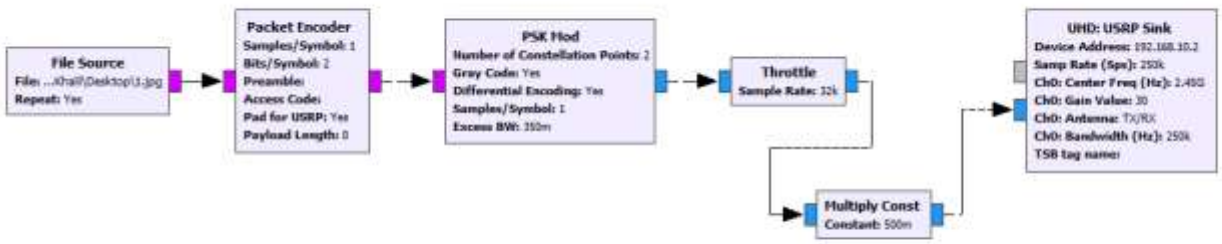

Fig. 5 Image transmitter using USRP Gnuradio

The receiver will be:

The received signals will be passed through PSK demodulation then the Packet decoder will convert the bits to back the real image.

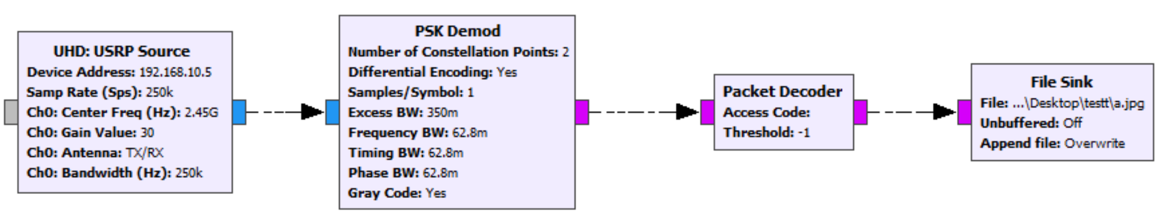

Fig. 6 Image Receiver using USRP Gnuradio

QPSK Wolf- Mathematica implementation:

Digital Modulation: Quadrature Phase-Shift Keying (QPSK) Signal Constellation and Eye Diagrams.

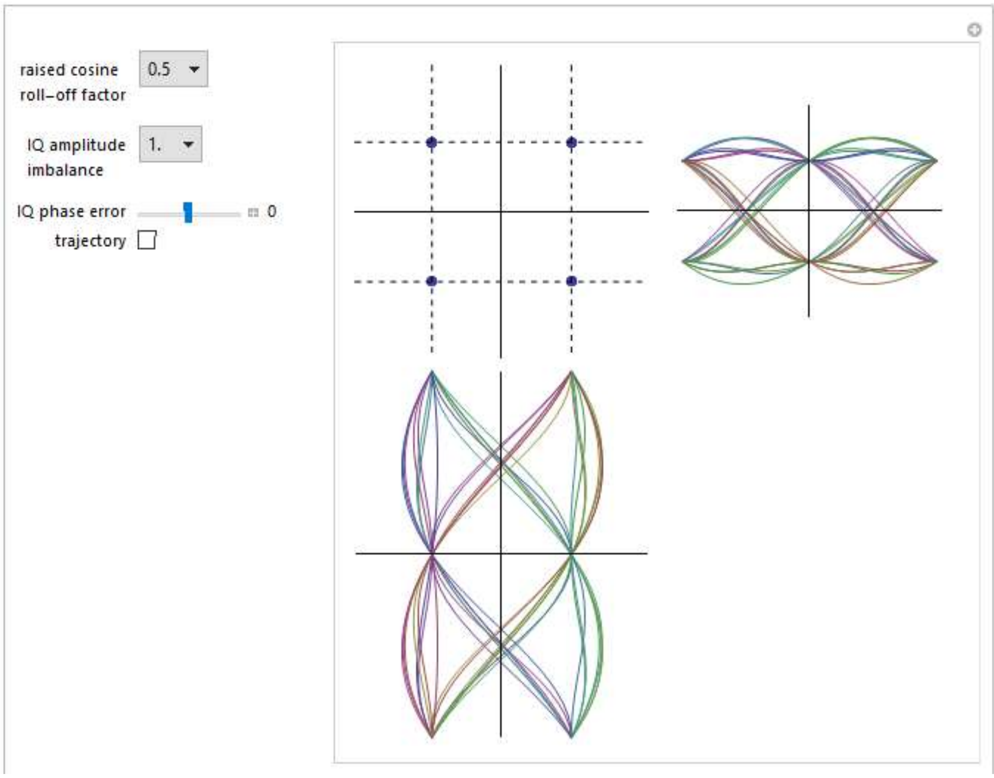

Fig. 7 Eye diagram

As we increase the raised cosine factor, the eye diagram become smoother.

<u>IQ amplitude imbalance (ideal case 1)</u>

When we decrease its value the upper bits become closer to each other and the lower bits too. But when we increase its value the right bits become closer to each other and the left ones.

<u>IQ phase error (ideal case 1)</u>

When its value > 0 the bits (00 and 11) become closer to each other and the bits (01 and 10) expand and vice versa.

Appendix for gnuradio:

File Source: read the input file

File Sink: drop the output file

Vector source: create a vector that can be transmitted, It is written in ASCII

Example: 65-66-67 (ABC)

Packet Encoder: Block is used with modulations (gmsk, dpsk, qam)

Preamble: string of 1's and 0's, leave blank for automatic.

It's a startup value for receiver and transmitter

Access Code: string of 1's and 0's, leave blank for automatic.

Bits/Symbol:

dbpsk -> 1

dqpsk -> 2

PSK modulation: modulate the signal using PSK modulation

Throttle: It settles the sample rate at a specific value to make the average rate does not exceed samples per sec. If we did not put it the program will try to put the maximum sample rate it can proceed which will lead to an error.

Limits the data throughput to the specified sampling rate. This prevents GNURadio from consuming all CPU resources

Channel Model: the channel which is used to transmit the file.

Noise voltage: to determine how much noise we need. It's better to make it as minimum as possible because if we increase the noise, the receiver will not be able to receive the file.

Epsilon: When pulse goes through the channel, epsilon doesn't make it sharp edged but go down with a slope.

Frequency offset: to match both frequencies in receiver and transmitter.

PSK Demodulation:

Number of constellation is 4 because qpsk spread the bits in four probabilities 00 01 10 11

Packet decoder: converts the received bits to the original image.