**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY**

---------------o0o---------------

****

**DIGITAL SIGNAL PROCESSING LABORATORY**

**MAJOR PROJECT**

**Quadrature Phase Shift Keying**

Instructor: **Prof. Dr Lê Tiến Thường**

Group 1:

Trình Nguyễn Anh Tuấn 1513879 Number 1

Phạm Quốc Kiên 1551077 Number 2

Class-day: Thursday

Digital Signal Processing Lab  
Major Project: Quadrature Phase Shift Keying (C5515)

Instructor: **Prof. Dr Lê Tiến Thường**

Group 1:

Trình Nguyễn Anh Tuấn 1513879 number 1

Phạm Quốc Kiên 1551077 number 2

Class-day: Thursday 7-9

Table of Contents

[Abstract 4](#_Toc25929613)

[**I.** **Introduction** 5](#_Toc25929614)

[**1.** **Summary** 5](#_Toc25929615)

[**2.** **Aim** 5](#_Toc25929616)

[**II.** **Theory** 5](#_Toc25929617)

[**1.** **Modulation** 5](#_Toc25929618)

[**2.** **Phase Shift Keying (PSK)** 6](#_Toc25929619)

[**a)** **Binary Phase Shift Keying (BPSK)** 7](#_Toc25929620)

[**b)** **Quadrature Phase Shift Keying (QPSK)** 9](#_Toc25929621)

[**3.** **Demodulation** 18](#_Toc25929622)

[**III.** **Simulation** 19](#_Toc25929623)

[**1.** **General Flowchart** 19](#_Toc25929624)

[**2.** **MATLAB Simulation** 20](#_Toc25929625)

[**a)** **Flowchart** 20](#_Toc25929626)

[**b)** **Source code** 20](#_Toc25929627)

[**c)** **Results** 23](#_Toc25929628)

[**3.** **CCSv4 (TMS320C5515) Simulation** 25](#_Toc25929629)

[**a)** **Flow chart** 25](#_Toc25929630)

[**b)** **Source code** 26](#_Toc25929631)

[**c)** **Result** 31](#_Toc25929632)

[**4.** **Compare results** 33](#_Toc25929633)

[**5.** **Compare sending signal and receiving signal** 35](#_Toc25929634)

[**a)** **Flow chart of demodulator** 35](#_Toc25929635)

[**b)** **Source code (MATLAB)** 37](#_Toc25929636)

[**c)** **Result** 39](#_Toc25929637)

[**IV.** **Conclusion** 41](#_Toc25929638)

[References 42](#_Toc25929639)

Abstract

Quadrature Phase-Shift Keying (QPSK) plays an important role in modern digital communication systems, it is one of the most widely used modulation scheme in transmission of signal from transmitter to receiver with high bandwidth efficiency. This report presents a comparative study of QPSK modulation, which has been simulated using MATLAB and CCSv4 (TMS320C5515).



1. **Introduction**
   1. **Summary**

Communications systems can be defined as transferring data from one place to another. If a signal is to be transmitted over some distance, it must be transformed or modulated to conform the required characteristics of transmission medium. This might be a wired connection, or a wireless/radio link, and optical. This report will be concentrated on QPSK, which is one of most popular digital techniques used satellite communication and sending data over cable networks, videoconferencing, cellular phone systems, and other form of digital communication over Radio Frequency (RF) carrier.

* 1. **Aim**

Study on one of digital techniques, quadrature phase shift keying, to broaden intellectual horizons about modulation techniques.

Simulator and compare quadrature phase shift keying signal in the MATLAB and CCSv4 (TMS320C5515).

1. **Theory**
2. **Modulation**

In electronic and telecommunications, modulation is process by which some characteristic of carrier wave in varied in accordance with an information bearing signal. Modulation technique is used to reduce the antenna height and also increase the range of communication, multiplexing and improve quality of reception. The modulation schemes are broadly classified into two categories analog modulation and digital modulation.

Analog modulation refers to the process of transferring an analog baseband (low frequency) signal, like audio or TV signal over a high frequency signal such as a radio frequency band. There are two kinds of analog modulation Amplitude Modulation (AM) and Frequency Modulation (FM).

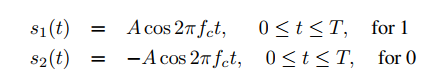
Digital modulation is a process that impresses a digital symbol into a signal suitable for transmission over a wired or wireless medium in order to receive that signal at receiving end correctly without any loss of information. There are three types of digital modulation Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) and Phase Shift Keying (PSK).   
 In next part is main part my group research, Phase Shift Keying, which will be discussed further.

1. **Phase Shift Keying (PSK)**

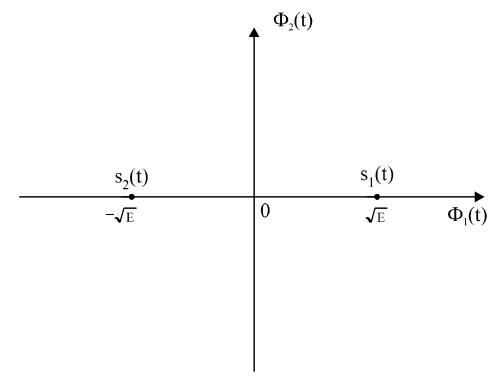
Phase Shift Keying is a large class of digital modulation schemes. PSK is widely used in the communication industry, which is one of the most popular techniques used for satellite communications and sending data over cable networks, videoconferencing, cellular phone system, and other form of digital communication over Radio Frequency carrier. First my team present coherent binary PSK and we discuss in great detail quadrature PSK.

* 1. **Binary Phase Shift Keying (BPSK)**

The simplest PSK is binary PSK, in which it has only two signal elements 0 and 180.

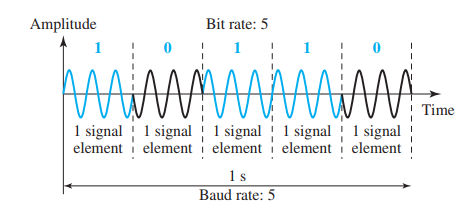


These signals are called antipodal. We should choose a correlation coefficient of -1, which leads to the minimum error probability.



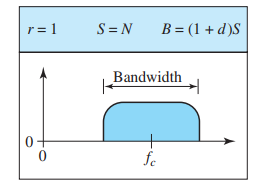
*Constellation Diagram.*

One of advantage of BPSK is less susceptible to noise. However, PSK needs more sophisticated hardware to be able to distinguish between phase. The waveform of a BPSK signal generated by modulator for a data steam [10110]. The waveform has a constant envelope like Frequency Shift Keying (FSK). Its frequency is constant too. In general, the phase is not continuous at bit boundaries.



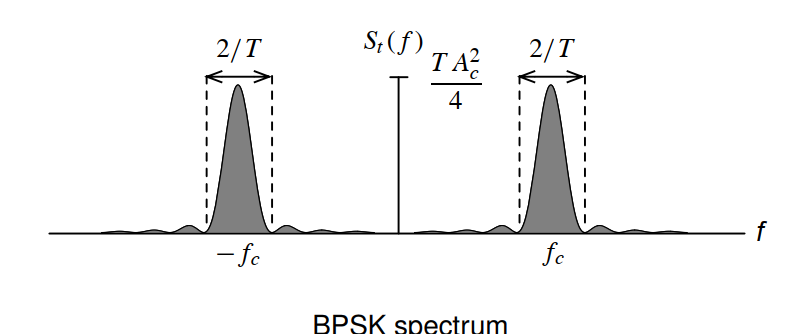
*Binary Phase Shift Keying Modulation Waveform.*

The bandwidth of BPSK is the same as that for binary Amplitude Shift Keying (ASK), but less than for Binary Frequency Shift Keying (BFSK). No bandwidth is wasted for separating two carrier signals.

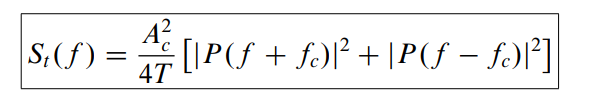


*Bandwidth of BPSK.*

Because the carrier is a pure sinusoid, the carrier spectrum is an impulse located at the carrier frequency. Convolution of any spectrum with a frequency impulse centers this spectrum about the frequency of the impulse. Therefore, the BPSK spectrum is the spectrum of the baseband symbols, centered about the carrier frequency. This waveform contains rectangular pulses having widths that are integer multiples of one symbol, T. This creates a spectrum that contains not only the fundamental symbol frequency and its odd harmonics, but also all integer sub-harmonics of the fundamental, along with their odd harmonics.



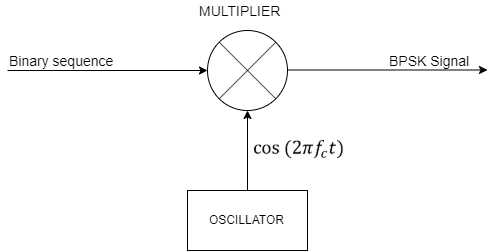
*Spectrum of BPSK.*



* 1. **Quadrature Phase Shift Keying (QPSK)**

QPSK is the most often used scheme since it does not suffer Bit Error Rate (BER) degradation while the bandwidth efficiency is increased. The scheme is called QPSK because it uses two separate BPSK modulations, one is in-phase, the other quadrature (out-of-phase). QPSK is one of the modulations schemes used in wireless communication system due to its ability to transmit twice the data rate fir a given bandwidth.

The coming bits are first passed through a serial-to-parallel conversion that sends one bit to one modulator and the next bit to the other modulator.

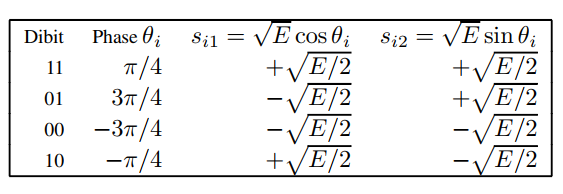


*BPSK Block Diagram.*

The signal is defined as



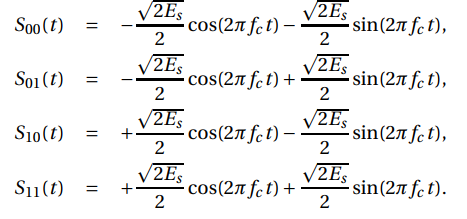
My team decided to use Grey code to map four phases of QPSK.



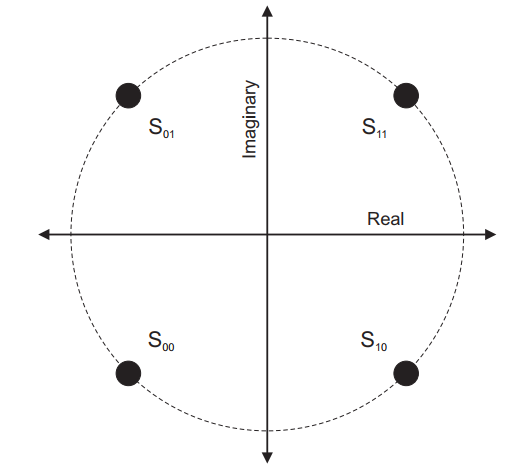
Using a trigonometric conversion equation:

,

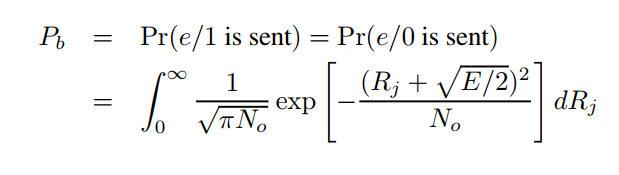
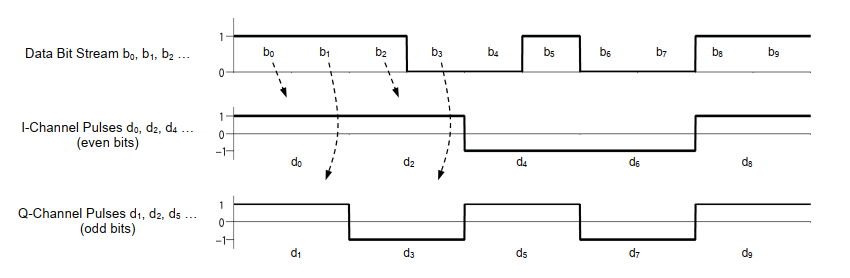
we have two separately ordinary BPSK signals for each case, one is in-phase BPSK and quadrature BPSK.



In constellation diagram, a signal element type is represented as a dot, the diagram has two axes. The ideal PSK constellation has 4 equidistant phase state and constant amplitude, the phases are separated by 90 degrees. The horizontal x-axis is related to the in-phase carrier (, the vertical y-axis is related to the quadrature carrier (). The starting point for grouping bits into dibits is completely arbitrary.

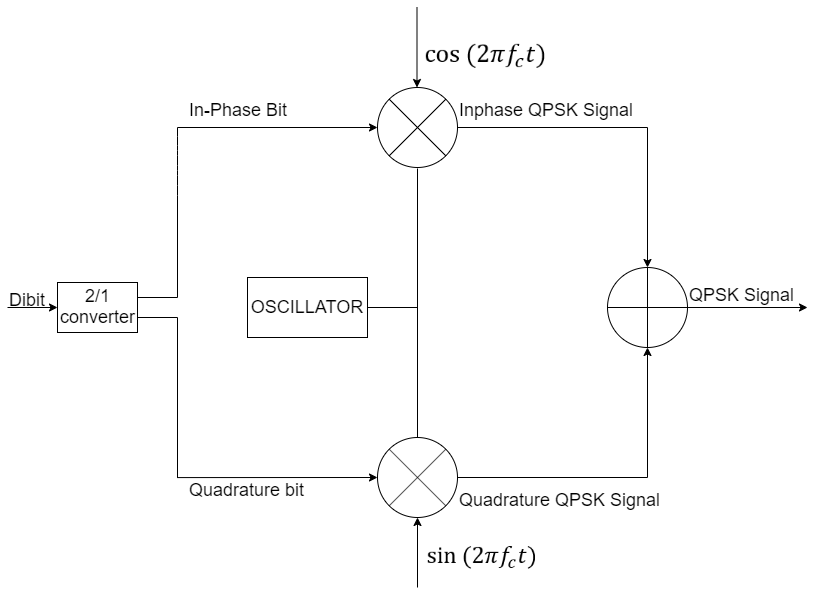
. 

The average bit error probability of the for each channel is:

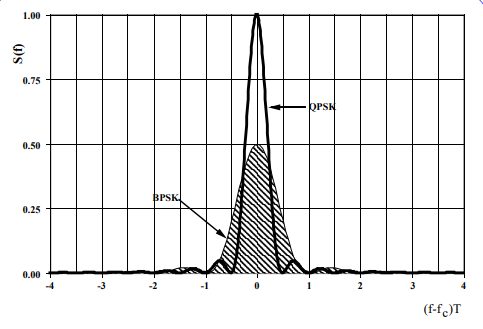


*Dibit distribution.*

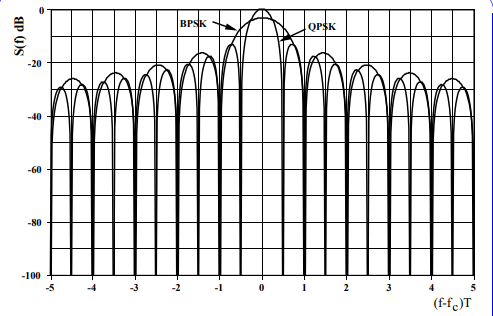
The even and odd bits will be used for in-phase pulse and quadrature pulse, respectively.

The Serial to Parallel Converter groups the incoming data into dibits (groups of two consecutive bits). Each time two bits have been clocked serially into its buffer, the Serial to Parallel Converter outputs one dibit in parallel at its two outputs. One bit of each dibit is sent to the I channel of the modulator; the other bit is sent to the Q channel of the modulator. Each channel of the modulator works independently to processes the stream of bits it receives.

*QPSK Block Diagram.*

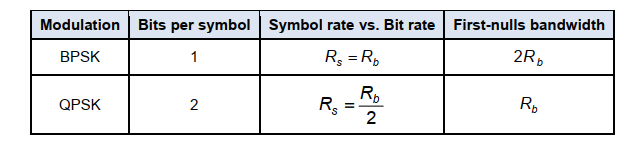


*Comparing spectrum of QPSK and BPSK.*

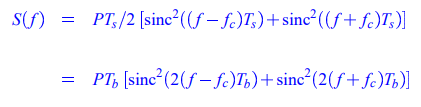


*Comparing spectrum of QPSK and BPSK (dB).*

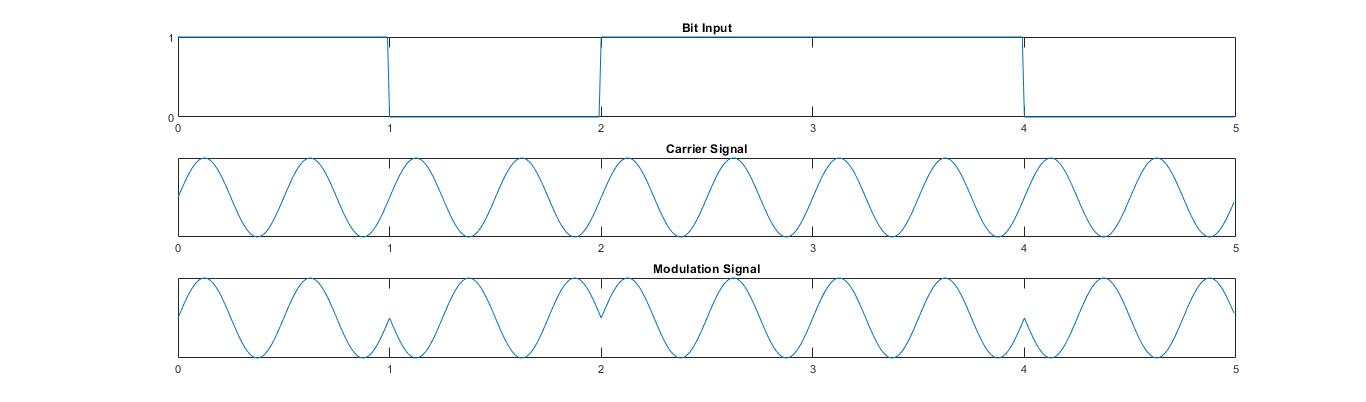
Compare BPSK and QPSK spectrum we can be easily seen QPSK had excellent power and bandwidth efficiency.



The bandwidth of QPSK is given by

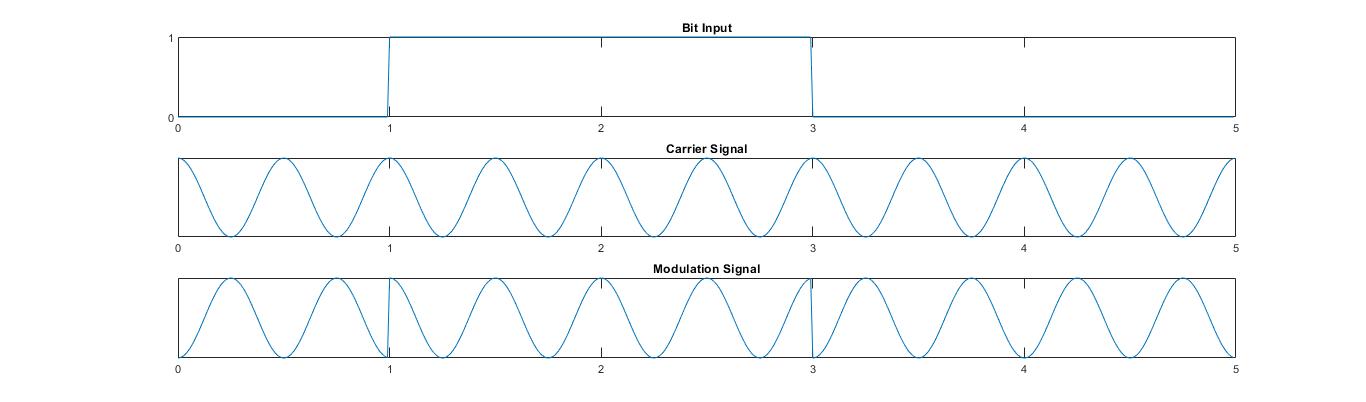


The waveform of this machine will be presented:



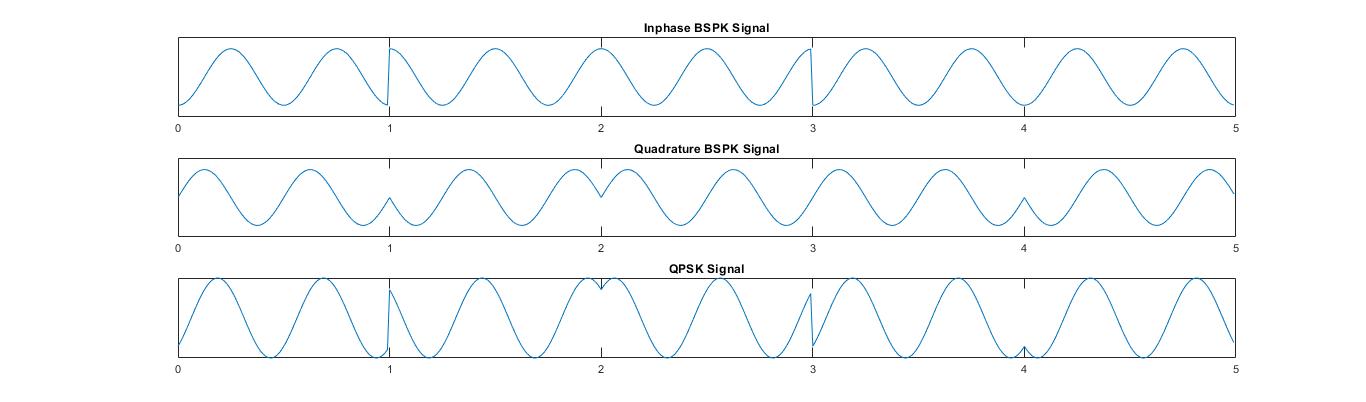
*In-phase BPSK Modulation.*

Firstly, the 1-2 converter (parallel to series converter) convers dibit to the two separate bits, one is in-phase BPSK bits, and other is quadrature BPSK bits. And The in-phase BPSK will be generate with the same BPSK processing. The phases will be jump over 180 degrees when the input change 0 to 1 or 1 to 0.



*Quadrature BPSK Modulation.*

Secondly, the quadrature BPSK bit will be generated similar to in-phase BPSK bits, but the carrier is different, this is sin waves compare with cos waves of in phase BPSK.

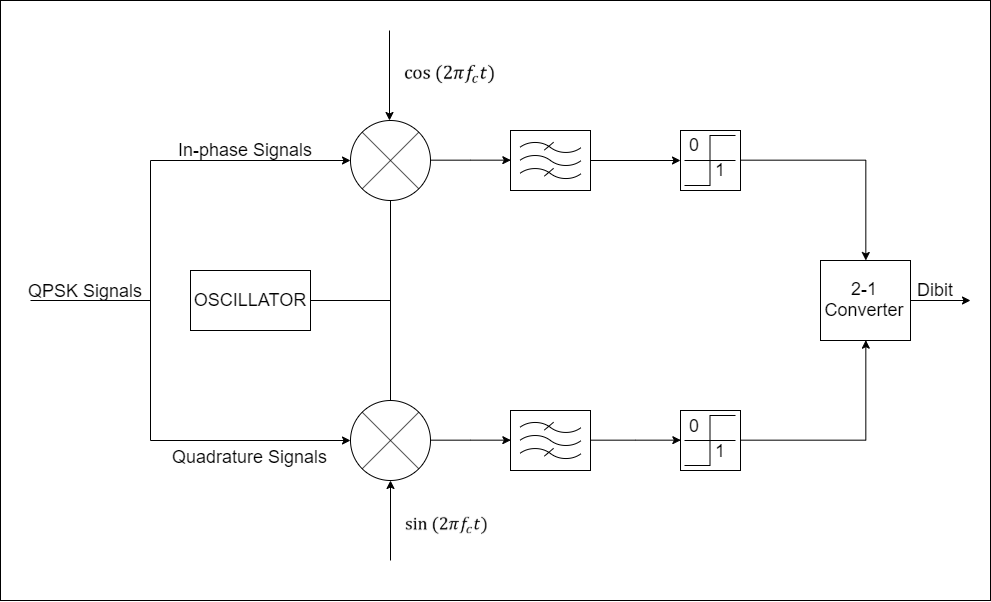


*QPSK is produced the sum of the two BPSK*

The final step is the summed of the two signals to produce QPSK. So, we can easily see the QPSK is generated by two basic BPSK signals.

1. **Demodulation**

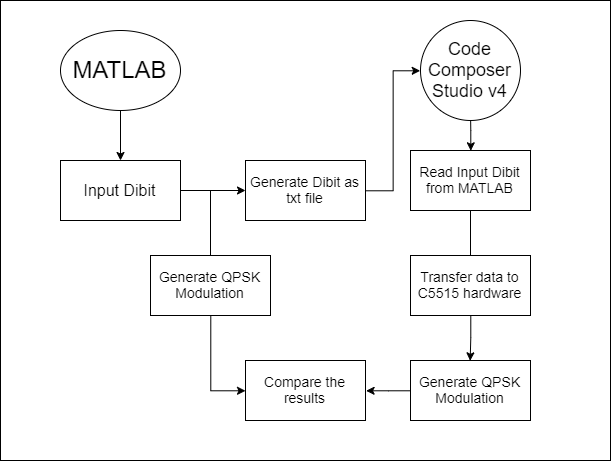
The block diagram for quadrature demodulator is show below:



*QPSK Demodulator.*

Prior to band pass filtering, the output of each multiplier will contain both baseband components and double-frequency component. And the bandpass filters will be removed the double-frequency components, leaving just baseband component. The I and Q outputs from the quadrature modulator can be processed separately as independent binary waveform. The pair of bits are recovered from the sending data. These signals after processing are passed to the 2-1 converter (parallel to serial converter).

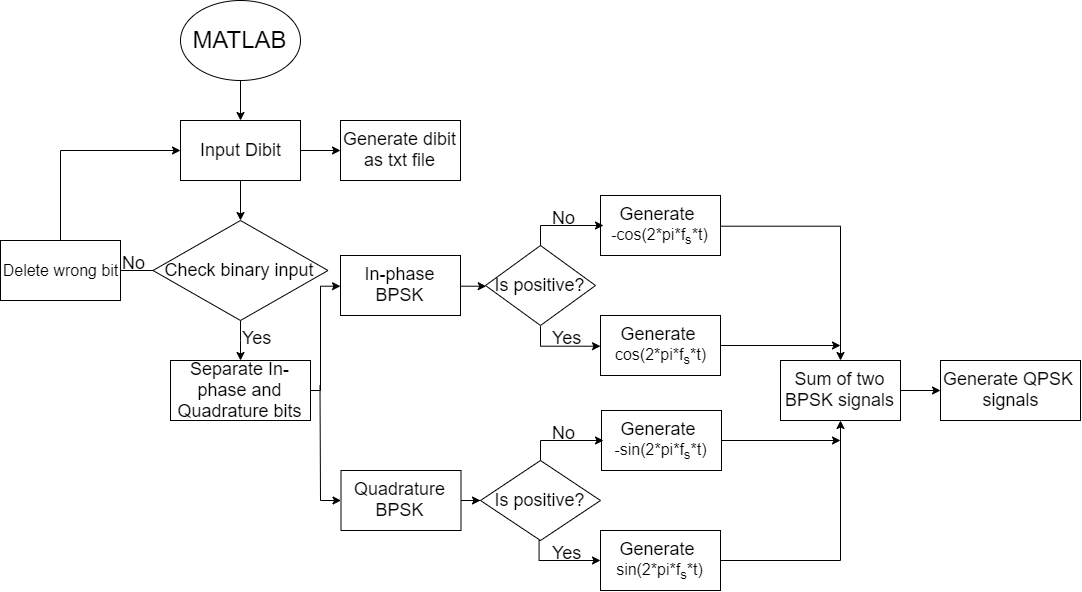
1. **Simulation**
   1. **General Flowchart**



*General Flowchart.*

* 1. **MATLAB Simulation**

1. **Flowchart**



*MATLAB Flowchart.*

1. **Source code**

clc

clear all

close all

data=input('Enter the data for qpsk');

last = length(data);

t=1;

f = 2;

fs = 100;

for i = 1:(last)

if(data(i) == 0)

dataw(i\*2 - 1) = 0;

dataw(i\*2) = 0;

end

if(data(i) == 1)

dataw(i\*2 - 1) = 0;

dataw(i\*2) = 1;

end

if(data(i) == 10)

dataw(i\*2 - 1) = 1;

dataw(i\*2) = 0;

end

if(data(i) == 11)

dataw(i\*2 - 1) = 1;

dataw(i\*2) = 1;

end

end

id1=fopen('A:\x(n).txt','w');

fprintf(id1, '%d', dataw);

fclose(id1);

id2=fopen('A:\lengthx(n).txt','w');

fprintf(id2,'%d', last);

fclose(id2);

%%

for start = 1:last

if (data(start)==00) %% 00

for j=0:1/fs:1-1/fs

q(t)=0;

p(t)=0;

o(t)=(-1/sqrt(2))\*(cos(2\*pi\*f\*j))-(1/sqrt(2))\*(sin(2\*pi\*f\*j)); %% 0 degree

t=t+1;

end

elseif (data(start)==01) %% 01

for j=0:1/fs:1-1/fs

q(t)=0;

p(t)=1;

o(t)=(-1/sqrt(2))\*(cos(2\*pi\*f\*j))+(1/sqrt(2))\*sin(2\*pi\*f\*j); %% 90 degree

t=t+1;

end

elseif (data(start)==10) %% 10

for j=0:1/fs:1-1/fs

q(t)=1;

p(t)=0;

o(t)=(1/sqrt(2))\*cos(2\*pi\*f\*j)-(1/sqrt(2))\*(sin(2\*pi\*f\*j)); %% 180 degree

t=t+1;

end

else %(data(start)==11) %% 11

for j=0:1/fs:1-1/fs

q(t)=1;

p(t)=1;

o(t)=(1/sqrt(2))\*cos(2\*pi\*f\*j)+(1/sqrt(2))\*sin(2\*pi\*f\*j); %% 270 degree

t=t+1;

end

end

end

x = 0:1/fs:(t-2)/fs;

subplot(3,1,1)

plot(x,q);

title('Inphase Bit Sequence');

subplot(3,1,2)

plot(x,p);

title('Quadrature Bit Sequence');

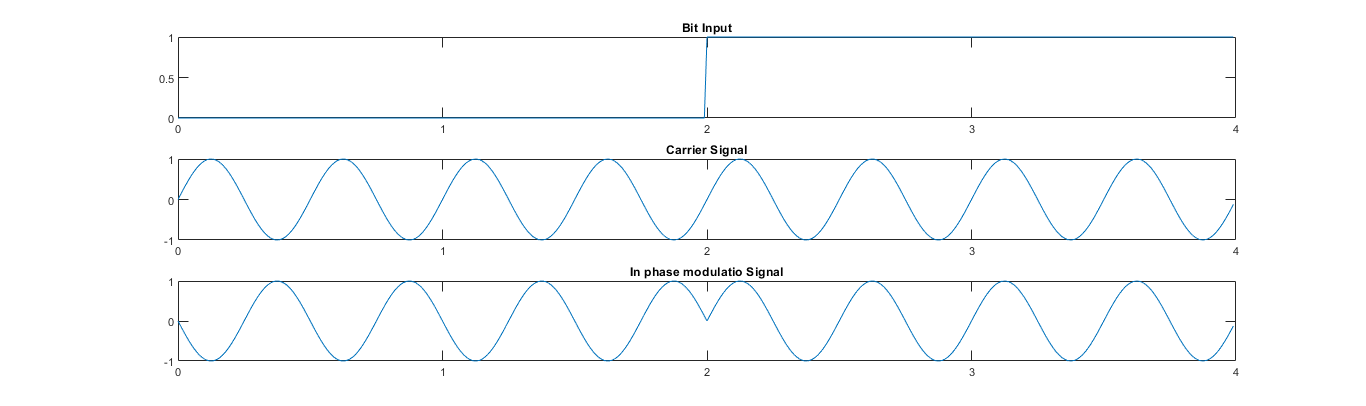
subplot(3,1,3)

plot(x,o);

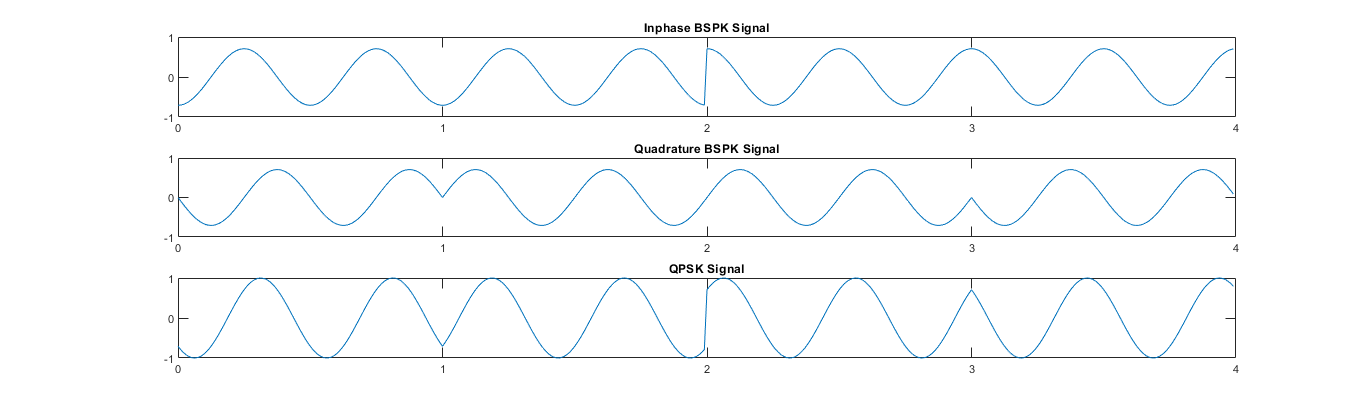
title('QPSK Signal Transmitted');

1. **Results**

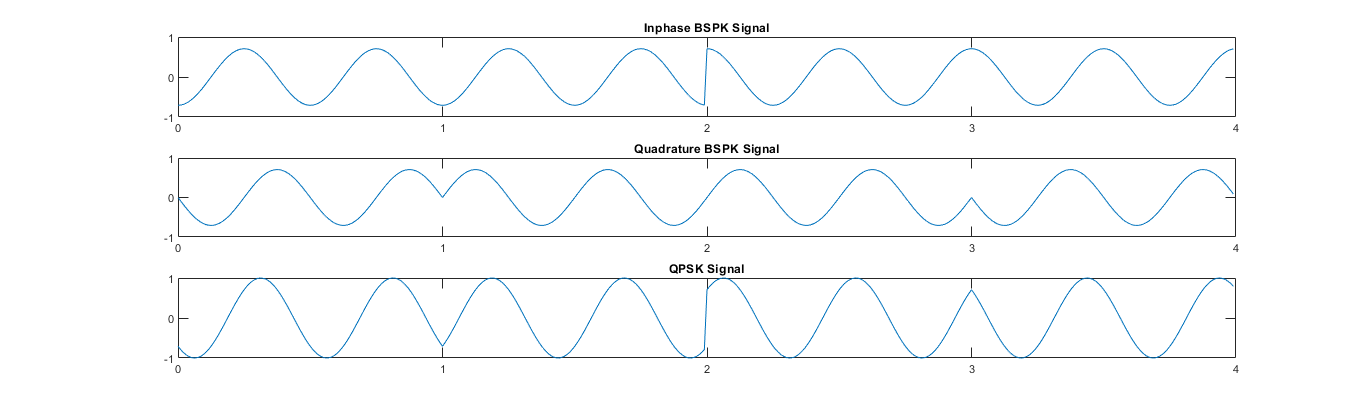
4-PSK bits input = [00 01 11 10]



*In-phase BPSK*



*Quadrature BPSK*

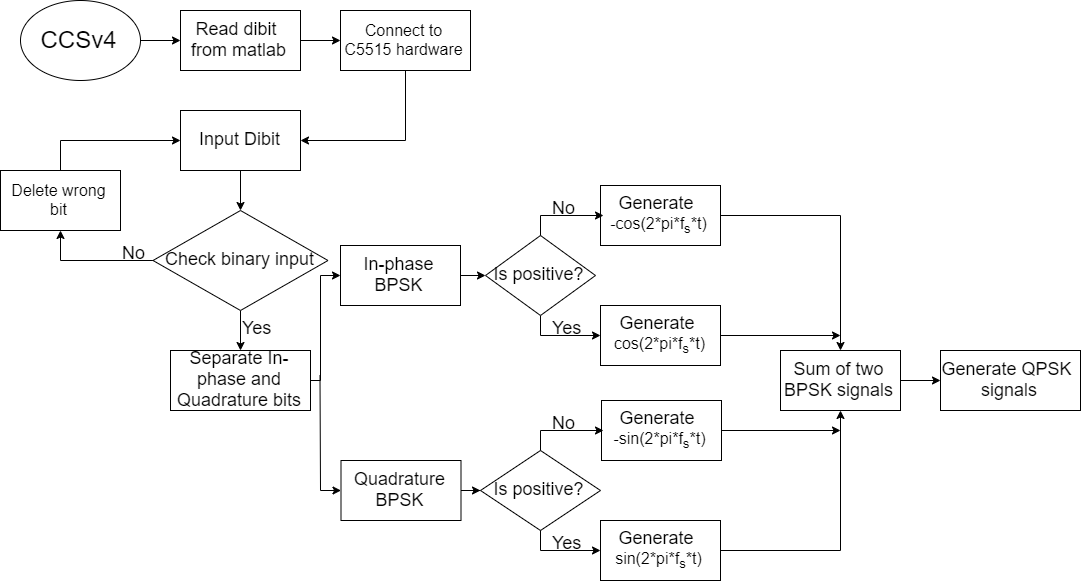


*QPSK Signals by Pure MATLAB.*

Comment: When we generate QPSK modulation with frequency =2\* , my team got the desirable results.

* 1. **CCSv4 (TMS320C5515) Simulation**

1. **Flow chart**



*CCSv4 and TSM320C5515 Flowchart.*

1. **Source code**

#include "stdio.h"

#include "usbstk5515.h"

extern Int16 AIC3204\_rset( Uint16 regnum, Uint16 regval);

#define XmitL 0x10

#define XmitR 0x20

#include"math.h"

#define F1 200

#define Fs 48000

#define PI 3.141592654

Int16 aic3204\_tone\_headphone( )

{

double\* signal;

double\* ptsig;

int x;

int nsample ;

int i,j,sample;

int length,len;

char bit[100];

char buffer[100];

int c;

double sign\_1,sign\_2;

FILE \*fp;

fp=fopen("A:\\x(n).txt","r");

fseek(fp,SEEK\_SET, 0);

length = 0;

while(1)

{

c = fgetc(fp);

if(feof(fp))

{

break;

}

length++;

}

fp = fopen("A:\\x(n).txt","r");

fread(buffer, length, 1, fp);

for(i = 0; i < length; i++)

{

bit[i] = buffer[i] - 48;

}

len = length/2;

nsample = Fs/F1;

signal = (double\*) malloc(nsample\*len\* sizeof(double));

for(i = 0; i < len; i ++)

{

if(bit[i\*2] == 1)

{

sign\_1 = 1;

}

else

{

sign\_1 = -1;

}

if(bit[i\*2 + 1] == 1)

{

sign\_2 = 1;

}

else

{

sign\_2 = -1;

}

for(x = 0; x<=nsample; x++)

{

signal[x+i\*nsample] = 1333\*((sign\_1/sqrt(2))\*cos(2\*PI\*x/nsample) + (sign\_2/sqrt(2))\*sin(2\*PI\*x/nsample));

}

}

ptsig = (short\*) malloc( nsample \* len \* sizeof(short));

for (i = 0; i < nsample\*len; i++)

{

ptsig[i] = (short) signal[i];

}

AIC3204\_rset( 0, 0 ); // Select page 0

AIC3204\_rset( 1, 1 ); // Reset codec

AIC3204\_rset( 0, 1 ); // Select page 1

AIC3204\_rset( 1, 8 ); // Disable crude AVDD generation from DVDD

AIC3204\_rset( 2, 1 ); // Enable Analog Blocks, use LDO power

AIC3204\_rset( 0, 0 );

/\* PLL and Clocks config and Power Up \*/

AIC3204\_rset( 27, 0x0d ); // BCLK and WCLK are set as o/p; AIC3204(Master);

AIC3204\_rset( 28, 0x00 ); // Data ofset = 0

AIC3204\_rset( 4, 3 );

AIC3204\_rset( 6, 7 ); // PLL setting: J=7

AIC3204\_rset( 7, 0x06 ); // PLL setting: HI\_BYTE(D=1680)

AIC3204\_rset( 8, 0x90 ); // PLL setting: LO\_BYTE(D=1680)

AIC3204\_rset( 30, 0x88 );

AIC3204\_rset( 5, 0x91 ); // PLL setting: Power up PLL, P=1 and R=1

AIC3204\_rset( 13, 0 );

AIC3204\_rset( 14, 0x80 );

AIC3204\_rset( 20, 0x80 );

AIC3204\_rset( 11, 0x82 ); // Power up NDAC and set NDAC value to 2

AIC3204\_rset( 12, 0x87 ); // Power up MDAC and set MDAC value to 7

AIC3204\_rset( 18, 0x87 ); // Power up NADC and set NADC value to 7

AIC3204\_rset( 19, 0x82 ); // Power up MADC and set MADC value to 2

AIC3204\_rset( 0, 1 ); // Select page 1

AIC3204\_rset( 0x0c, 8 ); // LDAC AFIR routed to HPL

AIC3204\_rset( 0x0d, 8 ); // RDAC AFIR routed to HPR

AIC3204\_rset( 0, 0 ); // Select page 0

AIC3204\_rset( 64, 2 ); // Left vol=right vol

AIC3204\_rset( 65, 0); // Left DAC gain to 0dB VOL; Right tracks Left

AIC3204\_rset( 63, 0xd4 ); // Power up left,right data paths and set channel

AIC3204\_rset( 0, 1 ); // Select page 1

AIC3204\_rset( 0x10, 0x00 );// Unmute HPL , 0dB gain

AIC3204\_rset( 0x11, 0x00 );// Unmute HPR , 0dB gain

AIC3204\_rset( 9, 0x30 ); // Power up HPL,HPR

AIC3204\_rset( 0, 0 ); // Select page 0

USBSTK5515\_wait( 100 ); // wait

AIC3204\_rset( 0, 1 ); // Select page 1

AIC3204\_rset( 0x34, 0x30 );

AIC3204\_rset( 0x37, 0x30 );// IN2\_R to RADC\_P through 40 kohmm

AIC3204\_rset( 0x36, 3 );

AIC3204\_rset( 0x39, 0xc0

AIC3204\_rset( 0x3b, 0 ); // MIC\_PGA\_L unmute

AIC3204\_rset( 0x3c, 0 ); // MIC\_PGA\_R unmute

AIC3204\_rset( 0, 0 ); // Select page 0

AIC3204\_rset( 0x51, 0xc0 );// Powerup Left and Right ADC

AIC3204\_rset( 0x52, 0 ); // Unmute Left and Right ADC

AIC3204\_rset( 0, 0 );

USBSTK5515\_wait( 200 ); // Wait

/\* I2S settings \*/

I2S0\_SRGR = 0x0;

I2S0\_CR = 0x8010; // 16-bit word, slave, enable I2C

I2S0\_ICMR = 0x3f; // Enable interrupts

/\* Play Tone \*/

for ( i = 0 ; i < 1000 ; i++ )

{

for ( j = 0 ; j < 1000 ; j++ )

{

for ( sample = 0 ; sample < nsample\*len ; sample++ )

{

while((XmitR & I2S0\_IR) == 0); // Wait for transmit

I2S0\_W0\_MSW\_W = (ptsig[sample]) ; // 16 bit left channel transmit

I2S0\_W1\_MSW\_W = (ptsig[sample]) ; // 16 bit right channel

}

}

}

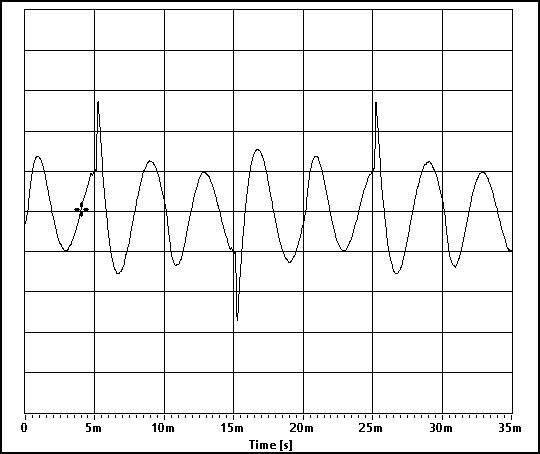
/\* Disable I2S \*/

I2S0\_CR = 0x00;

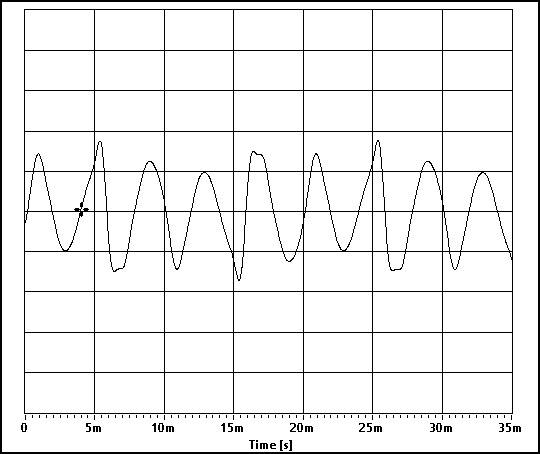
return 0;

}

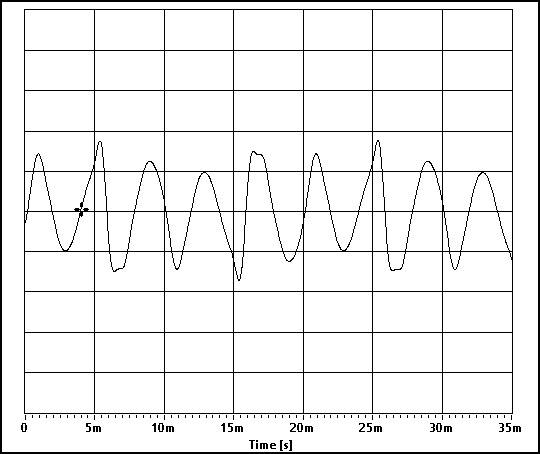
1. **Result**



*Signal on Oscilloscope without Filter.*



*Signal on Oscilloscope with Bandpass Filter*.

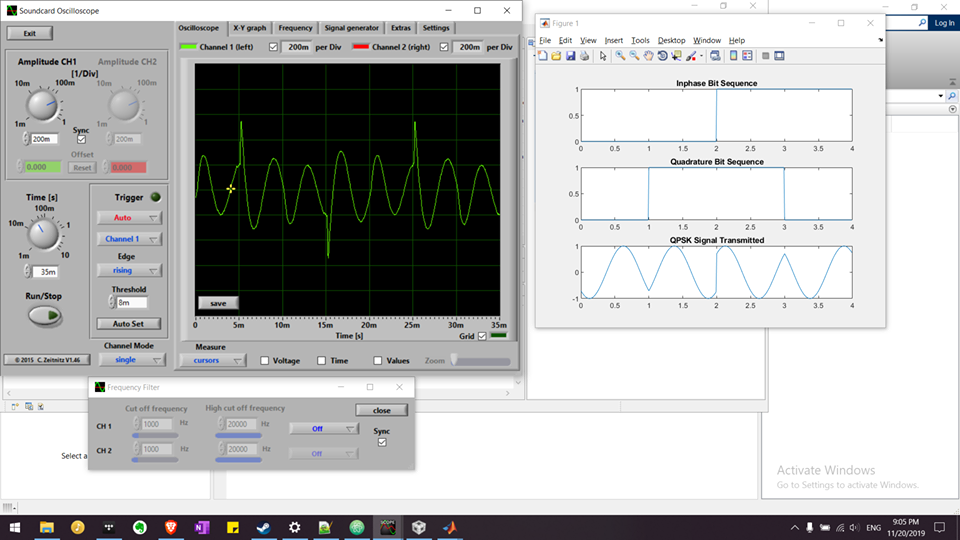


*Signal on Oscilloscope with Lowpass Filter Lowpass filter*

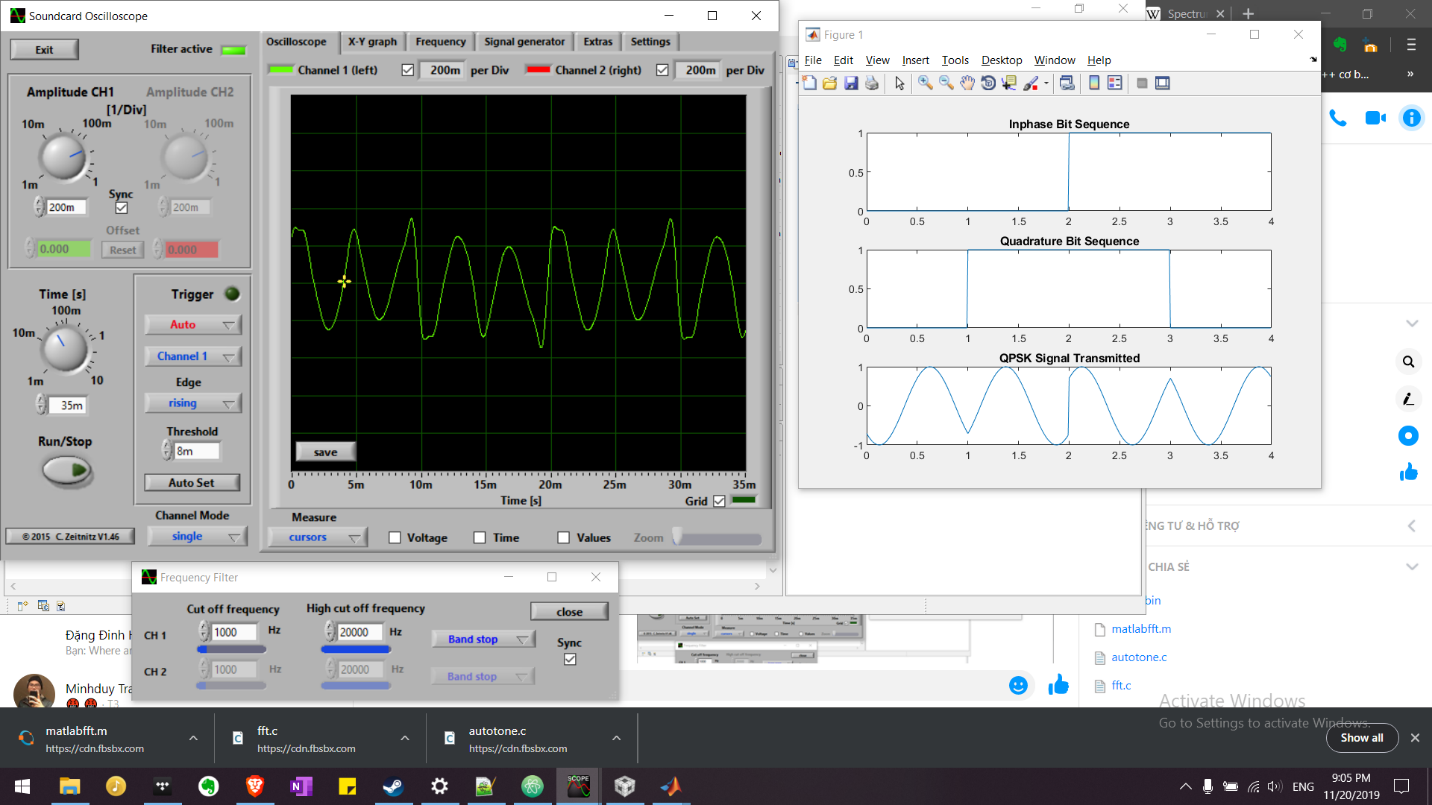
.

Comment: We tried some filter to get desirable results, however we can generate a exactly a jump of phase in C5515. My team guess the problems form handwave when it receives the signal from laptop by the cable and the bit error when modulation in theory.

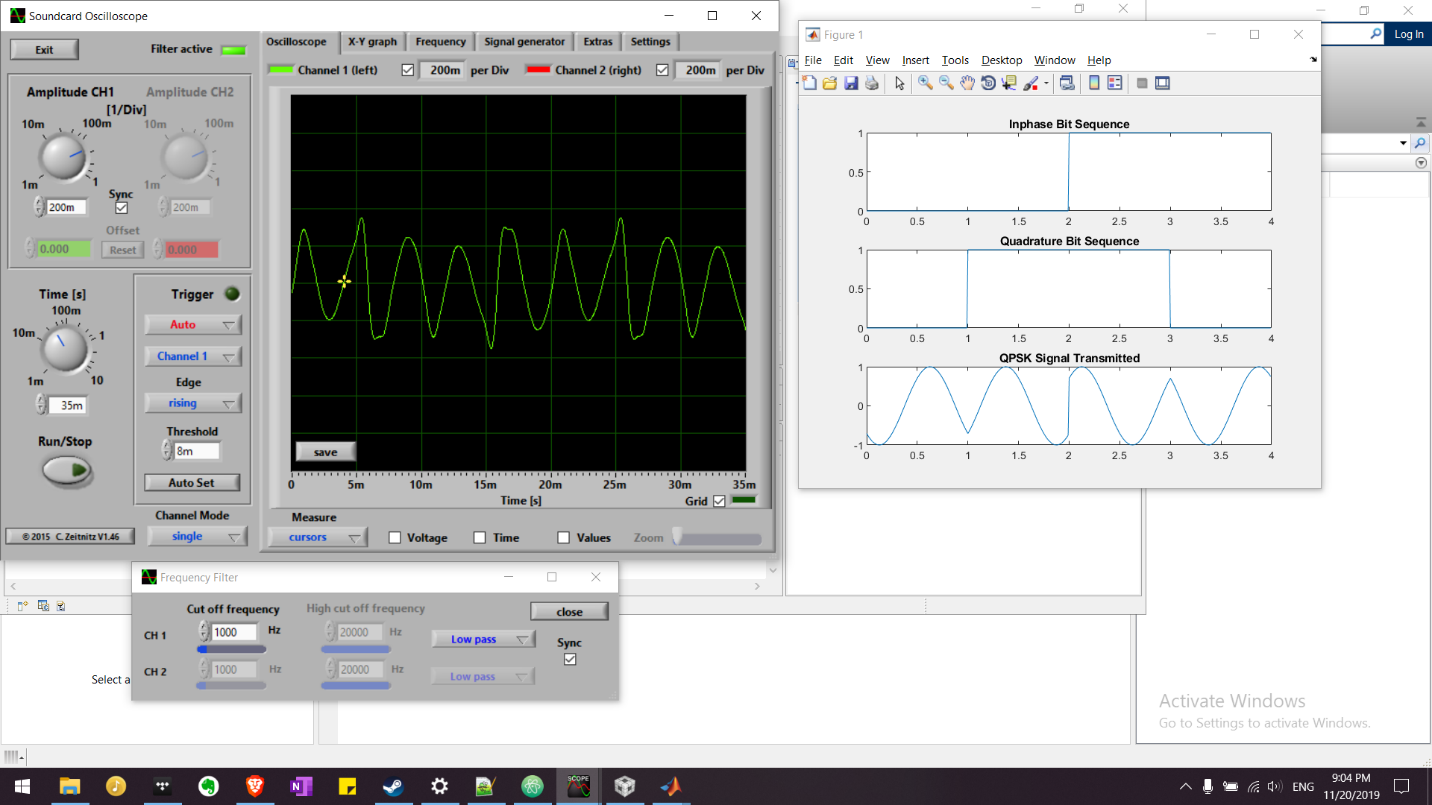
* 1. **Compare results**



*Signals without Filter.*

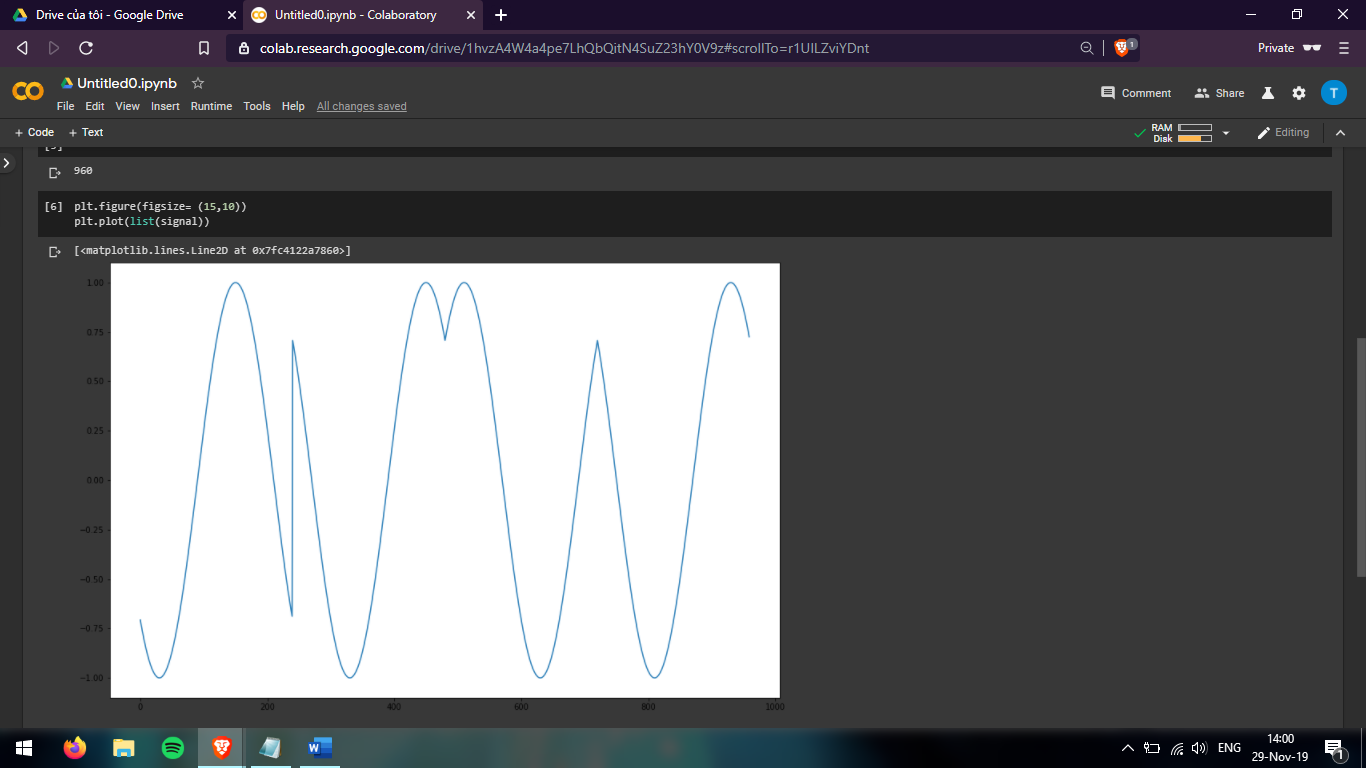


*Signal on Oscilloscope with Bandpass Filter*



*Signals without* Lowpass filter

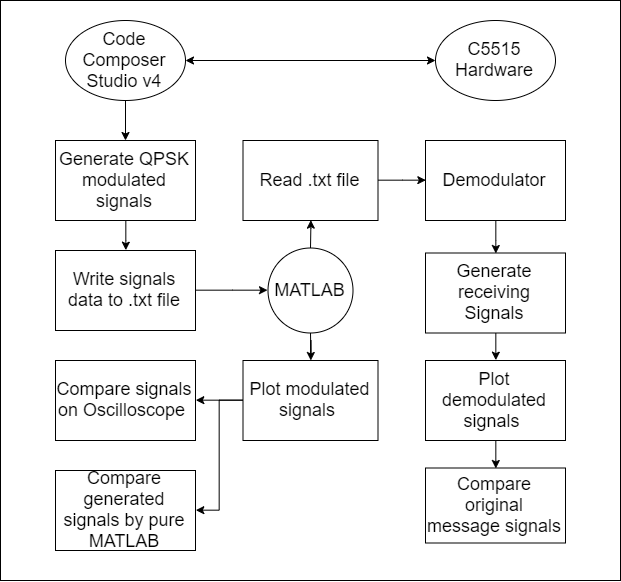
Comment: When compare the signals between MATLAB software and TSM320C5515 hardware. The result quite similar, but the C5515 generated the 4PSK have more different more than the theory, and my team tried to use two kinds of filter to reduce the wrong amplitude peak. However, we have checked an algorithm is not wrong by Python language.



* 1. **Compare sending signal and receiving signal**

According to comments of Professor, to prove our modulator and algorithm is true, we need to demodulate the QPSK modulated signals by TSM320C5515 Development Kit, and then comparing the results of sending signals and receiving signals.

* + 1. **Flow chart of demodulator**



* + 1. **Source code (MATLAB)**

clc;

close all;

clear all;

br = 1000;

f=br;

T=1/br;

t=T/120:T/120:T;

fileID = fopen('A:\signal.txt','r');

formatSpec = "%d %f";

sizeA = [1 Inf];

Tx\_sig = fscanf(fileID,formatSpec,sizeA);

data = Tx\_sig;

Rx\_data = [];

Rx\_sig = Tx\_sig;

for(i = 1:1:length(data)/2)

Z\_in = Rx\_sig((i-1)\*length(t)+1:i\*length(t)).\*cos(2\*pi\*f\*t);

Z\_in\_intg = (trapz(t,Z\_in))\*(2/T);

if(Z\_in\_intg>0)

Rx\_in\_data = 1;

else

Rx\_in\_data = 0;

end

Z\_qd = Rx\_sig((i-1)\*length(t)+1:i\*length(t)).\*sin(2\*pi\*f\*t);

Z\_qd\_intg = (trapz(t,Z\_qd))\*(2/T);

if (Z\_qd\_intg > 0)

Rx\_qd\_data = 1;

else

Rx\_qd\_data = 0;

end

Rx\_data = [Rx\_data Rx\_in\_data Rx\_qd\_data];

end

figure(1)

x1 = 0:length(Tx\_sig) - 1;

plot(x1, Tx\_sig);

title('Signal is generated by the C5515');

figure(2)

x2 = 0:length(Rx\_data)-1;

stem(x2,Rx\_data,'linewidth',3)

title('Information after Receiveing ');

axis([ 0 11 0 1.5]), grid on;

Source code generate .txt file of signals from oscilloscope by CCSv4

fp = fopen("A:\\signal.txt","w");

if(fp == NULL)

{

printf("Error!");

exit(1);

}

for(i = 0; i < j; i++)

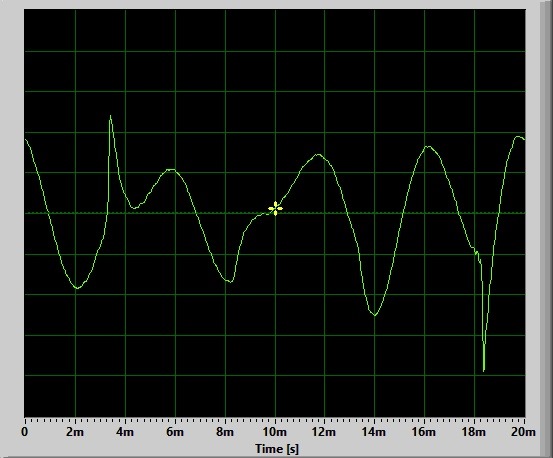
{

fprintf(fp,"%d ", signal[i]);

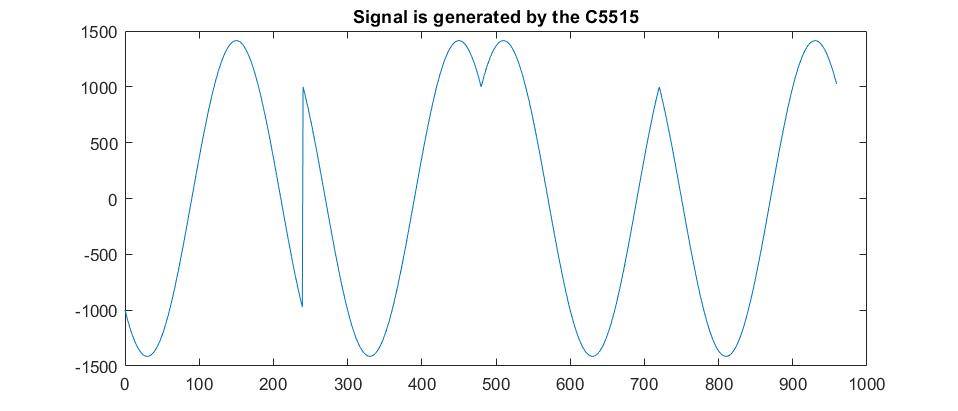
}

fclose(fp);

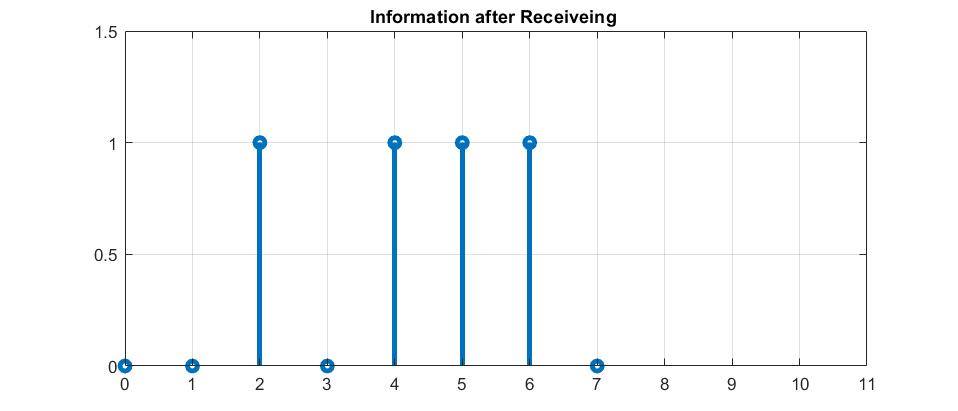
* + 1. **Result**



*Signals without no* *filter*



*Receiving Modulated Signal from c5515 read by MATLAB*



*Demodulated Signal by MATLAB*

Comment: After sending the signal modulated by C5515 to MATLAB to demodulate, the demodulated signal is the same the message signal [00 10 11 10]. Besides, the signal from C5515 which is represented by MATLAB is the same the modulated signal generated by pure MATLAB. From this proof, The QPSK modulated by C5515 through CCSv4 can be true.

1. **Conclusion**

After research on communication systems, some basic concept of Modulation. We are more understand Quadrature Phase Shift Keying, one of the most popular digital modulation used for satellite, video conferring, and other digital forms using Radio Frequency carrier. However, my team still have the problems to generate the ideal 4PSK.

References

Park, J. H., Jr., ‘‘On binary DPSK detection,’’ *IEEE Trans. Commun*., vol. 26, no. 4, April 1978,

pp. 484–486 Vinay K.Ingle, John G.Proakis.  *Digital Signal Processing using MATLAB*. Prentice Hall, 2010

Proakis, J., *Digital Communications*, 2nd ed., New York: McGraw-Hill, 1989.

Forouzan, Behrouz, *Data Communication and Network,* New York: McGraw-Hill, 2006

Sklar, B., *Digital Communications: Fundamentals and Applications,* Englewood Cliffs, New Jersey: Prentice Hall, 1988.

Ziemer, R. E., and R. L. Peterson, *Introduction to Digital Communication*, New York: Macmillan, 1992

Fuqin Xiong, *Digital Modulation Techniques,* 2nd ed., Boston: Artech House, 2006

Leis, John, *Communication system principle using MATLAB*, Hoboken: John Wiley & Sons, 2018