Lecture 2: Simulation of 16QAM systems

March 28 – April 19 2008

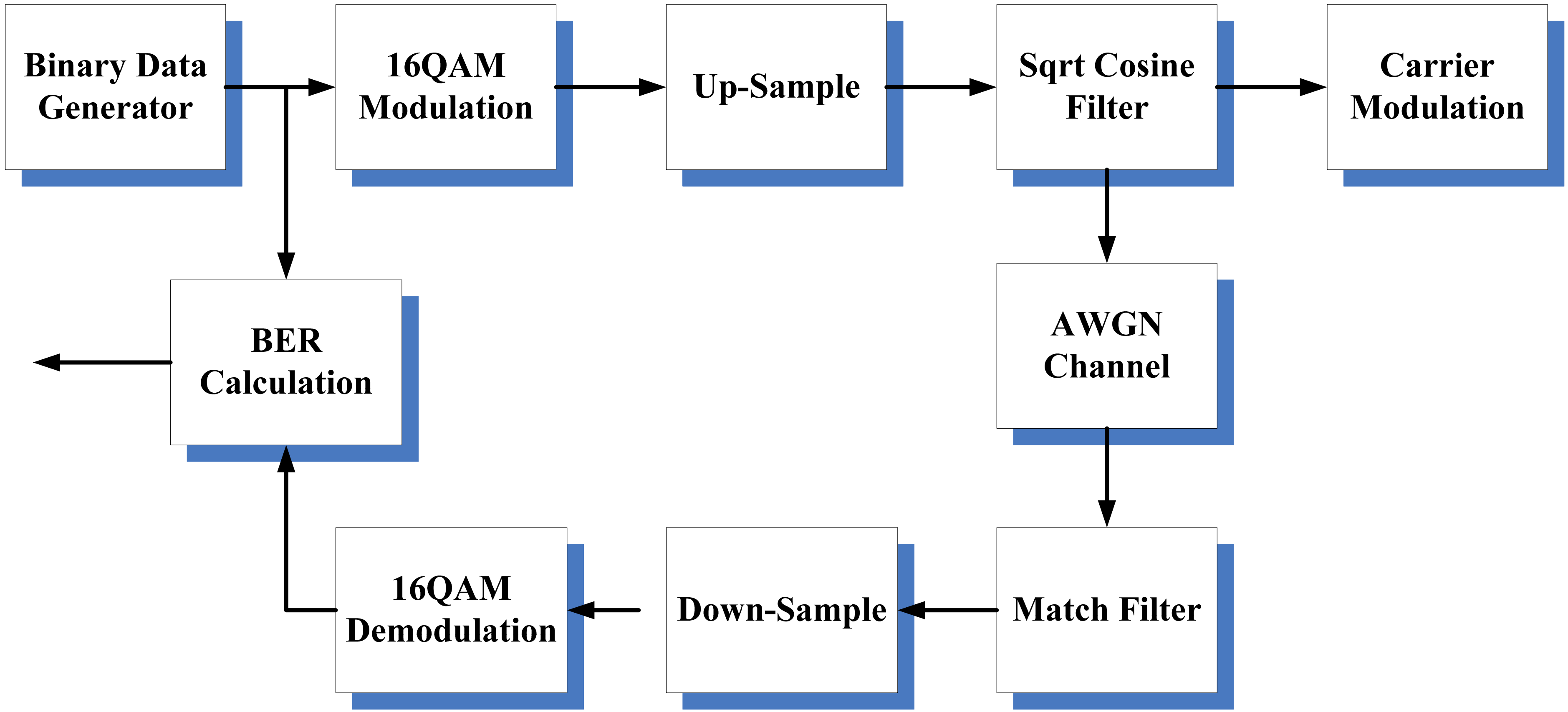
Yuping Zhao (Doctor of Science in technology)

Professor, Peking University
Beijing, China
Yuping.zhao@pku.edu.cn

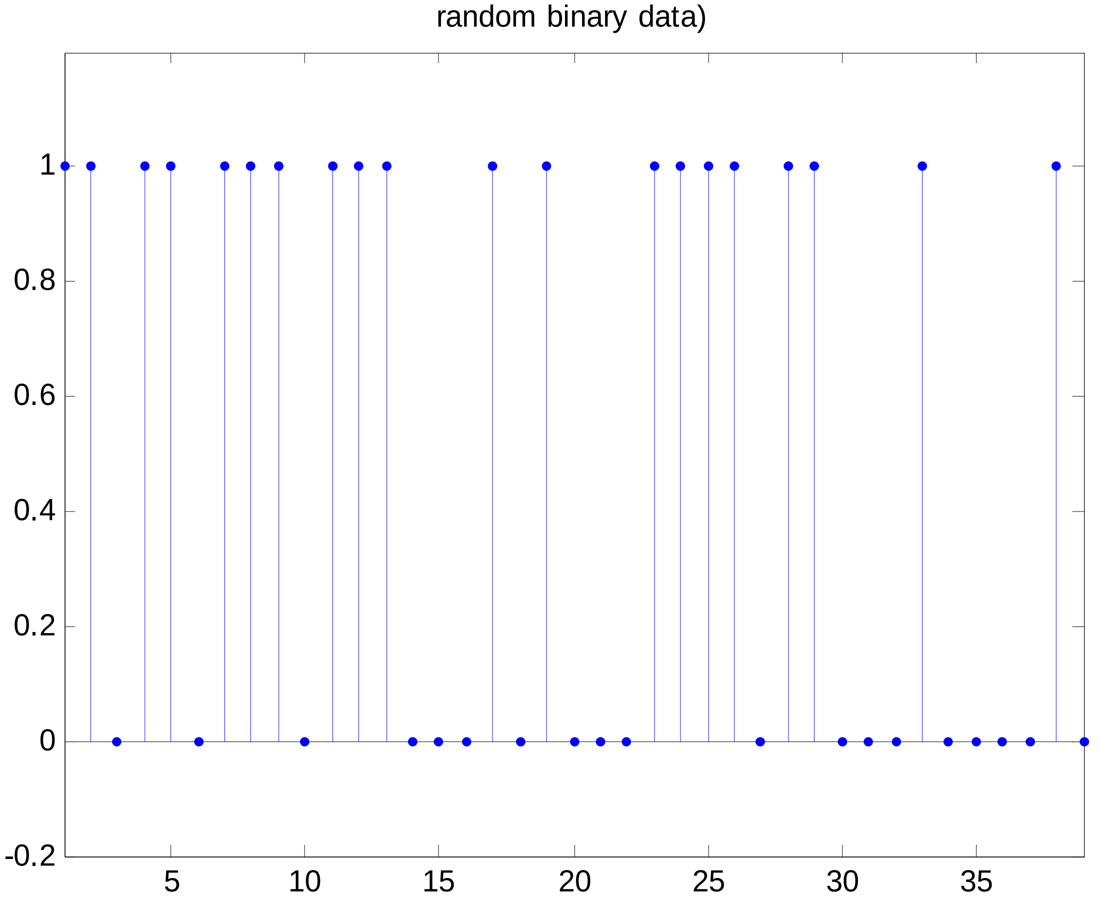
What we learn here?

- How to simulate the continues signals using discrete samples of the signals
- How to design the Matched filters that fulfill Nyquist conditions
- How to add the noise signals for given SNR, sampling rate, modulation level ?
- How to simulation Eye diagram
- How to evaluate the BER, Symbol error rate (theory and simulations)

Block diagram of the communication system



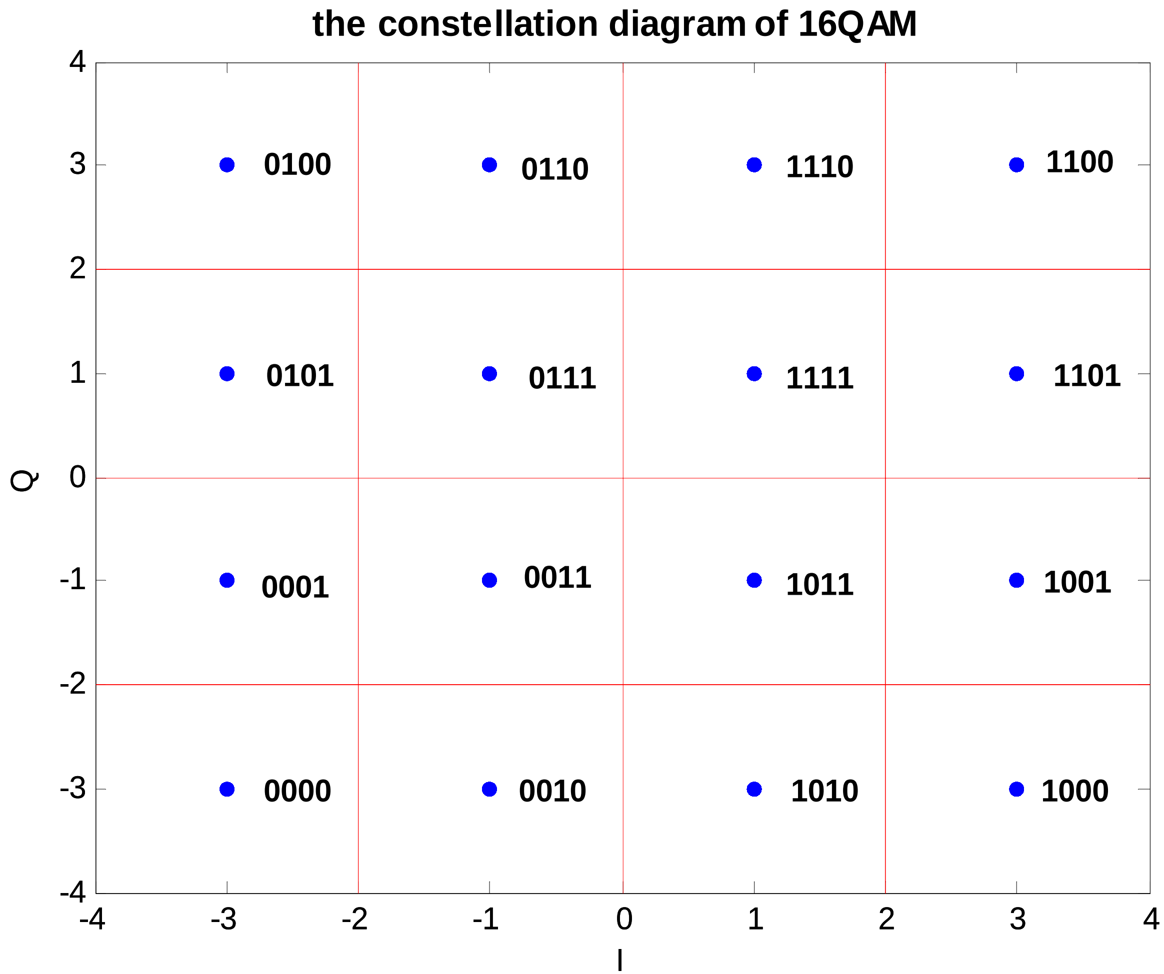
Generate binary data



x=randint(1,N), randomly generate binary data of length N

As the starting point, we suggest N=2000

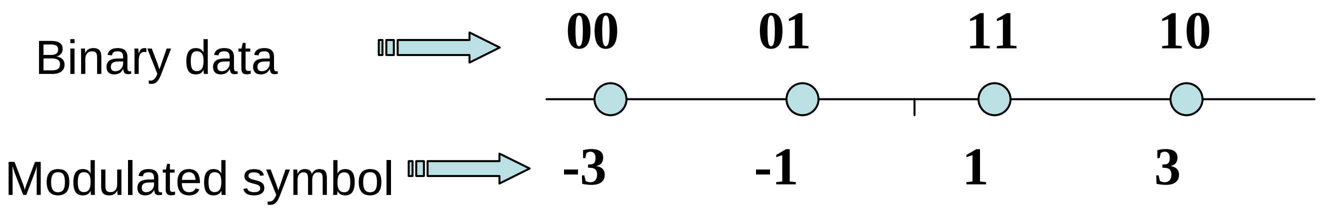
16QAM Modulation module



Each symbol contains 4 bits, the values in real and imaginary are 3， -1， 1， 3. The bits 1,3 corresponds to real part, the bits 2,4 corresponds to the imaginary part.

You may modulate real and imaginary parts separately

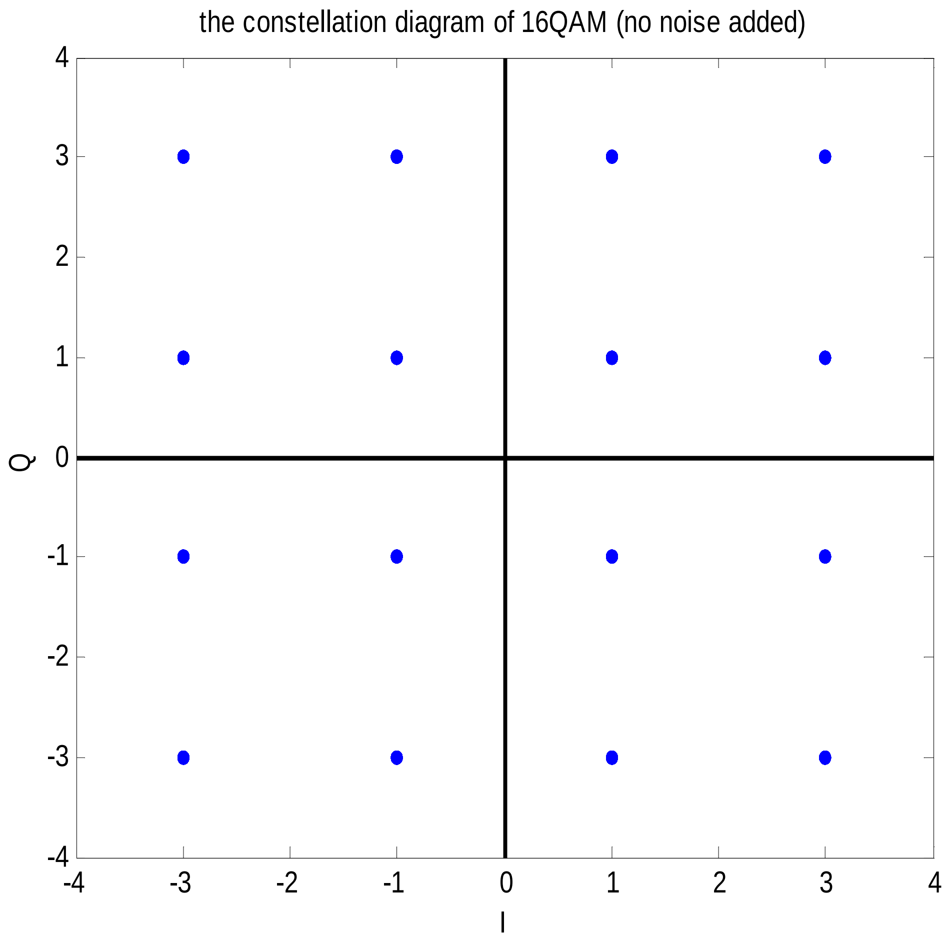
The real part corresponds to bits 1,3, using Gray mapping, the constellation is as follows



The imaginary part do the same way.

The modulated symbols are x = a + jb, or x = I + jQ

Transmitted signal constellation

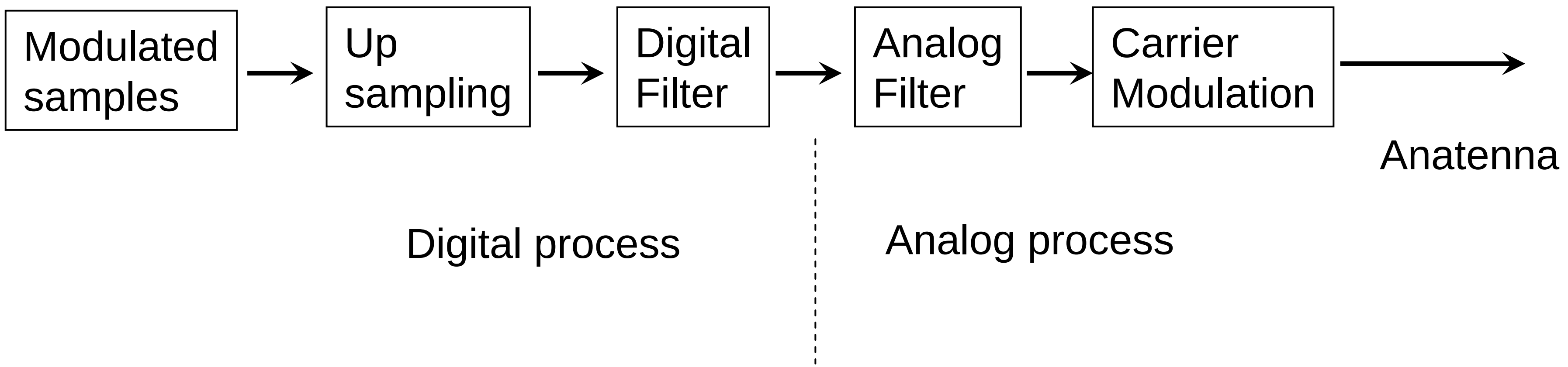


4. Up-sampling module

The reason for up sampling:

- The samples are impulse signals, its spectrum is extremely wide.
- The signals should pass the lowpass filters before transmission.
- In the hardware implementation, the continues signals is transmitted

Hardware process for the transmitted signals



Implementation

Assume the up-sampling rate is

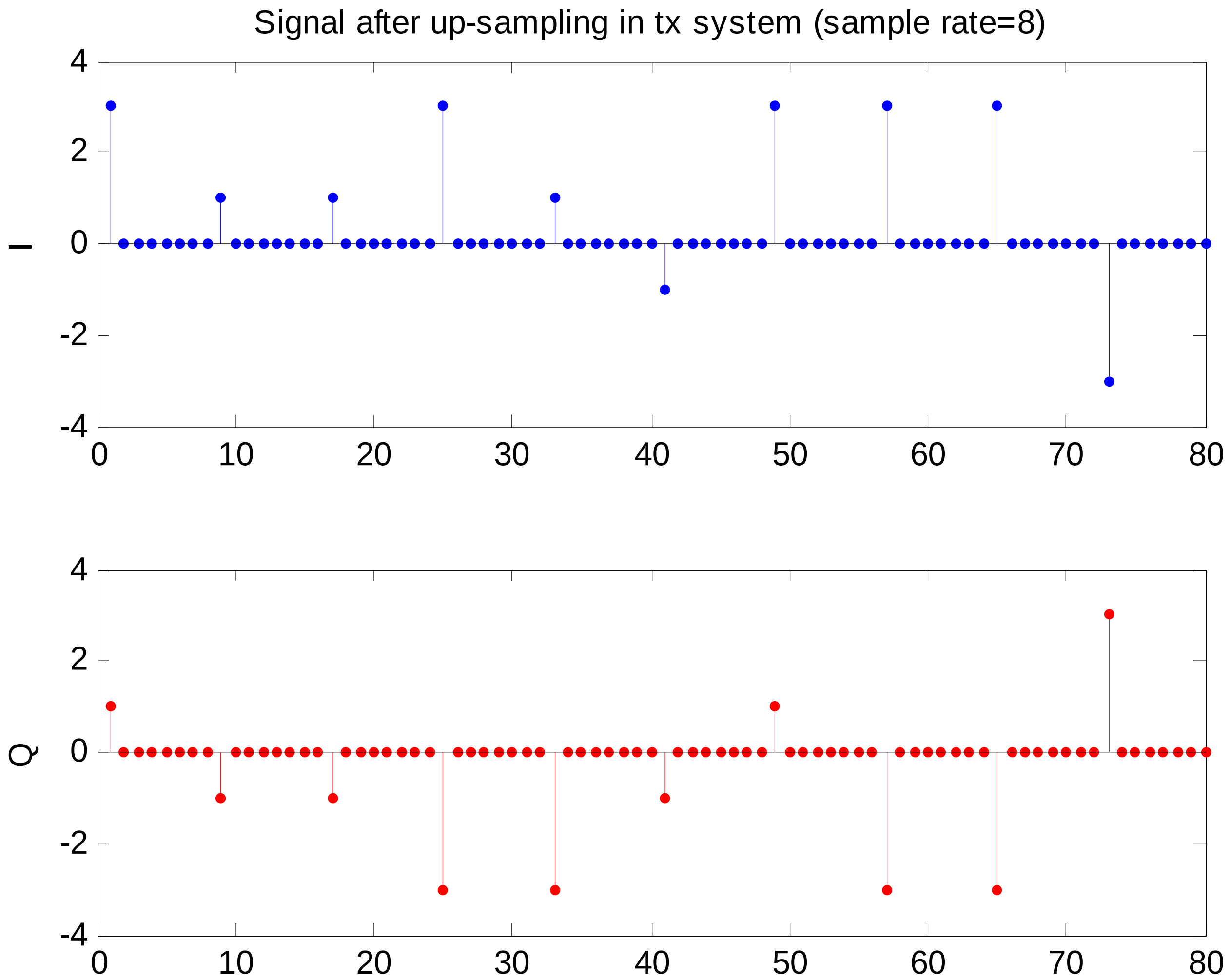
$$sample\_rate$$

Then we place zeros after each sample

$$sample\_rate - 1$$

- In the simulation, you can chose any up-sampling values, for example: 8
- In the hardware implementation, the typical value is 4
- For real and imaginary parts, do the up-sampling separately

The values can be +3,+1,-1,-3,0

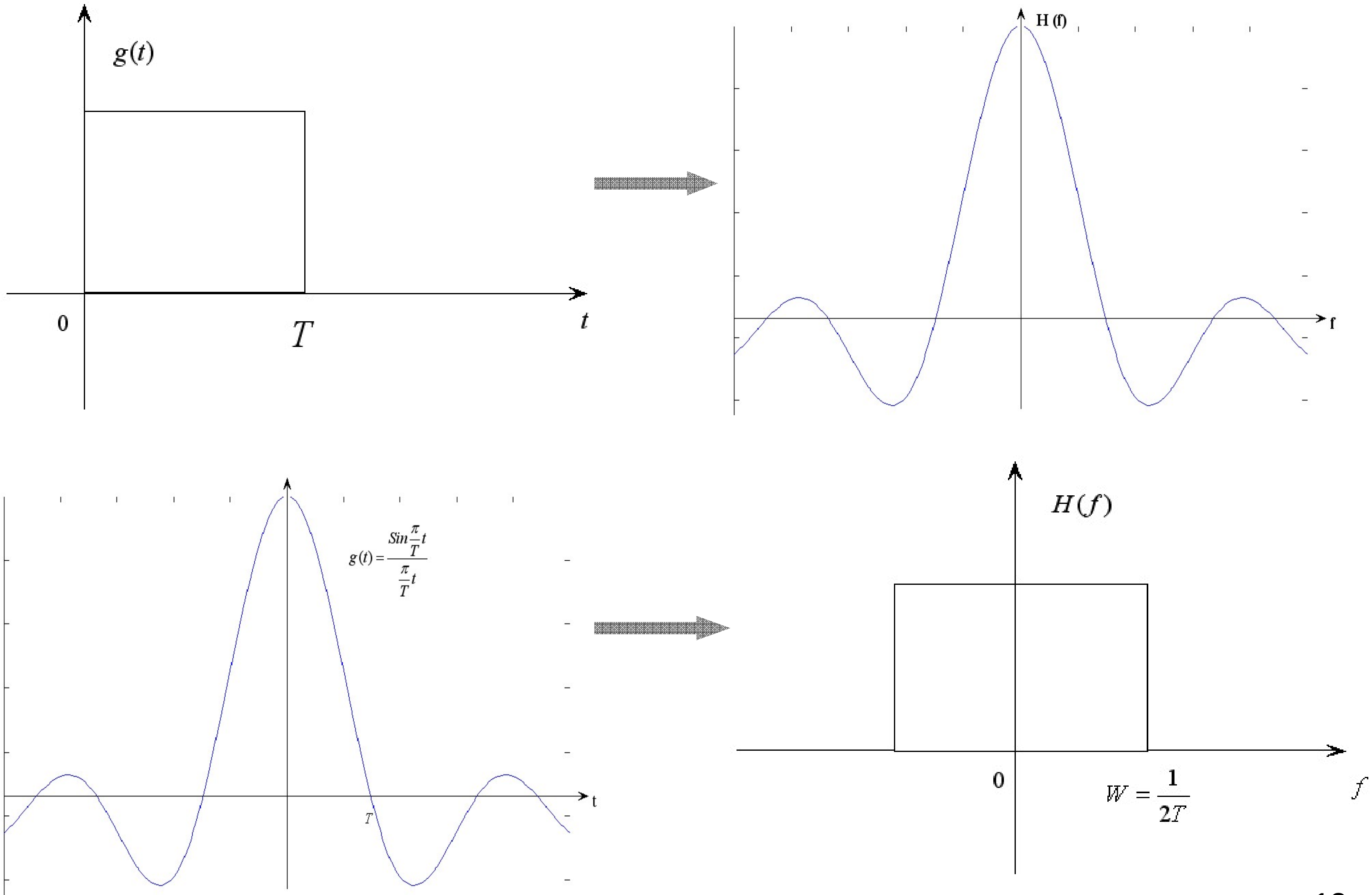


Low-pass filtering

To limit the transmitted signal within a certain band
The filtering is performed for real and imaginary parts separately

After the lowpass filtering, the time domain signal is not the impulse. It becomes continues signal shape.
The filter function and time domain signal shape is a pair of Fourier transform

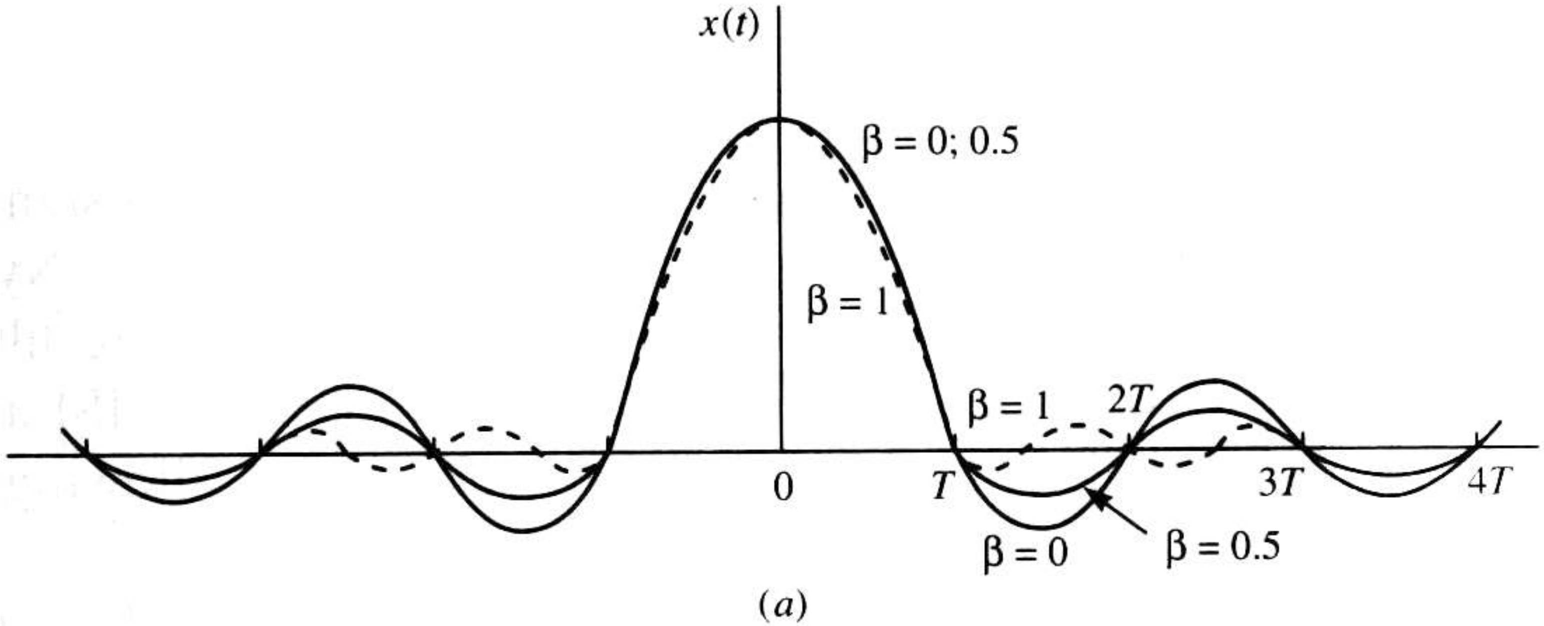
The design of lowpass filter

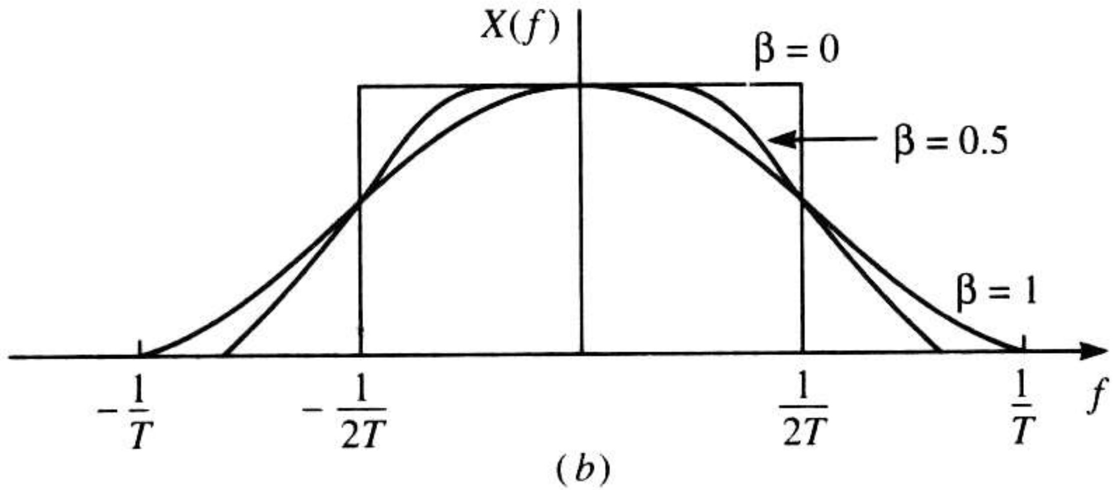


Rectangular filter in frequency band

$$X(f) = \begin{cases} T, (|f| < W) \\ 0, (otherwise) \end{cases},$$

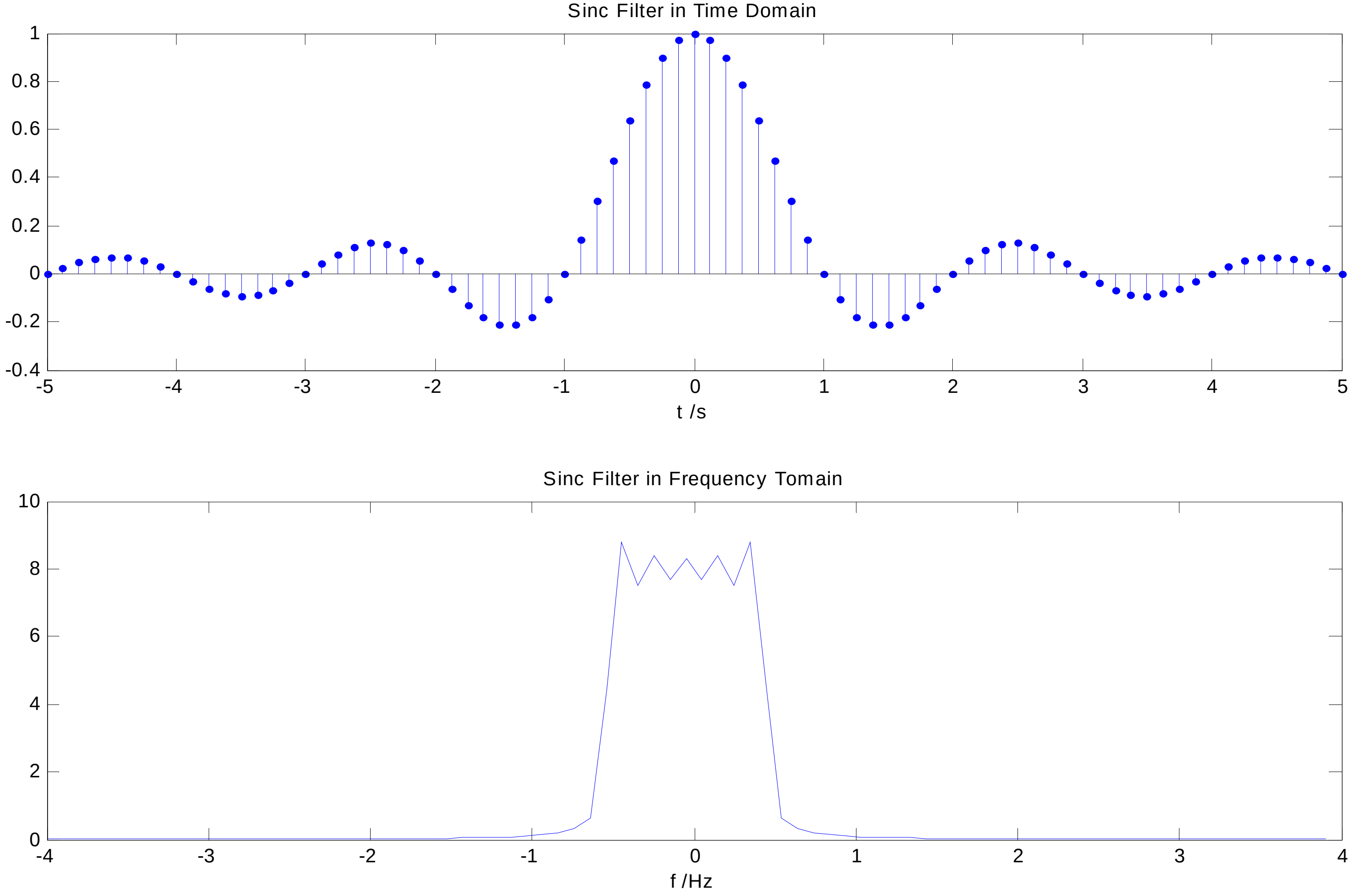
$$x(t) = \frac{\sin \pi t / T}{\pi t} = \mathrm{sinc}\left(\frac{\pi t}{T} \right)$$



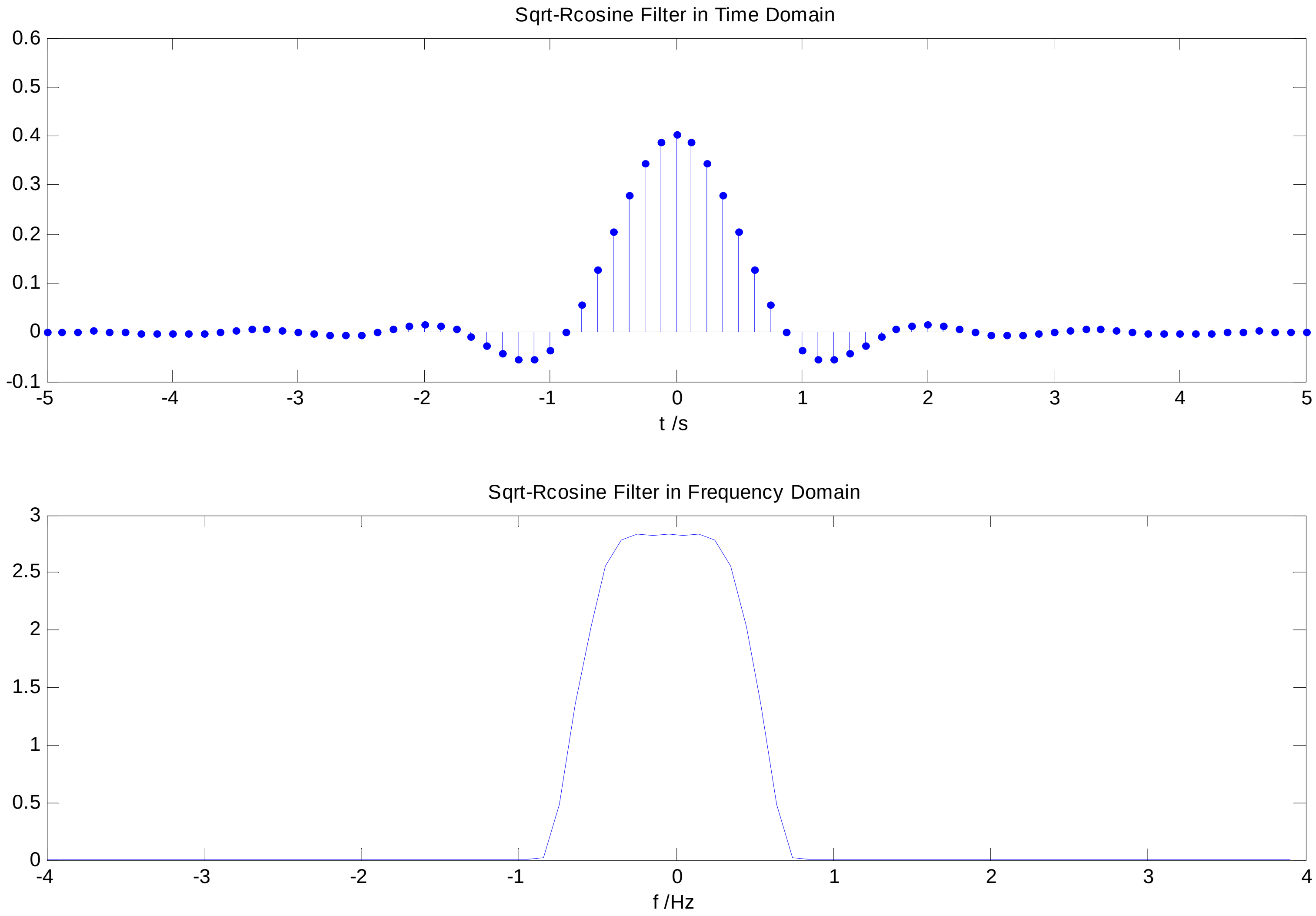


Pulses having a raised cosine spectrum.

Sinc Filter in time and frequency domain



Square root Cosine Filter in time and frequency domain



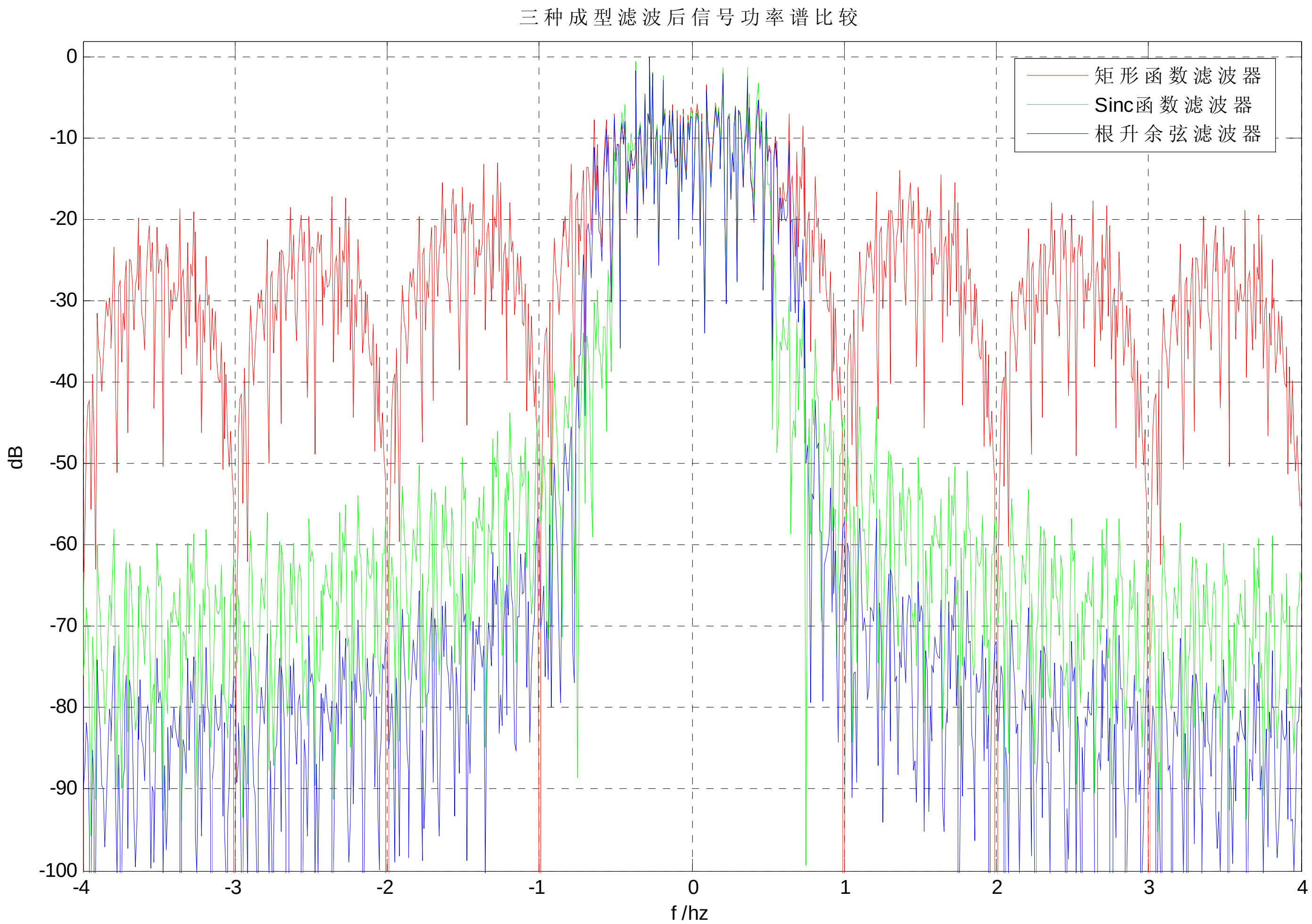
- Impulse response of Sinc and Sqrt_Cosine filter

Sinc Filter

$$h(t) = \frac{Sin\ (t)}{t}$$

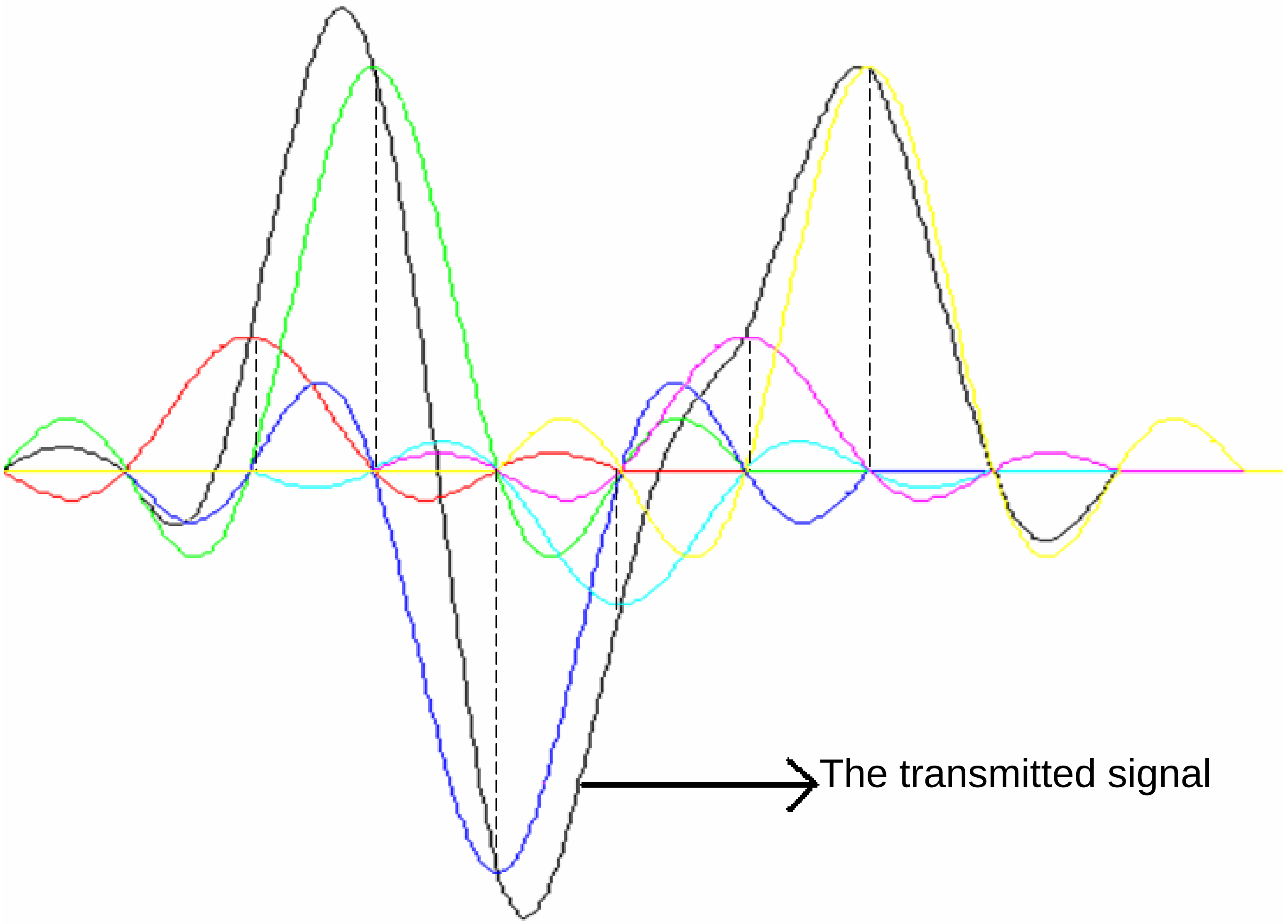
Sqrt_Cosine_Filter

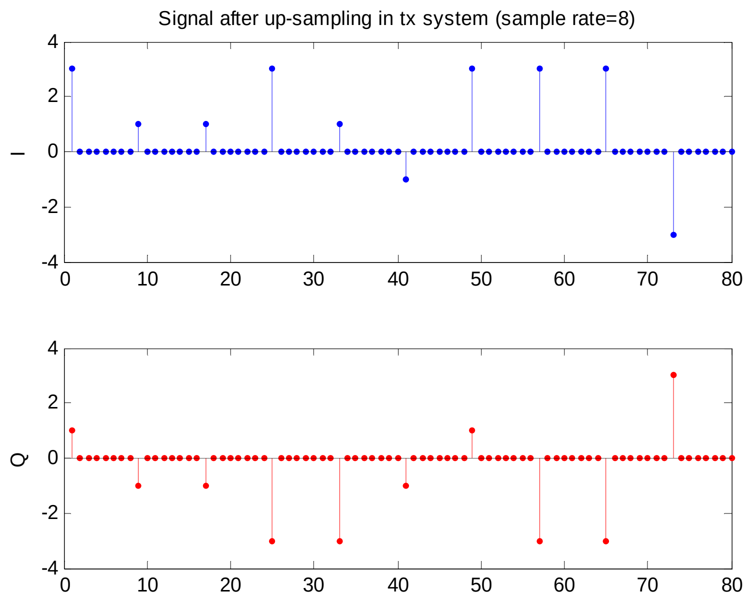
$$h(t) = 4R \frac{\cos((1+R)\pi t/T) + \dfrac{\sin((1-R)\pi t/T)}{4R\dfrac{t}{T}}}{\pi\sqrt{T}(1-(4Rt/T)^2)}$$

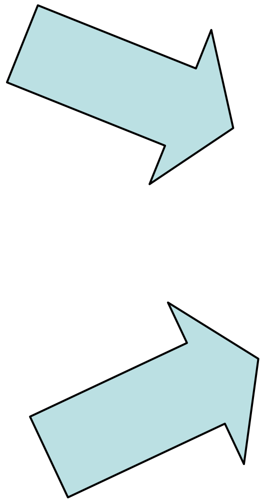


Simulation Spectrum of 3 filters

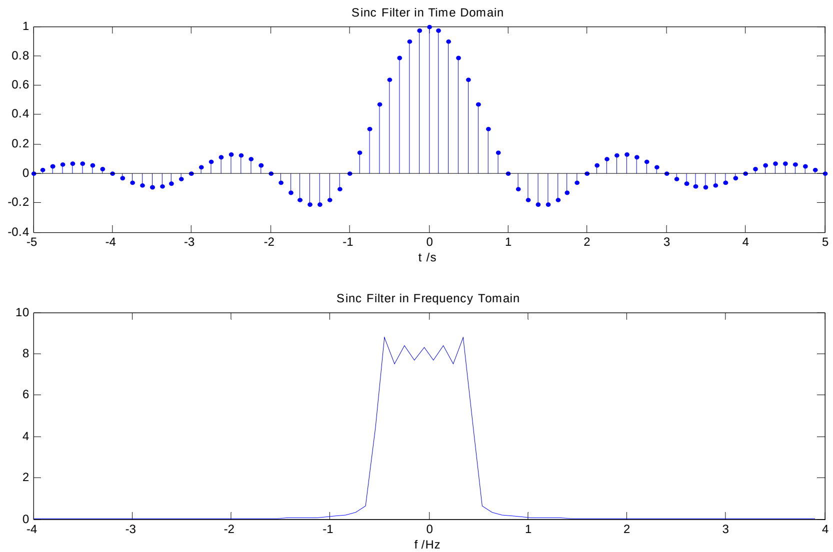
Symbols after up-sampling and lowpass filtering







convolution



you can use the filter function in matlab

rcosflt(X,Fd,Fs,'fir/sqrt',R,delay)
Example: rcosflt(X,1,fs,'fir/sqrt',R,delay)

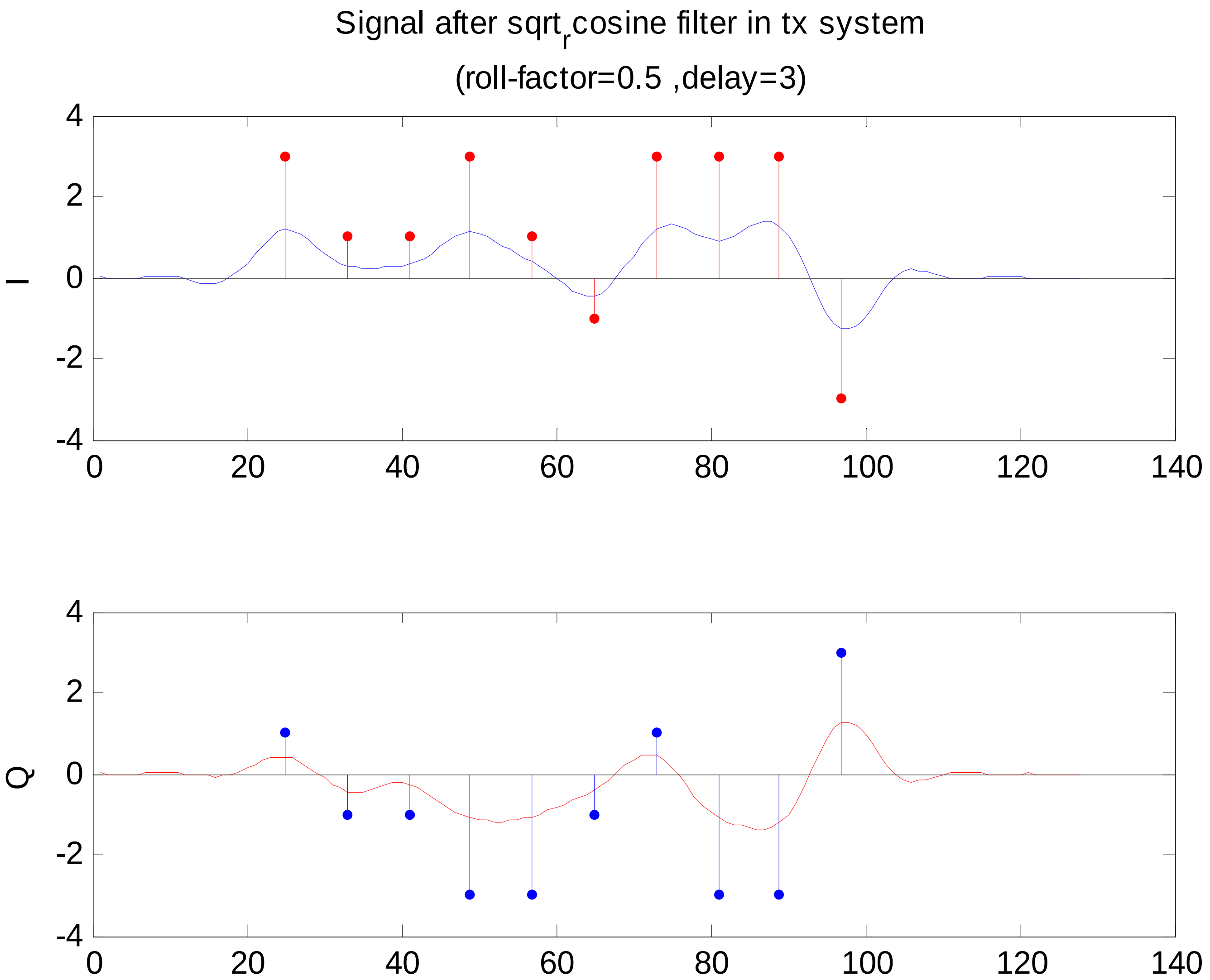
- X: input signal without up sampling
- The sample frequency for the input, X, is Fd (Hz).
- The sample frequency for the output, Y, is Fs (Hz). Fs must be an integer multiple of Fd. (note: if it is N times up-sampling, we can use Fd=1, Fs=N)
- The TYPE_FLAG gives specific filter design or filtering options.
- R: The rolloff factor, the value is from 0 to 1
- DELAY is number of sidelobs

Note: If the input signal X is up-sampled, the filter type should be 'fir/sqrt/fs',
For details, please use "help rcosflt" in MATLAB window

You may also program you own filter codes according to the given equation

```
function data_filtered=my_sqrt_filter(data_in,R,delay,sample_rate)
%data_in: up-sampled data; R: roll off factor; delay: number of sidelobe of each side
% sample_rate: up-sample rate
% give the time response of sqrt rcosine filter
    len_filter=delay*2*sample_rate+1;
    n=0:len_filter-1;
    n=-1*delay*sample_rate:delay*sample_rate;
    part1=cos((1+R)*pi*n/sample_rate);
    part2=sin((1-R)*pi*n/sample_rate)./(4*R*n/sample_rate);
    part3=1-(4*R*n/sample_rate).^2;
    hn=4*R*(part1+part2)./(pi*part3);
    % deal with the Inf points
    hn(delay*sample_rate+1)=4*R/pi+1-R;
    index=find(n==sample_rate/4/R | n==-1*sample_rate/4/R);
    hn(index)=(R*sqrt(2)/2/pi)*((2+pi)*sin(pi/4/R)+(pi-2)*cos(pi/4/R));
    % change the maximum value to make the raised cosine filter's maximum value=1
    max_value=max(conv(hn,hn));
    hn=hn/sqrt(max_value);
% the input data pass through the filter
    data_filtered=conv(data_in,hn);
```

The signal after the filter



Discussions

1. Why the amplitude of the original samples are not same as filtered one
2. can we have inter-symbol interference free transmission

Carrier modulation

Lowpass signal is

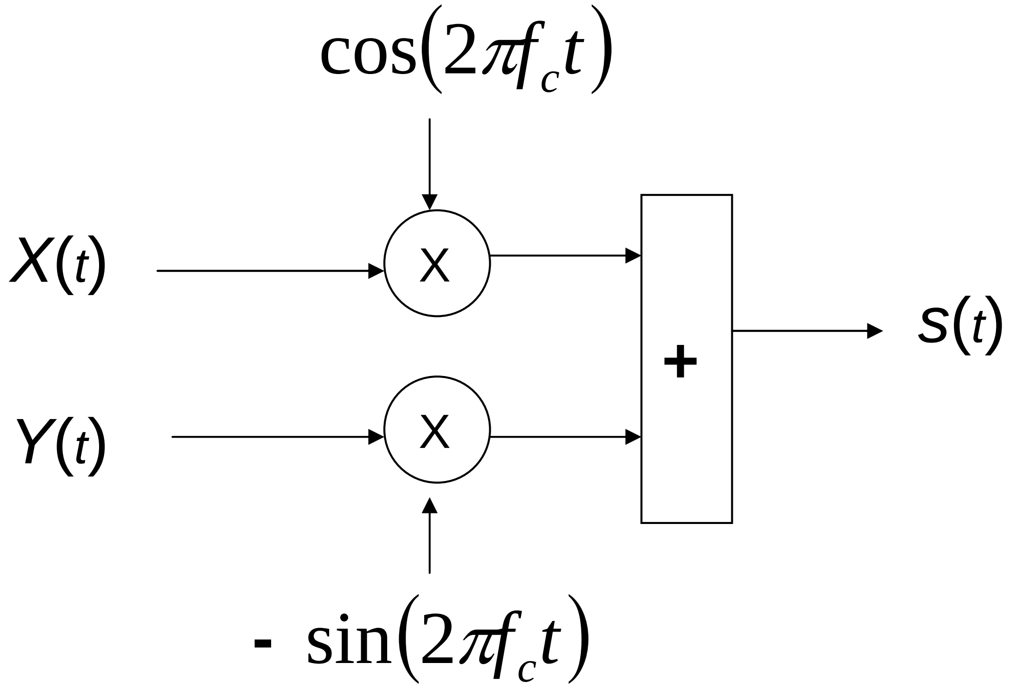
$$Z(t) = I(t) + jQ(t)$$

The carrier modulation is

$$S(t) = \mathrm{Re}\{Z(t)\exp(j2\pi fct)\}$$

The carrier modulated signal

$$s(t) = \mathrm{Re}\left[\left(x(t) + jy(t)\right)\exp\left(j2\pi f_c t\right)\right]$$
$$= x(t)\cos\left(2\pi f_c t\right) - y(t)\sin\left(2\pi f_c t\right)$$



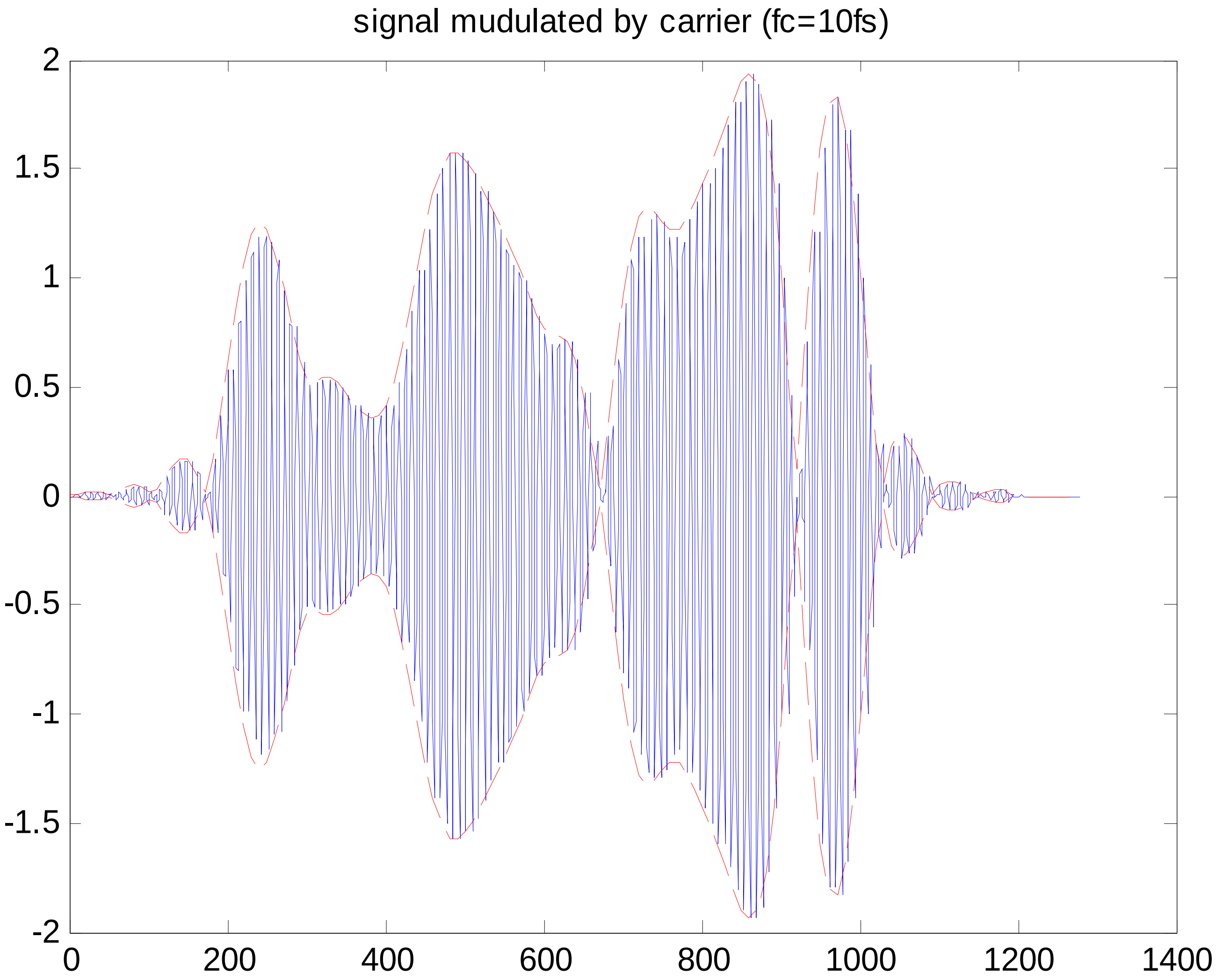
In the simulation, you may chose 10 times carrier modulation

$$fc = 10\,fs$$

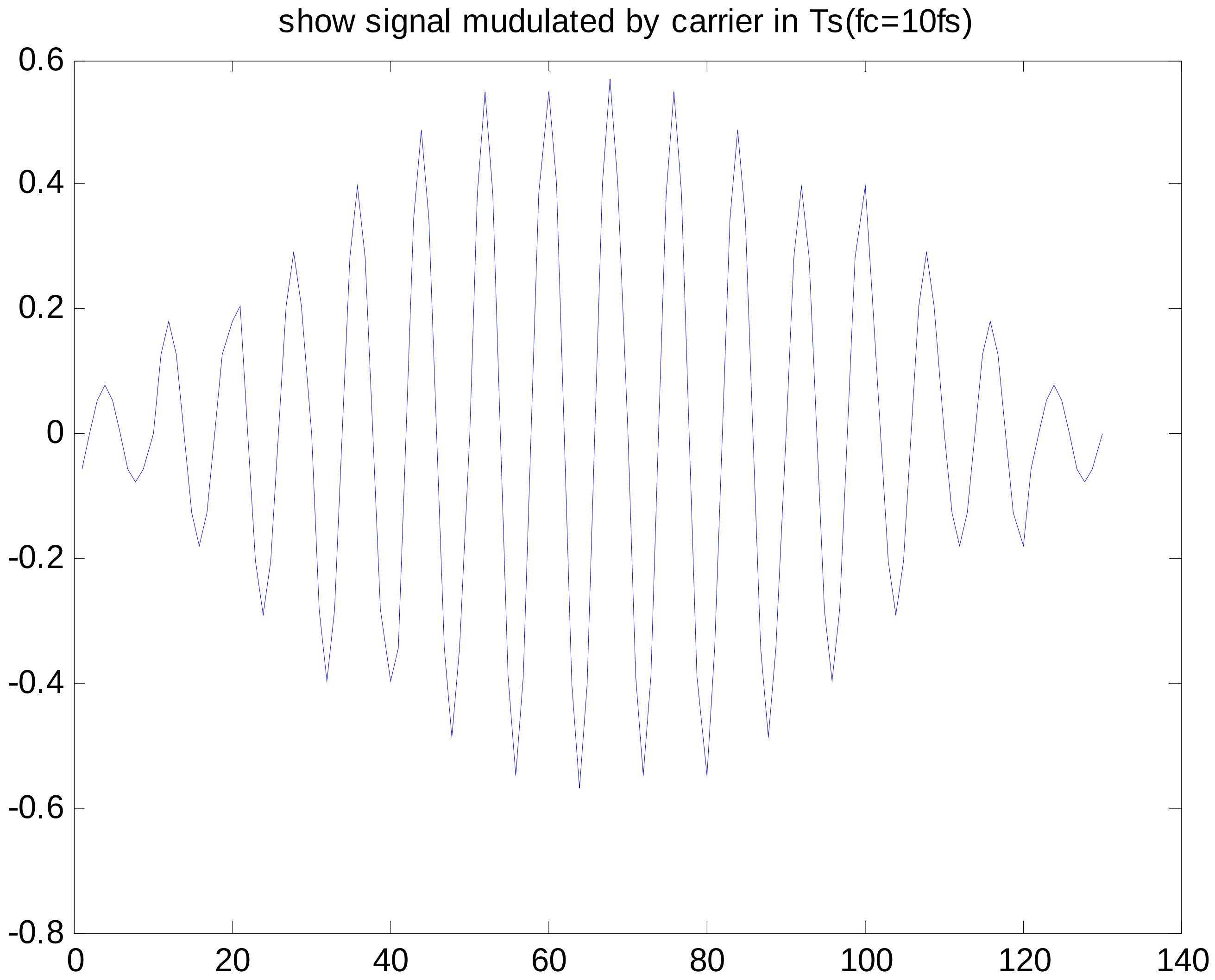
Note: The resolution of the baseband signals is not enough since it has only 8 time up sampling. However the carrier frequency modulation needs more resolutions

To solve problem you need make the interpolation for the baseband signals

The carrier modulated signals



The carrier modulated signals



Add the AWGN signal in baseband

- Using awgn() function
- y = awgn(x, snr, 'measured');

$$
\begin{aligned}
& snr(dB) \\
& = \frac{E_S}{N_0}(dB) - 10 \times \lg(Nsamp) \\
& = \frac{E_b}{N_0}(dB) + 10 \times \lg(\underline{k}) - 10 \times \lg(\underline{Nsamp})
\end{aligned}
$$

- For 16QAM, k = 4
- Nsamp is the over sampling rate, here is 8
- “Measured” means the function first measure the signal power, and then add the noise

Why consider the up-sampling rate when add AWGN signals?

Question : if transmitted same signal twice, at the receiver side make the combination of the signals, how the SNR of the received signals will change compare with transmit one time

The SNR for each transmission

$$SNR = \frac{E_S}{\sigma^2}$$

The received signals are

$$y_1 = x + n_1 \quad y_2 = x + n_2$$

Variance of the sum of random variables

Assume X_i, i=1,2,…n is iid random variables, its mean value is m_x， variance is $\sigma_x{}^2$。 **Define Y is the sum of X,**

$$Y = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Then

$$E\left[Y\right] = m_x$$

$$\sigma_y^2 = \frac{\sigma_x^2}{n}$$

$$\sigma_y = \frac{\sigma_x}{\sqrt{n}}$$

Answer:

1. **Transmit one time, the signal energy is** $E_s = |x|^2$ $\quad SNR = E_s / \sigma^2 = |x|^2 / \sigma^2$
2. **Transmit twice and add the received signal together** $y_1 + y_2 = 2x + n_1 + n_2$

   signal power $E_s^{'} = |2x|^2 = 4|x|^2 = 4E_s$ **noise power** $\sigma^{'2} = 2\sigma^2$

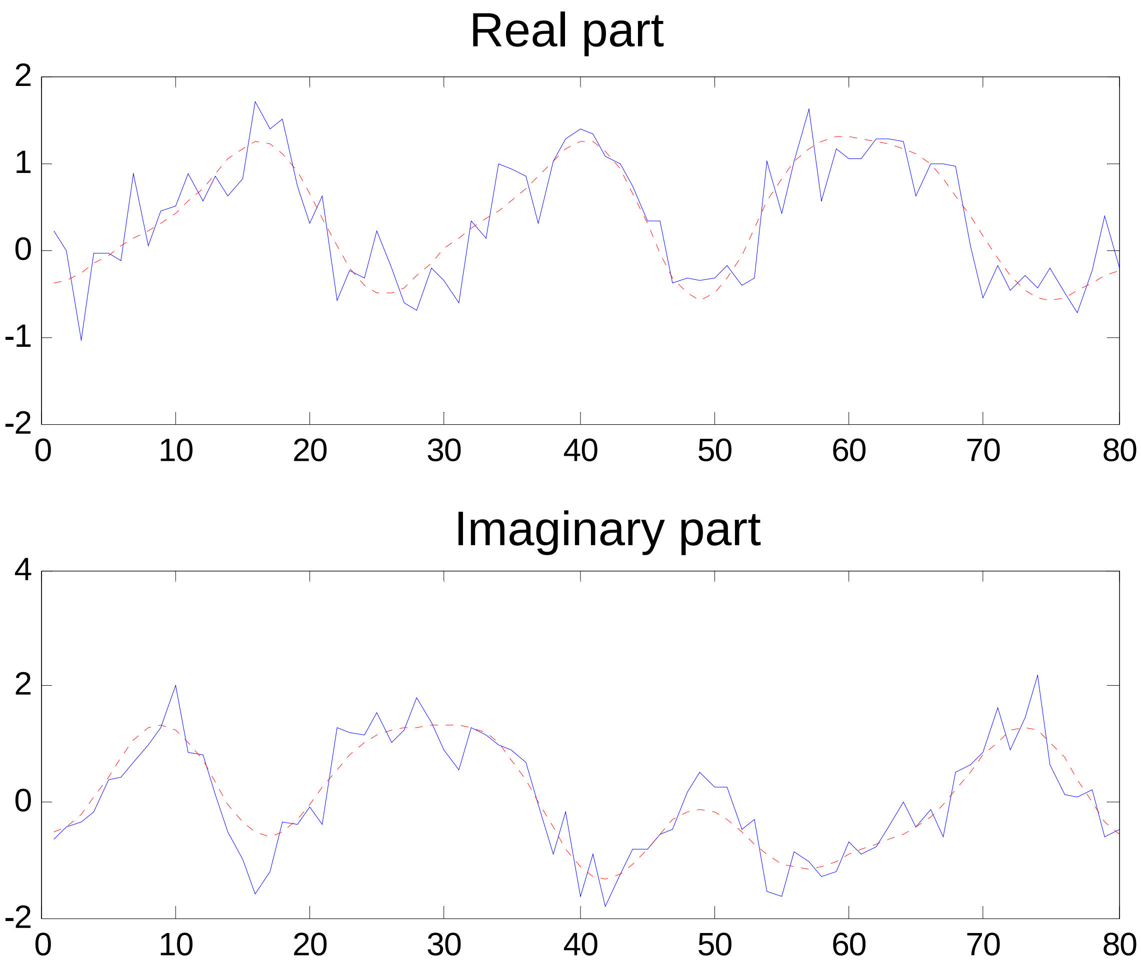
$$2\sigma^2$$

Therefore the SNR of transmit twice becomes

$$SNR' = \frac{4E_S}{2\sigma^2} = 2SNR$$

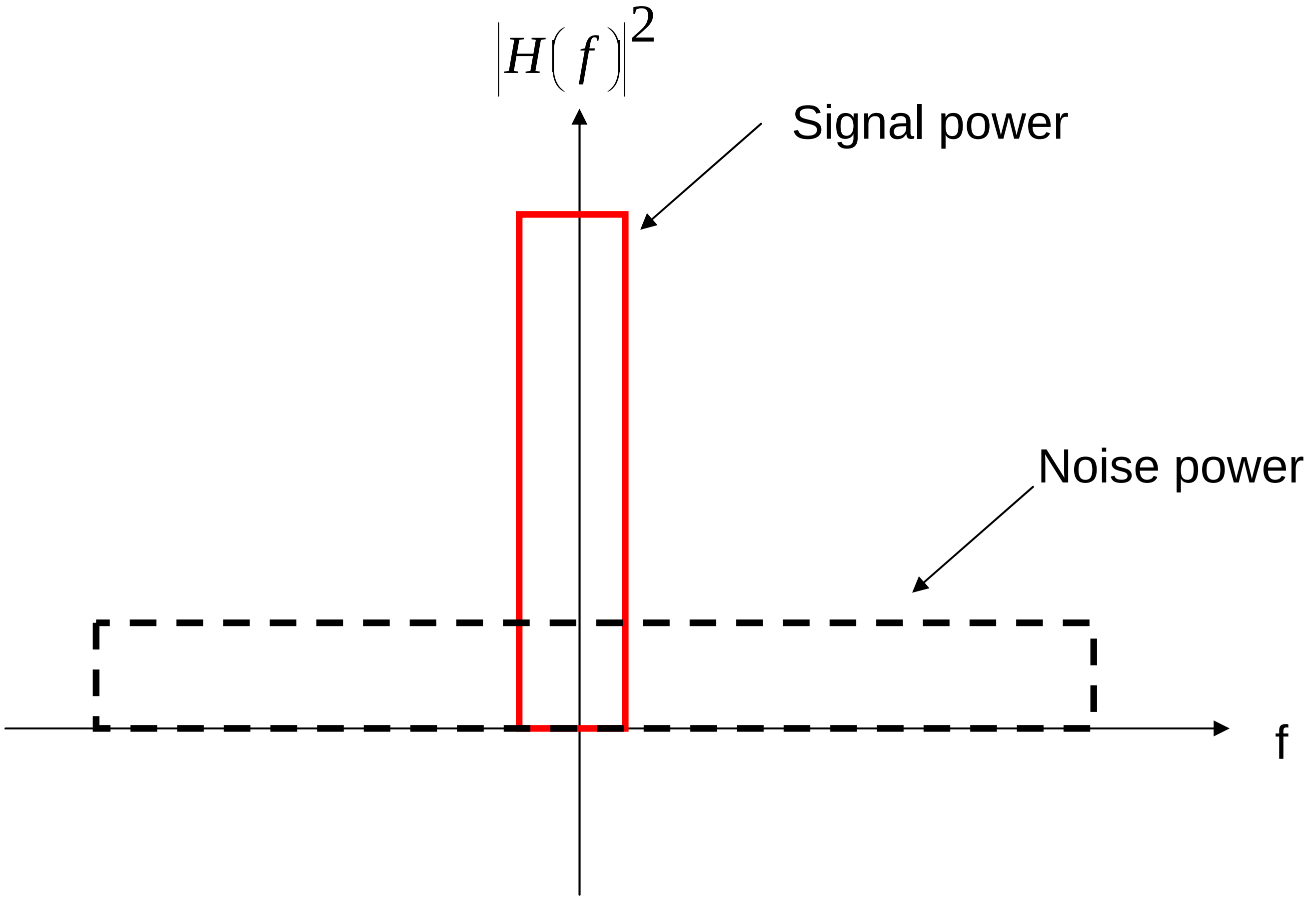
Conclusion: In AWGN Channel, transmit one signal twice will make SNR increase 3dB (two times)

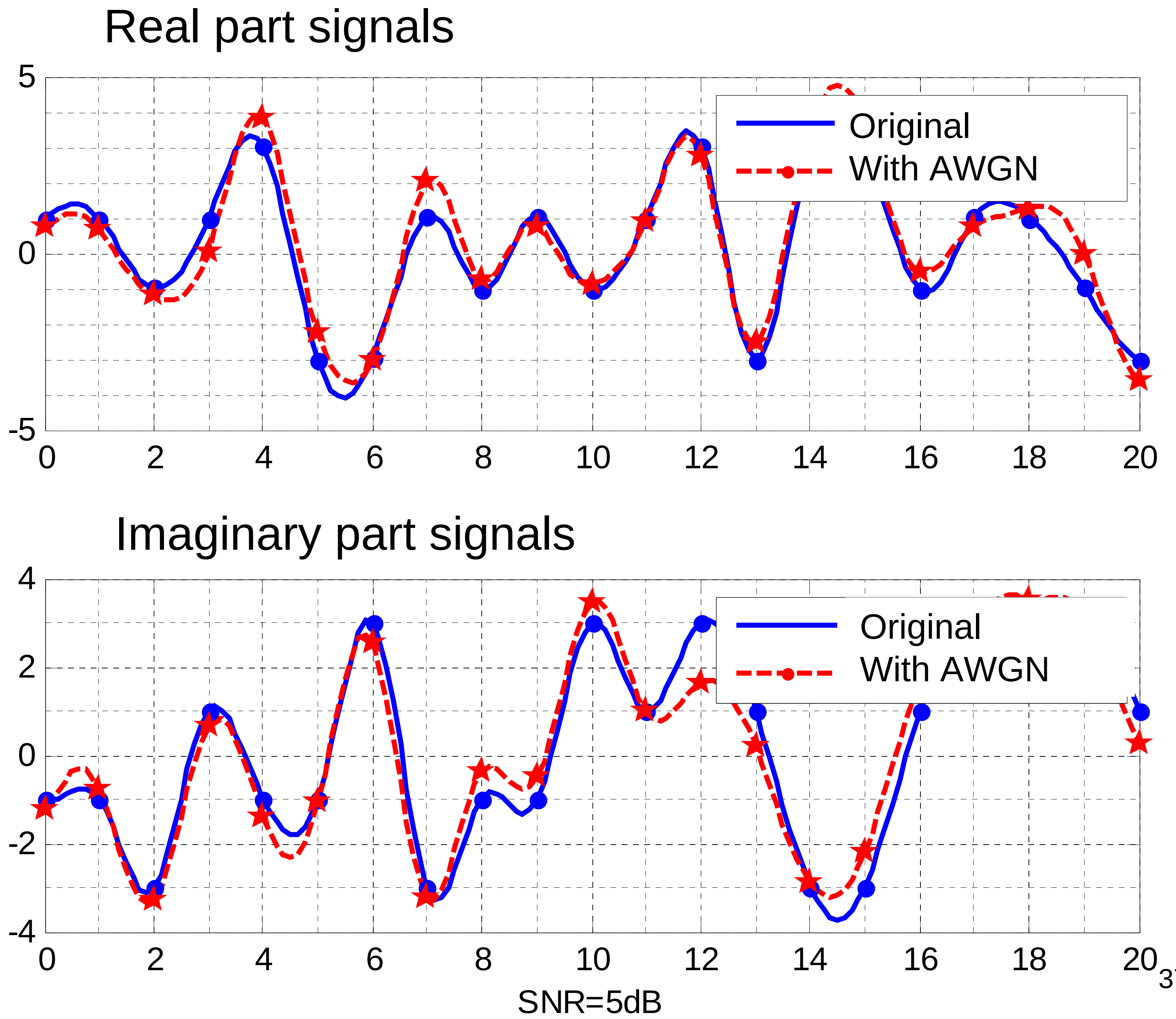
Received signal with noise

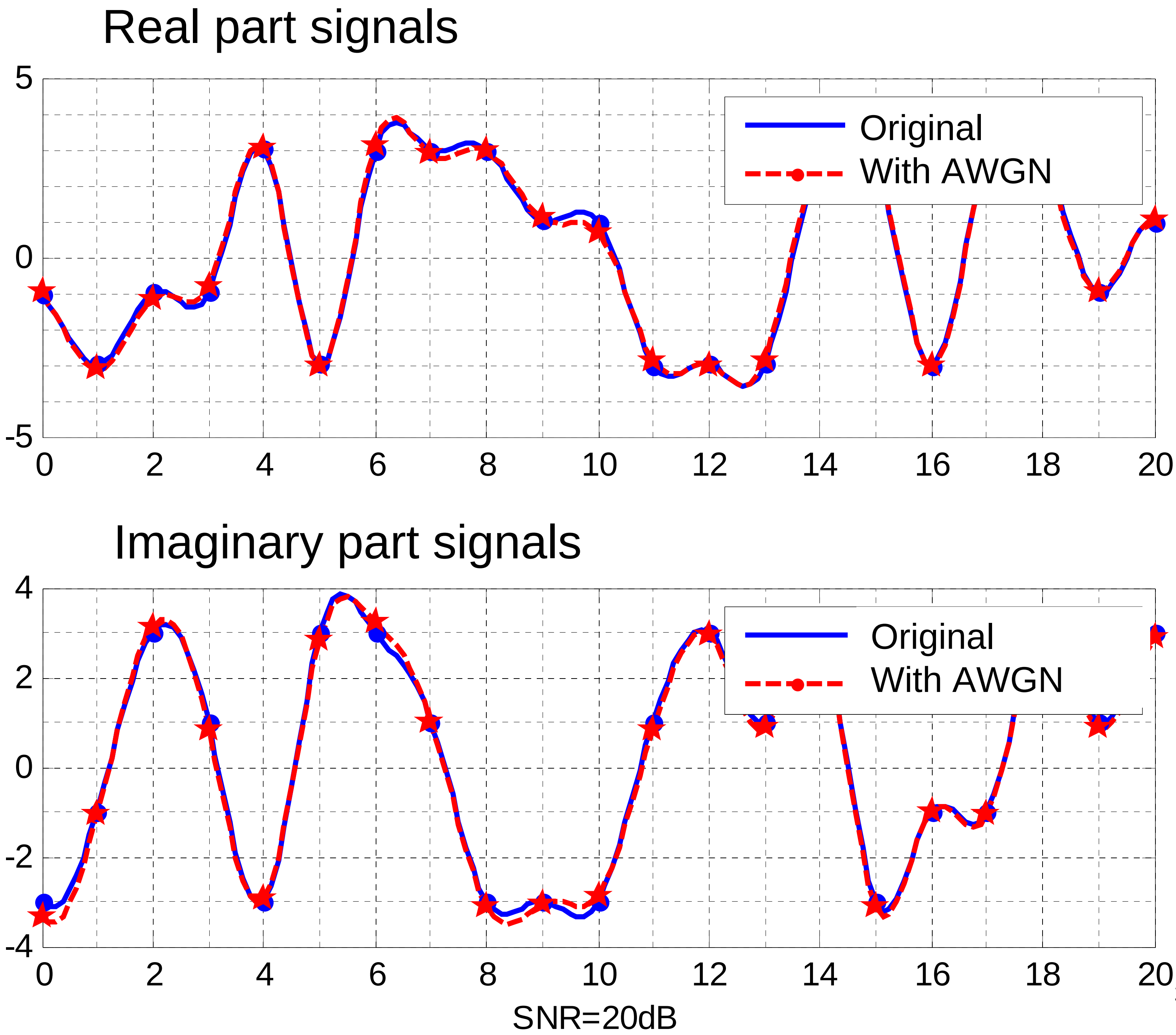


Filter at the receiver side

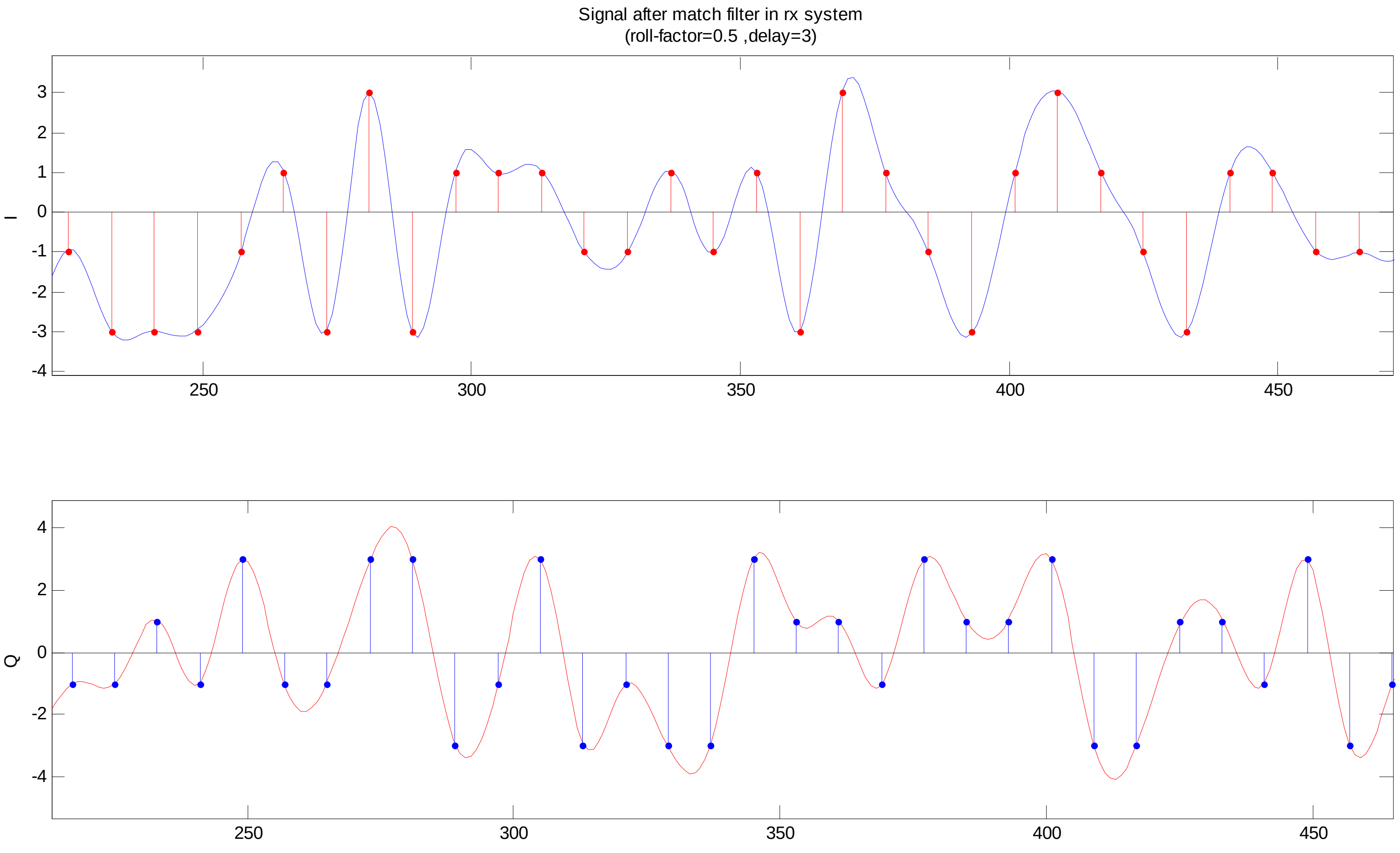
Use the matched filter as the transmitter side



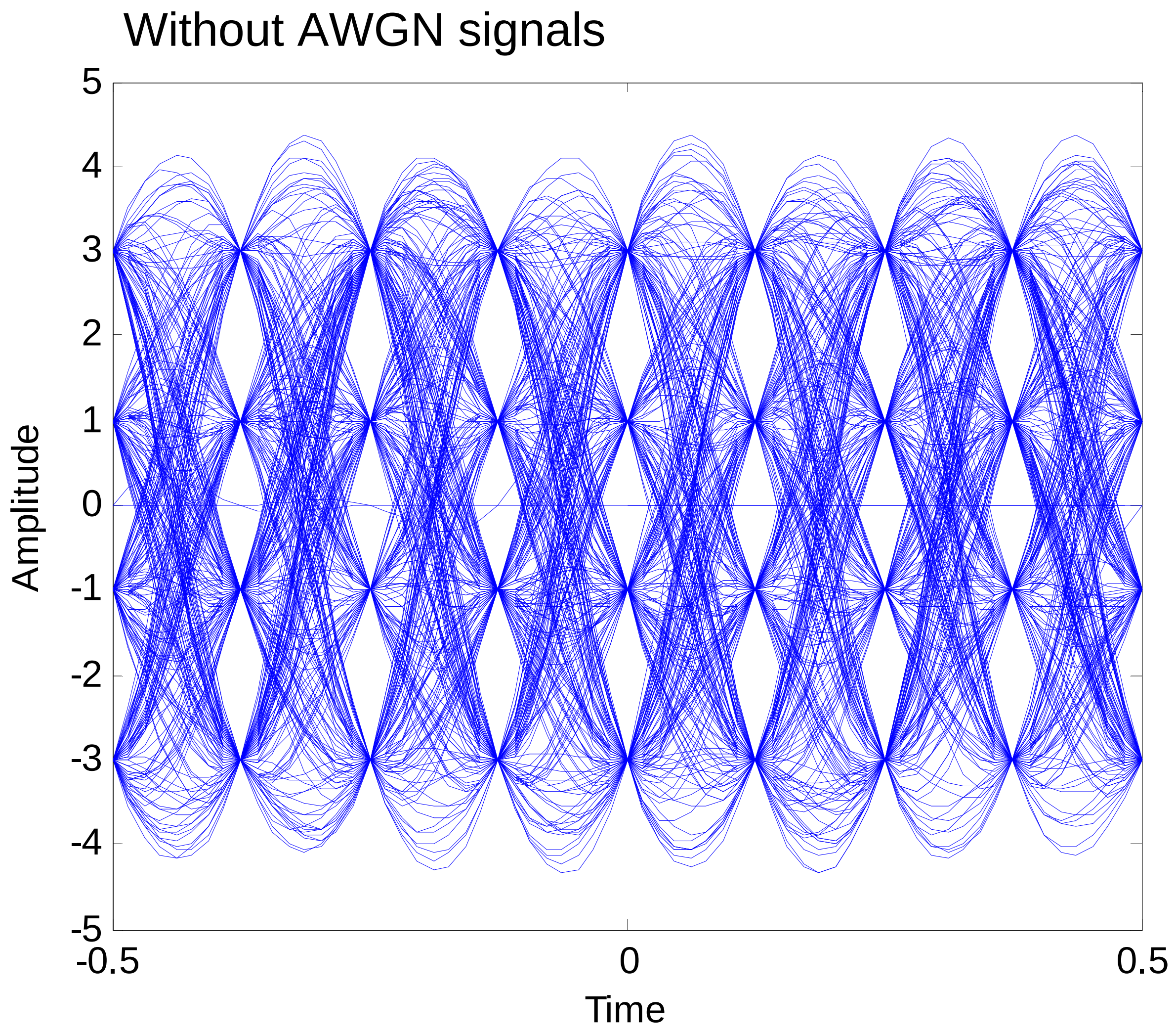

Real part signals
Original
With AWGN
5
0
-5
0 2 4 6 8 10 12 14 16 18 20
Imaginary part signals
Original
With AWGN
4
2
0
-2
-4
0 2 4 6 8 10 12 14 16 18 20
SNR=5dB


Real part signals
5
0
-5
0
2
4
6
8
10
12
14
16
18
20
Original
With AWGN
Imaginary part signals
4
2
0
-2
-4
0
2
4
6
8
10
12
14
16
18
20
Original
With AWGN
SNR=20dB

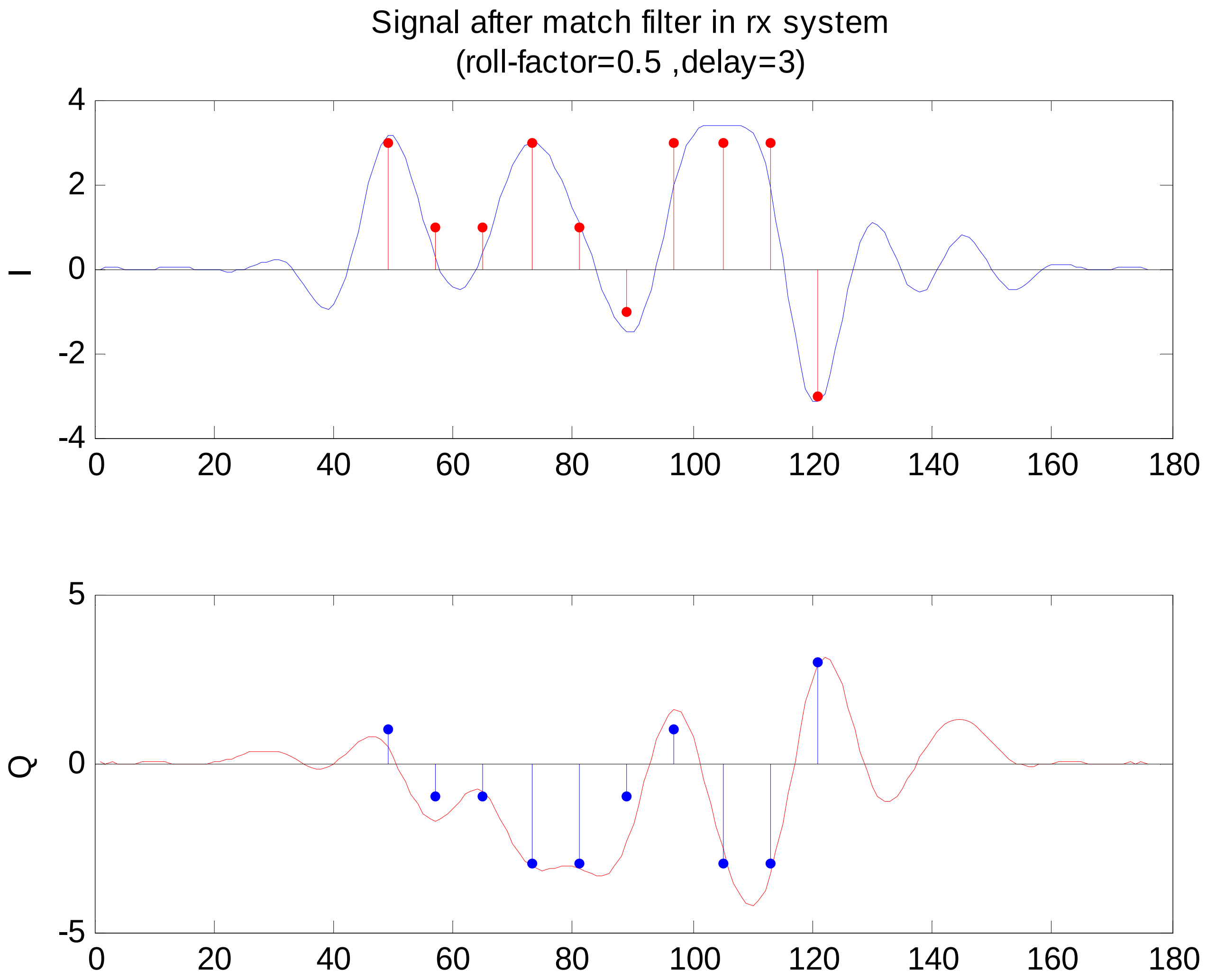
The signal after the receiver filter (without noise)



Eye diagram without noise

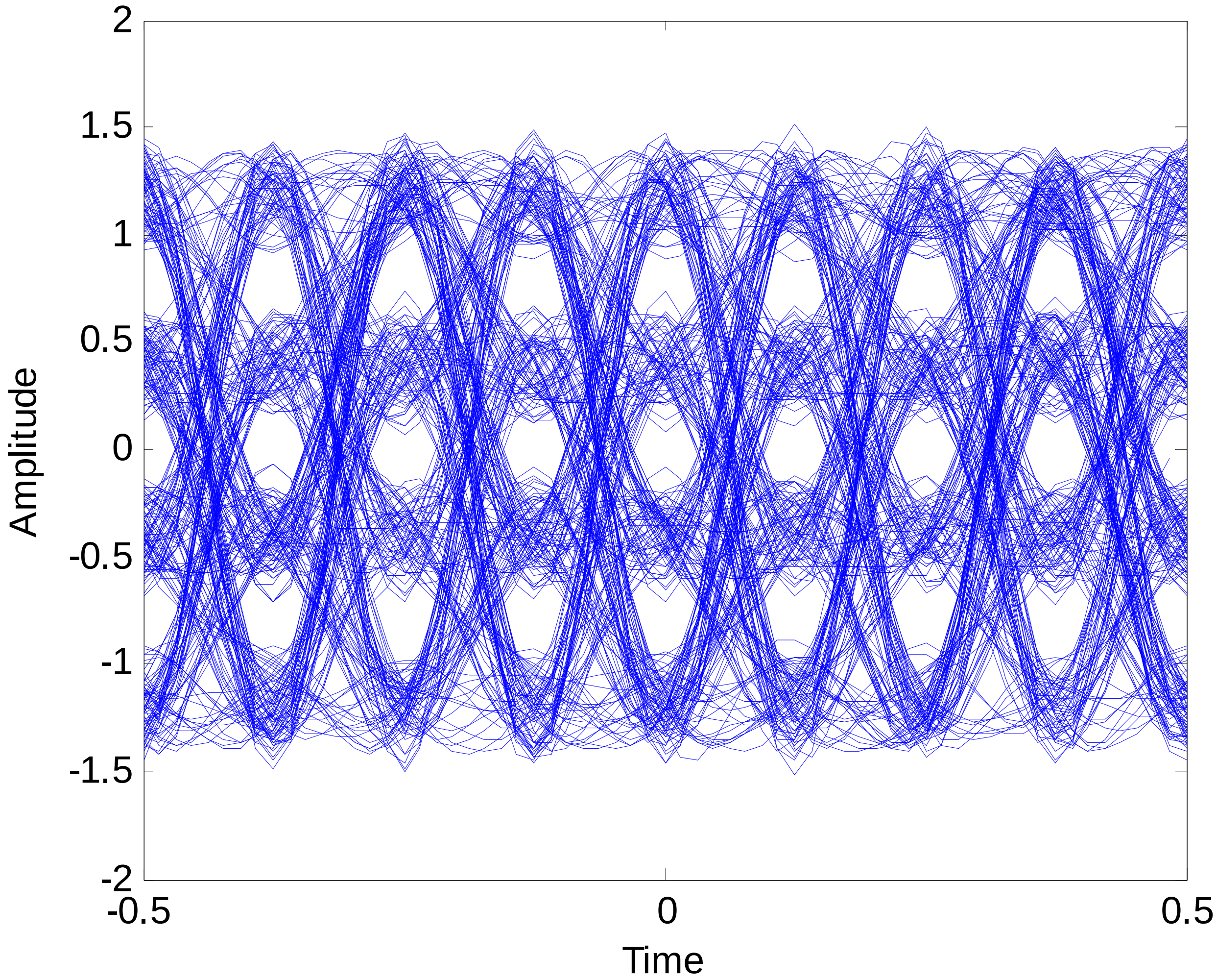


The signal after the receiver filter (with noise)

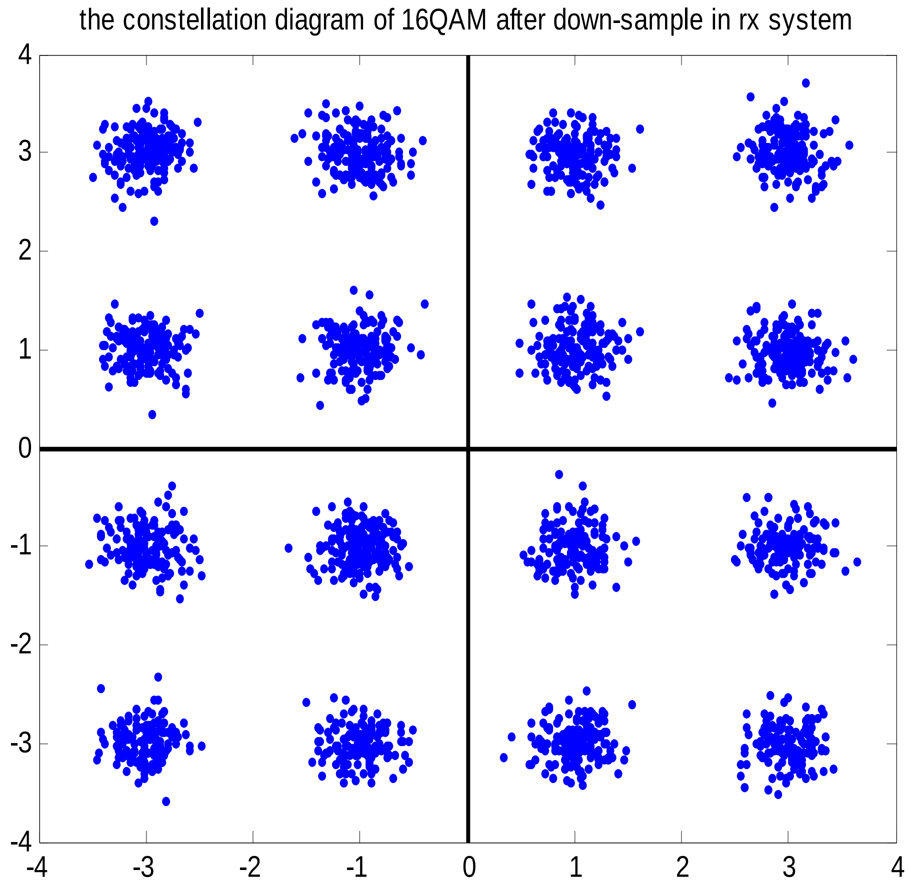


Eye diagram

With AWGN signals



Down sampling (sampling rate is 8)



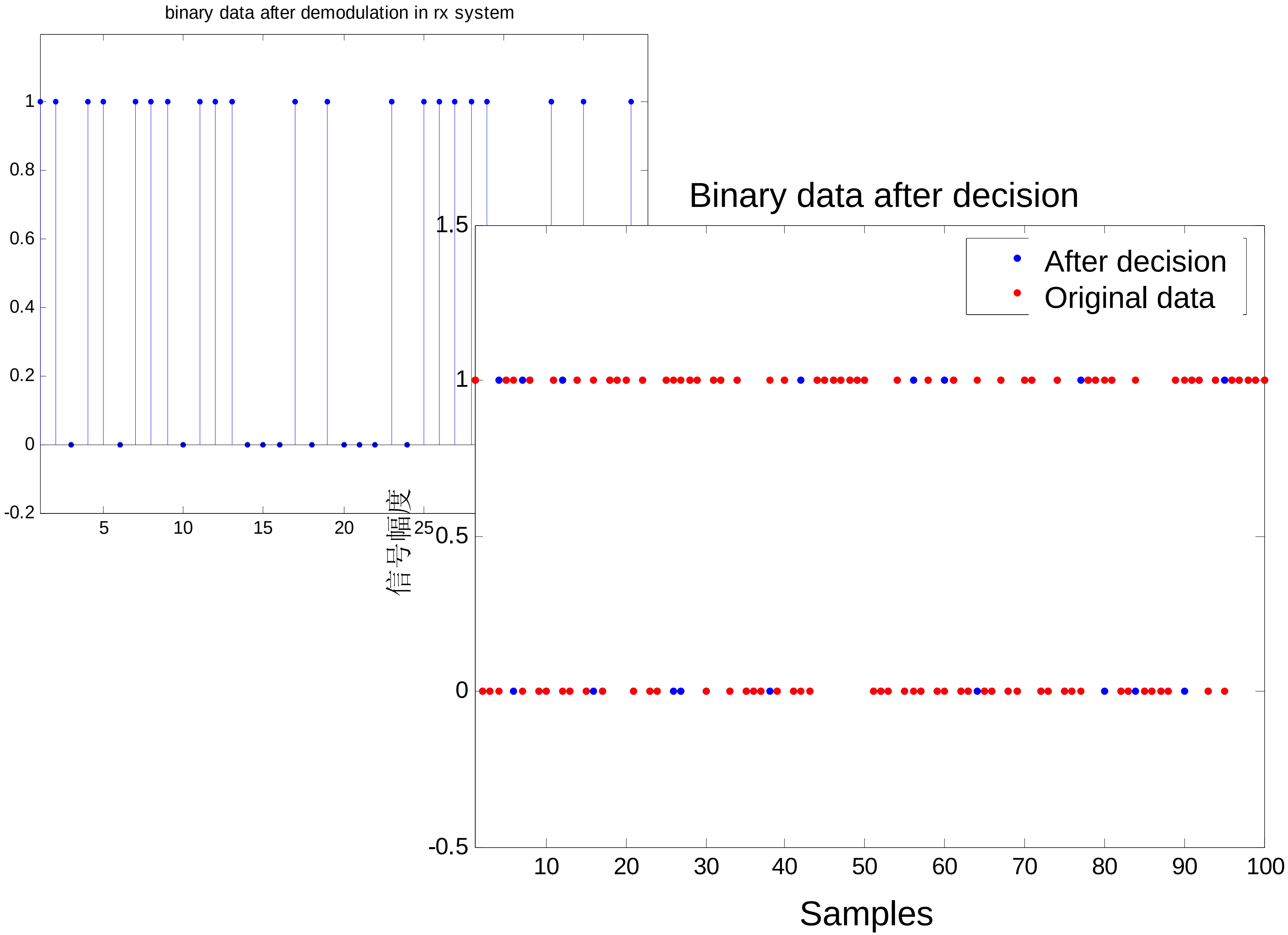
demodulation

Method 1
- Get the distances of the received samples to each signal point
- Chose the signal with minimum distance

Method 2
Bit by bit decision

	Bit s		Bit 2		Bit 3		Bit 4	
threshould	I>0	I<0	Q>0	Q<0	1>I>-1	I>1 or I<-1	1>Q>-1	Q>1 or Q<-1
decision	1	0	1	0	1	0	1	0

Get bit values after demodulation



BER comparison

Make the theoretical curve and simulated curve at same figure

$$P_M = 4Q\left(\sqrt{\frac{3}{M-1}\frac{\varepsilon_{av}}{N_0}}\right) - 4\left(Q\left(\sqrt{\frac{3}{M-1}\frac{\varepsilon_{av}}{N_0}}\right)\right)^2$$

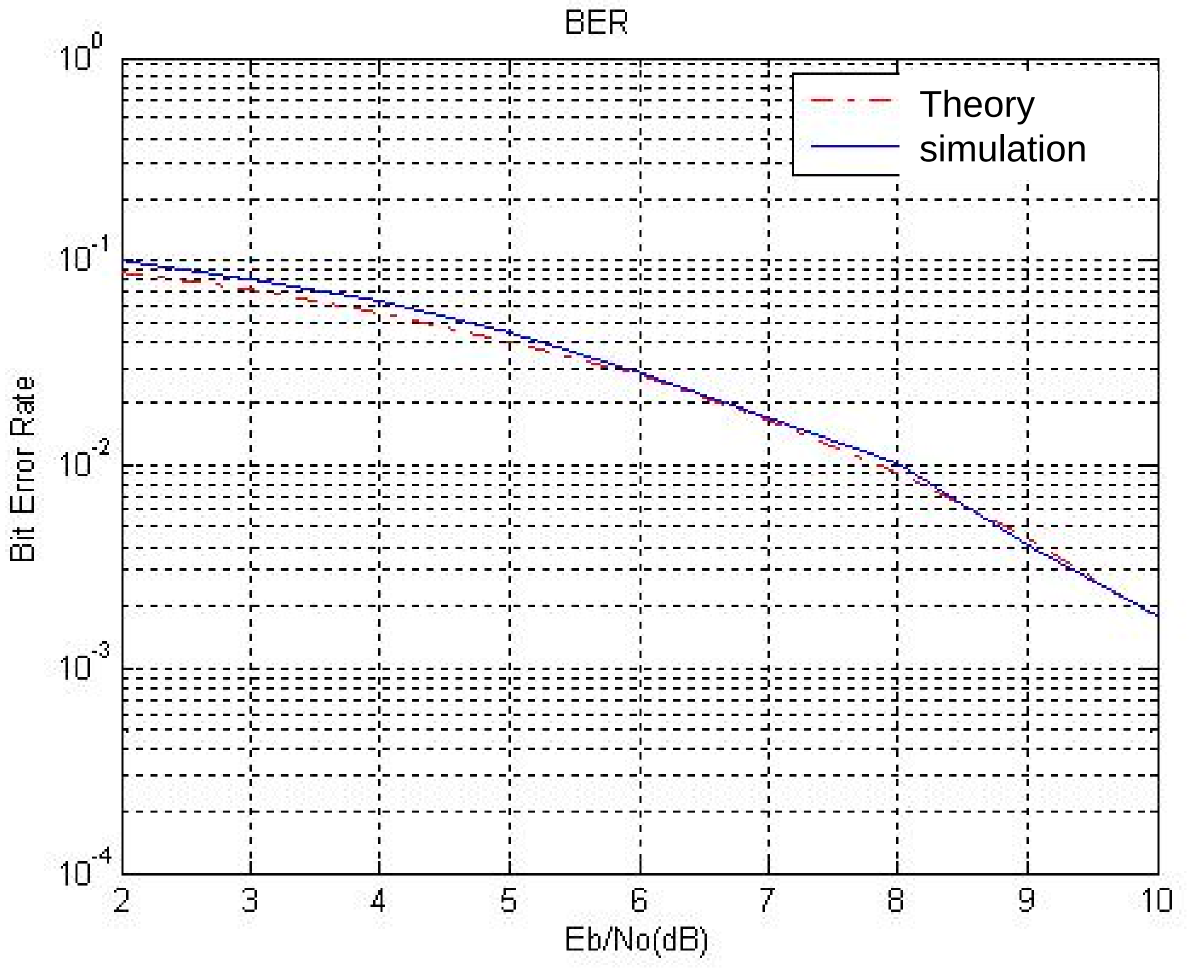
$$\leq 1 - \left[1 - 2Q\left(\sqrt{\frac{3}{M-1}\frac{\varepsilon_{av}}{N_0}}\right)\right]^2$$

The relation between Q function and erfc() fuction

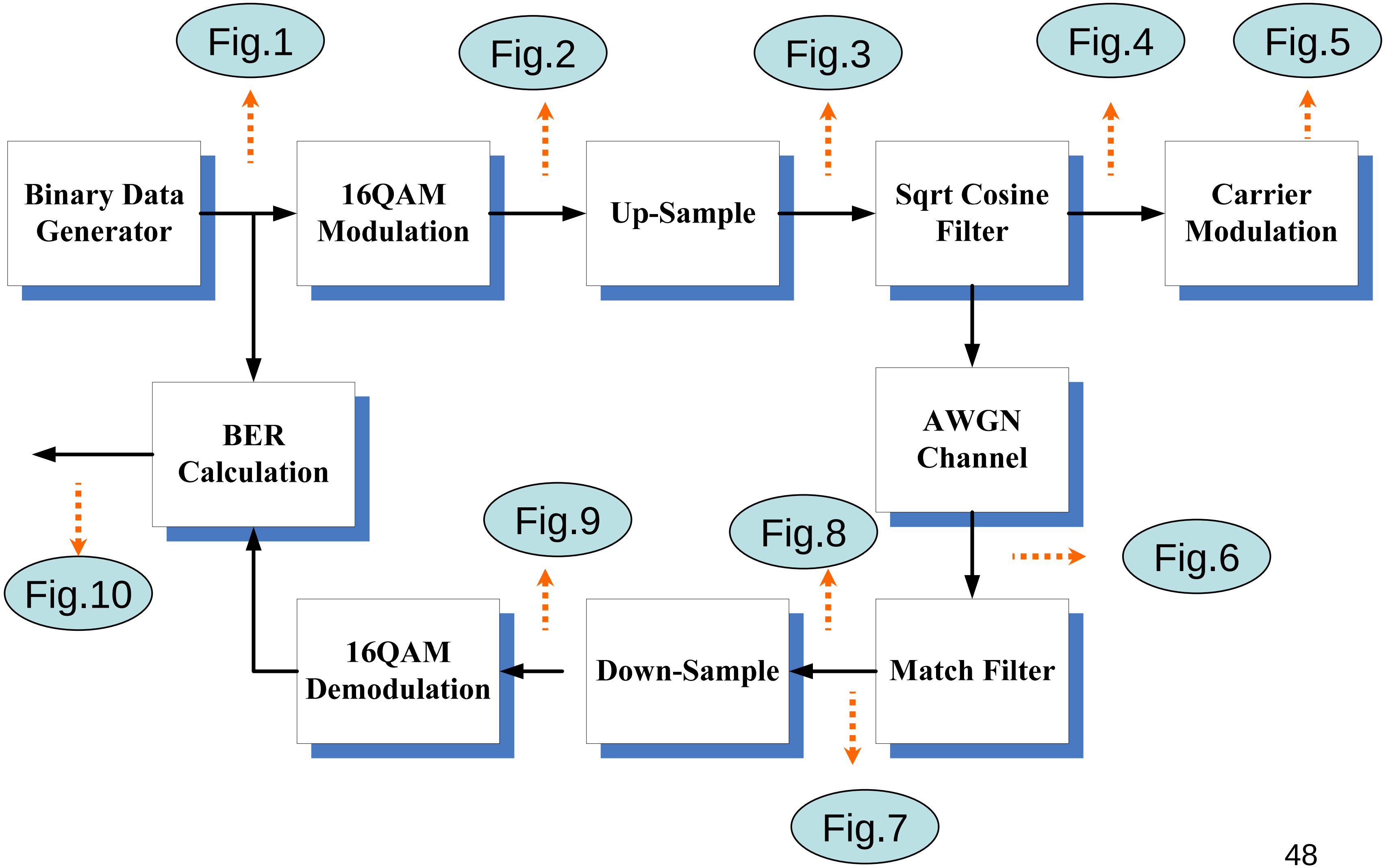
$$Q(x) = \frac{1}{2}\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)$$

Qfunc( ) and erf( ) in Matlab can be used

Simulation results



Block diagram of the communication system



Description for each figure

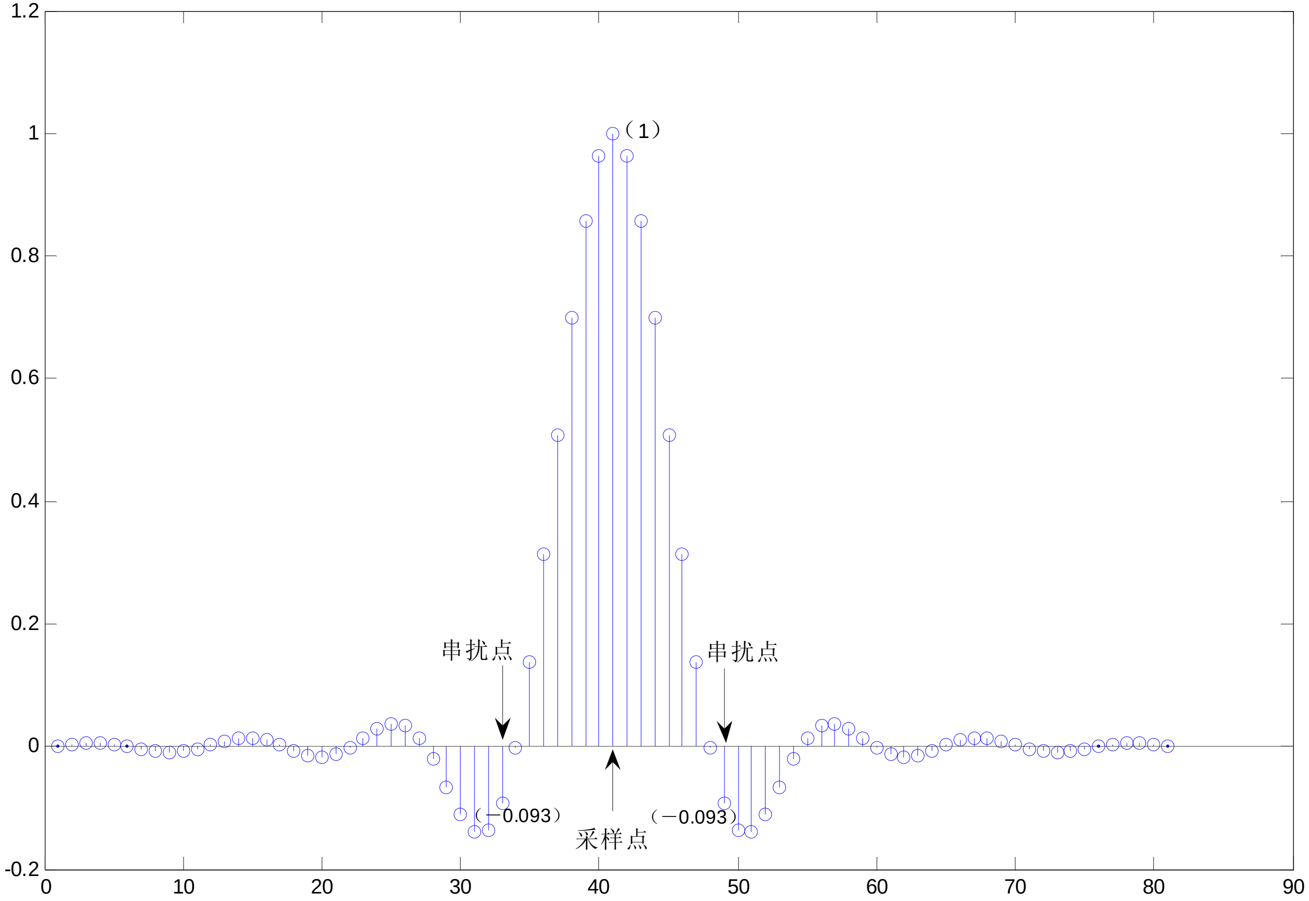
- Fig.1: Show the binary data
- Fig.2: Show the transmitted signal constellation
- Fig.3: Show the up-sampled signals (real and imaginary parts)
- Fig.4: Show the filtered baseband signals (real and imaginary parts)
- Fig.5: Show the carrier modulated signals (Real signals)
- Fig.6: Show the baseband signals with AWGN together with Fig.4
- Fig.7: Show the filtered baseband signals together with Fig.4
- Fig.8: Show eye diagram (Real or imaginary part)
- Fig.9: Show the constellation of the received signals
- Fig.10:Show the BER curve with respect to different SNR or Eb/No (Draw the theoretical and simulation curves at the same figure)

Further discussions

- Problem 1: Sqrt_Rcosine Filter & ISI
- Problem 2: Decision & BER under Gray Code
- Problem 3: Speed up your Simulation

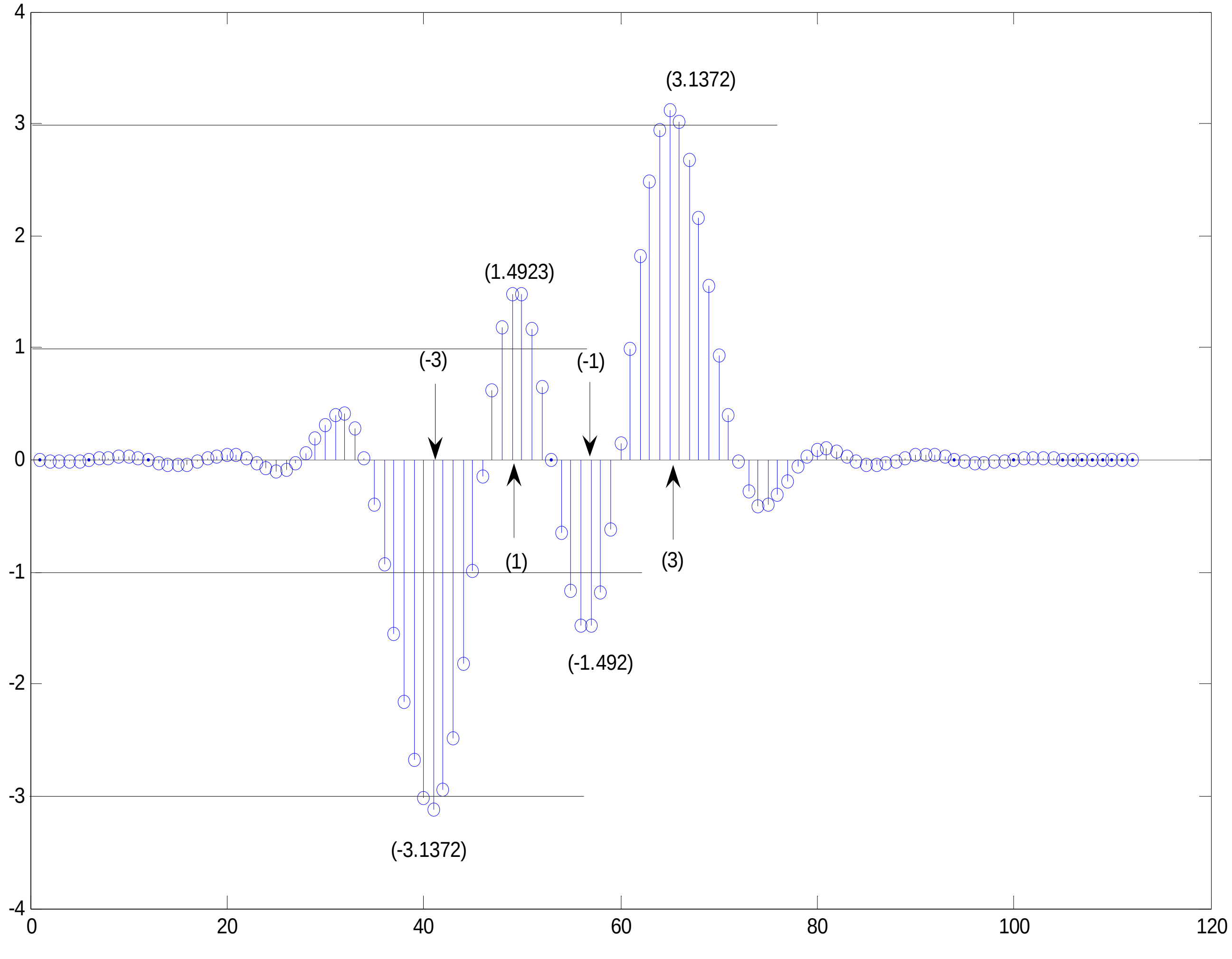
Problem 1

- Does Square Rcosine Filter has ISI?
- Simulation:

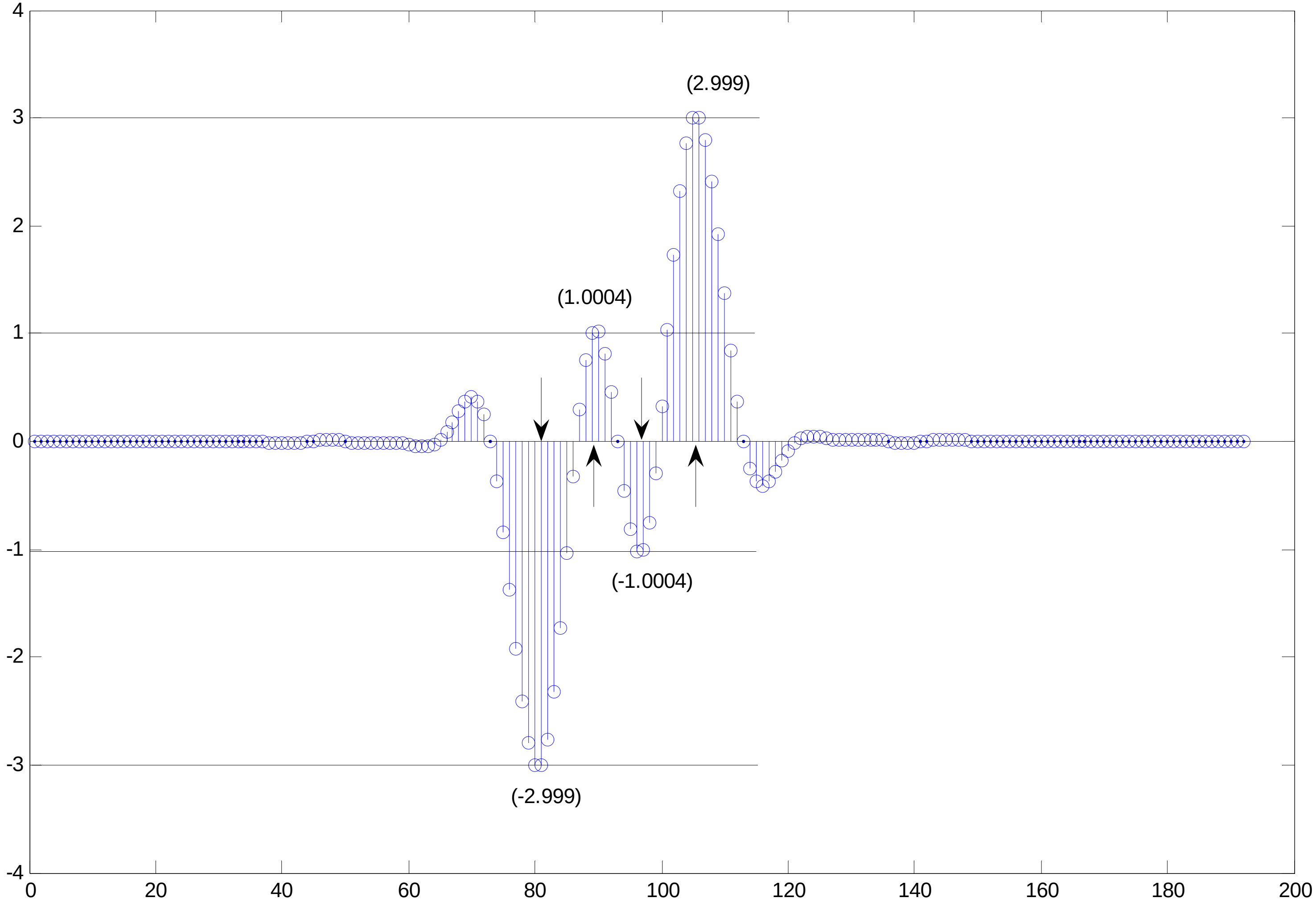


Normalize the value to 1 at the sampling point

- Pass the data sequence [-3,1,-1,3] after insertion into the filter:

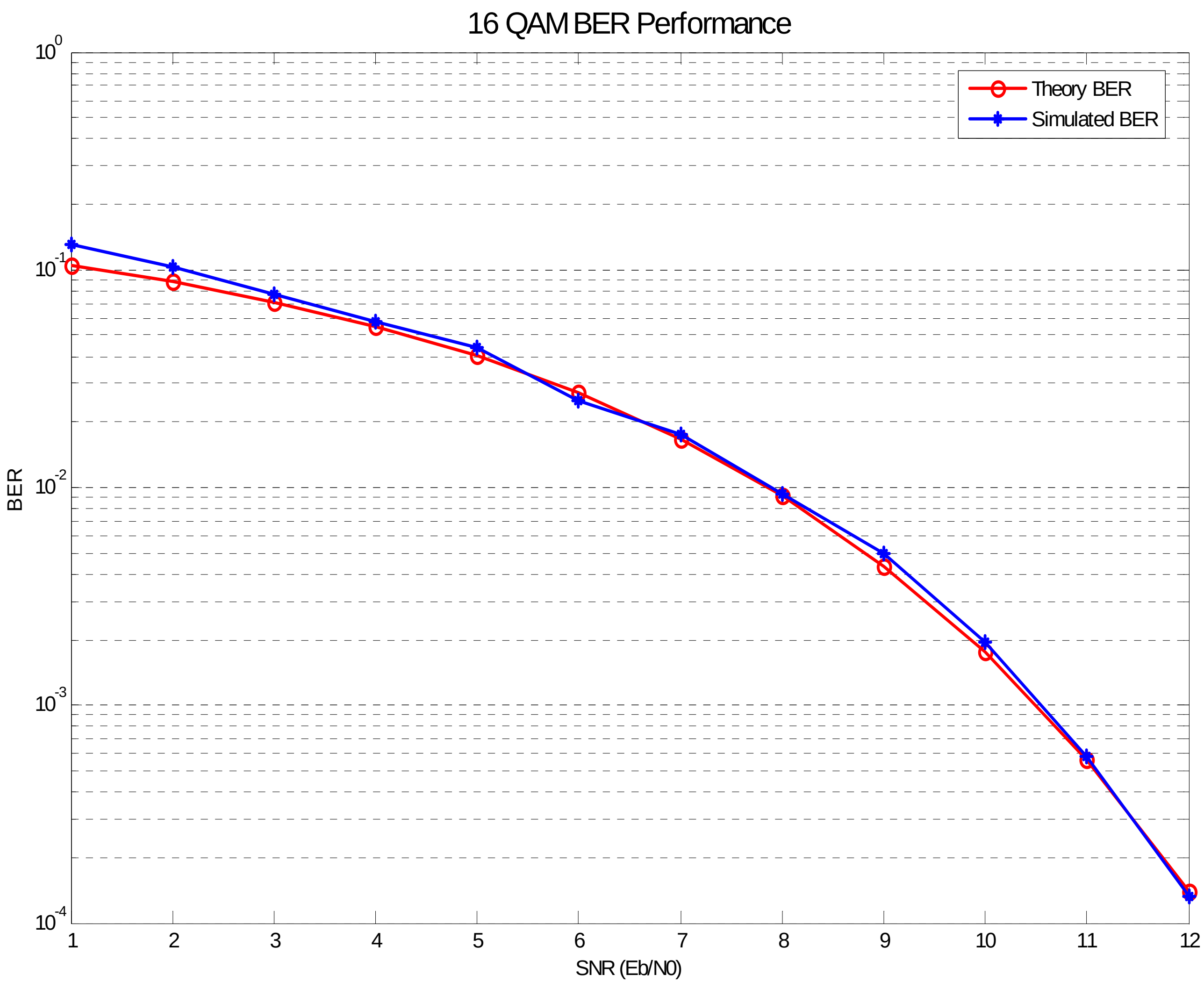


- Pass the filtered data into the same filter once more (Matched filter)



- Conclusion1
- Sqrt_Rcosine Filter has ISI, a pair of them does not have ISI, equals to a Rcosine Filter
- Similarly, Sinc Filter has no ISI, can also be implemented in pairs
- Rcosine Filter has no ISI, but a pair of them has ISI, thus we seldom use two in our system

Problem 2: ***Why simulated curve is bit above the theory one under lower SNR?***



Cause: we made an approximation under Gray coding.

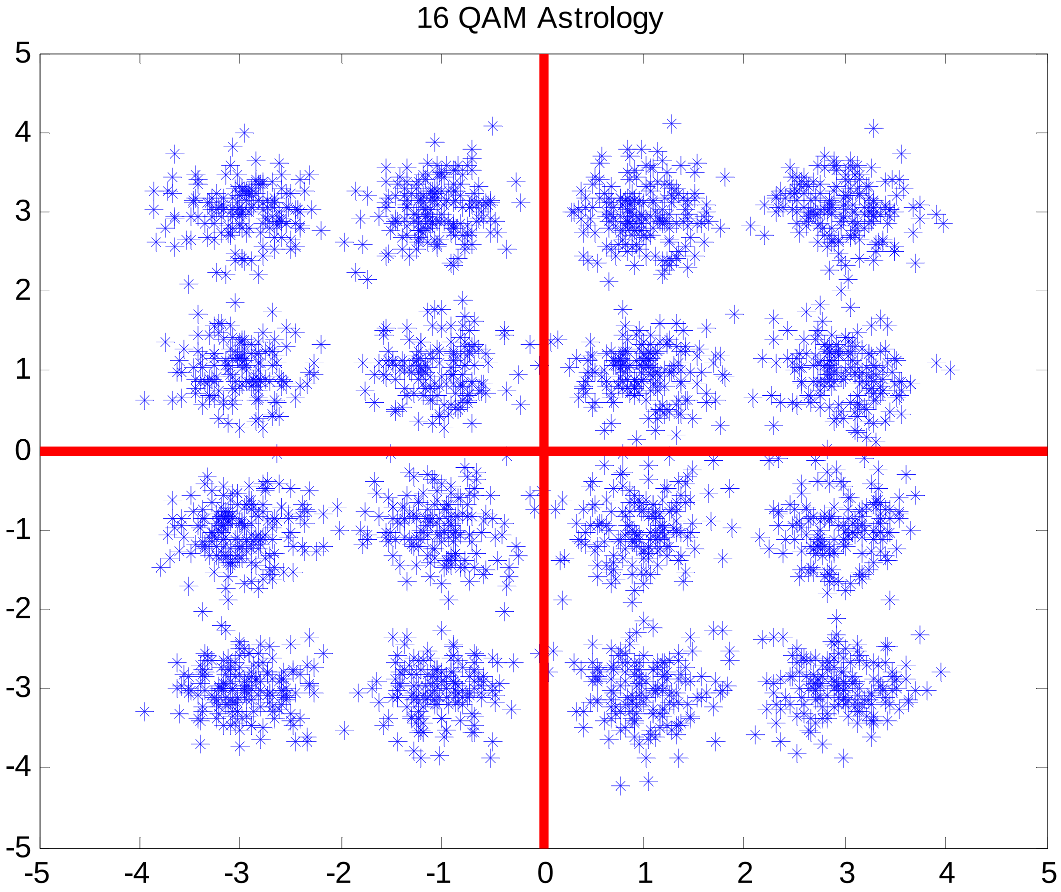
Assume the number of k information bits are mapped to one symbol

$$k = \log_2 M$$

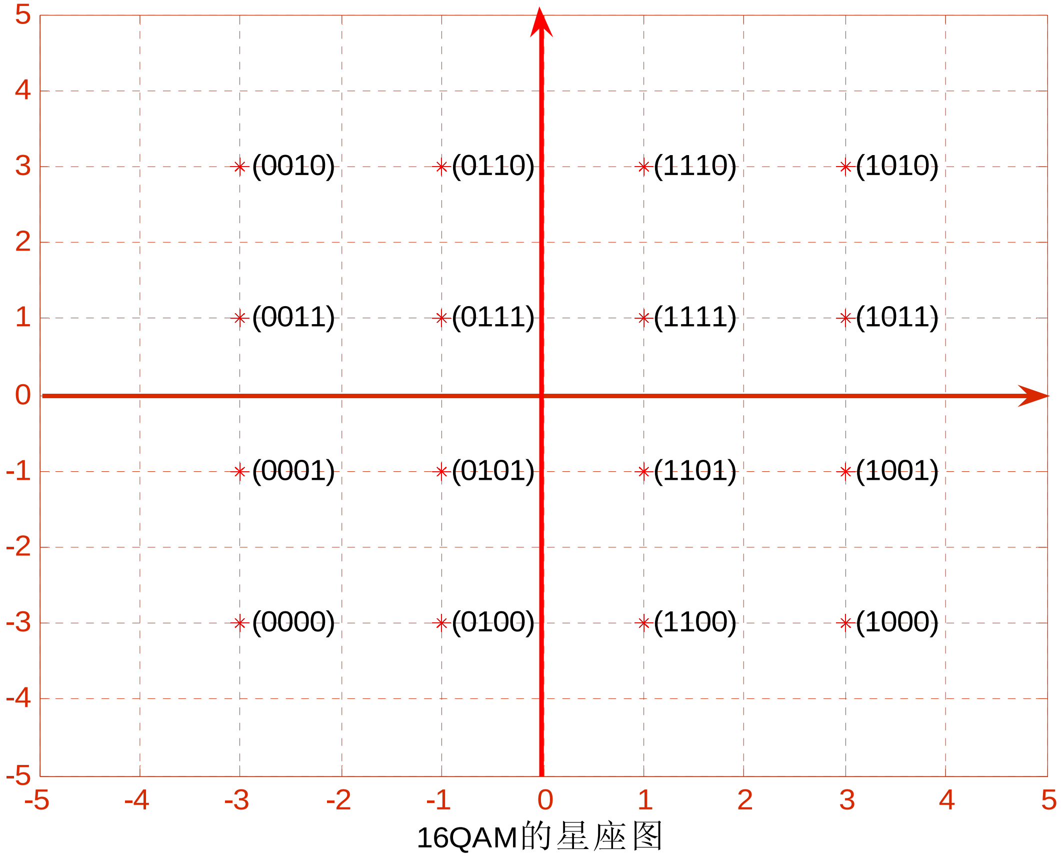
Using gray mapping, the BER and symbol error rate PER becomes

$$BER = \frac{PER}{k}$$

- Decision under Gray Coding



Standard 16 QAM Astrology



Answer:
When the SNR is very low, one symbol error might cause more than one bit error

Problem 3

- ***Speed up your Simulation***
- 1. Try to re-use the defined variables or arrays, use functions
- 2. How to calculate the proper number of simulated data?
- ***Method 1:***
- First, estimate the BER from theoretical curve under each SNR
- Then multiply it by 100 or more to get the proper number of data for each snr

Weak point: as SNR goes high, the simulation data will increase dramatically, and may finally exceed your computer memory limit

Easy way to determine the number of simulation data

First calculate the theoretical BER $Error\_theory$

According to rule method 1, then number is $\frac{1}{Error\_theory} \times 100$

For a very high SNR, give a fixed number, assume it is 5000, then

$$\frac{1}{Error\_theory} \times 100 < 5000$$

Stop condition for simulating BER

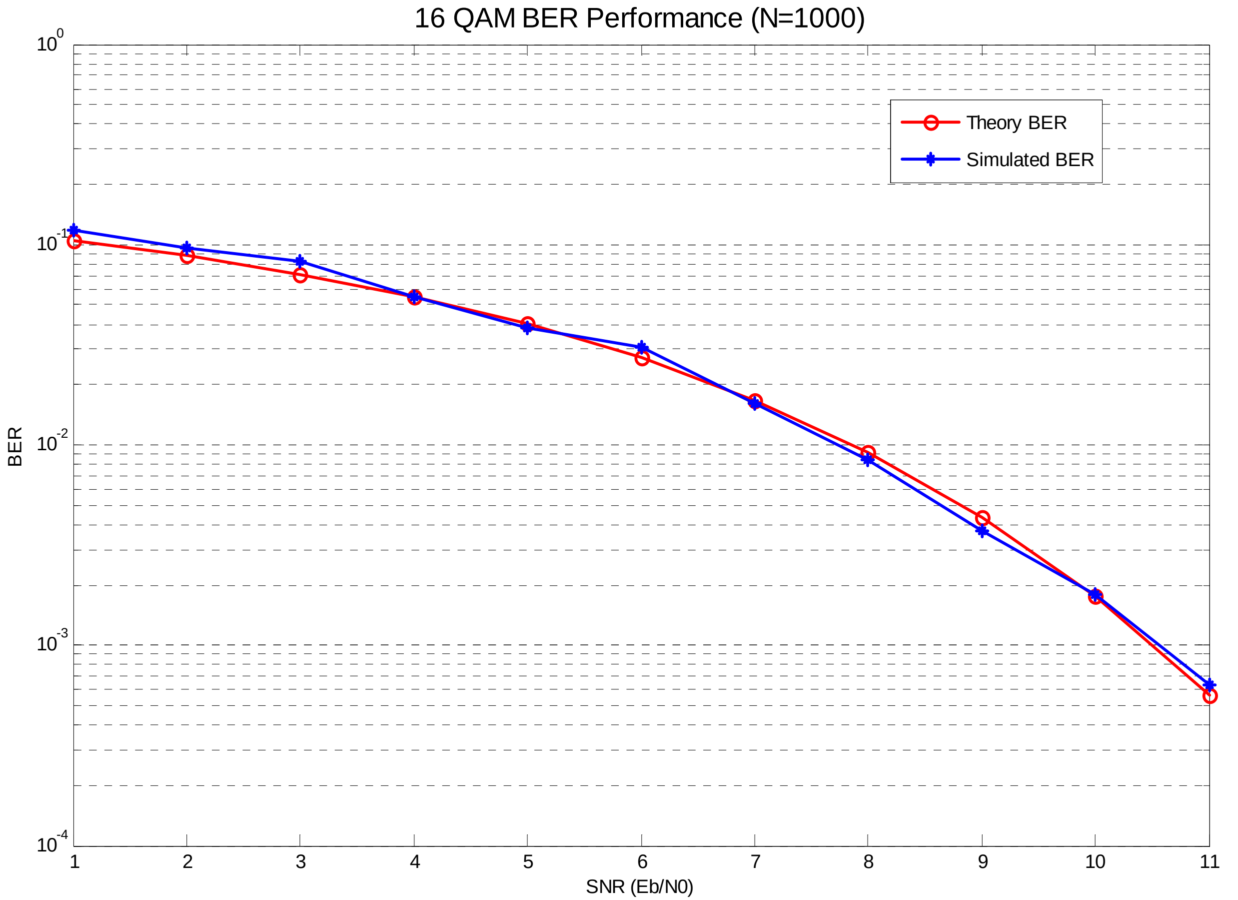
- When the simulation fulfills either of the following conditions, the simulation should be stopped:
  - The number of error bits reach to a certain value (in AWGN channel, 100 is enough)
  - The number of simulated bits reach to N (The N can decided according to the simulation requirement and the speed of computer, or based on the simulation time)

- ***Speed up your Simulation (Cont.)***
- ***Method 2:***
- Divide the total number of data need to be simulated into smaller groups, for example, N=1000
- Repeat the whole program and accumulate the error bits in each round
- When the accumulated error bits exceed 100 or the maximum number of data reached, stop the simulation

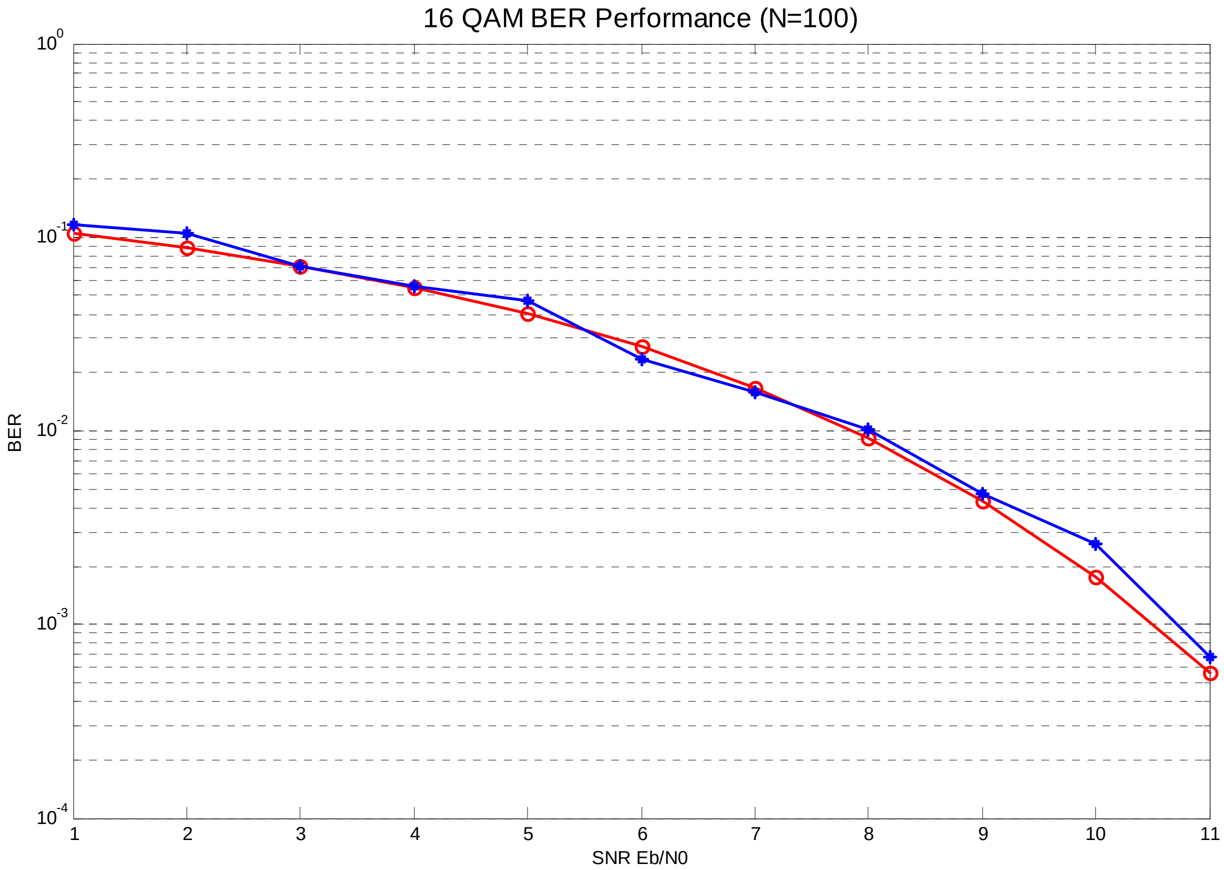
The reasons:

- Save computer memory: Since each variable will occupy some memory place, too many variables will slow down the operation
- The value of N is also cannot be too small, otherwise the it will always change the commends during the running
- Some compromise need to be considered

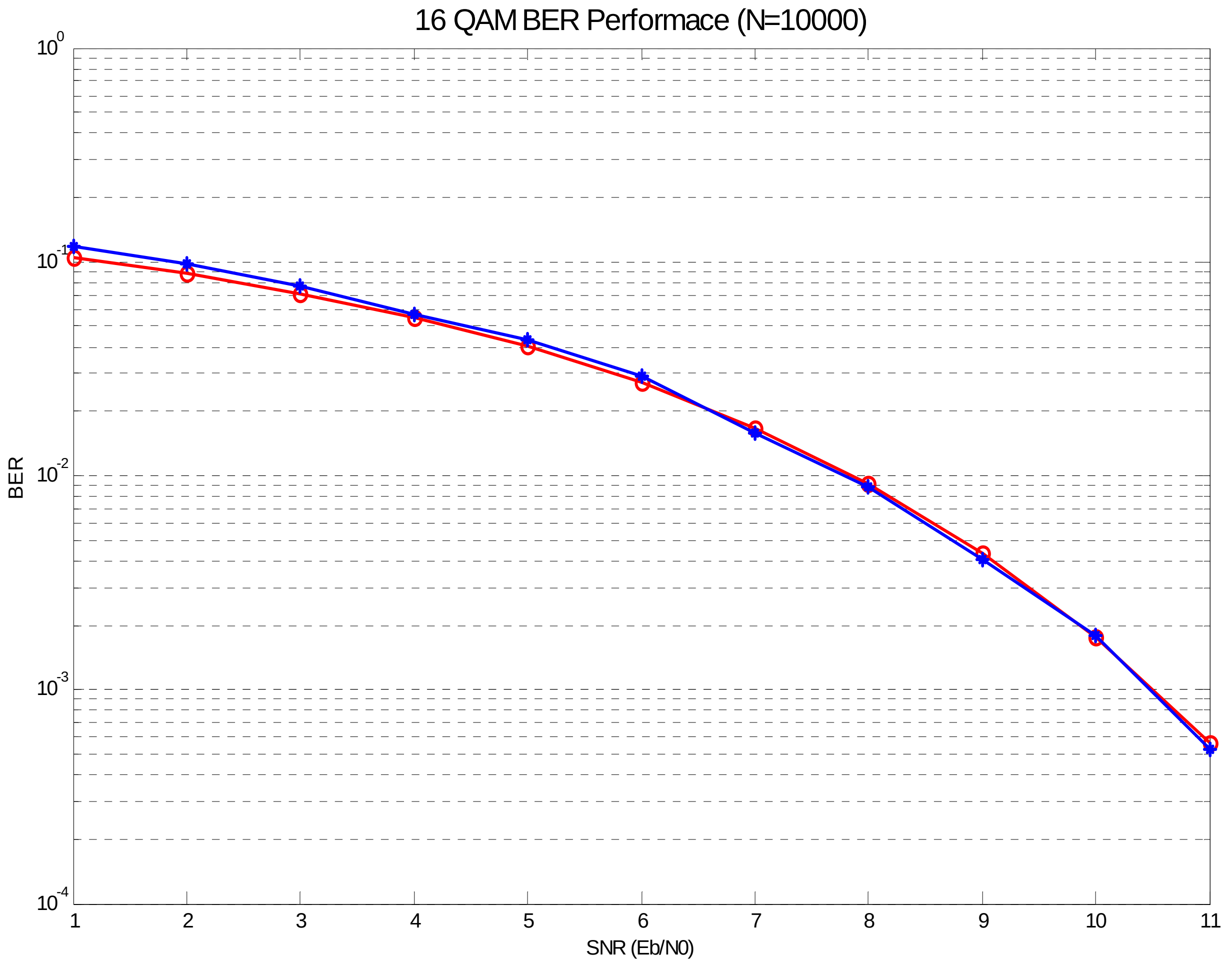
- ***Simulation carried out in method 2***



Let's see the effect of N with different value



If you DIY the noise, the N should not be too small to keep the noise precise.



In consideration of efficiency, N may not be too large.

Anyway, the value of N is just a trade off.

The way to speed up the Matlab simulate (2)

- Avoid to use “loop”, instead using matrix calculation
- Using the function provided Matlab rather than function written by yourself
- Using c language to build a function and inserted to the MATLAB

Example

```
clear;
tic
a (1:20000) = ones;
for i=1:20000
b(i)=a(20001-i);
end
toc
“-- Elapsed time is 0.405912 seconds.--”
-----------------------------------------------------------------------------------------------------------
clear;
tic
a (1:20000) = ones;
b=a(20000:-1:1);
toc
“-- Elapsed time is 0.004444 seconds.--”
```

Other Problems: Simulation results are not accurate

Simulation samples are not enough
Sampling points are not correct
The Gray mapping condition is not fulfilled

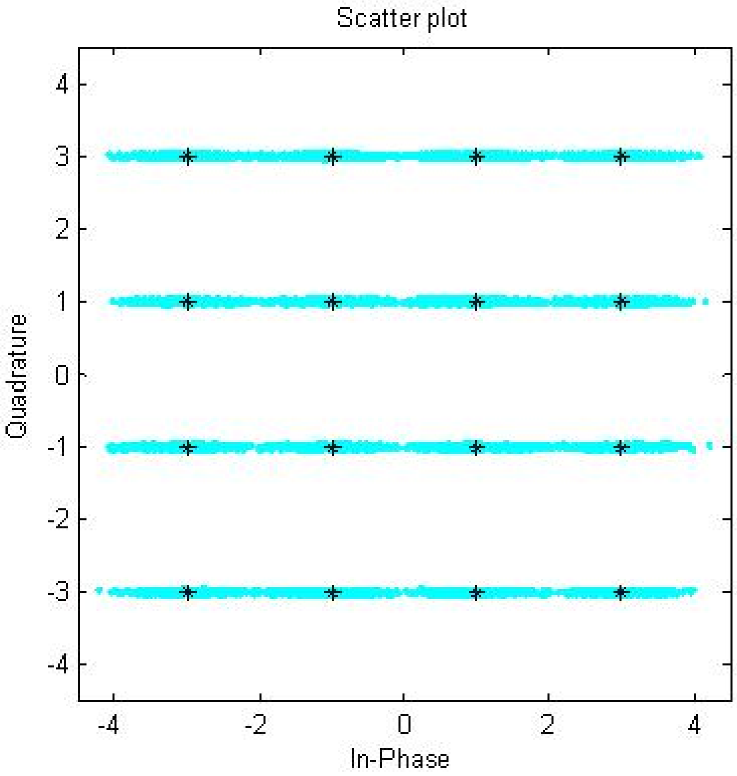
The way to locate your simulation errors:
1. Remove the AWGN module or set SNR be a very large value, check the corresponding figures at both transmitter and receiver side.

The way to locate your simulation errors:

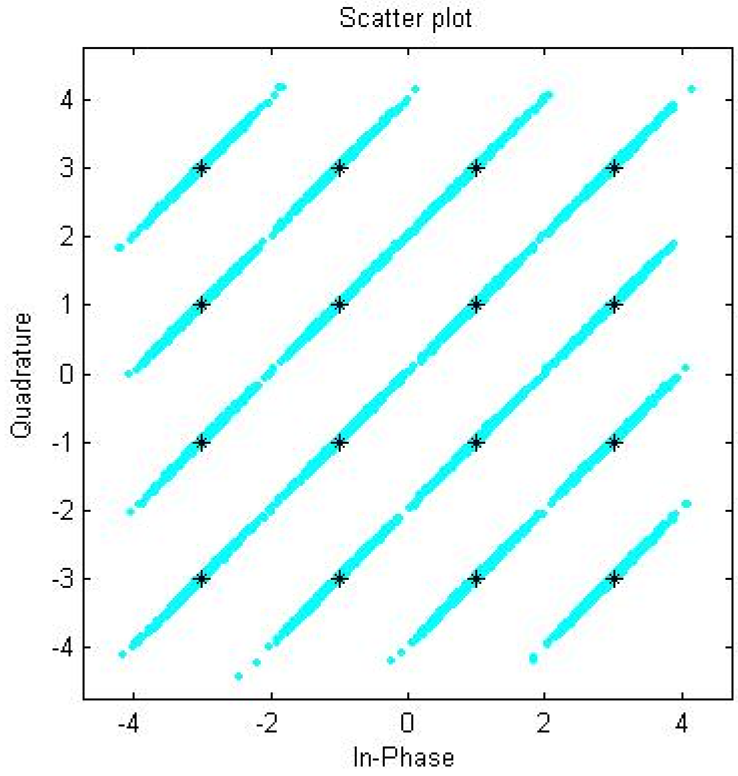
- Step 1: If BER = 0.5: Remove the AWGN module or set SNR be a very large value, check the corresponding figures at both transmitter and receiver side. Sometimes the delay between transmitted and received data cause error
  - Figure 2 and Figure 9 should be the same
  - Figure 3 and Figure 8 should be the same
  - Transmitted and received binary data should be same
  - Transmitted and received 16QAM data should be same

- Step 2: If BER is not same as theoretical value
  - Check if the symbol error rate is correct
    - Symbol error rate is not OK, check the constellation
    - Symbol error rate in OK, check the gray mapping
- Step 3:If the BER is not accurate if SNR is high
  - Check if the number of simulation point is enough

Possible problems in adding the AWGN



Add the AWGN only at the real part



The AWGN at the real and imaginary parts are the same

Thanks !