

A STUDY OF JAMMING ON QPSK MODULATION

Enbo Yang

Bachelor of Engineering
Telecommunication Engineering Major



Department of Electronic Engineering
Macquarie University

June 6, 2016

Supervisor: Associate Dr. Professor Sam Reisenfeld


ACKNOWLEDGMENTS

I would like to gratefully thank Associate Professor Dr. Sam Reisenfeld for their guidance in this entire semester, the guidance of detailed background tutorial, programming and final report is really helpful and effective, that helps my study of entire project work at Macquarie University.

STATEMENT OF CANDIDATE

I, (Enbo Yang), declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the Department of Electronic Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment at any academic institution.

Student's Name: Enbo Yang

Student's Signature: 

Date: 06/06/16

ABSTRACT

As the increased use of digital communication, the communication environment becomes more challenges than before. QPSK Modulation is widely used as a current digital communication method, in particular on the wireless communication network and high-performance communication links.

The purpose of this thesis is to follow with a systematical study on the digital communication system , modulation methods and jamming methods, from the simulated result, estimate the performance for different types of jamming which affect QPSK modulation, determine the performance of each jamming schemes in a particular condition, and how the limited jamming signal power will be used effectively. And this thesis has been estimated the effectiveness of tone jamming(single-tone jamming and multi-tone jamming) that affect the QPSK Modulation from comparing the bit error rate (BER) of the communication system.

Contents

Acknowledgments	iii
Abstract	vii
Table of Contents	ix
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Background	1
1.2 Digital Communication System	2
1.3 Quadrature Phase Shift Keying (QPSK)	4
1.4 Thesis Overview	4
1.5 Project Objectives	7
1.6 Project Baseline Review	7
1.6.1 Time Budget Review	7
1.6.2 Financial Budget Review	7
2 Literature Review	9
2.1 Basic knowledge of signal processing	9
2.1.1 Continuous-time signals and discrete time signals [20]	9
2.1.2 Discrete Time Sampling	10
2.1.3 Power and energy	11
2.1.4 Complex Envelope [6]	12
2.2 Quadrature phase shift keying(QPSK)	13
2.2.1 Power Spectrum Density(PSD) of QPSK	16
2.2.2 Signal Space	16
2.3 Jamming	18
2.3.1 Jamming Techniques	18
2.3.2 Noise jamming	18
2.3.3 Tone jamming	20
2.3.4 Other Jamming Techniques	22

3	Experimental Procedures	25
3.1	Experimental Devices	25
3.2	Matlab	25
3.3	Matlab Code Syntax	26
3.4	QPSK Simulation	28
3.5	QPSK with Single tone jamming	32
3.6	Single tone with frequency offset	32
3.7	QPSK with Two-tone jamming	33
3.8	QPSK with M-tone jamming	33
4	Results and Discussion	37
4.1	QPSK Simulation	37
4.2	QPSK with jamming channel	38
4.2.1	P _j vs P _s	38
4.2.2	Initial phase angle	39
4.3	Jamming with frequency offset	40
4.4	Two-tone jamming	45
4.5	Multi-tone Jamming	46
4.5.1	P _{ji} Distribution Methods	47
4.5.2	Multi-tone Jamming VS Single-tone Jamming	50
4.5.3	M number of tones	51
4.6	Code Implementation And Discussions	52
4.6.1	Code Performance	52
4.7	Conclusion Of Different Jamming Methods	52
5	Conclusions and Future Work	55
5.1	Conclusions	55
5.2	Future Work	55
6	Abbreviations	57
A	Matlab Coding	59
A.1	Overview	59
A.2	QPSK Sumulation	59
A.3	Single-Tone Jamming placed at centre frequency	62
A.4	Single-Tone Jamming with Frequency Offset	68
A.5	Two-Tone Jamming	73
A.6	Jammer P _{ji} with different distribution methods	78
A.7	M-tone jamming VS reference single-tone jamming	84
A.8	M-tone jamming influence with the number of M	89
B	Consulation Meetings Attendance Form	93
	Bibliography	93

List of Figures

1.1	Digital Communication System Block Diagram [17]	2
1.2	Mandatory modulation/coding modes [7]	4
1.3	Comparison of jamming techniques when multiple jammers attack a single 16-QAM modulated victim signal, JNR=10dB. [2]	5
2.1	Sinusoidal signal	10
2.2	Discrete-time sinusoidal signal	10
2.3	Spectrum of RF bandpass signal to be represented in baseband.	13
2.4	Spectrum of complex envelope in the baseband.	13
2.5	Modulation Results of QPSK [15]	15
2.6	QPSK Modulator [22]	16
2.7	PSD of MPSK	17
2.8	The signal space of QPSK	17
2.9	Different jamming methods present in the power spectrum	20
2.10	FHSS follower jamming timing	23
4.1	N=10,N=100,N=1000 for QPSK Simulation	38
4.2	The influence between different jamming power	39
4.3	The influence between different initial phases	40
4.4	initial check with frequency offset=0	40
4.5	Jamming with frequency offset=1,-1	41
4.6	Jamming with frequency offset=0.5	41
4.7	Jamming with frequency offset=0.2	42
4.8	Jamming with frequency offset=0.08	42
4.9	A conclusion of jamming with frequency offsets	43
4.10	Jamming with frequency offset=0.2,-0.2	43
4.11	Jamming with dFTs in a smaller Pj	44
4.12	1-tone vs 2-tone jamming case 1	45
4.13	1-tone vs 2-tone jamming case 2	46
4.14	1-tone vs 2-tone jamming better results case 2	46
4.15	1-tone vs 2-tone jamming case 3	47
4.16	1-tone vs 2-tone jamming case 4	47
4.17	Pji distribution method, Pj=0.1, deltafT=0.8, M=10	48

4.18	Pji distribution method, $P_j=0.1$, $\Delta f_T=0.08$, $M=10$	48
4.19	Pji distribution method, $P_j=0.05$, $\Delta f_T=0.4$, $M=10$	49
4.20	Pji distribution method, $P_j=0.05$, $\Delta f_T=0.2$, $M=10$	49
4.21	Pji distribution method, $P_j=0.01$, $\Delta f_T=0.8$, $M=10$	50
4.22	Pji distribution method, $P_j=0.01$, $\Delta f_T=0.08$, $M=10$	50
4.23	M-tone VS 1-tone, $P_j=0.01$, $\Delta f_T=0.1$, $M=10$	51
4.24	Influence with M tones	51

List of Tables

4.1 Relationship between EbNo and Pb	39
------------------------------------------------	----

Chapter 1

Introduction

Jamming is a common aspect which is used to disturb with communications or surveillance, for example, GPS jamming, radio jamming, radar jamming and deception and mobile phone jammer. In this project, jamming signals will be introduced in the communication channel between the process of modulation and demodulation. Moreover, for different jamming signal shapes, determine the performance of each jamming methods in a particular situation.

1.1 Background

More than ever before, the wars of the 21st century have proven to be reliant on the electromagnetic spectrum. The global electronic warfare (EW) market is estimated to be \$12.15 billion in 2014, and it is expected to reach \$15.59 billion by 2020. [8] The survivability of the platform provides the electronic cover of an EW system. The EW system has the function such as electronic attack, electronic support, and electronic protection. The needs of spectrum control and evaluation of systems into smaller systems have been the main driving forces for the market. Over the years, the system has become a vital necessity since there are no defined borders within the electromagnetic spectrum. Since the battlefield has changed to the urban environment, the challenge will be to ensure efficient management of the spectrum. Forced nations and organizations request and develop a suitable system.

Currently, the key challenge for information operations is EW systems are employed for offensive and defensive purposes. They are required for the interception, monitoring, direction finding (DF), collection, analysis, and recording of foreign non-communications (radar band) electromagnetic radiations. EW is a silent and invisible war, using the spectrum resource and electromagnetic power to intercept communications. For example, various airborne jamming systems can blind enemy air defence radar and interfere with their communications, data, and computer networks. [16]

1.2 Digital Communication System

Analog communication system uses an analog signal transferring information; analog signals have continuous value with time past. Analog signals are simple, and all natural signals are analog. Digital signals are represented by binary numbers and these values are discrete in time. Digital communication systems use a digital sequence between the source and the channel input (and similarly, between the channel output and final destination). Covering into the binary sequence from which data could transmit through over physical media, for example, cable, twisted wire pair, optical fibre, which is commonly used for Broadband Internet or electromagnetic radiation, such as wireless transmission. It is an essential part of the modern communication technology, and it is widely used for organizations with all the business communications. [4]

Some advantages of delivering data with digital sequence can be briefly summarized as below [4]:

- 1) The size of digital electronic hardware design is small than the size of analog electronic devices, current modern electronic hardware design towards miniaturization, meantime the cost decreased every year. It is cheap, and reliable than an analogy interface hardware.
- 2) A unified method, such as coding and decoding method between source and channel, then the coding and decoding block can be designed independently. It provides the design standardization which different designs easier spread and develop.

However, digital signals are easier corrupted by the excess noise, noise (or jamming) could change the value during digital demodulation, this causes the change or original data and makes the final output vary from their original data.

Figure 1.1. shows a block diagram of a typical digital communication system.

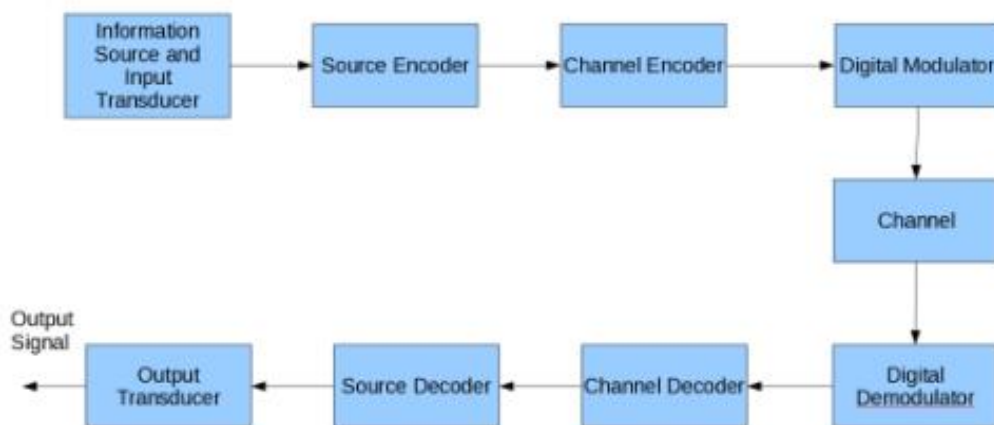


Figure 1.1: Digital Communication System Block Diagram [17]

This block diagram shows core elements of the digital communication system; each block represents one step in the digital communication system progress.

- **Information Source and Input Transducer:** The original input of the entire system, analog or digital signals deliver into the system, for example, audio and video are the typical analog signals and some digital signal such as teletype signal can also be the input. However, the analog signal needs a further covert since digital communication system only processes digital sequence. Then in the following block, a source Encoder is introduced.
- **Source Encoder** Source Encoder converts the input signal into digital form. These output of binary digits is called information sequence. This process requests the digital sequence use as few binary digits as possible to represent the signal. Normally the binary digits have a little or no redundancy during the process, and it will be added in the next channel encoder. The source coding algorithm determines the system performance with a higher code rate.
- **Channel Encoder:** Channel encoder adds some extra bits in the binary information sequence that can be used to overcome the effects of noise and interference through interface channel.
- **Digital Modulator:** Digital Modulator converts the binary sequence into electric signals then transmit to the channel. The digital modulator converts the binary sequences into signal waveforms. For example, QPSK is a modulation way which used in this project; one signal wave represents two bits. The modulation scheme plays a major role in deciding the data rate and immunity of signal towards the errors introduced by the channel.
- **Channel:** Channel is the transmitting medium between transmitter and receiver. It can be the atmosphere for a wireless system or optical channel for a wired system. Additive white Gaussian Noise(AWGN) is further discriminated by their property and characteristics. In the project, both AWGN and Jamming channel is introduced, and the influence of jamming channel will be estimated through simulation. Channel introduced many types of errors due to thermal noise.
- **Digital Demodulator:** The digital demodulator receives the modulated signal through the channel since the channel may corrupt the transmitted waveform, digital demodulator recovers the waveform to its represented binary information or data symbols. The demodulator should provide high BER that makes further process.
- **Channel Decoder:** Channel decoder received the demodulated binary sequence and tried to recover the original binary sequence by using the prefixed redundancy bits, to overcome the influence with the signal passing through channels. Channel decoder should also provide high BER that matches the demodulator.


- Source Decoder If the analog signal is required, the decoder could recover to their original signal shape from the origin by using the data sequence received by the channel decoder.
- Output Transducer: The output of signals in the desired format either analog or digital.

Then in the literature review part, some different schemes will be explained, and in this project, the modulation method,, and channels will be introduced in more detail.

1.3 Quadrature Phase Shift Keying (QPSK)

QPSK is a common Modulation method that widely used in practical, and QPSK is also utilized in this entire project. QPSK Modulation is the advanced version of BPSK that by using two set of BPSK Signals to represent two bits in one symbol. Therefore, the bandwidth efficiency is twice than the BPSK Modulation and still have the advantage of a good bits error performance; it is relatively robust than another large number of M-ary Modulation and meantime provided sufficient performance. A more detailed background knowledge will be contained in the next chapter. And QPSK is widely used in different kind of applications in real life, for example, the most common communication techniques GSM ,CDMA, LTE, 802.11 WLAN, 802.16 fixed and mobile WiMax, Satellite communication and CABLE TV applications. [25]

For example, some mandatory modulation modes are used in IEEE802.16-2004 WirelessMAN-OFDM PHY; these modes show in figure 1.2.



Mode. No.	Modulation	Block size (bytes)		RS code	CC code rate
		Uncoded	Coded		
1	BPSK	12	24	(12,12,0)	1/2
2	QPSK	24	48	(32,24,4)	2/3
3	QPSK	36	48	(40,36,2)	5/6
4	16-QAM	48	96	(64,48,8)	2/3
5	16-QAM	72	96	(80,72,4)	5/6
6	64-QAM	96	144	(108,96,6)	3/4
7	64-QAM	108	144	(120,108,6)	5/6

Figure 1.2: Mandatory modulation/coding modes [7]

1.4 Thesis Overview

In this project, QPSK Modulation with Gaussian Noise Channel simulation will be used as the initial simulation program; then this project will add a new jamming channel.

The motivation of the project is that current research paper is more on the theoretical analysis of jamming influence [19] [3], or some research papers focus on the specific type of modulation and jamming method [24] [26] [5] [1] (An example has been shown in figure 1.3). Therefore, The different jamming methods are estimated by simulation, parameters of jamming signal and the effectiveness of various jamming methods which can affect the final demodulation performance will be discussed.

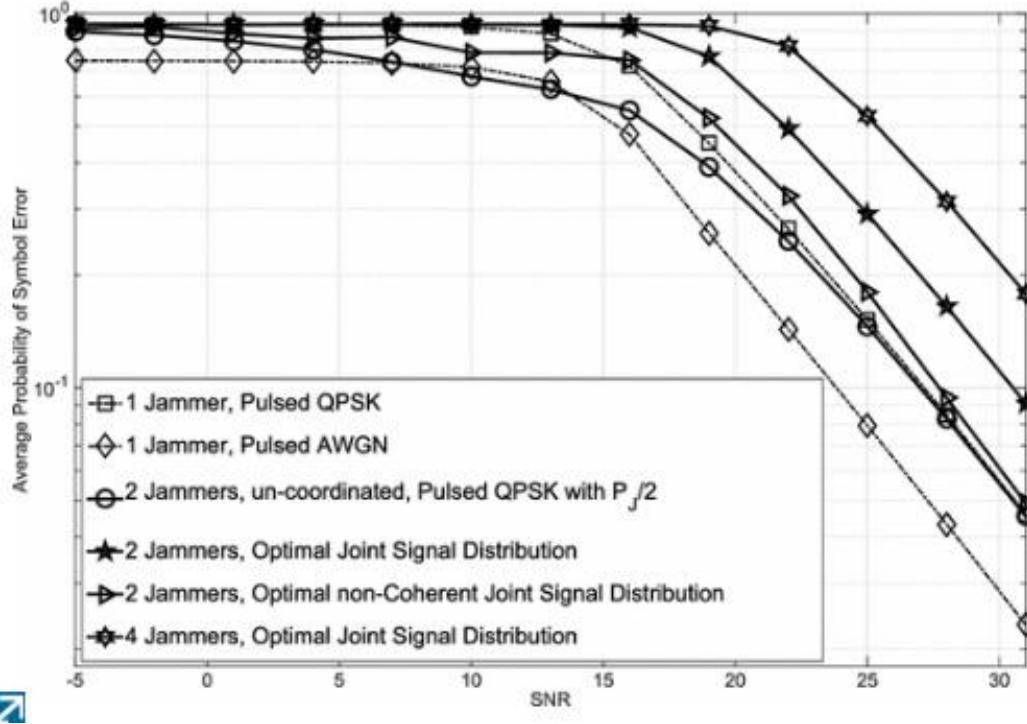


Figure 1.3: Comparison of jamming techniques when multiple jammers attack a single 16-QAM modulated victim signal, JNR=10dB. [2]

Chapter 1, has been explained the background knowledge, the digital communication system block diagram and its function introduction, the QPSK Modulation and the thesis progress and objectives.

Chapter 2 is the literature review part which will explain more in depth especially with equation notations and transformations that used in the following programming and simulation. Also some background knowledge, for example, the modulation and jamming methods will be introduced.

Chapter 3 will explain the device and current version of software which the project is used, and also will explain how each the codes work. A simulation with QPSK modulation with Gaussian Noise Channel of digital communication systems will be firstly introduced, then a single tone jamming signal will be added to the QPSK communication channel. After that, the single tone with an offset frequency covered will be used to the different situation. In the end, this project will simulate the jamming signal with the various

numbers of tones, different jamming power, the maximum edge offset frequency, and different power distribution allocation methods.

Chapter 4 will collect the results and discuss the effectiveness of each jamming signal methods.

Chapter 5 is the conclusion and future works that could do further research and development.

1.5 Project Objectives

The desired project objectives is listed below:

1. Some sufficient background study on digital communication system, modulation , demodulation methods, and channel (including both AWGN channel and jamming channel)
2. Be familiar with the basic QPSK Simulation program, have the ability to implement and modify the code for a certain purpose(For example implement different modulation methods estimate the performance of the communication system).
3. The study on different jamming types, choose the desired jamming method for QPSK Modulation.
4. Implement the jamming signal and jamming channel into the simulation code.
5. Implement the more advanced jamming channel(jamming with frequency offset, multi-tone jamming) into the simulation code step by step.
6. Estimate the effectiveness of jamming methods by simulation result(the simulation Pb result).
7. Determine the parameters that affect the final jamming performance.
8. Some further research on different jamming types against modulation method.

1.6 Project Baseline Review

1.6.1 Time Budget Review

The project covers entire 2016 semester one , from week 1 to week 12. Baseline plan was generated including twelve weeks plus two weeks during mid-semester break. And all estimated works are finished,and the percentage work completion is 100%.

1.6.2 Financial Budget Review

The project was initially allocated to a budget of \$300. The project did not require any purchased because in this project, the only simulation with Matlab software that provided from university, no hardware implementation required. Therefore, the financial budget is zero through the entire project.

Chapter 2

Literature Review

The following section shows all theories and equations that used during programming, firstly this section will explain the background knowledge of signal processing and communication system. Then some modulation methods such as MPSK and QPSK signals, some QPSK related background knowledge will be described in the following section; Next the sampling process for QPSK signals will also be introduced, at the end of the chapter, different jamming schemes and the equations with all the required complex envelope of jamming signal will be listed, which is the crucial part and the equations of complex envelope will be implementing into the code as the jamming signal interface.

2.1 Basic knowledge of signal processing

2.1.1 Continuous-time signals and discrete time signals [20]

Before we talk about modulation and demodulation, firstly we can see an example for two typical signals.

A continuous-time signal is a signal of the real variable t . For example, the sinusoidal signal:

$$x(t) = A \cos 2\pi f_0 t$$

Figure 2.1 is a function of the time variable t , with amplitude A and frequency f_0 .

A discrete-time signal is a sequence where the values of the index n are integers. For example, the discrete-time sinusoidal signal

$$x(n) = A \cos 2\pi f_0 n$$

with $f_0 = 1/12$ shows in figure 2.2

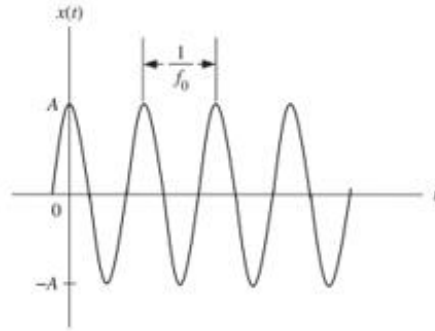


Figure 2.1: Sinusoidal signal

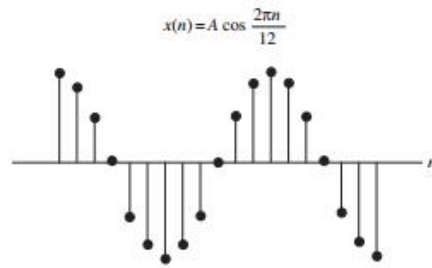


Figure 2.2: Discrete-time sinusoidal signal

From two figures, we can easily verify that there is some relationship between two signals since the discrete-time sinusoidal signal has the same shape with the continuous sinusoidal signal. The process that is transforming continuous-time signals to its discrete time signal form is called 'Sampling'.

Often a discrete-time signal is obtained from a continuous-time signal $x(t)$ by sampling at the time instants separated by a sampling interval T . Thus, $x(n) = x(nT)$. For example, if $x(t) = \cos 2\pi f_0 t$, then $x(n) = x(nT) = \cos 2\pi f_0 nT$. Specifically, $f_0 T = 1/12$ for figure 2.2.

2.1.2 Discrete Time Sampling

Sampling converts a continuous-time analog signal into its digital signals form, with a similar shape. The value of digital signals after sampling is the same with their original analog signal at a certain of time. In addition, the sampling requires some additional conditions because many signals may have more set of samples, the sampling could be failed if the sampled signal represent multiple meanings.

According to WOKSS's sampling theorem [21], reconstructing a band-limited $x(t)$ require the sampling frequency $f_s \leq 2f_{max}$ for $x(t)$, f_{max} is the maximum frequency of $x(t)$ and f_s represent the sampling rate. $f_s = 2f_{max}$ is called the Nyquist sampling rate.

[Theorem 1](WOKSS's sampling theorem) If $x(t) \in L_2(-\infty, \infty)$ has the maximum frequency f_{max} , and the sampling frequency f_s satisfies

$$f_s \geq 2f_{max}$$

Digital Modulated Signals are usually in the form of a time sequence of continuous time pulse; the sampling theorem may be used to obtain a discrete time representation of continuous-time signals and continuous time additive noise.

A continuous time signal, $x_c(t)$, $t_0 \leq t < t_1$, may be represented by,

$$x[k] = x_c(k\Delta T + \frac{\Delta T}{2}), k = 0, 1, 2, \dots, N - 1$$

where, $\Delta T = \frac{t_1 - t_0}{N - 1}$, and N is the number of time samples in time interval $t_1 - t_0$.

$f_s = \frac{1}{\Delta T}$, where f_s is the sampling frequency in samples/s. For $x_c(t)$ being completely band-limited to B Hz, and $f_s > 2B$, the Sampling Theorem grants that $x_c(t)$ can be reconstructed from $x[k]$.

2.1.3 Power and energy

Since the total jamming power P_j and total jamming power to signal power P_j/P_s plays important roles in the entire project. Firstly the definition of power and energy is introduced. The average power in watts delivered by a voltage signal $v(t)$ in volts to a resistive load R in ohms is

$$P = \frac{V_{rms}^2}{R}$$

where V_{rms} is the root mean square(rms) value of $v(t)$, defined as

$$V_{rms} = [\lim_{T \rightarrow \infty} \frac{1}{2T} \int_T^T v^2(t) dt]^{1/2}$$

In a communication system, the value of the resistive load is typically assumed to be

$$R = 1\Omega$$

Thus the average (normalized) power in watts is given by

$$P = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_T^T v^2(t) dt$$

Elements in equations like $\sqrt{2P_s}$ and $\sqrt{2P_j}$ is calculated from the definition of the average power equation when the amplitude of a sinusoid signal is A.

For example, in this project, the ratio of Jamming power to signal power P_j/P_s for the complex envelope(This will be explained in the following section) of single tone jammer can be represented as [20]

$$X\tilde{(t)} = X(\tilde{n}T_s) = \sqrt{2P_s}[1 + \sqrt{\frac{P_j}{P_s}}e^{j2\pi\Delta f nT_s}e^{j\Theta-\phi}] \quad (2.1)$$

2.1.4 Complex Envelope [6]

The representation of complex envelope provides a mathematical definition which the envelope and angles of signals are presented. Because in general the amplitude or angle or both is used for carrying information and data can be recovered after demodulation. It means the complex envelope $\tilde{x}(t)$ still carries the information we need. The complex envelope notation is important in telecommunications. And also the complex envelope requires the minimum sampling rate and the minimum sampling rate of complex envelope are represented with a complex-valued function.

The complex envelope $x(t)$, the RF bandpass signal is decomposed into a slowly-varying complex envelope

$$\tilde{x}(t) = x_I(t) + jx_Q(t)$$

and an exponential signal Formula as defined by

$$x(t) = R[\tilde{x}(t) \exp(j\omega_c t)]$$

The real and imaginary components of the complex envelope are denoted by the real-valued signals $x_I(t)$ and $x_Q(t)$.

The complex envelope of a signal can be written in canonical form as:

$$\tilde{x}(t) = x_I(t) \cos(\omega_c t) - x_Q(t) \sin(\omega_c t) \quad (3)$$

Also, it can be rewritten in both Cartesian or polar form

$$\tilde{x}(t) = x_I(t) + jx_Q(t) = a(t) \exp[j\theta(t)] \quad (4)$$

Figure 2.3 shows the spectrum of RF bandpass signal to be represented in baseband. and figure 2.4 shows the spectrum of the complex envelope in the baseband, from the RF bandpass signal $x(t)$. Compared with these two graphs, The complex envelope resets the center frequency f_c and the minimum sampling rate is halved from $2B$ to B . The complex envelope of the signal is the equivalent low-pass signal, and the required sampling rate for the complex envelope of the signal is related to the bandwidth of the signal and not the centre frequency.

The Complex Envelope of Gaussian Noise

Using the complex representation provided above, a noise process formula can be written as [23]

$$n(t) = \text{Re}[\eta(t)e^{j2\pi f_0 t}]$$

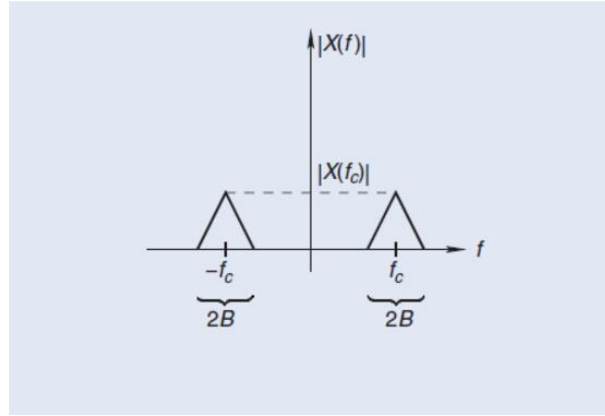


Figure 2.3: Spectrum of RF bandpass signal to be represented in baseband.

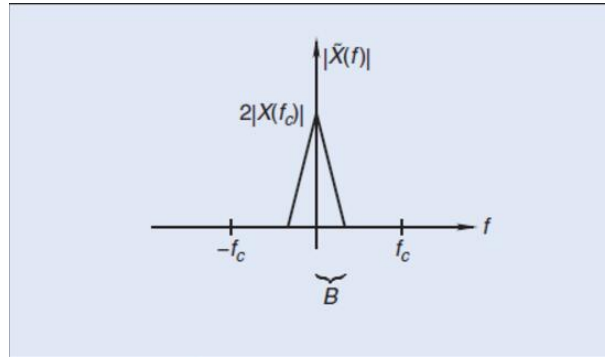


Figure 2.4: Spectrum of complex envelope in the baseband.

2.2 Quadrature phase shift keying(QPSK)

The digital communication system block diagram has shown the purpose of Modulation. There are varying kinds of modulation that currently used, the difference between modulation methods is the using either amplitude, frequency or phase shift to represent the required transmission information. For example, in this project, the phase shift keying (PSK) is using the phase shift different to distinguish the symbol that represents either 1 or 0; Amplitude shift keying (ASK), normally we called on-off keying(OOK), the square signal with a voltage level could represent 1 and the voltage is 0 could represent 0; Frequency shift keying(FSK) similar like the simplest form frequency modulation(FM), the frequency difference can represent binary data.

In addition, the modulation methods have been talked above are some basic modulation, M-ary modulation is needed and used for improving the modulation performance with one symbol represents more binary bits. For example, the QPSK is used in the project; it is a 4-ary PSK Modulation that uses two sets of BPSK signals that one symbol can represent 2 bits to form a QPSK Modulation; Similarly, M-ary amplitude shift keying(MASK) divides the voltage level to smaller pieces that can represent more bits than ASK. Also, there are other M-ary modulation methods such as differential M-ary phase

shift keying(DMPSK), M-ary frequency shift keying (MFSK). However, a significant number of M-ary will increase the difficulty of demodulation in the following procedure, in particular with a larger value of noise channel and jamming channel. Choose the proper modulation method while designing a digital communication system is the main channeling towards the developer. Therefore, the simulation for example Matlab simulation with AWGN could be used to get the correct modulation method, estimate the bandwidth requirements and throughput capacity of the communication system, meantime some channel problems may be identified during simulation. For example, the channel linearity could be identified to make more convince for practical hardware design. The channels may include filtering of the signals; Channel fading, some interference that hard to avoid, and many impairments which could cause the degradation of bit error rate towards the ending digital communication system. Overall, the simulation is one way to estimate the digital communication performance during developing, and this project is using QPSK as the modulation method to estimate the jamming effectiveness of the tone-jamming scheme.

This project is only used QPSK Modulation, a more detailed information is provided in the following section, some are the basic background knowledge, and some are assistant tools that could help the understanding of QPSK, especially the understanding of what actually happens while implementing the code. Before we explain QPSK, Phase shift keying(PSK) or M-ary phase shift keying(MPSK) will be introduced first. PSK the most frequently used binary modulation, such as satellite communications, cellar communications, and wireless local area network. The set of two digital signals [20]

$$s_1(t) = A\cos 2\pi f_c t, iT_b \leq t < (i+1)T_b \quad (2.2)$$

$$s_2(t) = -A\cos 2\pi f_c t = A\cos(2\pi f_c t + \pi), iT_b \leq t < (i+1)T_b \quad (2.3)$$

can represent bit 1 and bit 0 respectively. Therefore the phase shift π radians represent bits 1 and 0. This kind of modulation technique by using phase shift is called binary phase shift keying or simply PSK.

QPSK is widely used in current practical applications, and it has the advantage with high spectral efficiency and moderate high power efficiency than M-ary amplitude shift keying(MASK) although MASK has high spectral efficiency. QPSK provides the constant amplitude requirement that provides the nonlinearity and decreased the design difficulty for electronic circuits such as power amplifier, satellite transponder. QPSK could even halve the spectral usage, increasing the spectral efficiency than PSK.

A generalization of PSK is M-ary phase shift keying (MPSK) where M data symbols are represented by M signals with distinct phase shifts. The MPSK signals can be represented as follows: [20]

$$s_n(t) = A\cos 2\pi f_c t + (2n-1)\frac{\pi}{M}, n = 1, 2, \dots, M \quad (2.4)$$

where the frequency $f_c > 1/T_s$

From the definition of PSK and MPSK, the QPSK can be expanded from the case $M=4$; $M=4$ means four signals are used for representing for distinct symbols 00,01,10,11. In this case, one symbol represent 2 bits, $k=2$ and the signals could be written as:

$$s_1(t) = A \cos(2\pi f_c t + \frac{\pi}{4}) \quad (2.5)$$

$$s_2(t) = A \cos(2\pi f_c t + \frac{3\pi}{4}) \quad (2.6)$$

$$s_3(t) = A \cos(2\pi f_c t + \frac{5\pi}{4}) \quad (2.7)$$

$$s_4(t) = A \cos(2\pi f_c t + \frac{7\pi}{4}) \quad (2.8)$$

The representation of these four signals of QPSK has been implemented into me QPSK Simulation Code; four phase angles represent 00,01,10,11 respectively from top to bottom.

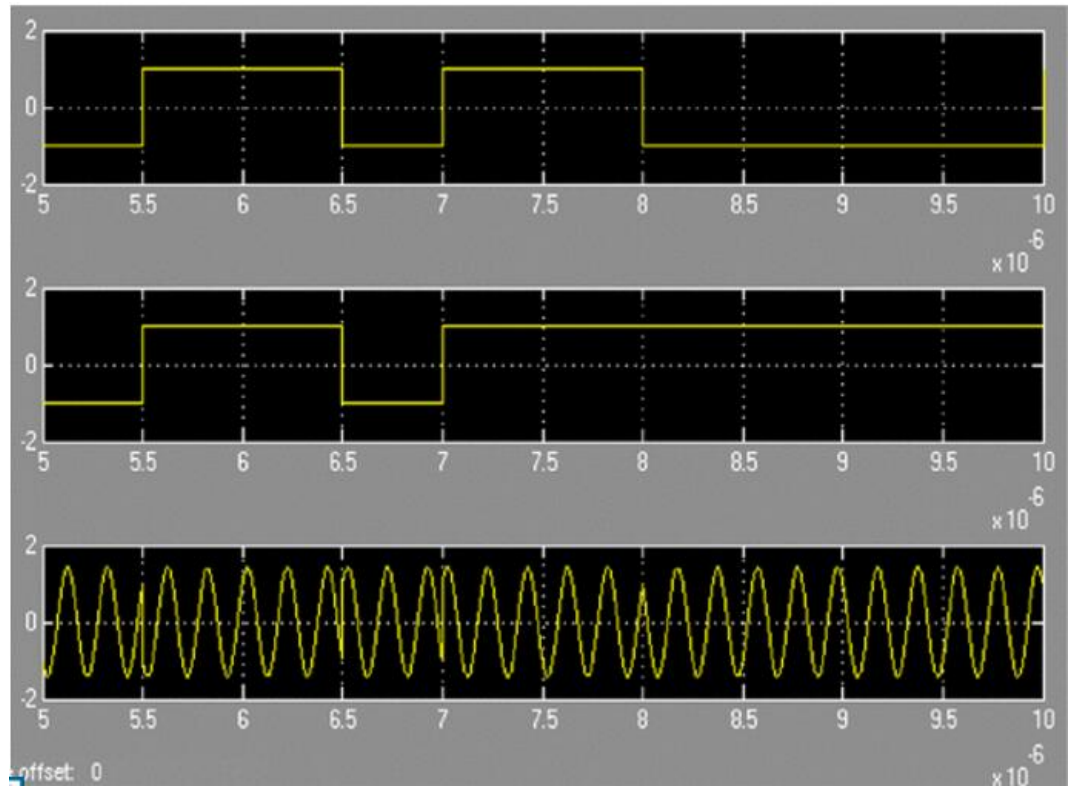


Figure 2.5: Modulation Results of QPSK [15]

Figure 2.5 shows an example of modulation results that simulated from Matlab Simulink tool boxes with a QPSK Matlab Modulator Model, from the result, we can see two sets of binary sequences signal are located on the top of the QPSK Modulated signal, after modulation; QPSK Signal consists of a sinusoidal signal with only phase shifting difference. And figure 2.6 shows a typical Modulator of QPSK process, from this graph we could see two channels are used and QPSK modulated signal is obtained from two channels.

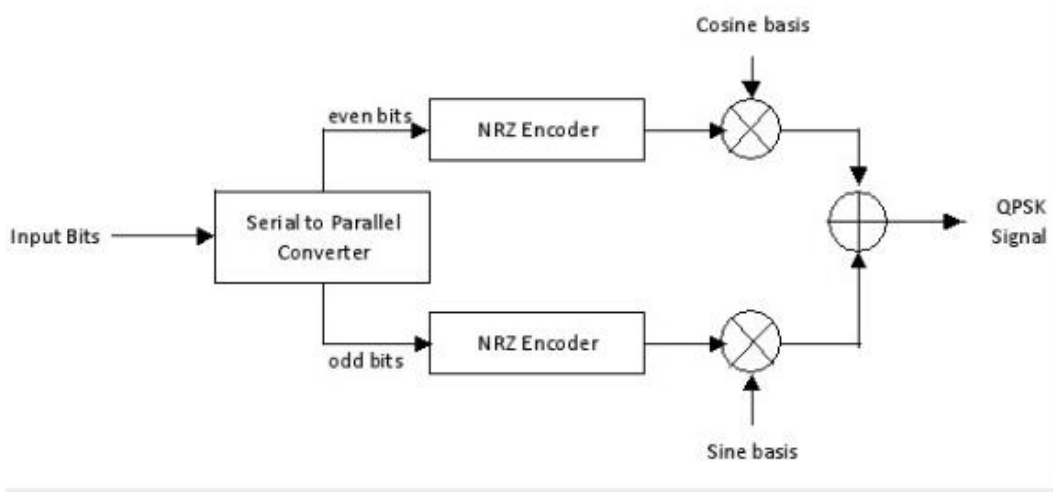


Figure 2.6: QPSK Modulator [22]

2.2.1 Power Spectrum Density(PSD) of QPSK

The power spectrum density (PSD) is another way which helps the understanding of QPSK Signal and Jamming signal; we can easily identify the relationship between power and frequency spectrum, ignoring the influence with the time interval domain. In this project, the PSD graph of the complex envelope of QPSK is frequently used together with the jamming power spectrum.

The PSD $S(f)$ of the MPSK signal $s(t)$ can be represented as

$$S(f) = \frac{A^2}{4}[S_L(f - f_c) + S_L(f + f_c)] \quad (2.9)$$

where $S_L(f)$ is the PSD of the complex envelope $s_L(t)$, and the PSD $S_L(f)$ of the complex envelope is

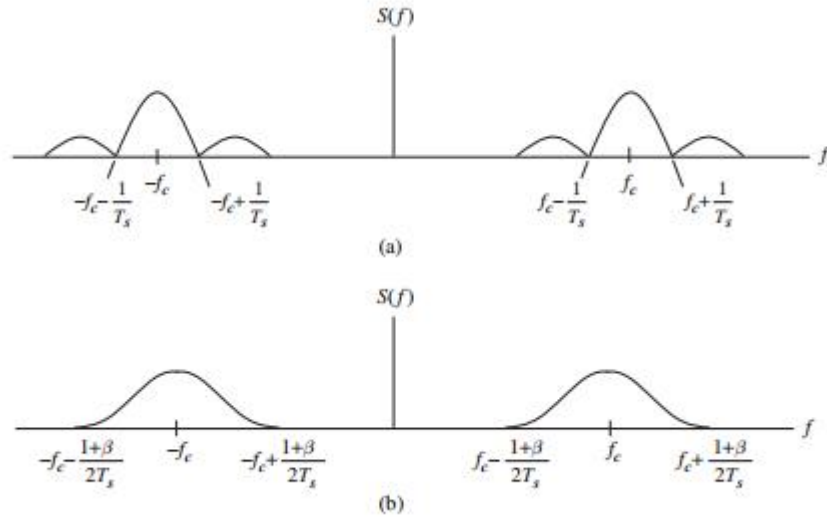
$$S_L(f) = S_{L,I}(f) + S_{L,Q}(f) = \frac{|P(f)|^2}{T_s} \quad (2.10)$$

The complex envelope $s_L(t)$ is the sum of the power spectral density $S_{L,I}$ of the in-phase (real) component and the power spectral density $S_{L,Q}$ of the quadrature (imaginary) component of $s_L(t)$. [20]

Figure 2.7 shows the PSD of MPSK with unit amplitude squared pulse and with raised-cosine pulse.

2.2.2 Signal Space

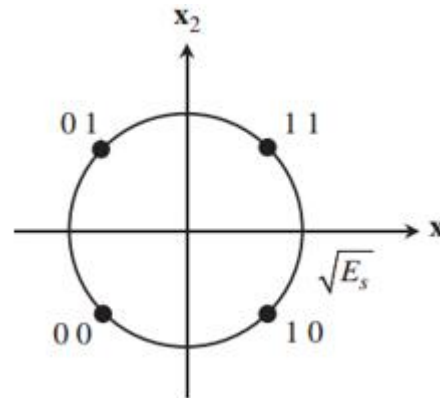
A set of finite energy digital signals can be represented as a corresponding set of vectors, which we can ignore the time dependency. These vectors consult a signal space. Signal space can be used to help the understanding and estimate the phase state during sampling.

**Figure 2.7:** PSD of MPSK

The MPSK signal space is a two-dimensional space, which the euclidean distance between two adjacent MPSK signals can be represented as

$$\|s_k - s_m\| = \sqrt{2E_s} \left[1 - \cos(k - m) \frac{2\pi}{M} \right] \quad (2.11)$$

The QPSK signal space can intuitively represent the modulation and sampling process. The single space of QPSK with symbols representation is shown in the following figure 2.8. [20]

**Figure 2.8:** The signal space of QPSK

2.3 Jamming

Jamming is a technique that widely used for interfering, disturbing or blocking to the wireless communication or surveillance by a jamming channel. And it is joint with the topic of Electronic warfare (EW).

EW refers to the use of electromagnetic resources such as electromagnetic spectrum and electromagnetic energy to implement an electronic attack or defense purpose for especially some military operations.

Electronic Warfare can be divided into three characteristics. Electronic Attack (EA) and Electronic Protection (EP) and Electronics Support (ES).

EA involves the aggressive action in EW, which uses some EM energy, directed energy (lasers, radio frequency weapons, particle beams)), or anti-radiation weapons, attack the enemies' target. EM energy refers to the jamming progress and can be performed on communications systems or radar systems which will reduce the performance of systems and destroy their network capability. EP includes the action that enhances the electronic defense, and ES is the operation with located the required target by measuring or detecting the signal characters, it is the subsection belongs to the early stage of EA operation. [16]

2.3.1 Jamming Techniques

This section will describe some common jamming methods that currently used. And different jamming techniques have their own advantages but some limitations will also be found. Therefore, there is not a certain best jamming technique, how properly use with different jamming techniques is important to affect a different kind of source signals. Two fundamental waveforms are used for jamming; one way is the signal modulated with signal tones, or just increase the background noise in the spectrum in which the communication occur. And the different techniques are classified with the distribution of total jamming power and the frequency spectrum range which the jamming signal placed. [18]

2.3.2 Noise jamming

Noise jamming includes all the jamming techniques related with the use of random noise waveform to modulate the jamming carrier signal. The purpose of the noise jamming is trying to increase the noise between communication channels for example from sender to receiver; the increased noise will reduce the probability that the receiver could get the correct data from the origin. The Gaussian noise channel is the general type of noise sources designed into communication and jamming systems.

Broadband noise jamming, also called full band jamming or barrage jamming, this method covers the entire range of frequency spectrum for the communication system; Theatrically, a strong enough jamming power with spread to over entire frequency spectrum could block any kinds of communication, in practice, the jamming power is limited, how efficiently allocated the power resources will be the changeling topic. Broadband noise jamming is generally used for EP purpose that affects the adversary ES action.

The background noise level is increased significantly during jamming, and it could at least decrease the communication range, in some contain conditions, it could block the communication.

Broadband noise jamming has a direct influence on the adversary's communication system, the required capacity of the communication system is raised by the nose jamming. For example, the capacity of a channel corrupted by Gaussian noise is

$$C = W_{ss} \log_2 \left(1 + \frac{R}{P_T} \right)$$

W_{ss} was the signal bandwidth, R is the average signal power, and P_t is the total average nose present given by $P_T = W_{ss}(N_0 + J_0)$. From this equation, the capability of a channel is easily decreased with the rise of P_t ; one way could solve this problem is to increase the signal average power R to reduce the impact with noise jamming.

The bit error probability P_b for a coherently demodulated BFSK system in the presence of AWGN is

$$P_b = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

The full-band noise jamming has the noise power spectral density J_0 . Then the bit error probability for a coherent BFSK system with the interference of full-band noise jamming can be defined as

$$P_b = Q\left(\sqrt{\frac{E_b}{N_0 + J_0}}\right)$$

Similarly, the bit error probability P_b for a coherent BPSK system is

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

The bit error probability after add the full-band noise jammer can be defined as:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0 + J_0}}\right)$$

It can also represent the bit error probability of QPSK system.

Partial-band jamming focus on the part of the frequency spectrum, the total jamming power is concentrated to a smaller section of frequency range than the full range, then this method further increases the jamming effectiveness. Partial-band jamming is the counterpart of pulsed jamming of a DS signal. [18]

2.3.3 Tone jamming

The jamming technique with one or more jamming tones placed in the spectrum, different strategies are varying about where and how much power should be distributed in the spectrum. Single tone jamming means only one tone placed into the spectrum. The phase of the jamming tone is one of the main parameters that affect the final performance. The total jamming power splits into the small part, the multi number of tones could spread the total jamming power to its spectrum; that increase the effectiveness of jamming especially to FH signal. The jamming signal over a single frequency was shown in figure 2.9.

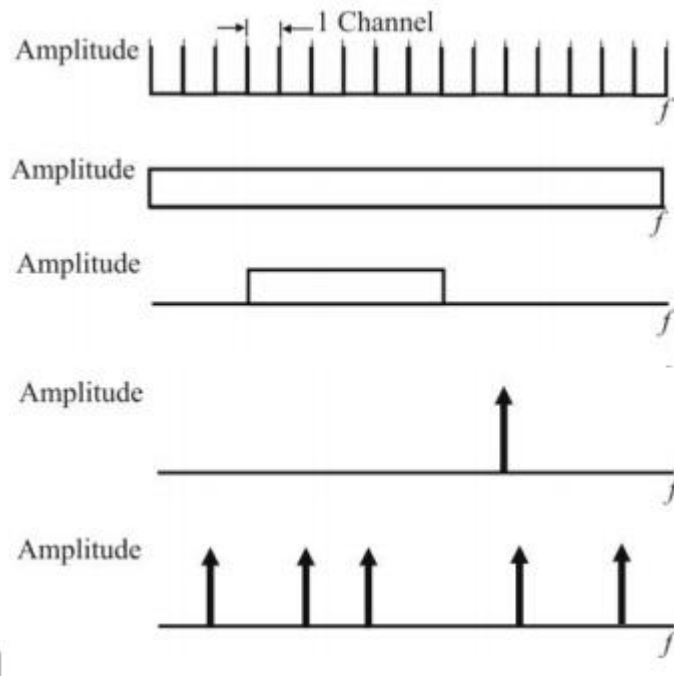


Figure 2.9: Different jamming methods present in the power spectrum

Figure 2.9 shows jamming strategies in the power spectrum. from top to bottom, it is the original signal channel, full-band noise jamming, partial band noise jamming, single-tone jamming, and multi-tone jamming respectively. [18]

The complex envelope representations for each jamming case is shown below, and these complex envelope equations will be implemented to the simulation code.

Firstly, the output of QPSK signal with AWGN can be represented as:

$$\begin{aligned} y(t) &= x(t) + n(t) \\ &= \sqrt{\frac{2E_s}{T}} \cos(2\pi f_0 t + \phi) + n(t) \end{aligned} \quad (2.12)$$

With jamming interface ,the representation can be written as:

$$\begin{aligned} y(t) &= x(t) + n(t) + J(t) \\ &= \sqrt{\frac{2E_s}{T}} \cos(2\pi f_0 t + \phi) + n(t) + \sqrt{2P_j} \cos(2\pi f_0 t + \theta_k) \end{aligned} \quad (2.13)$$

Just add a single tone jamming channel in.

The complex envelope of QPSK with AWGN and jamming channel can be rewritten with:

$$\begin{aligned} \tilde{y}(t) &= \sqrt{\frac{2E_s}{T}} e^{j\phi} + \tilde{n}(t) + \sqrt{2P_j} e^{j\theta_k} \\ &= \sqrt{\frac{4E_b}{T}} e^{j\phi} + \tilde{n}(t) + \sqrt{2P_j} e^{j\theta_k} \end{aligned} \quad (2.14)$$

Since $E_s = 2E_b$, and also E_s/T is the power of signal, P_s .

The following section will list all the necessary equations that implemented to the final code.

The single tone jammer placed at the central frequency of QPSK power spectrum is:

$$J(t) = \sqrt{2P_j} e^{j\theta_k} \quad (2.15)$$

The single tone jammer with a frequency offset is:

$$J(t) = \sqrt{2P_j} \cos(2\pi(f_0 + \Delta f)t + \theta_k) \quad (2.16)$$

P_j/P_s is the ratio of total jamming power to signal power. In these parameters, P_j/P_s , θ_k can be assign with different values; θ_k is the initial phase angle of jamming signal, and the initial value can be set with $1/4\pi$, $3/4\pi$, $5/4\pi$ and $7/4\pi$ respectively.

The complex envelope conversion from the previous equation can be rewritten as:

$$\tilde{J}(t) = \sqrt{2P_j} e^{j2\pi\Delta f t} e^{j\theta_k} \quad (2.17)$$

The complex envelope of two-tone jamming can be represented as:

$$\tilde{J}(t) = \sqrt{\frac{2P_j}{2}} e^{j2\pi\Delta f t_1} e^{j\theta_{k1}} + \sqrt{\frac{2P_j}{2}} e^{j2\pi\Delta f t_2} e^{j\theta_{k2}} \quad (2.18)$$

In this equation, $\Delta f t_1, \Delta f t_2$ represents the frequency offset belongs to their own jamming tones, and $\theta k_1, \theta k_2$ is the initial phases similar like single tone jamming.

The complex envelope of M-tone jamming can be represented as the submission of several single tones; it can be written as:

$$J(t) = \sum_{m=1}^M \sqrt{\frac{2P_j}{M}} e^{j2\pi\Delta f t_m} e^{j\theta_k m} \quad (2.19)$$

In this equation, similar to two-tone jamming, $\Delta f t_m$ represents the frequency offset belongs to their own jamming tones, and θk_m is the initial phases of tones, and the total jamming power is divided uniquely by P_j/M , a more efficient way to distribute jamming power is followed by sinc distribution function.

$P_{j,i}$ is the power for each jamming tones from total jamming power. From

$$P_{j,i} = \frac{\text{sinc}^2(f_i T_s)}{\sum_1^M \text{sinc}^2(f_i T_s)} P_j \quad (2.20)$$

instead of $P_{j,i} = \frac{P_j}{M}$.

Therefore, the complex envelope of M-tone jamming with improved individual jamming power distribution method is:

$$J(t) = \sum_{m=1}^M \sqrt{\frac{2\text{sinc}^2(f_i T_s)}{\sum_1^M \text{sinc}^2(f_i T_s)}} P_j e^{j2\pi\Delta f t_m} e^{j\theta_k m} \quad (2.21)$$

2.3.4 Other Jamming Techniques

There are some other jamming techniques that normally not used for jamming the QPSK Signal. Nevertheless, these are common techniques that commonly mentioned in some paper; the following section will introduce five types of jamming techniques, which are Swept Jamming, Pulse Jamming, Pulse Jamming, Follower Jamming and Smart Jamming. [18]

- **Swept Jamming:** A technique that similar to full-band or partial-band noise jamming. A relatively narrowband signal that as narrow as a tone. This swept jamming had a narrow frequency range that centered on a specific frequency. It is a technique that normally use to against a Frequency-Hopping Spread Spectrum Signal Detection(FHSS) target net is located.
- **Pulse Jamming:** A technique that similar to partial-band noise jamming, which the pulsed jamming could change the on/off state that not always active like partial-band jamming. Then pulse jamming could increase the power efficiency with a lower average power cycle. The duty cycle of pulse jamming could be set with specific modulation method that against with.

- **Follower Jamming:** A more advanced jamming technique that has the ability to identify the signal target, then jam at the detected frequency range. Follower Jamming is also referred to as responsive jamming, repeater jamming, and repeat back jamming. It is usually used to against the FHSS signal. A follower jamming timing is showed in figure 2.10.

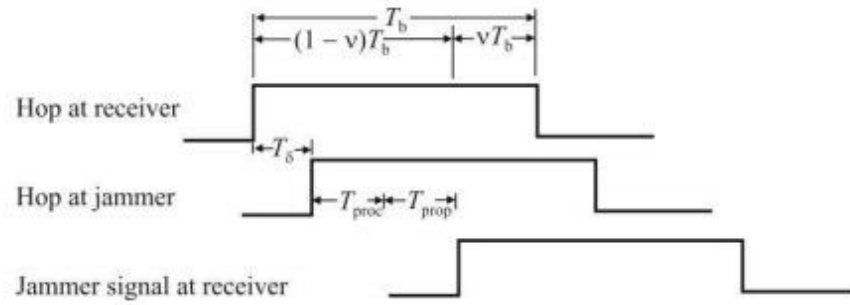


Figure 2.10: FHSS follower jamming timing

- **Smart Jamming:** Smart Jamming only affect the digital signals, which some types of communication systems required synchronized during operations. For example, some systems use a separate channel to synchronizing ; this code could be affected by smart jamming which could degrade the synchronization progress.

Chapter 3

Experimental Procedures

The project is divided into five parts, Matlab is used for simulating the QPSK Modulation with Gaussian Noise Channel and Jamming signal. The first part is the simulation and the comparison of QPSK Simulation with theoretical result, then a simple single-tone Jammer will be added in the communication channel. Next the jamming signal with different initial angel phase states and offset frequencies will be added into the original code. Then two jamming signals with equally divide jamming power will be distributed symmetric between the output signal central frequency. After that the two jamming signals will be extended to the situation with M-tone jamming, later, the influence of different jamming power distribution methods, frequencies offset and jamming power to signal power ratios will be compared and discussed in the end. Finally, with the simulation results that prided above, compare the results and find the best effectiveness cases for certain conditions.

3.1 Experimental Devices

This project is focusing on the code implementation and simulation with Matlab, therefore no electronics hardware required except a relatively high performance computer. In this project, two computers with Matlab installed in are used for simulation; The computability is important for this project since the simulation at late stage such as multi-tone jamming will spend large amount of time to get the estimate result which will explain in next chapter.

3.2 Matlab

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. The Matlab

version 2015b is used in the project, the newest version is 2016a till the submission date of report, some functions may vary between versions.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems. Especially Simulink is widely used for communication network design and simulation. [10]

3.3 Matlab Code Syntax

The following document shows the code syntax which is used in the program for simulation.

colon,: [9] is one of the most useful operators in MATLAB. It can create vectors, subscript arrays, and specify for iterations. The colon operator uses the following rules to create regularly spaced vectors for scalar values i , j , and k :

$j:k$ is the same as $[j, j+1, j+2, \dots, j+m]$, where $m = \text{fix}(k-j)$. In the case where both j and k are integers, this is simply $[j, j+1, \dots, k]$. This syntax returns an empty matrix when $j > k$.

$j:i:k$ is the same as $[j, j+i, j+2i, \dots, j+m*i]$, where $m = \text{fix}((k-j)/i)$. This syntax returns an empty matrix when $i == 0$, $i > 0$ and $j > k$, or $i < 0$ and $j < k$.

Colon can be used to create a vector of indices to select rows, columns, or elements of arrays, $A(j:k)$ is $A(j), A(j+1), \dots, A(k)$. In the simulation program, colon is used to create vectors for example:

`Eb_No_dB=[-2:1:11];` Generate a 1×11 vector `Eb_No_dB` from 1 to 11

`nn=[1:1:Nspb];` create a vector that preparing for calculating the jammer phase angle

`Eb_No_dB=[Eb_No_dB.l:delta_Eb_No:Eb_No_dB.h];` Generate an advanced version `Eb_No_dB` that can manually set the lowest and highest value of `Eb_No_dB`, which can be a floating number.

And for loop iterations are widely used in the simulation program.

for `iiEbNo=1:length(Eb_No);` Create a loop calculate the entire simulation with `Eb_No_dB` value from -2 to 11.

for `ii=1:blocklength;(Above data(ii))` generate the binary data filled in block;

for `ii=1:2:blocklength;` Generate vectors that divided in half and distributed into two series that can generate the QPSK phases;

for `ii=1:blocklength/2;` is used for symbol detection, from the QPSK demodulation

, summarize the different symbols and calculate the bit errors divided the total bits to estimate the bit error rate

randn: [13] Array of randn values Syntax $R = \text{randn}(\text{sz}, \text{arraytype})$. Argument sz , values m, n or $[m \ n]$ specifies size as an m -by- n matrix. Argument arraytype , distributed specifies distributed array; codistributed specifies codistributed array, using the default distribution scheme; gpuArray specifies gpuArray.

In the project simulation, randn is used to generate the random numbers with normal distribution of binary data and the noise vector.

$\text{data} = \text{randn}(1, \text{blocklength})$ create a $1 \times \text{blocklength}$ (in this project blocklength is equal to 256) vector called data that contain numbers followed with normal distribution.

$n = \text{sigma_n} * (\text{randn}(1, \text{Nspb} * \text{blocklength} / 2) + j * \text{randn}(1, \text{Nspb} * \text{blocklength} / 2))$; generate the noise vector, the Gaussian noise channel.

$\text{randn}(1, \text{Nspb} * \text{blocklength} / 2)$ create a 1 multiply by $\text{Nspb} * \text{blocklength} / 2$ (in this project Nspb is 16) which each symbol in a block will be created and used for create Gaussian noise vector.

rand [12] is the Uniformly distributed random numbers, returns a single uniformly distributed random number in the interval $(0, 1)$. In the code, we use $2 * \pi * \text{rand}$ to get the random number in the interval $(0, \pi)$ to calculate jamming phase angles.

ones [11] create array of all ones, and **zeros** [14] create array of all zeros. For example, in the code, $\text{ones}(1, \text{Nspb})$ filled the space for x , the input signal.

$S = \text{zeros}(1, \text{Nspb})$; is prepared to the M-tone jamming signal.

3.4 QPSK Simulation

The first code is the comparison between Pb Performance (Bit Error Rate) QPSK Simulation and QPSK Modulation Theoretical result. It is the demo code from the previous course 'Introduction to Simulation of Digital Communication Systems'.

clc clears all input and output from the Command Window display, giving a "clean screen." **clear** removes all variables from the current workspace, releasing them from system memory.

Primary parameters include the initial **Eb_No** dB value, which is the energy per bit to noise power spectral density ratio. It is a normalized signal-to-noise ratio (SNR) measure, also known as the "SNR per bit". It is especially useful when comparing the bit error rate (BER) performance of different digital modulation schemes without taking bandwidth into account.

Blocklength is the block numbers of data that will be generated in the following code, these data will create a series of binary data and is used for estimating the bit error rate and find the influence with Eb/No value and jamming signal. In this code, the blocklength is set to 256 and it should be an even number to make it simple with following QPSK Modulation Division.

Nspb is the number of samples per baud (symbol), and it is equal to 16 in simulation.

N is the number of bit errors for the error determination, the bit error rate estimation can be shown in equation X, the more N value will get a better estimate. In practice, $N \geq 10$ and for a good estimate, $N \geq 100$. However, it will take time when using large N values, at the situation with less calculation on the easier code, for example the single-tone jamming signal simulation, $N=1000$ will be used to estimate a smooth graph and more reliable result. $N=10$ can be used to get a quicker results and find the trend and draft graph during simulation, because most of times, both code problems and simulation will take a long period of time and it will be easily going to infinitely run. $N=100$ will be used to estimate the performance with M-tone jamming signal, because $N=1000$ will take more than one day to run a single time.

Eb_No=10.^(Eb_No_dB./10); is a derived parameter that simple convert the dB value to its numeral value for Eb_No dB value.

```
for iiEbNo=1:length(Eb_No);
```

```
...
```

```
end
```

The loop that contain the simulation from -2dB to 9dB of Eb/No value, in one loop.

N_errors=0, assign a variable N_errors that will be used for countering the number of bit errors.

N_bits=0, assign a variable N_errors that will be used for countering the number of total bits.

Sigma_n=sqrt(Nspb/ (4*Eb_No(iiEbNo))), establish a noise standard deviation that will be used to generate the noise vector, the Gaussian Noise Channel.

condition=0 is used in the loop to decide whether the simulation end, the initial value is equal to 0 means the code will keep running until

```
if N_errors>N
    condition=1
```

The condition to terminate the simulation when the simulation collect enough bit errors which greater than N, the number of bit errors for the error determination.

N_bits=N_bits+blocklength; Because the initial value of N_bits is equal to 0, in the loop with condition==0, the number of total bits is equal to the blocklength of data.

```
data=randn(1,blocklength);
for ii=1:blocklength
    if data(ii)>0
        data(ii)=1;
    else
        data(ii)=0;
    end
```

Assign the vector data, generated random value data followed with normal distribution. And then create the new vector data(ii), that covert the vector data to a series of random binary numbers. For all of the blocklength (256) numbers within the data vector, change all the positive value to 1, and all the negative values to 0.

```
for ii=1:2:blocklength
    a=data(ii);
    b=data(ii+1);
    if a==0 & b==0
        phi(ii/2+1/2)=pi/4;
    elseif a==1 & b==0
        phi(ii/2+1/2)=3*pi/4;
    elseif a==1 & b==1
        phi(ii/2+1/2)=5*pi/4;
    else
        phi(ii/2+1/2)=7*pi/4;
    end
```

Generate the QPSK phase values by using the binary data generated above. According the QPSK modulation theory in (2.2), one phase angle value can present two bits. Then

this loop converts all the 256-bits binary data to its corresponding phases. Generate the

```
n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*blocklength/2));
```

Gaussian noise vector. According to (2.14), the complex envelope of n .

```
for ii=1:blocklength/2
    x((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Ps)*exp(j*phi(ii)).*ones(1,Nspb);
end
```

Generate signal vector for the block, since QPSK signal represent two bits in one phase value. The loop runs only half of the blocklength, and QPSK signal is generated with the QPSK Complex Envelope (2.14).

$y=x+n$; is the additive communications channel that add the QPSK signal channel and Gaussian Noise Channel, this will be add a jamming channel together at the following jamming simulations.

```
for ii=1:blocklength/2
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
```

Is the QPSK signal demodulation with the matched filter detection method followed with (2.14) $iir=0$ assign the variable iir for the demodulator, the real part of the output signal will be demodulated the odd number of bits, and the virtual part will be demodulated the even numbers of bits. After all the process with a loop with half of the blocklength.

A new vector `datar` will be created that contained the series of demodulated binary data vector.

errornumber=**sum(mod((data+datar),2))**, compare the different between `data` and `datar`, submission the number of different binary data.

N_errors=**N_errors+errornumber**; Then `N_errors`, the number of error difference will be added in to the total number of bit errors.

PB(iiEbNo)=**N_errors/N_bits** , the BER can be easily calculated by the total number of bits error divided by the total number of bits.

PB_theory=**qfunc(sqrt(2*Eb_No))**; is the theoretical results with (5) of QPSK signal modulation with gaussian noise channel. Then

`figure(1)`

`semilogy(Eb_No_dB,PB,Eb_No_dB,PB_theory)`

Plot the semilogy graph with the comparison between `PB_theory` and `PB`, the simulated graph.

To be conclude, the primary parameters `Eb_No` range and the number of bit errors for the error determination `N` affect the computational work, large number will get a better estimate however it also takes more time. And the simulation code computed all the simulation within `Eb_No_dB` range, at the first code generates series of 256-bit random of data, and convert to binary data, then these data is used to consult QPSK phase angles, next QPSK phases as the input signal channel and add a Gaussian noise channel, after that the matched filter detection will demodulate the merged new data stream, because the nose vector is involved in the communication procedure, there will be some errors occur, and the final system performance can be estimated by calculating the BER value of the system. In the following part, a new channel jamming channel will be added into the communication link, therefore the communication link quality and BER can indicate the effectiveness of the jamming channel.

3.5 QPSK with Single tone jamming

$\mathbf{ja}((ii-1)*N_{spb}+1:(ii*N_{spb}))=\sqrt{2*P_j}*\exp(j*\phi_{ij})$; A simple form jamming with the complex envelope (2.15), is introduced in the communication channel.

P_j is the jamming power, and P_s is the signal power. Then we can get the jamming power to signal power ratio is equal to P_j because normally in the project we set the P_s is equal to 1.

ϕ_{ij} is the initial phase state of the jamming signal.

$\mathbf{y}=\mathbf{x}+\mathbf{n}+\mathbf{ja}$; The additive communications channel (with jammer), the jamming channel is also added after the AWGN channel.

Then we can manually set set different P_j and ϕ_{ij} to see the effect of jamming.

Meantime, $PB(iiEbNo)$ is the first loop estimates the jamming performance with $P_j=0.01$, jamming power is increased to $P_j=0.05$ and 0.5 in following similar loops. More loops can be added into the code and see the difference between jamming parameters.

In this project, for instance, $PB2(iiEbNo)$, $PB3(iiEbNo)$ is estimated in the code, and also the PB_theory indicates the circumstance with no jamming channel as the reference line. Then a comparison graph can be plotted with

`semilogy(Eb_No_dB,PB3,Eb_No_dB,PB2,Eb_No_dB,PB,Eb_No_dB,PB_theory)`

The following code use the same plotting method to compare the different result by adding more estimate loops, and plot it in the same graph. Jamming performance can be easily identified by see the Pb_No performance.

3.6 Single tone with frequency offset

Before implement the code, the Eb_No_dB range is modified with a more complex form that can type float number in it.

$\mathbf{nn}=[1:1:N_{spb}]$; create the column that calculate the jamming phase angle with frequency offset, in the following code $\Delta fT.*nn(1:N_{spb})./N_{spb}$; for example $1/16*\Delta fT$, $2/16*\Delta fT$. The frequency offset will equally distributed in the sampling process.

θ is the initial phase, and ϕ is the phase angle of the frequency offset, it is the random number between 0 to 2π

$\phi=2.*\pi.*rand$ $\theta=2.*\pi.*rand$;

Then the jamming signal with frequency offset can represent as

```
ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2.*Pj).*exp(j*theta) ...
.*exp(j.*2.*pi.*deltafT.*nn(1:Nspb)./Nspb).*exp(j.*tsi);
```

The complex envelope of jamming signal is shown in (2.17).

At the end, we can find the difference between frequency offset values(δfT), the previous code is the jamming signal within the frequency central, and the range is δfT function is -1 to 1. For example, 0.9 0.5 and 0.1 means the frequency offset is closer to central frequency with $\delta fT = 0.1$. This simulation will estimate the influence with different frequency offsets.

3.7 QPSK with Two-tone jamming

Next, the code is further developed to the two-tone jamming situation.

The complex envelop of two jamming signal has shown in (2.18), and the code implication is shown below

```
ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2.*Pj./2).*exp(j*theta1) ...
.*exp(j.*2.*pi.*deltafT1.*nn(1:Nspb)./Nspb).*exp(j.*tsi1)+ ...
sqrt(2.*Pj./2).*exp(j*theta2) ...
.*exp(j.*2.*pi.*deltafT2.*nn(1:Nspb)./Nspb).*exp(j.*tsi2);
```

In this code, the frequency offset is normally symmetrical between central frequency, $tsi1$ $tsi2$ and $\theta1$ $\theta2$ is the same as previous code but set with separate random numbers for each jammer. And the power of jamming signal is equally distributed to two jammers, the overall power of jammers is still P_j , the same as the single tone jamming.

Similarly, like the single tone with frequency offset, firstly the difference between frequency offset values(δfT) that distributed between the QPSK signal will be estimated. And the effectiveness with different jamming methods can be estimated in the simulation.

3.8 QPSK with M-tone jamming

Multi-tone jamming is a more complex version that expand the two tones jammer with M numbers of tones which the frequency allocation is uniformly distributed to the entire spectrum. This project will firstly determine the different power distribution methods of M -tone jamming, one is the total jamming power will distribute equally into each tones, the jamming power of each tones is P_{ji} , another way to distribute the jamming power, is the sinc function. It means for every P_{ji} , the closer tone to the central frequency, the larger P_{ji} will be distributed. Then with a better scheme, find different jammer parameters that affect the jamming performance.

The primary parameters are the maximum frequency offset, the power strength of P_j towards P_s , which is the power ratio between jammer and input signal, and the numbers of M tones that used in the jamming progress.

```
deltafT_h=0.8;
deltafT_l=-deltafT_h;
delta_deltafT=deltafT_h/(M-1)*2;
deltafT=[deltafT_l:delta_deltafT:deltafT_h];
```

The maximum and minimum frequency offset is set to 0.8, which is the edge frequency M -tone jammers can reach. This code is similar to the `Eb_No_dB` setting but the minimum frequency is the negative maximum frequency since the distribution is symmetrical, meantime, `delta_deltafT`, the interval should be doubled to match the entire frequency distribution.

$M=10$; is the number of jamming tones that will allocate to the spectrum.

```
for m=1:M
    tsi(m)=2.*pi.*rand % i
    theta(m)=2.*pi.*rand
end
```

Running a loop that filling the vector from 1 to M that contain all the random phases will be used in the following code.

```
S=zeros(1,Nspb);
for m=1:M
    mm=sqrt(2.*Pj./M).*exp(j*theta(m))...
    .*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)./Nspb).*exp(j.*tsi(m))
    S=mm+S;
    ja((ii-1)*Nspb+1:(ii*Nspb))=S;
    tsi(m)=tsi(m)+2.*pi.*deltafT(m); %next block initial state additi
end
```

S is the temporary parameter that consult the final jamming signal, this code is the expanded version from two-tone jamming, it runs m times that form a M -tone jamming at the end, and for each tone, the phase angles is different, and these angles will be continuous during one $Nspb$ period.

From the above code, the jamming power distributed is uniform because P_j/M has been shown in (2.19). And for the next loop, the jamming code is

```
mm=sqrt(2.*(sinc(deltafT(m))^2)./SINC.*Pj).*exp(j*theta(m))...
.*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)./Nspb).*exp(j.*tsi(m))
```

$(sinc(deltafT(m))^2)./SINC.*P_j$ means P_j is followed with sinc function distribution has shown in (2.20), and the complex envelop of multi-tone jamming with improved sinc individual jamming power distribution method has shown in (2.21).

Then in the project, the comparison between different P_j distribution method, frequencies offset, power to signal ratio, the number of M for multi-tone jammer, and also the comparison to the simplest jamming, one tone jamming within the frequency offset. Finally, we will find the best performance jamming technic in different situations, the best performance is determined by the highest E_b/N_0 that affect the final decoding.

In order to find jamming effectiveness with the number of M influence, some parameters are fixed to find the relationship E_b/N_0 VS M .

Eb_No_dB=8; the SNR is 8 or 9 to get a better estimate.

M=[4,10,20,30,40,50,100,200,500,1000]; Then in the code, series of M number of tones can be simulated in order and plot at the end.

```
SINC=0
for m=1:M(iM)
    SINC=SINC+(sinc(deltafT(m)))^2
end
for m=1:M(iM)
    mm=sqrt(2.*(sinc(deltafT(m))^2)./SINC.*Pj).*exp(j*theta(m))...
    .*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)./Nspb).*exp(j.*tsi(m))
    S=mm+S;
    ja((ii-1)*Nspb+1:(ii*Nspb))=S;
    tsi(m)=tsi(m)+2.*pi.*deltafT(m); %next block initial state additive
```

There are some modifications in the code, the SINC adds all of jamming power, form the total jamming power. Then vectors are covered in the loop which is trying to calculate the E_b/N_0 values, and plot all the results at the end.

```
figure(1)
plot(M,PB)
grid
xlabel('Number of M')
ylabel('BER')
title('Number of M Tone Jamer effects')
legend('Simulated P_B(Pj=0.05)')
```

In addition, the result between each time may vary because too many random numbers are running during the code. And SNR, edge frequency offset, especially the jamming power strength will give the communication channel huge impact on the final performance.

Chapter 4

Results and Discussion

The objective of project is to find the best case of the jamming could affect QPSK modulation; Jamming performance can be estimated by estimating the probability of bit errors in communication system, the bit error rates(BER). The QPSK modulation theoretical results can be used as the reference level to correct especially code errors, generally implementing any jammers in communication channel will increase the BER, and when the simulation result is less than the QPSK the theoretical result; Then some structural and systematical errors may be in the program. Next the QPSK signal with a single tone jammer placed upon the central frequency of power spectrum density will be another reference level because it is the simplest jamming technique to estimate the difference among the effectiveness of jamming methods.

4.1 QPSK Simulation

Procedures of the entire project has been introduced in the previous chapter in detail. First code is the QPSK Simulation, the basic code of QPSK Modulation and Demodulation with AWGN channel. Then the simulated result will be compared to the theoretical results in (1). As we mentioned above, N is the number of bit errors for the error determination, a large N value could estimate a better result, following figures show the QPSK Simulation, the simulated estimate result is compared with theoretical result, all parameters are setting with the same value except $N=10,100,1000$, respectively.

From the simulated result showed in figure 4.1, increasing E_b/N_0 , the symbol energy could reduce P_b , and QPSK Simulation results is coinciding with theoretical results. As N increasing, a better estimate result will get. $N=10$ will get a roughly correct trend but with fluctuation, it can be used to see the trend roughly and for debugging purpose; $N=100$ is used to get estimate in general, when the code is not too complex, from the graph, the blue line of the simulated P_b is mainly coinciding with the theoretical P_b red line, with a little fluctuate; $N=1000$ is the best to show a very smooth graph with high-quality estimate.

The following sections will explain different tone-jamming schemes, single-tone jamming, two-tone jamming will be estimated with $N=1000$, for m-tone jamming, the cal-

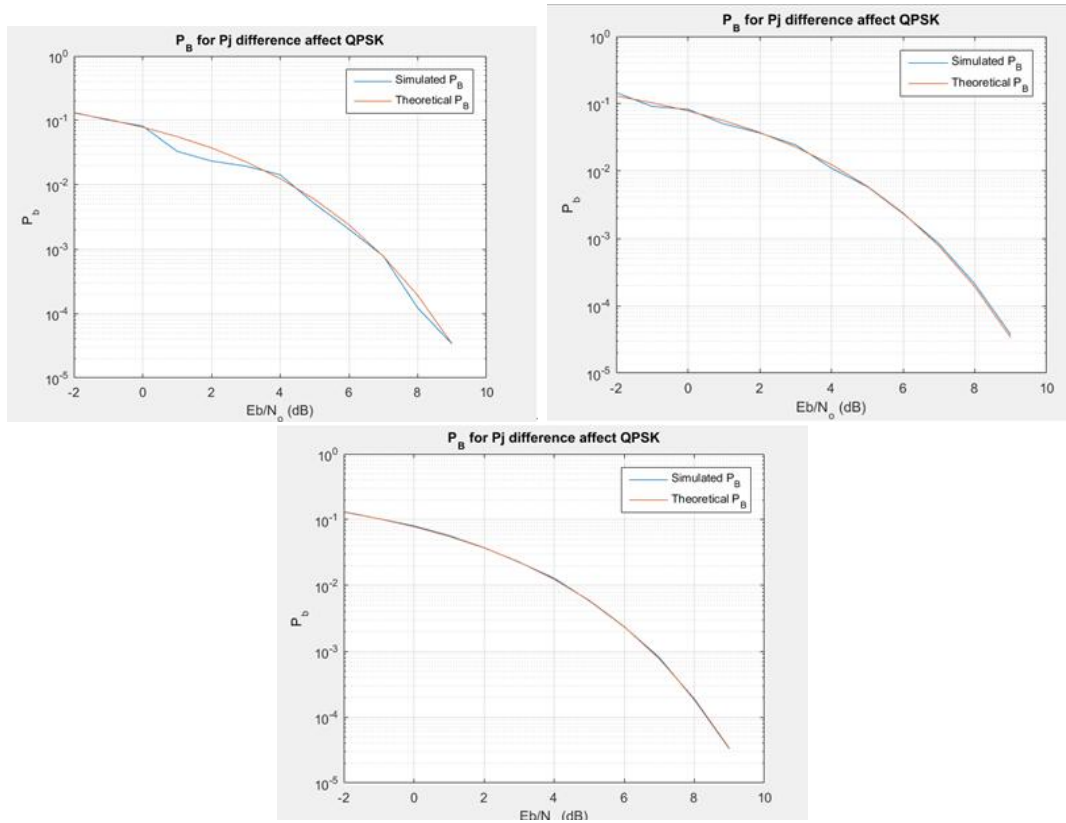


Figure 4.1: $N=10, N=100, N=1000$ for QPSK Simulation

culated quantity is huge, especially for 100 tones, $N=100$ will be used for the reasonable estimate.

4.2 QPSK with jamming channel

The following results will show firstly the influence with a total power of jammer to signal, then the influence with the initial phases state.

4.2.1 P_j vs P_s

From figure 4.2 we can see four lines have been shown in the graph, three simulated lines with jamming power and one theoretical line. As the P_j increase, some significant changes affect the BER, the theoretical P_B is at the bottom, and the highest value of $P_j=0.5$ is at the top, and almost form a straight line with the consistent value of 0.25. The results from $E_b/N_0 = 7\text{dB}$ to 9dB is shown in Table 4.1.

Ratio represents the P_B value from $P_j=0.05$ divided by $P_j=0.01$; From above results, we can see the ratio is further increased with the rise of E_b/N_0 . And with the growth of jamming power, a greater growth of error rates can be estimated from simulated results.

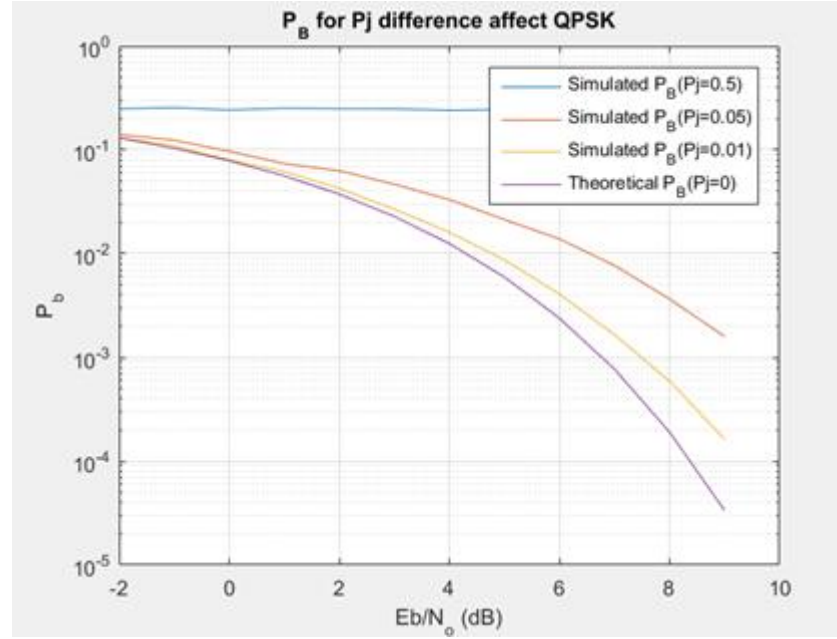


Figure 4.2: The influence between different jamming power

Table 4.1: Relationship between EbNo and Pb

EbNo	7dB	8dB	9dB
Pj=0.05	0.007697	0.003658	0.001594
Pj=0.01	0.001661	0.0005921	0.0001637
Pj=0	0.0007727	0.0001909	3.363e-05
Ratio(Pb)	4.633955	6.17801	9.737324

To conclude, the power of jamming power to the power of signal has a huge impact on the jamming performance. However, with some portable devices, the battery power is limited, how effectively allocate the jamming power will be the challenge towards jammer design.

This simulation runs several times; other similar results can be found in the later demonstration.

4.2.2 Initial phase angle

Another character of single-tone jammer shows in figure 4.3.

Four lines coincide together with different initial phase angles set-up. It indicates that there is no difference between the value of jammer initial phases.

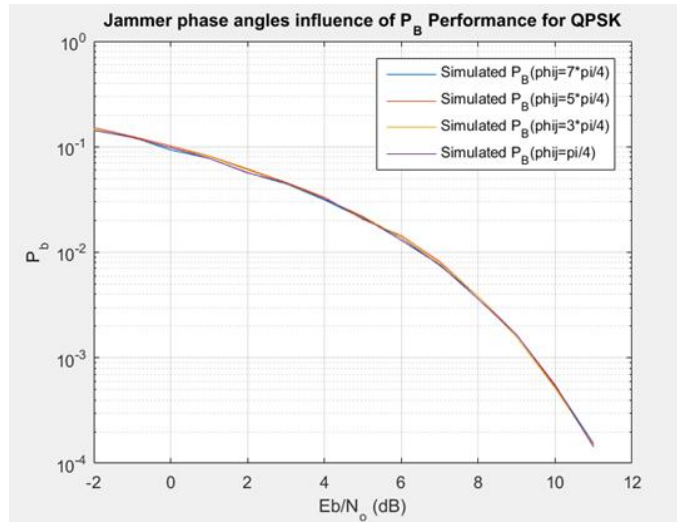


Figure 4.3: The influence between different initial phases

4.3 Jamming with frequency offset

This section, the jamming signal will distribute with frequency offset away from the central frequency of QPSK signal.

Firstly, set $\Delta f T = 0$, to check the correctness of the program. The result shows in figure 4.4.

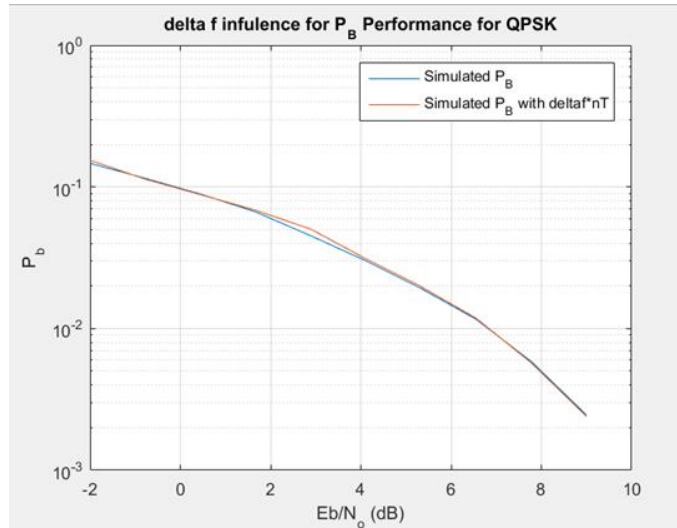


Figure 4.4: initial check with frequency offset=0

Two lines mainly coincide together, the blue line is the simulated P_B without frequency offset, and the red line is the $\Delta f T = 0$, it means the jamming power spectrum is still placed on the central frequency. The simulated result is expected and verify the $\Delta f T = 0$ is correct in code.

Then, another extreme condition will be estimated first, $\Delta f T = 1$ or -1 shows in figure 4.5

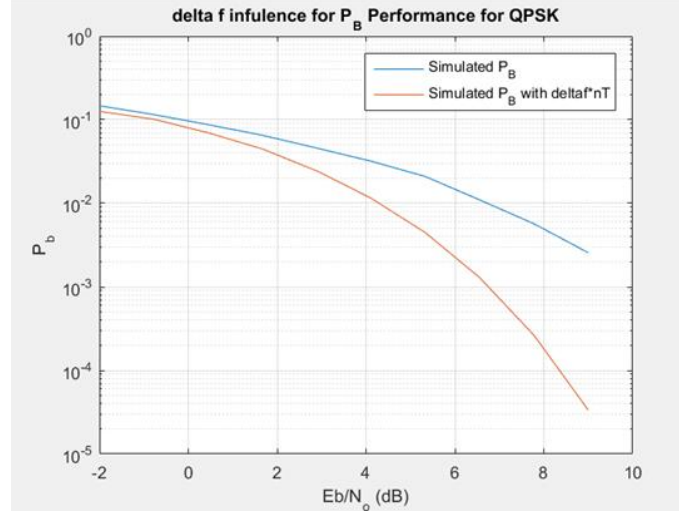


Figure 4.5: Jamming with frequency offset=1,-1

Blue line is the P_b with reference jamming signal, and the red line is P_b with jamming signal located at the edge of the QPSK power spectrum. Blue line is located above the entire red line and the big gap between two graphs indicates that the frequency offset value affects the jamming effectiveness of QPSK signal.

Decrease the frequency offset $\Delta f T = 0.5$, shows in figure 4.6

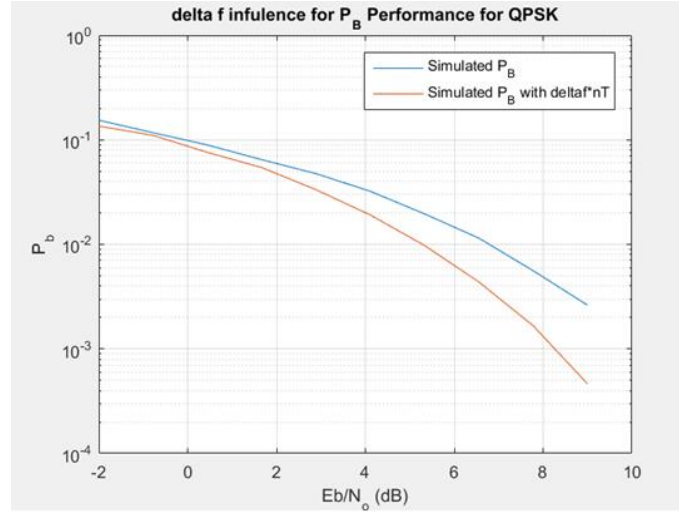


Figure 4.6: Jamming with frequency offset=0.5

From figure 4.6 we can see the blue line is the same reference jamming signal, the gap between two lines is reduced, and a better P_b performance for $\Delta f T = 0.5$ estimate compared with the estimate of $\Delta f T = 1$.

Continually decrease the frequency offset to 0.2, shows in figure 4.7

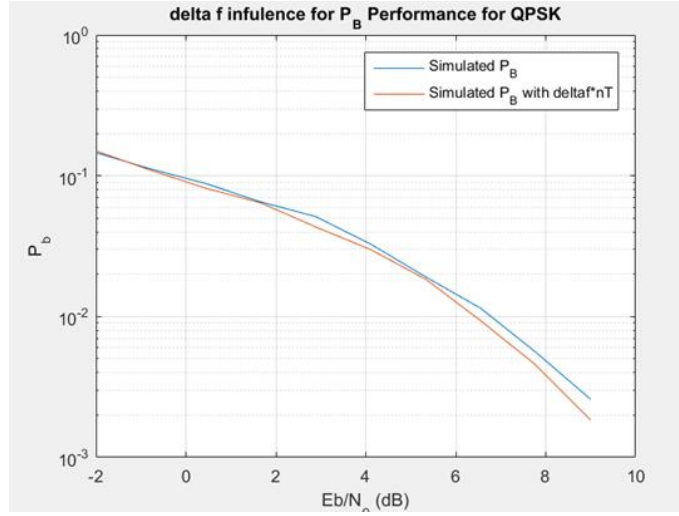


Figure 4.7: Jamming with frequency offset=0.2

A better performance for $\Delta f T = 0.2$ compared with previous frequency offsets.

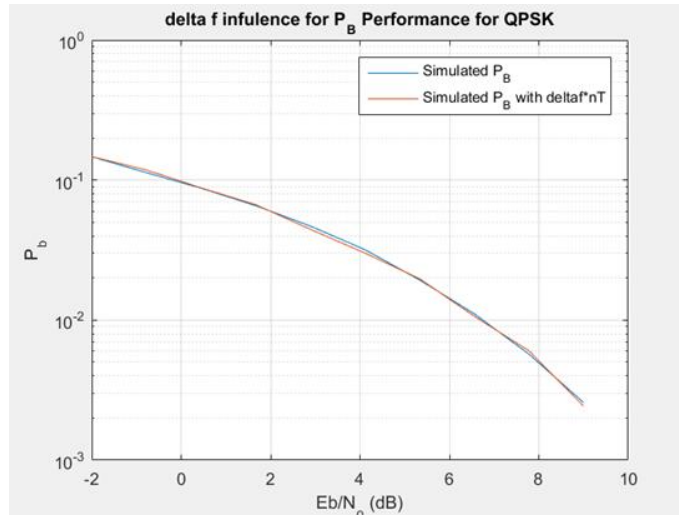


Figure 4.8: Jamming with frequency offset=0.08

Figure 4.8 is the frequency offset $\Delta f T = 0.08$, the gap between two lines is continually merging, and there is little difference between the simulations.

A conclusion of difference frequency offsets, the comparison result, shows in figure 4.9.

Figure 4.9 shows the relationship between frequency offsets and P_b performance, the increasing $\Delta f T$ will decrease the jammer performance for single-tone jamming.

In addition, two symmetric values between central frequency will get the same estimate result, for example, in figure 4.5. P_b between two symmetric jamming signals is same

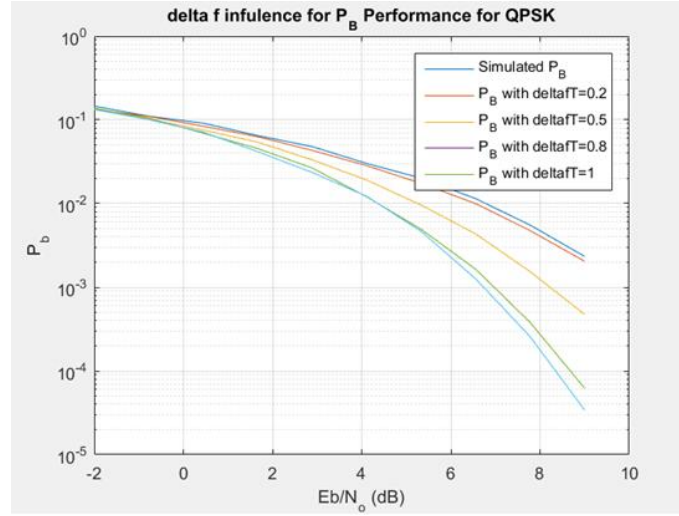


Figure 4.9: A conclusion of jamming with frequency offsets

between two results. Also when $\text{delta}fT=0.2$ and $\text{delta}fT=-0.2$, the estimated results shows in figure 4.10

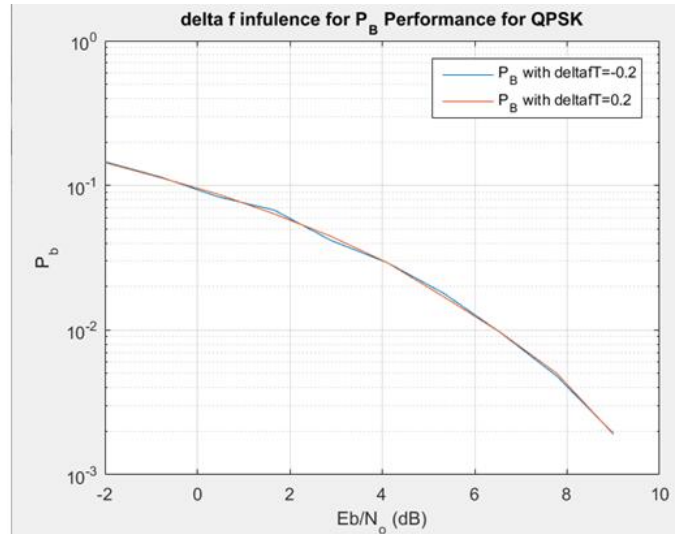


Figure 4.10: Jamming with frequency offset=0.2,-0.2

All previous simulation is running with $P_j=0.1$, in the following part, a small total jamming power will be implemented into the simulation. Figure 4.11 shows the same frequency offset with the previous section.

Similarly, the increasing $\text{delta}fT$ decrease the jamming effectiveness for single tone jamming with small jamming power.

To conclude, for single tone jamming within the same jamming power, the jamming power spectrum located at the central of the QPSK spectrum will get the highest P_b

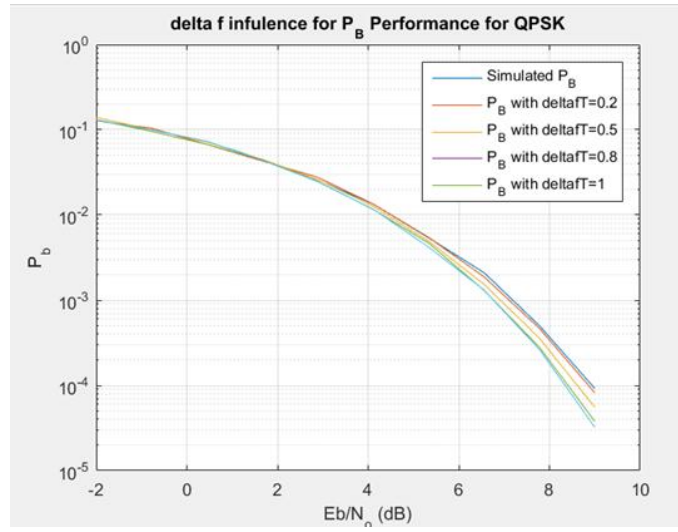


Figure 4.11: Jamming with dFTs in a smaller P_j

result. And the frequency offset has an adverse impact on the jammer performance. The next part will show the jamming performance with two-tone situation.

4.4 Two-tone jamming

The test progress is similar to the single-tone jamming; different Δf value will be tested during the estimate, find the effectiveness of two-tone jamming in different situations. All of the simulations in this section is estimated with $N=1000$, and E_b/N_0 dB from -2 to 9. Different jamming power to signal power ratios range will be test respectively.

Firstly, when $P_j=0.05$, $\Delta f=0.5$, the simulation graph has been shown in figure 4.12.

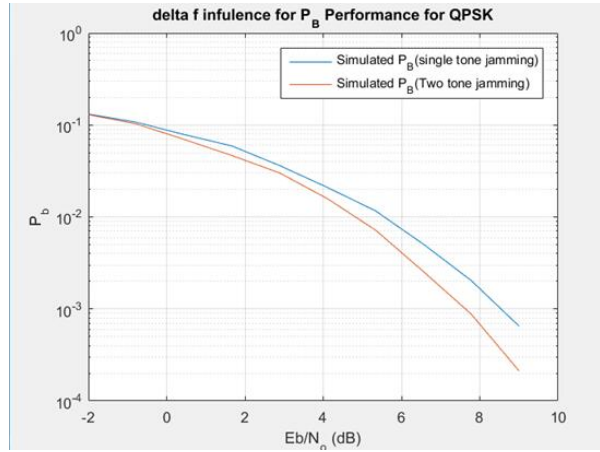


Figure 4.12: 1-tone vs 2-tone jamming case 1

From this graph we can see two lines represent the P_b performance for each jammer, the blue line is the reference jammer of single tone jamming within the central frequency, and the red line is the two jammer tones. The total jamming powers for two methods are the same $P_j=0.05$. It obviously shows the reference single tone jamming has a better performance than the two jamming tones.

Then decrease the two jamming tones to $\Delta f=0.2$, the simulation graph shows in figure 4.13 .

This time, two lines mainly coincide together, it means with frequency offset 0.2, there is little difference between two jamming methods. The interspace between two graphs may due to the random progress during simulation, the following chart showed the simulation with same parameters. A better result can be found in several times of running, for example a similar result with better estimated shows in figure 4.14.

Then we continually decrease the frequency offset to 0.1; the simulation is shown in following figure 4.15.

This result is varying with the previous estimate; The result with frequency offset is greater than the reference single-tone jammer. Two lines divergent after 7dB and two tone jamming has a better performance than the single town jamming. This result is not occasional it can be estimated by running several times, the result is still the same, the effectiveness of two- tone jamming is better during a high E_b/N_0 dB environment.

The effectiveness can be even larger when P_j is relatively large, for example, $P_j=0.2$

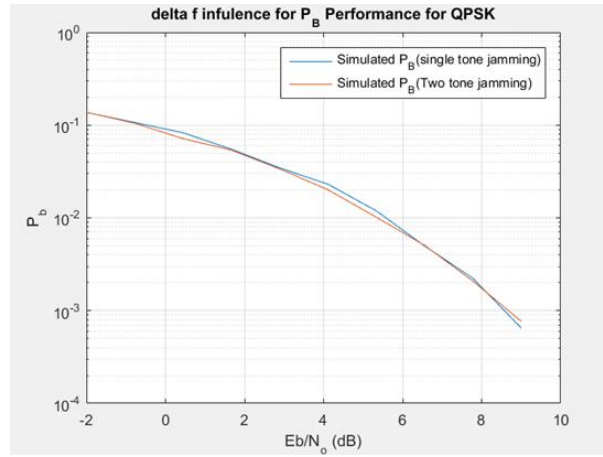


Figure 4.13: 1-tone vs 2-tone jamming case 2

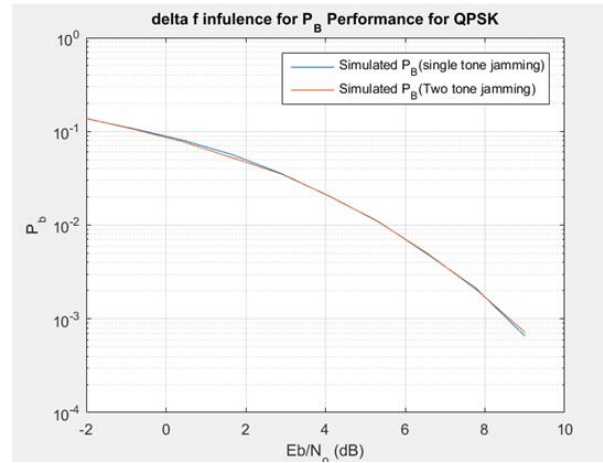


Figure 4.14: 1-tone vs 2-tone jamming better results case 2

From figure 4.16. It is obviously shown that there is a large gap between two jamming methods and the two tone jamming method has a better performance after 6dB.

To conclude, the effectiveness of two-tone jammer can be estimated by using with great E_b/N_0 environment with relatively large P_j/P_s . The frequency offset should be closer to the central frequency to get better performance. Large frequency offset will bring a worse impact on the final jamming performance.

4.5 Multi-tone Jamming

Next, a more complex version of the jamming method is expanded from the two-tone jamming scheme from previous section, the number of M tones will be distributed around QPSK signal. The relative primary parameters are estimated in the following section, which is the two P_{ji} distribution methods, and Δf_T , the maximum and minimum

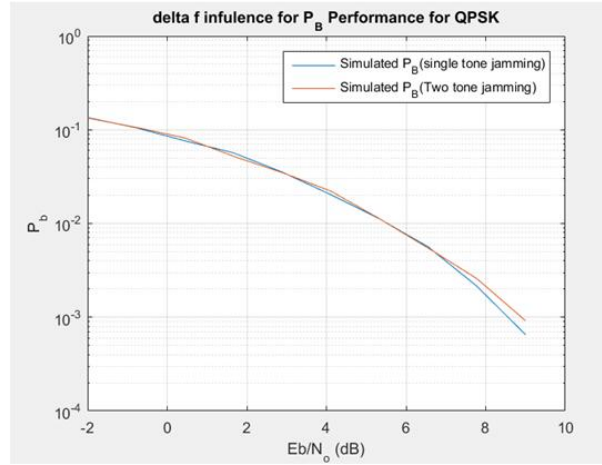


Figure 4.15: 1-tone vs 2-tone jamming case 3

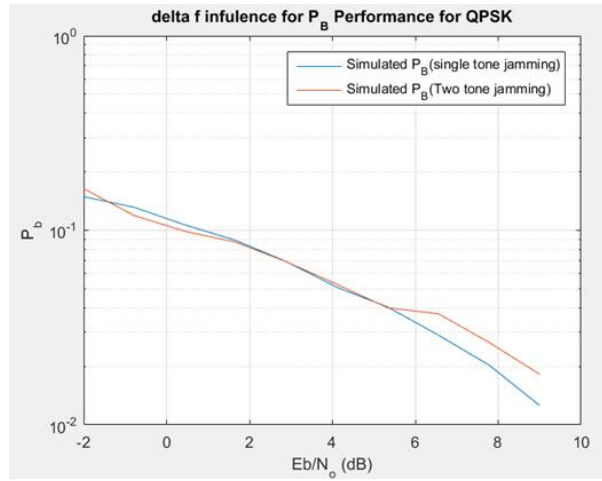


Figure 4.16: 1-tone vs 2-tone jamming case 4

frequency offset, the total jamming power that can be allocated with, and the different number of M tones. These parameters will be estimated, and the best case will be found for the best jamming performance.

4.5.1 P_j Distribution Methods

The simulation results show in following figures, from figure 4.17 to figure 4.22. Some parameters setting is chosen, these could cover some general cases with in all the range of values to see the difference of P_b performance.

From top to bottom of graphs, these parameters are setting with:

$P_j=0.1$, $\Delta f T=0.8$, $M=10$

$P_j=0.1$, $\Delta f T=0.08$, $M=10$

$P_j=0.05$, $\Delta f T=0.4$, $M=10$

$P_j=0.05$, $\Delta fT=0.2$, $M=10$

$P_j=0.01$, $\Delta fT=0.8$, $M=10$

$P_j=0.01$, $\Delta fT=0.08$, $M=10$

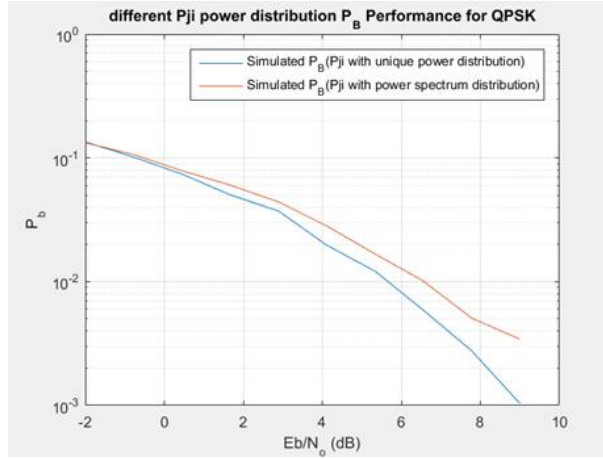


Figure 4.17: Pji distribution method, $P_j=0.1$, $\Delta fT=0.8$, $M=10$

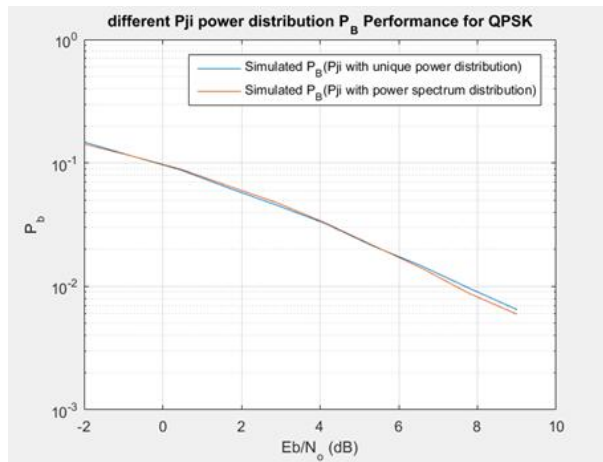


Figure 4.18: Pji distribution method, $P_j=0.1$, $\Delta fT=0.08$, $M=10$

These figures indicate that, the blue line is Pji distributed uniquely and the red line is Pji distributed with sinc function, the more complex distribution method. Pji distribution with sinc function either the same or has a better efficiency than the unique jamming power distribution. Better performance with sinc distribution is estimated. The result may be due to the higher jamming power allocated close to the central frequency; a Pji sinc distribution has a greater power closer to the QPSK central frequency than the Pji with unique distribution.

Other related simulation results could be found in later demonstration, which show the simulation with different parameter combinations between frequency offset and total jamming power.

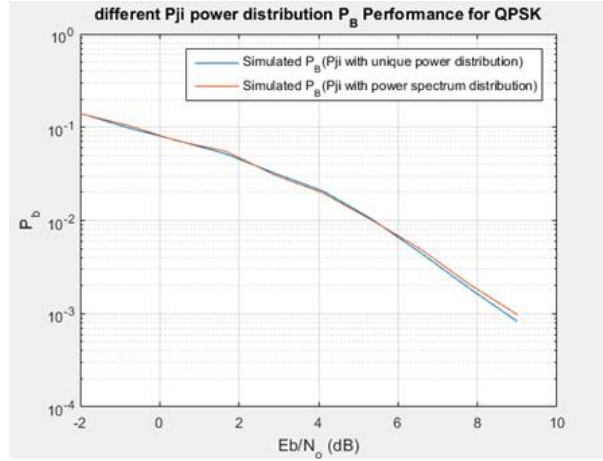


Figure 4.19: Pji distribution method, $P_j=0.05$, $\Delta fT=0.4$, $M=10$

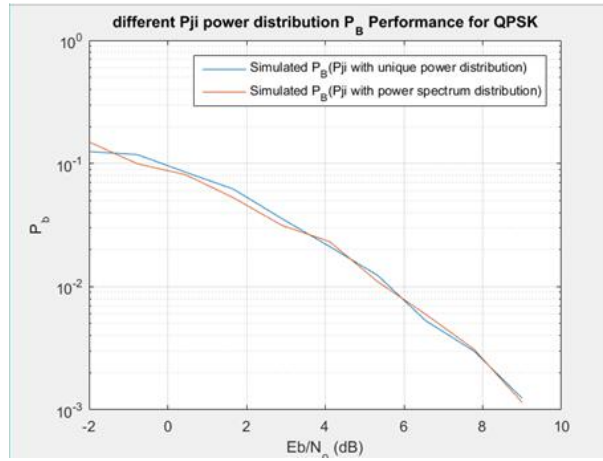


Figure 4.20: Pji distribution method, $P_j=0.05$, $\Delta fT=0.2$, $M=10$

There is little difference between two Pji distribution methods, when frequency offset value is too small, because the narrow frequency range centralizes all the jamming power around QPSK central frequency, the effect will be less with different distribution methods.

In addition, if the P_j value is set to less in the lower frequency range, there will be little difference between two methods because two jamming methods will be both less effective compared with reference jammer. This case is estimated by $P_j=0.01$, $\Delta fT=0.08$, $M=10$, a narrow frequency offset with small total jamming power, and with relatively large number of tones. Shown in figure 4.22.

In conclusion, we generally use the sinc function to allocate the Pji in M-tone jamming, because better effectiveness can be found in the sinc distribution function than the unique Pji, for most of the situations, in particular for a wider frequency jamming range, the distribution with sinc function could get better performance.

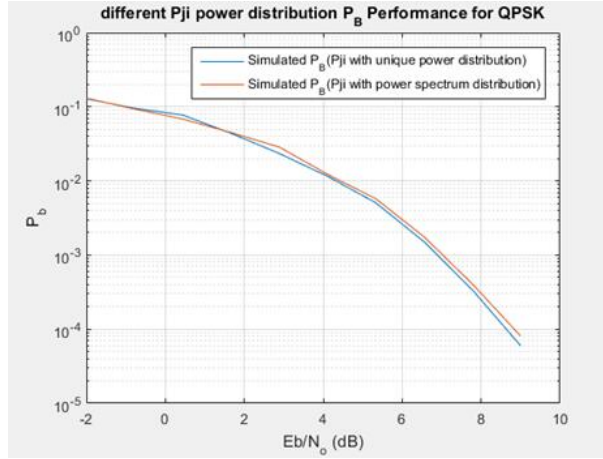


Figure 4.21: Pji distribution method, $P_j=0.01$, $\Delta fT=0.8$, $M=10$

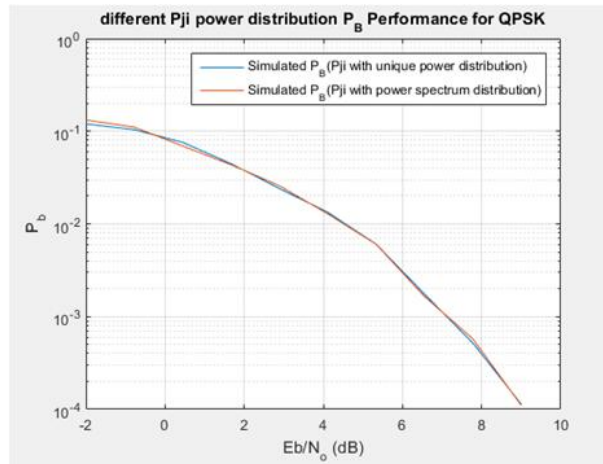


Figure 4.22: Pji distribution method, $P_j=0.01$, $\Delta fT=0.08$, $M=10$

4.5.2 Multi-tone Jamming VS Single-tone Jamming

For a clear review, as we mentioned in the experimental procedure, several sets of parameters for m-tone jamming has been estimated. Figure 4.23 is an example with the setting $M=10$, $P_j=0.1$, $\Delta fT=0.1$. For other related results will be showing in later demonstration.

From the graph, it is easily found that a clear blue line places above the red line, the blue line represents the M-tone jamming has a better performance than the reference single-tone jamming without frequency offset.

It indicates that the multi-tone jamming scheme could improve the jamming performance that affect the QPSK modulation.

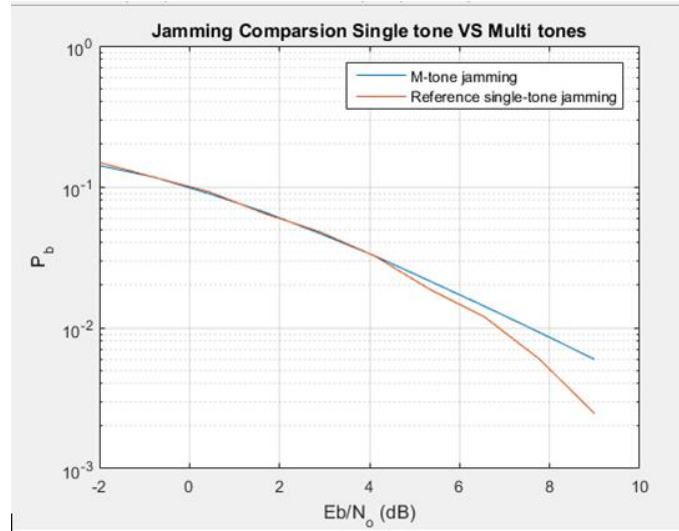


Figure 4.23: M-tone VS 1-tone, $P_j=0.01$, $\Delta T=0.1$, $M=10$

4.5.3 M number of tones

At the end of the section, we tried to fix the other parameters, leave M , find the relationship between M tones and P_b , see the effectiveness of multi tones jamming, the result shows in figure 4.24.

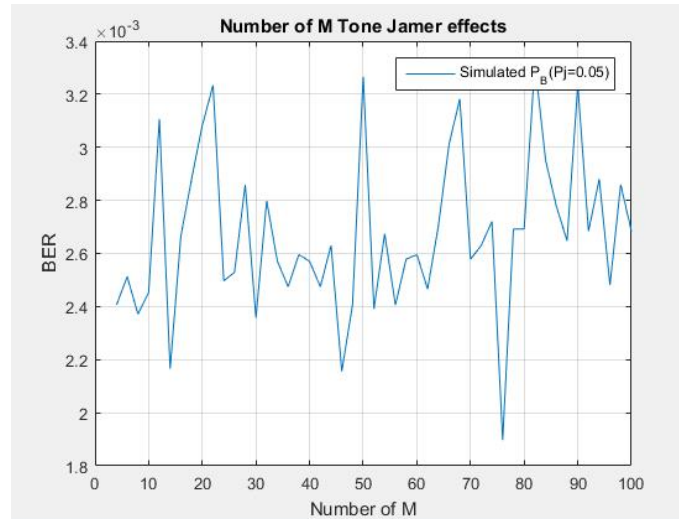


Figure 4.24: Influence with M tones

$M=4,8$ will give a large impact on QPSK signal modulation, continuous increasing M will also increase the performance a little. The fluctuation of the line due to the random progress during simulation, and it is normal since the P_b , the range of y-axis is located with very tiny numbers.

4.6 Code Implementation And Discussions

The following section will explain the problem during code writing, during the progress of writing code, there are some errors at the early stage. Firstly, sometimes running a code will drop into the infinite loop, which is the problem with stop condition, and this issue is easily confusing with some codes spending the relatively considerable time to execute. Therefore, as we suggest in the previous progress part, we can use a small range of parameters to estimate the result during coding. For example, set E_b/N_0 range from -2dB to 7dB, lower the error determination condition to $N=10$, meantime increase the P_j to a larger number 0.1 or 0.2 since a massive jamming power will produce more bit errors during modulation, then the error determination will be faster to reach the requirement. Also, M is setting with 4 to 10 to get a sufficient estimate. And additionally, the number of vectors is too many during programming, trace the result from the workspace, solve every problem especially the vector dimension matching, that is the most frequently happening problem during programming because the vector operation is complex and challenging to determine the problem quickly.

4.6.1 Code Performance

The computing complexity of the simulation is increasing during the development of coding. The following parameters show the effect the code running speed.

E_b/N_0 dB, especially after 9dB, the estimate P_b value is less, and it's hard to collect enough bit errors to terminate the code progress.

N , as mentioned, the time complexity for N is linear, a small N can be used for easier stage, and $N=1000$ gives a good estimate.

P_j , large P_j affect the communication channel; more bit errors could be collected, and N condition could be reached faster.

M , M -tone jamming will spend further time on simulation, the number of M will give an enormous impact on simulation speed, for every symbol estimated during modulation, the M tones calculate once and filled random phases. Most of the estimate will take several hours even one day, and sometimes the result may be dissatisfied; some different parameters will be changed and repeat the progress will take even a longer time.

4.7 Conclusion Of Different Jamming Methods

So far, we estimate all the results with single-tone jamming, single-tone jamming with frequency offset, two-tone jamming, m -tone jamming. All related parameters have been estimated and discussed; the result could be concluded as the following list:

1. P_j , total jamming power and P_j/P_i , the total jamming power to the total signal power ratio give the modulation communication channel a huge impact on the final performance; Increasing P_j could dramatically improve the jamming performance, and increase the BER(bit error rates), that could finally cause the decrease of the

entire communication system performance, decrease the probability that receiver could get the correct information.

2. Frequency offset Δf , ΔfT in the code, also affect the jamming effectiveness; In general, increase the value with ΔfT will cause the jamming tone placed away from central frequency, this could weaken the final jamming performance that affects the QPSK Modulation. However for m-tone jamming place the jamming tones and total jamming power around central frequency (0.08 - 0.1), could get a better performance over the single tone jamming placed at the central frequency. The reason of m-tone jamming could get a better result is with some certain conditions, the required phase angle value is triggered with the effect of m-tone jamming, these phases is affected in several times with large amount of random process; At the end, the final reconstructed signal is just over the next state of modulation determining condition.
3. In the current case with QPSK Simulation, for M-tone jamming, the appropriate number of m-tone jamming is setting with 8-10, there are some big difference between 2,4,8 respectively; Continually increase M would have a little effect on the final m-tone jamming performance; Although there are little difference for relatively large number of M-tone jamming, the scheme of how to place the multi tones will be more effect with some advanced modulation method, and multi-tone jamming will be used as the anti-anti-jamming scheme in the future development. This means the relatively large number of multi-tone jamming will be applied for against some advanced modulation method which already implemented some 'anti-jamming scheme.'

Finally, different tone jamming schemes have been estimated and discussed, the 10-tone jamming with frequency offset= 0.08 has the best performance compared with other jamming schemes in the same total jamming power to total signal power ratio. The final estimate multi-tone jamming is succeeded over the reference performance of single-tone jamming without frequency offset.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In conclusion, this project has been introduced all the background information behind the jamming signal and digital communication system, a robust understanding of digital communication system helps the implementation of programming the simulation code; The modulation and channel are two critical blocks that should put more concern into this project. The progress of QPSK Modulation is used in this project as the initial modulated signal. And the jamming channel is involved in the communication channel between modulation and demodulation. Different jamming methods ,including the one-tone jamming within central frequency, the one-tone jamming with frequency offset, two-tone jamming, and M-tone jamming is estimated by the performance with system Pb, the bit error rate; A higher bit error rate represents better performance towards the jamming method which is used. And moreover, the related parameters affect the jamming performance has also been estimated during simulation.

Therefore, for every jamming methods, P_j , the power of jamming plays a major role in jamming effectiveness, the jamming effectiveness will be increased as the large P_j interfere in the jamming channel.

For a certain type of jamming method, the secondary parameter that affects the jamming performance is the amount of jamming power that distributed closer to the central frequency; it includes the jamming frequency offset and the distribution method.

Also ,for multi-tone jamming, the number of M tones can further increase the jamming performance by distributed the individual jamming power around the central frequency of QPSK power spectrum.

5.2 Future Work

So far, the effectiveness of tone jamming that affects QPSK modulated signal has been estimated. However, there are still some works can be estimated. The future work will focus on the following points:

- Currently , only tone jamming is estimated, some other types such as the noise jamming, can be used for jamming QPSK signal, the jamming performance of partial band noise jamming could be estimated in the following works.
- In this project ,only QPSK Modulation , a common type of modulation is used as the original modulation method. There are many type of modulation methods , for example Amplitude shift keying(ASK),frequency shift keying(FSK),M-array ASK(MASK),M-Array-FSK(MFSK),Code shift keying(CSK), Continuous phase modulation(CPM),Trellis coded modulation(TCM). The different modulation methods also have their property, and the best jamming scheme matches the specific modulation method.
- Moreover, both modulation and jamming method has their advanced version for the anti-jamming or anti-anti-jamming purpose, for example, the digital system design Direct Sequence Spread Spectrum Systems(DSSS) is to prevent the interference of single-tone jamming. The multi-tone jamming, how effectively distribute jamming power over the DSSS is the task could estimate in the future. Some other anti-jam signal detection technique for instance Frequency-Hopping Spread Spectrum Signal Detection(FHSS), for FHSS,it could also be divided into Fast Frequency Hopping Systems(FFHSS) and Slow Frequency Hopping Systems(SFHSS); These are some advanced anti-jamming modulation that widely used in practical especially in wireless communications, need more research on the related topics.

Chapter 6

Abbreviations

ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
DF	Direction Finding
DMPSK	Differential M-ary Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum Systems
EA	Electronic Attack
EP	Electronic Protection
ES	Electronic Support
EW	Electronic Warfare
FFHSS	Fast Frequency-Hopping Spread Spectrum Signal Detection
FHSS	Frequency-Hopping Spread Spectrum Signal Detection
FM	Frequency Modulation
FSK	Frequency Shift Keying
MASK	M-ary Amplitude Shift Keying
MFSK	M-ary Frequency Shift Keying
MPSK	M-ary Phase Shift Keying
OOK	On-Off Keying
PSD	Power Spectrum Density
PSK	Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
SFHSS	Slow Frequency-Hopping Spread Spectrum Signal Detection

Appendix A

Matlab Coding

A.1 Overview

All the Matlab codes implementing during the entire project will be provided here for the reference of readers.

A.2 QPSK Sumulation

```
%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%First QPSK Simulation

clc
clear
%
-----

% Primary parameters
Eb_No_dB=[-2:1:11]; %Eb/No
blocklength=256; % The bit error rate is obtained in blocks of
    data
Nspb=16;
N=1000; %The number of bit errors for the error determination
%
-----

% Derived parameters
Eb_No=10.^(Eb_No_dB./10);
```

```

for iiEbNo=1:length(Eb_No);
    N_errors=0; % Counter for the number of bit errors
    N_bits=0; % Counter for the number of bits
    sigma_n=sqrt(Nspb/ (4*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength); % Generated random binary
            data
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
        for ii=1:2:blocklength % Generate QPSK
            phases
            a=data(ii);
            b=data(ii+1);
            if a==0 & b==0
                phi(ii/2+1/2)=pi/4;
            elseif a==1 & b==0
                phi(ii/2+1/2)=3*pi/4;
            elseif a==1 & b==1
                phi(ii/2+1/2)=5*pi/4;
            else
                phi(ii/2+1/2)=7*pi/4;
            end
        end
        % Generate noise vector
        n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
            blocklength/2));
        for ii=1:blocklength/2 % Generate signal vector for the
            block
            x((ii-1)*Nspb+1:(ii*Nspb))=exp(j*phi(ii)).*ones(1,
                Nspb);
        end
        y=x+n; %Additive Communications Channel(with jammer)
        iir=0; % Counter on the bit number
        for ii=1:blocklength/2 % Symbol detection

```

```

        u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
        v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
        iir=iir+1;
        %bit detection
        if u>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
        iir=iir+1;
        if v>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
    end
    errornumber=sum(mod((data+datar),2)); % Number of bit
        errors in block
    N_errors=N_errors+errornumber; % Total Number of bit
        errors
    if N_errors>N % Condition to terminate the simulation
        condition=1
    end
end
PB(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

PB_theory=qfunc(sqrt(2*Eb_No));
figure(1)
semilogy(Eb_No_dB,PB,Eb_No_dB,PB_theory)
grid
xlabel('Eb/No_(dB)')
ylabel('P_b')
title('P_B_for_Pj_difference_affect_QPSK')
legend('Simulated_P_B','Theoretical_P_B')

```

A.3 Single-Tone Jamming placed at centre frequency

```

%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%The influence of BER with jammer signal power

clc
clear
%
-----

% Primary parameters
Eb_No_dB=[-2:1:9]; %Eb/No
blocklength=256; % The bit error rate is obtained in blocks of
    data
Nspb=16;
N=1000; %The number of bit errors for the error determination
%
-----

% Derived parameters
Eb_No=10.^(Eb_No_dB./10);

for iiEbNo=1:length(Eb_No);
    N_errors=0; % Counter for the number of bit errors
    N_bits=0; % Counter for the number of bits
    sigma_n=sqrt(Nspb/(4*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength); % Generated random binary
            data
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
    end
end

```

```

for ii=1:2:blocklength % Generate QPSK
    phases
    a=data(ii);
    b=data(ii+1);
    if a==0 & b==0
        phi(ii/2+1/2)=pi/4;
    elseif a==1 & b==0
        phi(ii/2+1/2)=3*pi/4;
    elseif a==1 & b==1
        phi(ii/2+1/2)=5*pi/4;
    else
        phi(ii/2+1/2)=7*pi/4;
    end
end
% Generate noise vector
n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
    blocklength/2));
for ii=1:blocklength/2 % Generate signal vector for the
    block
        x((ii-1)*Nspb+1:(ii*Nspb))=exp(j*phi(ii)).*ones(1,
            Nspb);
        phi_j=pi/4; % Phase angle for the jammer signal
        Pj=0.01; % Power of jammer signal, in this loop, Pj=0.01
        j_a((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Pj)*exp(j*phi_j);
    %Generate jamming signal vector
end
y=x+n+j_a; %Additive Communications Channel(with jammer)
iir=0; % Counter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    %bit detection
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end

```

```

        end
    end
    errornumber=sum(mod((data+datar),2)); % Number of bit
        errors in block
    N_errors=N_errors+errornumber; % Total Number of bit
        errors
    if N_errors>N % Condition to terminate the simulation
        condition=1
    end
end
PB(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

%Below is a similar loop, the power of jammer signal is 0.05
for iiEbNo=1:length(Eb_No);
    N_errors=0;
    N_bits=0;
    sigma_n=sqrt(Nspb/ (4*Eb_No(iiEbNo)));
    condition=0;
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1, blocklength);
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
    end
    for ii=1:2:blocklength
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
end
end

```

```

n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
    blocklength/2));

for ii=1:blocklength/2
    x((ii-1)*Nspb+1:(ii*Nspb))=exp(j*phi(ii)).*ones(1,
        Nspb);

    phi_j=pi/4;
    Pj=0.05; % Power of jammer signal, in this loop, Pj=0.05
    ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Pj)*exp(j*phi_j);

end

y=x+n+ja;
iir=0;
for ii=1:blocklength/2
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2));
N_errors=N_errors+errornumber;
if N_errors>N
    condition=1
end
end
PB2(iiEbNo)=N_errors/N_bits
end

%Below is a similar loop, the power of jammer signal is 0.5
for iiEbNo=1:length(Eb_No);

```

```

N_errors=0;
N_bits=0;
sigma_n=sqrt(Nspb/ (4*Eb_No(iiEbNo)));
condition=0;
while condition==0
    N_bits=N_bits+blocklength;
    data=randn(1,blocklength);
    for ii=1:blocklength
        if data(ii)>0
            data(ii)=1;
        else
            data(ii)=0;
        end
    end
    for ii=1:2:blocklength
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
    n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
        blocklength/2));

    for ii=1:blocklength/2
        x((ii-1)*Nspb+1:(ii*Nspb))=exp(j*phi(ii)).*ones(1,
            Nspb);

        phij=pi/4;
        Pj=0.5; % Power of jammer signal, in this loop, Pj=0.5
        ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Pj)*exp(j*phij);
    end

    y=x+n+ja;

```



```

    iir=0;
    for ii=1:blocklength/2
        u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
        v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
        iir=iir+1;
        if u>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
        iir=iir+1;
        if v>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
    end
    errornumber=sum(mod((data+datar),2));
    N_errors=N_errors+errornumber;
    if N_errors>N
        condition=1
    end
end
PB3(iiEbNo)=N_errors/N_bits
end

PB_theory=qfunc(sqrt(2*Eb_No));
figure(1)
semilogy(Eb_No_dB,PB3,Eb_No_dB,PB2,Eb_No_dB,PB,Eb_No_dB,
    PB_theory)
%Plot 4 graph ,compare with different Pj
grid
xlabel('Eb/No (dB)')
ylabel('P_b')
title('P_B for P_j difference affect QPSK')
legend('Simulated_P_B(Pj=0.5)', 'Simulated_P_B(Pj=0.05)', '
    Simulated_P_B(Pj=0.01)', 'Theoretical_P_B(Pj=0)')

```

A.4 Single-Tone Jamming with Frequency Offset

```

%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%Jammer with offset frequency

clc
clear
%
-----

% Primary parameters
Eb_No_dB_l=-2;
Eb_No_dB_h=9;
N_points=10;
delta_Eb_No=(Eb_No_dB_h-Eb_No_dB_l)/(N_points-1);
Eb_No_dB=[Eb_No_dB_l:delta_Eb_No:Eb_No_dB_h]; %dB range from -2
    to 9
N=1000; %The number of bit errors for the error determination
blocklength=256; % The bit error rate is obtained in blocks of
    data
Nspb=16;
nn=[1:1:Nspb]; %calculate the jammer phase angle

Ps=1; % Power of x signal
Pj=0.1; % Power of jammer signal
deltafT=0.1; %offset frequency delta f * T

%
-----

% Derived parameters

Eb_No=10.^(Eb_No_dB./10);

for iiEbNo=1:length(Eb_No);
    N_errors=0; % Couter for the number of bit errors
    N_bits=0; % Couter for the number of bits
    sigma_n=sqrt(Ps*Nspb/(2*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained

```

```

while condition==0
    N_bits=N_bits+blocklength;
    data=randn(1,blocklength); % Generated random binary
    data
    for ii=1:blocklength
        if data(ii)>0
            data(ii)=1;
        else
            data(ii)=0;
        end
    end
    for ii=1:2:blocklength % Generate QPSK phases
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
    % Generate noise vector
    n=sigma_n.*(randn(1,Nspb*blocklength/2)+ ...
        j*randn(1,Nspb*blocklength/2));

    tsi=2.*pi.*rand % initial phase state of jammer
    theta=2.*pi.*rand;
    for ii=1:blocklength/2
        x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2.*Ps) ...
            * exp(j*phi(ii)).*ones(1,Nspb);

        %phase angle of jammer

        ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2.*Pj).*exp(j*theta)
        ...
        .*exp(j.*2.*pi.*deltafT.*nn(1:Nspb)./Nspb).*exp(j.*
            tsi);

```

```

    tsi=tsi+2.*pi.*deltafT; %next block initial state
    additive
end

y=x+n+ja; %Additive Communications Channel(with jammer)
iir=0; % Counter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2)); % Number of bit
    errors in block
N_errors=N_errors+errornumber; % Total Number of bit
    errors
if N_errors>N % Condition to terminate the simulation
    condition=1
end
end
PB(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

%Below is a similar loop, without frequency offset
for iiEbNo=1:length(Eb_No);
    N_errors=0;
    N_bits=0;
    sigma_n=sqrt(Ps*Nspb/(2*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0;
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength);
    end
end

```

```

    for ii=1:blocklength
        if data(ii)>0
            data(ii)=1;
        else
            data(ii)=0;
        end
    end
    for ii=1:2:blocklength
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
    end

    n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
        blocklength/2));
    theta=2.*pi.*rand;
    for ii=1:blocklength/2
        x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2*Ps)*exp(j*phi(ii))
            .* ones(1,Nspb);

        ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Pj)*exp(j*theta);

    end
    y=x+n+ja;
    iir=0;
    for ii=1:blocklength/2
        u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
        v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
        iir=iir+1;
        if u>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
        iir=iir+1;
    end

```

```

        if v>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
    end
    errornumber=sum(mod((data+datar),2));
    N_errors=N_errors+errornumber;
    if N_errors>N
        condition=1
    end
end
PB2(iiEbNo)=N_errors/N_bits
end

PB_theory=qfunc(sqrt(2*Eb_No));
figure(1)
semilogy(Eb_No_dB,PB2,Eb_No_dB,PB)
grid
xlabel('Eb/No (dB)')
ylabel('P_b')
title('delta f influence for P_B Performance for QPSK')
legend('Simulated P_B', 'Simulated P_B with delta f * nT')

```

A.5 Two-Tone Jamming

```

%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%Two Jamming Tones + frequency offset

clc
clear
%
-----

% Primary parameters
Eb_No_dB_l=-2;
Eb_No_dB_h=9;
N_points=10;
delta_Eb_No=(Eb_No_dB_h-Eb_No_dB_l)/(N_points-1);
Eb_No_dB=[Eb_No_dB_l:delta_Eb_No:Eb_No_dB_h]; %dB range from -2
to 9
N=1000; %The number of bit errors for the error determination
blocklength=256; % The bit error rate is obtained in blocks of
data
Nspb=16;
nn=[1:1:Nspb]; %calculate the jammer phase angle
Ps=1; % Power of x signal
Pj=0.2; % Power of jammer signal
deltafT1=0.1; %offset frequency of jamming tone 1
deltafT2=0.1; %offset frequency of jamming tone 2

%
-----

% Derived parameters

Eb_No=10.^(Eb_No_dB./10);

for iiEbNo=1:length(Eb_No);
    N_errors=0; % Counter for the number of bit errors
    N_bits=0; % Counter for the number of bits
    sigma_n=sqrt(Ps*Nspb/(2*Eb_No(iiEbNo))); % Noise standard
    deviation
    condition=0; % N bit errors not yet obtained

```

```

while condition==0
    N_bits=N_bits+blocklength;
    data=randn(1,blocklength); % Generated random binary
    data
    for ii=1:blocklength
        if data(ii)>0
            data(ii)=1;
        else
            data(ii)=0;
        end
    end
    for ii=1:2:blocklength % Generate QPSK phases
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
    end
    % Generate noise vector
    n=sigma_n.*(randn(1,Nspb*blocklength/2)+ ...
        j*randn(1,Nspb*blocklength/2));

    tsi1=2.*pi.*rand % initial phase state of jammer
    tsi2=2.*pi.*rand
    theta1=2.*pi.*rand;
    theta2=2.*pi.*rand;
    for ii=1:blocklength/2
        x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2.*Ps) ...
            * exp(j*phi(ii)).*ones(1,Nspb);

        ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2.*Pj./2).*exp(j*theta1
        ) ...
            .*exp(j.*2.*pi.*deltafT1.*nn(1:Nspb)./Nspb).*exp(j.*
            tsi1)+ ...
            sqrt(2.*Pj./2).*exp(j*theta2) ...

```



```

        .*exp(j.*2.*pi.*deltafT2.*nn(1:Nspb)./Nspb).*exp(j.*
            tsi2);
    tsi1=tsi1+2.*pi.*deltafT1; %next block initial state
        additive
    tsi2=tsi2+2.*pi.*deltafT2;
end

y=x+n+ja; %Additive Communications Channel(with jammer)
iir=0; % Couter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2)); % Number of bit
    errors in block
N_errors=N_errors+errornumber; % Total Number of bit
    errors
if N_errors>N % Condition to terminate the simulation
    condition=1
end
end
PB(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

%Below is a similar loop, single tone without frequency offset
for iiEbNo=1:length(Eb_No);
    N_errors=0;
    N_bits=0;
    sigma_n=sqrt(Ps*Nspb/(2*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0;

```

```

while condition==0
    N_bits=N_bits+blocklength;
    data=randn(1,blocklength);
    for ii=1:blocklength
        if data(ii)>0
            data(ii)=1;
        else
            data(ii)=0;
        end
    end
    for ii=1:2:blocklength
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
end

n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
    blocklength/2));
theta=2.*pi.*rand;
for ii=1:blocklength/2
    x((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Ps)*exp(j*phi(ii))
        .*ones(1,Nspb);
    ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Pj)*exp(j*theta);
end

y=x+n+ja;
iir=0;
for ii=1:blocklength/2
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end

```

```

        end
        iir=iir+1;
        if v>=0
            datar(iir)=0;
        else
            datar(iir)=1;
        end
    end
    errornumber=sum(mod((data+datar),2));
    N_errors=N_errors+errornumber;
    if N_errors>N
        condition=1
    end
end
PB2(iiEbNo)=N_errors/N_bits
end

figure(1)
semilogy(Eb_No_dB,PB2,Eb_No_dB,PB)
grid
xlabel('Eb/No (dB)')
ylabel('P_b')
title('delta_f influence for P_B Performance for QPSK')
legend('Simulated P_B(single tone jamming)', 'Simulated P_B(Two tone jamming)')

```

A.6 Jammer Pji with different distribution methods

```

%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%Jammer Pji with M power distribution vs sinc^2 power
    distribution

clc
clear
%
-----

% Primary parameters
Eb_No_dB_l=-2;
Eb_No_dB_h=9;
N_points=10;
delta_Eb_No=(Eb_No_dB_h-Eb_No_dB_l)/(N_points-1);
Eb_No_dB=[Eb_No_dB_l:delta_Eb_No:Eb_No_dB_h];

N=200; %The number of bit errors for the error determination
blocklength=256; % The bit error rate is obtained in blocks of
    data
Nspb=16;
nn=[1:1:Nspb];
Ps=1; % Power of x signal
Pj=0.01; % Power of jammer signal

M=10;
deltafT_h=0.08;
deltafT_l=-deltafT_h;
delta_deltafT=deltafT_h/(M-1)*2;
deltafT=[deltafT_l:delta_deltafT:deltafT_h];

%
-----

% Derived parameters

Eb_No=10.^(Eb_No_dB./10);

```

```

%Below is the first loop, with unique power spectrum
distribution
for iiEbNo=1:length(Eb_No);
    N_errors=0; % Counter for the number of bit errors
    N_bits=0; % Counter for the number of bits
    sigma_n=sqrt(Ps*Nspb/ (2*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength); % Generated random binary
            data
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
        for ii=1:2:blocklength % Generate QPSK phases
            a=data(ii);
            b=data(ii+1);
            if a==0 & b==0
                phi(ii/2+1/2)=pi/4;
            elseif a==1 & b==0
                phi(ii/2+1/2)=3*pi/4;
            elseif a==1 & b==1
                phi(ii/2+1/2)=5*pi/4;
            else
                phi(ii/2+1/2)=7*pi/4;
            end
        end
        % Generate noise vector
        n=sigma_n.*(randn(1,Nspb*blocklength/2)+ ...
            j*randn(1,Nspb*blocklength/2));

            for m=1:M
                tsi(m)=2.*pi.*rand % initial phase state of jammer
                theta(m)=2.*pi.*rand
            end

        for ii=1:blocklength/2
            x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2.*Ps) ...

```

```

        * exp(j*phi(ii)).*ones(1,Nspb);

S=zeros(1,Nspb);
for m=1:M
    mm=sqrt(2.*Pj./M).*exp(j*theta(m))...
        .*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)./Nspb).*exp(j
        .*tsi(m))
    S=mm+S;
    ja((ii-1)*Nspb+1:(ii*Nspb))=S;
    tsi(m)=tsi(m)+2.*pi.*deltafT(m); %next block initial
        state additive
end

end

y=x+n+ja; %Additive Communications Channel(with jammer)
iir=0; % Couter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2)); % Number of bit
    errors in block
N_errors=N_errors+errornumber; % Total Number of bit
    errors
if N_errors>N % Condition to terminate the simulation
    condition=1
end
end
end

```

```

PB1(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

%Below is a similar loop, with power spectrum distribution
for iiEbNo=1:length(Eb_No);
    N_errors=0; % Counter for the number of bit errors
    N_bits=0; % Counter for the number of bits
    sigma_n=sqrt(Ps*Nspb/ (2*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength); % Generated random binary
            data
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
        for ii=1:2:blocklength % Generate QPSK phases
            a=data(ii);
            b=data(ii+1);
            if a==0 & b==0
                phi(ii/2+1/2)=pi/4;
            elseif a==1 & b==0
                phi(ii/2+1/2)=3*pi/4;
            elseif a==1 & b==1
                phi(ii/2+1/2)=5*pi/4;
            else
                phi(ii/2+1/2)=7*pi/4;
            end
        end
        end
    % Generate noise vector
    n=sigma_n.*(randn(1,Nspb*blocklength/2)+ ...
        j*randn(1,Nspb*blocklength/2));

        for m=1:M
            tsi(m)=2.*pi.*rand % initial phase state of jammer
            theta(m)=2.*pi.*rand
        end

```

```

for ii=1:blocklength/2
    x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2.*Ps) ...
        * exp(j*phi(ii)).*ones(1,Nspb);

S=zeros(1,Nspb);
SINC=0
for m=1:M
    SINC=SINC+(sinc(deltafT(m)))^2
end
for m=1:M
    nm=sqrt(2.*(sinc(deltafT(m))^2)/SINC.*Pj).*exp(j*
        theta(m)) ...
        .*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)./Nspb).*exp(j
        .*tsi(m))
    S=nm+S;
    ja((ii-1)*Nspb+1:(ii*Nspb))=S;
    tsi(m)=tsi(m)+2.*pi.*deltafT(m); %next block initial
        state additive
end

end

y=x+n+ja; %Additive Communications Channel(with jammer)
iir=0; % Couter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
end

```

```

        errornumber=sum(mod((data+datar),2)); % Number of bit
            errors in block
        N_errors=N_errors+errornumber; % Total Number of bit
            errors
        if N_errors>N % Condition to terminate the simulation
            condition=1
        end
    end
    PB2(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

PB_theory=qfunc(sqrt(2*Eb_No));
figure(1)
semilogy(Eb_No_dB,PB1,Eb_No_dB,PB2)
grid
xlabel('Eb/No_(dB)')
ylabel('P_b')
title('different_Pji_power_distribution_P_B_Performance_for_QPSK')
legend('Simulated_P_B(Pji_with_unique_power_distribution)',
        'Simulated_P_B(Pji_with_power_spectrum_distribution)')

```

A.7 M-tone jamming VS reference single-tone jamming

```
%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%M-tone jamming VS reference single-tone jamming

clc
clear
%
-----

% Primary parameters
Eb_No_dB_l=-2;
Eb_No_dB_h=9;
N_points=10;
delta_Eb_No=(Eb_No_dB_h-Eb_No_dB_l)/(N_points-1);
Eb_No_dB=[Eb_No_dB_l:delta_Eb_No:Eb_No_dB_h];

N=1000; %The number of bit errors for the error determination
blocklength=256; % The bit error rate is obtained in blocks of
    data
Nspb=16; % Setting 1-16 columns to the jammer
nn=[1:1:Nspb];
Ps=1; % Power of x signal
Pj=0.1; % Power of jammer signal, in this loop, Pj=0.1

M=10;
deltafT_h=0.1;
deltafT_l=-deltafT_h;
delta_deltafT=deltafT_h/(M-1)*2;
deltafT=[deltafT_l:delta_deltafT:deltafT_h];

%
-----

% Derived parameters

Eb_No=10.^(Eb_No_dB./10);
```

```

for iiEbNo=1:length(Eb.No);
    N_errors=0; % Counter for the number of bit errors
    N_bits=0; % Counter for the number of bits
    sigma_n=sqrt(Ps*Nspb/ (2*Eb.No(iiEbNo))); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength); % Generated random binary
            data
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
        for ii=1:2:blocklength % Generate QPSK phases
            a=data(ii);
            b=data(ii+1);
            if a==0 & b==0
                phi(ii/2+1/2)=pi/4;
            elseif a==1 & b==0
                phi(ii/2+1/2)=3*pi/4;
            elseif a==1 & b==1
                phi(ii/2+1/2)=5*pi/4;
            else
                phi(ii/2+1/2)=7*pi/4;
            end
        end
        % Generate noise vector
        n=sigma_n.*(randn(1,Nspb*blocklength/2)+ ...
            j*randn(1,Nspb*blocklength/2));

        for m=1:M
            tsi(m)=2.*pi.*rand % initial phase state of jammer
            theta(m)=2.*pi.*rand
        end

        for ii=1:blocklength/2
            x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2.*Ps) ...
                * exp(j*phi(ii)).*ones(1,Nspb);

```

```

S=zeros(1,Nspb);
SINC=0
for m=1:M
    SINC=SINC+(sinc(deltafT(m)))^2
end
for m=1:M
    mm=sqrt(2.*(sinc(deltafT(m))^2)/SINC.*Pj).*exp(j*
        theta(m))...
    .*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)/Nspb).*exp(j
        .*tsi(m))
    S=mm+S;
    ja((ii-1)*Nspb+1:(ii*Nspb))=S;
    tsi(m)=tsi(m)+2.*pi.*deltafT(m); %next block initial
        state additive
end

end

y=x+n+ja; %Additive Communications Channel(with jammer)
iir=0; % Couter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2)); % Number of bit
    errors in block
N_errors=N_errors+errornumber; % Total Number of bit
    errors
if N_errors>N % Condition to terminate the simulation

```

```

        condition=1
    end
end
PB(iiEbNo)=N_errors/N_bits % bit error rate estimate
end

%Below is a similar loop, the reference single-tone jamming
for iiEbNo=1:length(Eb_No);
    N_errors=0;
    N_bits=0;
    sigma_n=sqrt(Ps*Nspb/ (2*Eb_No(iiEbNo))); % Noise standard
        deviation
    condition=0;
    while condition==0
        N_bits=N_bits+blocklength;
        data=randn(1,blocklength);
        for ii=1:blocklength
            if data(ii)>0
                data(ii)=1;
            else
                data(ii)=0;
            end
        end
        for ii=1:2:blocklength
            a=data(ii);
            b=data(ii+1);
            if a==0 & b==0
                phi(ii/2+1/2)=pi/4;
            elseif a==1 & b==0
                phi(ii/2+1/2)=3*pi/4;
            elseif a==1 & b==1
                phi(ii/2+1/2)=5*pi/4;
            else
                phi(ii/2+1/2)=7*pi/4;
            end
        end
        n=sigma_n.*(randn(1,Nspb*blocklength/2)+j*randn(1,Nspb*
            blocklength/2));
        theta=2.*pi.*rand;
        for ii=1:blocklength/2
            x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2*Ps)*exp(j*phi(ii)
                ).*ones(1,Nspb);
        end
    end
end

```

```

    ja((ii-1)*Nspb+1:(ii*Nspb))=sqrt(2*Pj)*exp(j*theta);

end
y=x+n+ja;
iir=0;
for ii=1:blocklength/2
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2));
N_errors=N_errors+errornumber;
if N_errors>N
    condition=1
end
end
PB2(iiEbNo)=N_errors/N_bits
end

figure(1)
semilogy(Eb_No_dB,PB,Eb_No_dB,PB2)
grid
xlabel('Eb/N_o_(dB)')
ylabel('P_b')
title('Jamming_Comparsion_Single_tone_VS_Multi_tones')
legend('M-tone_jamming','Reference_single-tone_jamming')

```

A.8 M-tone jamming influence with the number of M

```

%A/Prof Sam Reisenfeld
%QPSK Simulation over an AWGN chanel
%ELEC426, Autumn, 2012
%Modified by Enbo Yang ,QPSK Jammers
%M VS Eb/No

clc
clear
%
-----

% Primary parameters
Eb_No_dB=8;
M=[4,10,20,30,40,50,100,200,500,1000];
N=100;%The number of bit errors for the error determination
blocklength=256; % The bit error rate is obtained in blocks of
    data
Nspb=16; % Setting 1-16 columns to the jammer
nn=[1:1:Nspb];
Ps=1;% Power of x signal
Pj=0.05;% Power of jammer signal, in this loop, Pj=0.05
deltafT_h=0.08;

%
-----

% Derived parameters
Eb_No=10.^(Eb_No_dB./10);

for iM=1:length(M);
    deltafT_l=-deltafT_h;
    delta_deltafT=deltafT_h/(M(iM)-1)*2;
    deltafT=[deltafT_l:delta_deltafT:deltafT_h];

    N_errors=0; % Couter for the number of bit errors
    N_bits=0; % Couter for the number of bits
    sigma_n=sqrt(Ps*Nspb/(2*Eb_No)); % Noise standard
        deviation
    condition=0; % N bit errors not yet obtained

```

```

while condition==0
    N_bits=N_bits+blocklength;
    data=randn(1,blocklength); % Generated random binary
    data
    for ii=1:blocklength
        if data(ii)>0
            data(ii)=1;
        else
            data(ii)=0;
        end
    end
    end
    for ii=1:2:blocklength % Generate QPSK phases
        a=data(ii);
        b=data(ii+1);
        if a==0 & b==0
            phi(ii/2+1/2)=pi/4;
        elseif a==1 & b==0
            phi(ii/2+1/2)=3*pi/4;
        elseif a==1 & b==1
            phi(ii/2+1/2)=5*pi/4;
        else
            phi(ii/2+1/2)=7*pi/4;
        end
    end
    end
    % Generate noise vector
    n=sigma_n.*(randn(1,Nspb*blocklength/2)+ ...
        j*randn(1,Nspb*blocklength/2));

        for m=1:M(iM)
            tsi(m)=2.*pi.*rand % initial phase state of jammer
            theta(m)=2.*pi.*rand
        end

    for ii=1:blocklength/2
        x((ii-1)*Nspb+1:(ii*Nspb))= sqrt(2.*Ps) ...
            * exp(j*phi(ii)).*ones(1,Nspb);

    S=zeros(1,Nspb);
    SINC=0
    for m=1:M(iM)
        SINC=SINC+(sinc(deltafT(m)))^2
    end

```



```

    for m=1:M(iM)
        mm=sqrt(2.*(sinc(deltafT(m))^2)./SINC.*Pj).*exp(j*
            theta(m))...
        .*exp(j.*2.*pi.*deltafT(m).*nn(1:Nspb)./Nspb).*exp(j
            .*tsi(m))
        S=mm+S;
        ja((ii-1)*Nspb+1:(ii*Nspb))=S;
        tsi(m)=tsi(m)+2.*pi.*deltafT(m); %next block initial
            state additive
    end

end

y=x+n+ja; %Additive Communications Channel(with jammer)
iir=0; % Couter on the bit number
for ii=1:blocklength/2 % Symbol detection
    u=real(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    v=imag(sum(y((ii-1)*Nspb+1:(ii*Nspb))));
    iir=iir+1;
    if u>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
    iir=iir+1;
    if v>=0
        datar(iir)=0;
    else
        datar(iir)=1;
    end
end
errornumber=sum(mod((data+datar),2)); % Number of bit
    errors in block
N_errors=N_errors+errornumber; % Total Number of bit
    errors
if N_errors>N % Condition to terminate the simulation
    condition=1
end

end
PB(iM)=N_errors/N_bits % bit error rate estimate
end

```

```
figure(1)
plot(M,PB)
grid
xlabel('Number of  $M$ ')
ylabel('BER')
title('Number of  $M$  Tone Jammer effects')
legend('Simulated  $P_B(P_j=0.05)$ ')
```

Appendix B

Consulation Meetings Attendance Form

Consultation Meetings Attendance Form

Week	Date	Comments (if applicable)	Student's Signature	Supervisor's Signature
2	10.03.16	Simulation	Enbo	Sam Rasmussen
4	24.03.16	Jamming	Enbo	Sam Rasmussen
7	29.04.16	Matlab Simulation	Enbo	Sam Rasmussen
8	03.05.16	Matlab Simulation	Enbo	Sam Rasmussen
9	11.05.16	Jamming Theory and Matlab Simulation	Enbo	Sam Rasmussen
10	18.05.16	Simulation Results Report	Enbo	Sam Rasmussen
10	20.05.16	Simulation Results Report	Enbo	Sam Rasmussen
11	26.05.16	Report	Enbo	Sam Rasmussen
12			Enbo	

Bibliography

- [1] R. Amuru, S.; Buehrer, “Optimal jamming against digital modulation.”
- [2] —, “Optimal jamming against digital modulation,” 2015.
- [3] —, “Optimal jamming against digital modulation,” 2015.
- [4] R. Gallager, *Principles of Digital Communication*, 2008.
- [5] Y. Z. J. Kou, “Adaptive multi-tone jamming suppression for dsss communications based on compressive sensing,” 2015.
- [6] G. K. I. K. C. Lau, “Theory and application of software defined electronics: Design concepts for the next generation of telecommunications and measurement systems,” 2012.
- [7] S.-G. Li, Juan; Hggman, “Performance of ieee802.16-2004 based system in jamming environment and its improvement with link adaptation,” 2006.
- [8] MarketsandMarkets, “Electronic warfare market by category (electronic support, electronic attack, electronic protection), platform (airborne, naval, land, unmanned), product, technology, portable, and region - global forecast to 2021,” 2016.
- [9] T. MathWorks. colon, :. [Online]. Available: <http://au.mathworks.com/help/matlab/ref/colon.html>
- [10] —. (2016) Matlab official website. [Online]. Available: <http://au.mathworks.com/products/matlab/>
- [11] —. (2016) ones. [Online]. Available: <http://au.mathworks.com/help/matlab/ref/ones.html?searchHighlight=ones>
- [12] —. (2016) rand. [Online]. Available: <http://au.mathworks.com/help/matlab/ref/rand.html?searchHighlight=rand>
- [13] —. (2016) randn. [Online]. Available: <http://au.mathworks.com/help/matlab/ref/randn.html?searchHighlight=randn>

- [14] ——. (2016) zeros. [Online]. Available: <http://au.mathworks.com/help/matlab/ref/zeros.html?searchHighlight=zeros>
- [15] W. C. B. s. A. A. Memon, Tayab D.; Ghangro, “Quadrature phase shift keying modulator & demodulator for wireless modem.”
- [16] C. of the Joint Chiefs of Staff (CJCS). (2007) ”joint publication 3-13.1 electronic warfare. [Online]. Available: <https://fas.org/irp/doddir/dod/jp3-13-1.pdf>
- [17] A. S. Patankar. (2009) Tutorial i : Basic elements of digital communication system. [Online]. Available: <https://wirelesscafe.wordpress.com/2009/07/10/tutorial-i-basic-elements-of-digital-communication-system/>
- [18] R. Poisel, *Modern Communications Jamming Principles and Techniques*, 2011.
- [19] C. Sen, “Digital communications jamming,” 2000.
- [20] TriT.Ha, *Theory and Design of Digital Communication Systems*, 2011.
- [21] T. Ueda, H.; Tsuboi, “A sampling theorem for periodic functions with no minus frequency component and its application,” 2013.
- [22] M. Viswanathan. (2010) Qpsk modulation and demodulation. [Online]. Available: <http://www.gaussianwaves.com/2010/10/qpsk-modulation-and-demodulation-2/>
- [23] R. Viswanathan, “On the autocorrelation of complex envelope of white noise.”
- [24] H. H. Wei, “A study of gps jamming and anti-jamming,” 2009.
- [25] R. W. World. (2012) Qpsk modulation-quadrature phase shift keying modulation. [Online]. Available: <http://www.rfwireless-world.com/Terminology/QPSK.html>
- [26] Q. L. M. L. X. K. N. Zhao, “Disrupting mimo communications with optimal jamming signal design,” 2015.

PROFESSIONAL AUTHORITY FORM

ALL FIELDS MUST BE COMPLETED BY A REGISTERED HEALTH PROFESSIONAL
OR PROFESSIONAL WITHIN MACQUARIE UNIVERSITY CAMPUS WELLBEING AND SUPPORT SERVICES

1. INSTRUCTIONS TO THE PROFESSIONAL AUTHORITY PROVIDING DOCUMENTATION

We appreciate your help in providing information regarding the student's health condition(s). The information that you provide will enable the University to determine the impact of the impairment on the student's ability to meet academic assessment requirements. Within the limits of confidentiality, this form and/or certificate, must describe the nature and impact of the student's problem so that an assessment of the possible effects on academic performance can be made.

2. PERSONAL DETAILS OF STUDENT

Student number:

Family Name:

Yang

Other Names:

Enbo

3. CONSULTATION

Date of Consultation:

Duration of condition

From:

19-5-16

To:

23-5-16

NOTE: For chronic health conditions, only complete this form if there has been a considerable and unpredictable exacerbation of symptoms that have impacted the student's academic functioning.

Nature of condition: Please provide a plain English description of any restrictions on the student's academic functioning (eg reading, writing, learning, memory, concentration etc) as a result of the health condition(s); details of the medical diagnosis are NOT required:

Respiratory tract infection which may impact on concentration

Impact of condition: Please indicate your evaluation of the likely impact of the health condition(s) on the student's ability to attend class, learn, retain or complete assessment requirements by marking the scale below.

1
Able to study

2

3
Limited capacity to study

4

5
Unable to study

Signature of Professional Authority:

Name and Title:

DR EMILY HO (MBBS)

Provider or Registration Number:

Phone Number:

DR EMILY HO
PROV. NO 224396AX
165 PACIFIC HWY.
HORNSBY 204776000

The University collects the personal information from the student identified above to enable their request for special consideration to be considered. Personal information held by the University is subject to the Privacy and Personal Information Protection Act 1998. Where the information collected includes health information as defined under the Health Records and Information Privacy Act 2002, references to personal information in this notice will be taken to also apply to health information. The University may disclose the personal information provided on this form to other University staff involved in the processing and assessment of the request for special consideration. The University will not otherwise disclose the information unless it has the student's consent or such disclosure is permitted or required by law. The provision of personal information is voluntary, but if a student does not provide the information requested, the University may be unable to process their request. Students who wish to access or inquire about the handling of their personal information may do so by contacting the University Privacy Officer by email at privacyofficer@mq.edu.au.

INSTRUCTIONS FOR STUDENTS: THIS PAF NEEDS TO BE UPLOADED TO ASK.MQ.EDU.AU AS PART OF YOUR APPLICATION



MACQUARIE
University

PROFESSIONAL AUTHORITY FORM

ALL FIELDS MUST BE COMPLETED BY A REGISTERED HEALTH PROFESSIONAL
OR PROFESSIONAL WITHIN MACQUARIE UNIVERSITY CAMPUS WELLBEING AND SUPPORT SERVICES

1. INSTRUCTIONS TO THE PROFESSIONAL AUTHORITY PROVIDING DOCUMENTATION

We appreciate your help in providing information regarding the student's health condition(s). The information that you provide will enable the University to determine the impact of the impairment on the student's ability to meet academic assessment requirements. Within the limits of confidentiality, this form and/or certificate, must describe the nature and impact of the student's problem so that an assessment of the possible effects on academic performance can be made.

2. PERSONAL DETAILS OF STUDENT

Student number: 42839734

Family Name: Yang

Other Names: Enbo

3. CONSULTATION

Date of Consultation:

6 June, 2016

Duration of condition

From: 2-6-16 To: 6-6-16

NOTE: For chronic health conditions, only complete this form if there has been a considerable and unpredictable exacerbation of symptoms that have impacted the student's academic functioning.

Nature of condition: Please provide a plain English description of any restrictions on the student's academic functioning (eg reading, writing, learning, memory, concentration etc) as a result of the health condition(s); details of the medical diagnosis are NOT required:

upper respiratory infection affecting concentration

Impact of condition: Please indicate your evaluation of the likely impact of the health condition(s) on the student's ability to attend class, learn, retain or complete assessment requirements by marking the scale below.

1

Able to study

2

3

Limited capacity to study

4

5

Unable to study

Signature of Professional Authority:

Name and Title:

Provider or Registration Number:

Phone Number:

94776000

DR EMILY HO
PROV. NO 224396AX
165 PACIFIC HWY
HORNSBY 94776000
Stamp of Professional Authority

The University collects the personal information from the student identified above to enable their request for special consideration to be considered. Personal information held by the University is subject to the Privacy and Personal Information Protection Act 1998. Where the information collected includes health information as defined under the Health Records and Information Privacy Act 2002, references to personal information in this notice will be taken to also apply to health information. The University may disclose the personal information provided on this form to other University staff involved in the processing and assessment of the request for special consideration. The University will not otherwise disclose the information unless it has the student's consent or such disclosure is permitted or required by law. The provision of personal information is voluntary, but if a student does not provide the information requested, the University may be unable to process their request. Students who wish to access or inquire about the handling of their personal information may do so by contacting the University Privacy Officer by email at privacyofficer@mq.edu.au.

INSTRUCTIONS FOR STUDENTS: THIS PAF NEEDS TO BE UPLOADED TO ASK.MQ.EDU.AU AS PART OF YOUR APPLICATION