DEPARTMENT OF Electronic & Computer Engineering

Module Code: EE2626
Module Name: Principles of Communications
Title of Assessment: Experimental Report

School of Engineering & Design
Electronic and Computer Engineering

Principles of Communications

Experimental Report

Brunel Student ID: 2161047

Name: Xukang Liu

1. Introduction

The objective of this experiment is to comprehend the intricacies involved in analog and digital modulation as well as demodulation processes. It aims to validate the pivotal elements influencing coherent detection performance. The MATLAB software simulations are utilized to meet the experiment's specifications, facilitating a more profound comprehension of the fundamental principles in Communication.

This experiment comprises two distinct sections: the analog modulation system and the digital modulation system. The analog modulation system primarily focuses on exploring the modulation and demodulation processes of the Double Sideband (DSB) signal. Meanwhile, the digital modulation system centers on investigating the modulation and demodulation of the Binary Phase Shift Keying (BPSK) signal.

2. Theory

2.1. Analog modulation systems

$$\text{Linear modulation}\begin{cases} AM \\ DSB \\ SSB \end{cases}$$

Time domain representation of AM signals:

$$S_{\mathrm{AM}}(t) = [A_0 + m(t)]\cos(\omega_c t + \varphi_0)$$

Time domain representation of DSB signals:

$$S_{DSB}(t) = m(t)\cos\omega_c t$$

$$\textbf{Demodulation}\begin{cases} \text{Coherent demodulation} \\ \text{Noncoherent demodulation} \end{cases}$$

Coherent demodulation of AM signals:

$$m_0(t) = \frac{1}{2}[A_0 + m(t)]$$

Coherent demodulation of DSB signals:

$$m_0(t) = \frac{1}{2}m(t)$$

Coherent demodulation of SSB signals:

$$m_0(t) = \frac{1}{4}m(t)$$

2.2. Digital modulation systems

$$\textbf{Binary digital modulation}\begin{cases} 2ASK \\ 2FSK \\ 2PSK,\ 2DPSK \end{cases}$$

Binary absolute phase shift keying (BPSK) time domain representation:

$$S_{\mathrm{2PSK}}(t) = \sum_n a_n g(t - nT_s)\cos\omega_c t = s(t)\cos\omega_c t$$

Where

$$a_n = \begin{cases} +1,\ P \\ -1,\ 1-P \end{cases} \qquad S_{\mathrm{2PSK}}(t) = \begin{cases} \cos(\omega_c t + 0),\ P \\ \cos(\omega_c t + \pi),\ 1-P \end{cases}$$

Experiment 1: Analog Modulation Systems

3. Experimental Steps and Results

3.1. Plot the time-domain waveforms of DSB signals

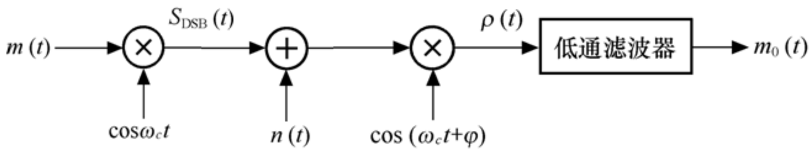


Figure 1: DSB modulation and demodulation block diagram

Figure 1 illustrates the theoretical model of DSB signal modulation and demodulation. According to this model, it can be seen that the modulation process of DSB is the baseband signal multiplied by a high-frequency carrier signal. In my experiment, I set **the baseband analog signal** is:

$$m(t) = \cos(2\pi f_m t), \quad f_m = 1kHz$$

And the **high-frequency carrier signal** is:

$$c(t) = \cos(2\pi f_c t), \quad f_c = 10kHz$$

The mathematical model of the DSB modulated signal is given by:

$$s_{\mathrm{DSB}}(t) = c(t) \times m(t)$$

The DSB modulated signal can be obtained by directly using the baseband signal multiplied by the carrier signal. Figure 2 shows the modulating signals, carrier signals, and modulated signals. To plot the envelope of the DSB signal, I also used the **Hillport transform**. Figure 3 shows the corresponding code for this section.

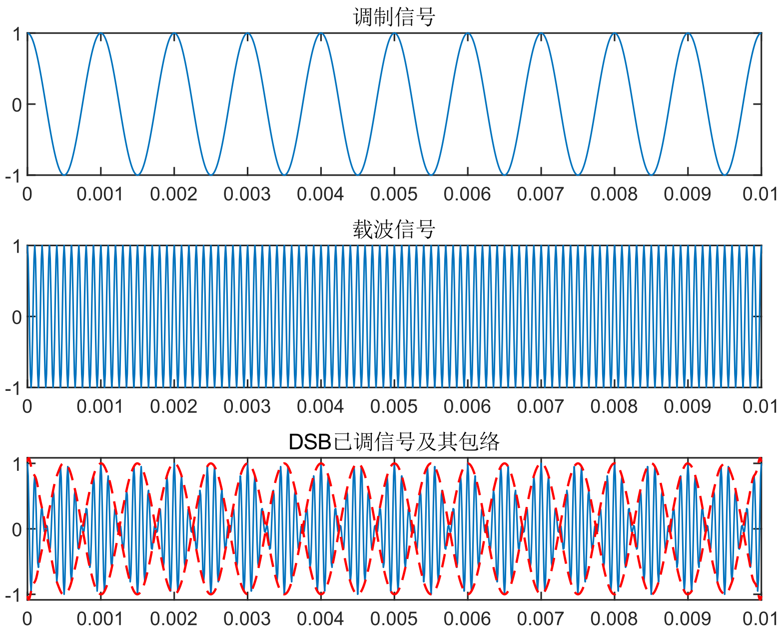


Figure 2

```
%% DSB信号的调制
clear; clc;
Ac=1; % 载波振幅
Am=1; % 信号振幅
f_c=10000; % 载波频率
f_m=1000; % 信号频率
fs = 10.*f_c; % 采样频率
t=0:1/fs:0.01; % 时间范围
m=Am.*cos(2*pi.*f_m.*t); % 调制信号
c=Ac.*cos(2*pi.*f_c.*t); % 载波信号
uDSB=m.*c; % DSB已调信号

% 计算包络线
envelope = abs(hilbert(uDSB));
```

Figure 3: Code section

3.2. Plot the frequency-domain waveforms of DSB signals

```
%% DSB调制信号的频谱
% 定义FFT长度
N = 2048;

M=fftshift(fft(m,N));
C=fftshift(fft(c,N));
UDSB=fftshift(fft(uDSB,N));

f=(0:N-1)*fs/N;
f=f-fs/2;
figure(2);
subplot(3,1,1);
plot(f,abs(M));xlabel('f');xlim([0 15000]);title('调制信号的频谱图');
subplot(3,1,2);
plot(f,abs(C));xlabel('f');xlim([0 22000]);title('载波信号的频谱图');
subplot(3,1,3);
plot(f,abs(UDSB));xlabel('f');xlim([0 22000]);title('已调信号的频谱图');
```

Figure 4: Code section

A waveform in the time domain can be converted to the frequency domain using the Fourier transform. However, since the computer can only deal with discrete variables, the form of discrete Fourier transform is taken. In MATLAB, the fast Fourier transform can be used for time to frequency domain conversion.

Figure 4 shows the corresponding code section. This code snippet performs spectral analysis of a DSB (Double Sideband) modulated signal using the Fast Fourier Transform (FFT) algorithm in MATLAB or a similar environment.

The variable N is defined as 2048, which specifies the length of the Fast Fourier Transform (FFT) to be used for spectral analysis. ‘fft()’ computes the FFT of the input signal. ‘fftshift()’ is used to shift the zero-frequency component (DC component) to the center of the spectrum for better visualization. Figure 5 illustrates the spectrogram of the modulating signal, the carrier signal, and the modulated signal.

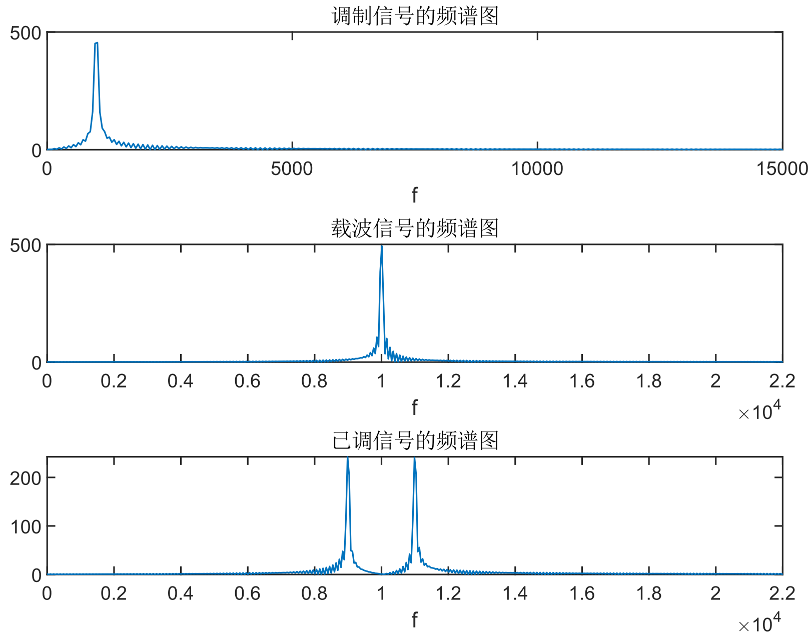


Figure 5

3.3. Coherent detection for DSB signals

The only demodulation method of DSB signal is coherent demodulation. The reason is that the envelope of DSB signal is no longer consistent with the change law of modulation signal, and simple envelope detection cannot be used to recover the signal, but coherent demodulation is also called synchronous detection (much more complex than envelope detection). Coherent demodulation is the process of restoring the original baseband signal by adding a synchronous pulse between the modulated signal and the carrier signal, so that the phase of the two signals is consistent.

```
%% DSB已调信号的相干解调，与本地载波相乘
s=uDSB.*c;
S=fftshift(fft(s,N));
figure;
subplot(2,1,1);
plot(t,s);xlim([0,0.01]);xlabel('t');grid on; title('与载波相乘后的时域波形图');
subplot(2,1,2);
plot(f,abs(S));xlim([0,25000]);xlabel('f');grid on; title('与载波相乘后的频谱图');
```

Figure 6: Code section

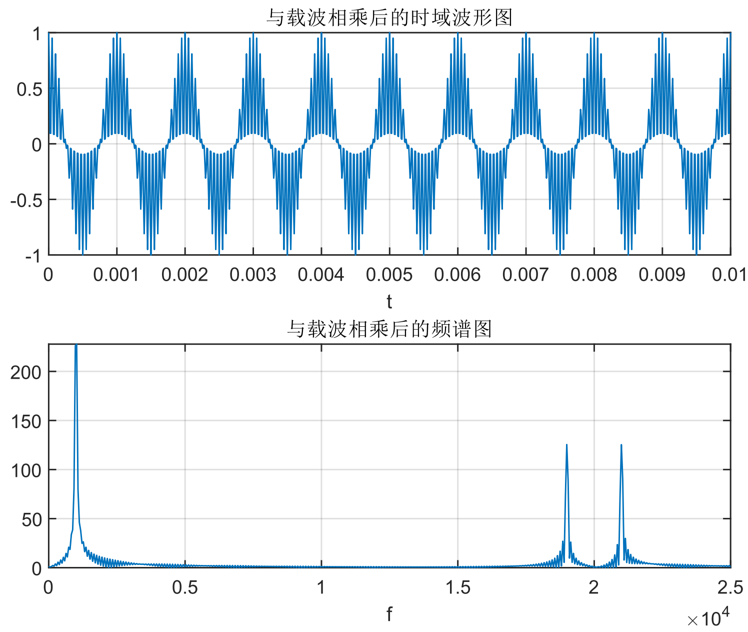


Figure 7

3.4. Design a low pass filter to remove the high-frequency components and obtain the demodulated signal

For the DSB signal, after the multiplication of the local carrier and the received signal, the high-frequency signal needs to be filtered out by a low-pass filter, and the original baseband signal can be recovered without distortion. Its demodulated output signal is：

$$m_0(t) = \frac{1}{2}m(t)$$

According to the derivation of the mathematical model of demodulation, the cut-off frequency of the low-pass filter should be at $f_c < f < 2f_c$, where f_c is the frequency of the carrier signal. Therefore, I used a low-pass filter with a cutoff frequency of $\omega_c = 1.5 \times 2 \times f_m / f_s$. This setting is to leave some margin so that the cutoff frequency increases slightly with respect to the signal bandwidth, avoiding the edge of the signal spectrum being truncated. As shown in figure 7, we want to filter out the high frequency component. Figures 8 and 9 show the corresponding code and the recovered original signal, respectively.

```
%% 设计低通滤波器滤除高频分量，得到解调信号
wc=1.5*2*f_m/fs;
B=fir1(32,wc);
% figure;
% freqz(B,1,N,fs);

%% 再通过低通滤波器，恢复原信号
[h, w]=freqz(B, 1, N);
so=filter(B,1,s);
SO=fftshift(fft(so,N));
figure;
plot(w*fs/(2*pi),20*log10(abs(h)));grid on;
title('低通滤波器的频率响应图');
figure;
subplot(211);
plot(t,so);xlim([0 0.01]);xlabel('t');grid on; title('解调器输出信号的时域波形图');
subplot(212);
plot(f,abs(SO));xlabel('f');axis([0 22000 0 130]);grid on;title('解调器输出信号的频谱图');
```

Figure 8: Code section

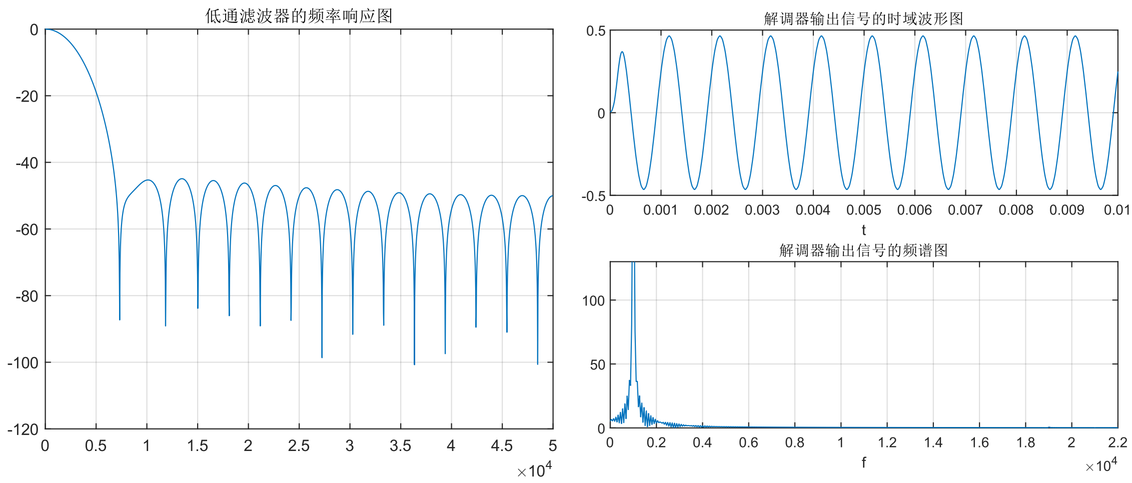


Figure 9: The frequency response and the output signal

3.5. Coherent detection for asynchronous local carriers

In the process of demodulation, different phase differences will have different effects on the demodulated signal, which is mainly reflected in the quality and characteristics of the demodulated signal. Different phase differences will cause the amplitude of the demodulated signal to change. When the phase difference is close to **zero**, the amplitude of the demodulated signal may be the **largest**. However, when the phase difference **increases**, the amplitude of the demodulated signal will **decrease**.

In this experiment, I used 4 different phase differences, which are: $\frac{\pi}{8}$, $\frac{\pi}{4}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$. Figure 10 shows the corresponding code section. This code simulates different phase differences during demodulation. It mixes demodulated carrier signals with the modulated signal and filters the resulting signal using the same low-pass filter to restore the original modulated signal. This helps observe how varying phase differences affect the demodulation outcome for comparisons of the recovered signal's quality or characteristics.

```
%% 在解调载波中引入相位差
pd1=pi/8; pd2=pi/4; pd3=pi/3; pd4=pi/2; %
c_demod1 = uDSB.*cos(2*pi*f_c*t + pd1); %
c_demod2 = uDSB.*cos(2*pi*f_c*t + pd2);
c_demod3 = uDSB.*cos(2*pi*f_c*t + pd3);
c_demod4 = uDSB.*cos(2*pi*f_c*t + pd4);
% 通过低通滤波器恢复原始信号
% 在这里使用您之前定义的低通滤波器 B 对解调信
so1 = filter(B, 1, c_demod1);
so2 = filter(B, 1, c_demod2);
so3 = filter(B, 1, c_demod3);
so4 = filter(B, 1, c_demod4);
```

Figure 10: Code section

Figure 11 shows the output waveform under different phase difference. It can be found that as the phase difference increases, the amplitude of the output waveform gradually decreases, and by 90° there is no output waveform.

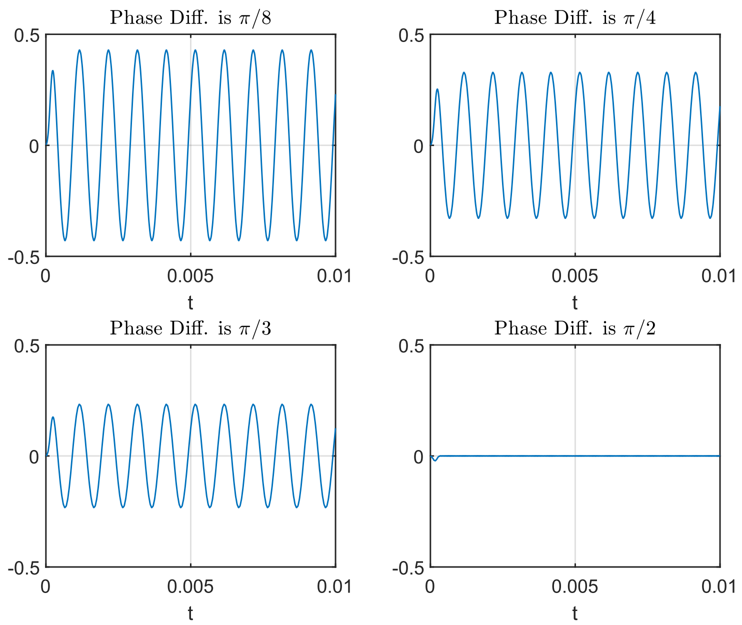


Figure 11: Output waveform under different phase difference

3.6. Plot the time-domain waveforms for AM and SSB modulation

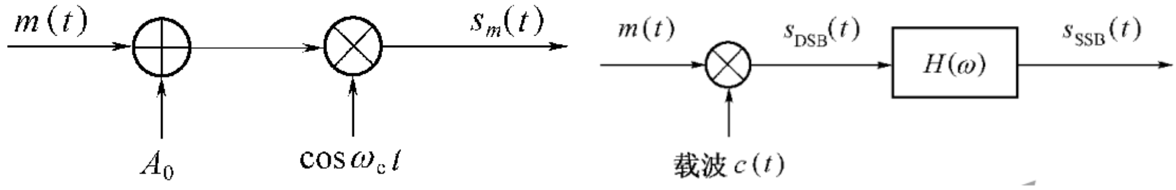


Figure 12: The block diagram of AM (left) and SSB (right) modulation

Figure 12 illustrates the generation process of AM signal and SSB signal. The time domain expression of AM signal is as follows:

$$S_{AM}(t) = [A_0 + m(t)]\cos(\omega_c t + \varphi_0)$$

For the convenience of analysis in the experimental simulation process, setting $\varphi_0 = 0$, but it does not affect the generality of the discussion.

Since the upper and lower sidebands of the DSB signal are completely symmetric and they can carry all the information of the modulated signal, only one sideband needs to be transmitted, so the SSB signal is introduced. There are two ways of generating SSB signals, which are **filtering** and **Hilbert transform**. In this experiment, I used the filtering method to generate SSB signals. The required single sideband signal can be generated by selecting the lower sideband of the DSB signal after the previously generated DSB signal is processed by low-pass filtering.

Figure 13 shows the corresponding code section. This code is mainly used to plot the time domain waveforms of AM (amplitude modulation) signals and SSB (single sideband modulation) signals, and to calculate and plot the envelope of the AM signal. Figure 14 shows the AM signal and SSB signal.

```
%% 绘制AM和SSB调制的时域波形
A =1;
s_am=(A+m).*c;
s_ssb=filter(B,1,uDSB); %DSB信号通过滤波器，产生SSB信号

% 计算包络线
envelope = abs(hilbert(s_am));
figure;
subplot(211);
plot(t,s_am);xlabel('t');grid on; title('AM信号的时域图');
hold on;
plot(t,envelope,'r--','LineWidth',1.0); % 添加上包络线
plot(t,-envelope,'r--','LineWidth',1.0); % 添加下包络线
hold off;
subplot(212);
plot(t,s_ssb);xlim([0.005 0.01]);xlabel('t');grid on; title('下边带信号的时域图');
```

Figure 13: Code section

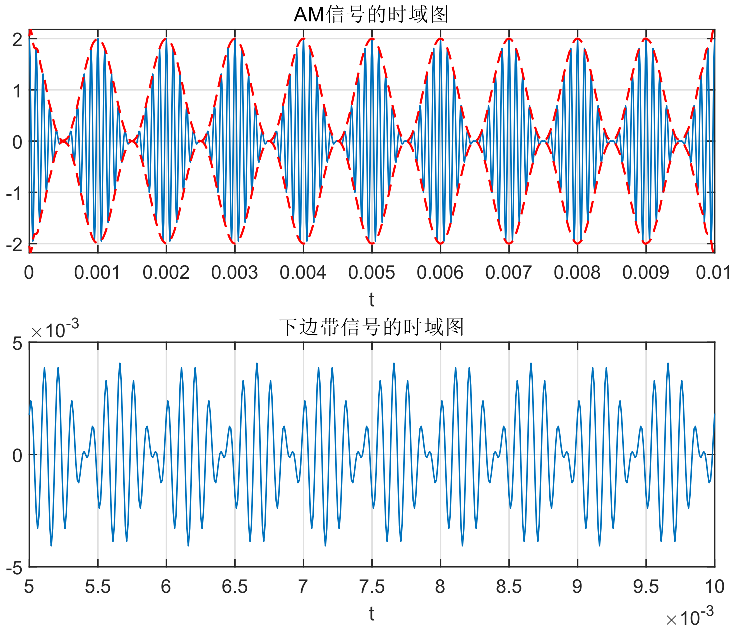


Figure 14: AM signal and SSB signal

The total code of analog modulation system is in appendix A.

Experiment 2: Digital Modulation Systems

4. Experimental Steps and Results

4.1. BPSK signal waveforms with rectangular pulse and roll-off pulse

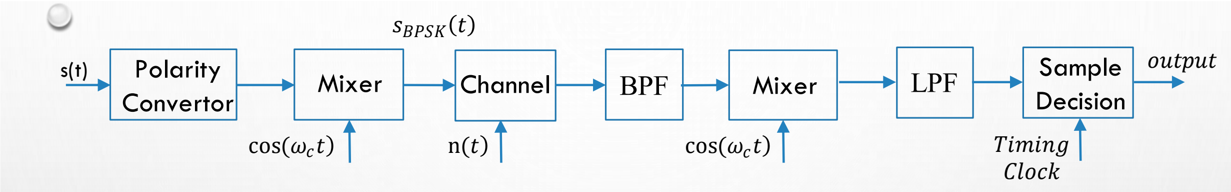


Figure 15: BPSK signal modulation and demodulation block diagram

Figure 15 shows the process of BPSK signal modulation and demodulation. The signal transmitted by 2PSK is a bipolar signal, and its time domain expression is as follows:

$$S_{2\text{PSK}}(t)=\sum_n a_n g(t-nT_s)\cos\omega_c t=s(t)\cos\omega_c t$$

In this experiment, the symbol transmission rate is $R_b=100\ Baud$, which means $T_s=\frac{1}{R_b}=0.01s$. The carrier frequency is $f_c=1\text{kHz}$ and the roll-off factor is $\alpha=1$.

```
Tm=0.01; %符号周期Tm=1/Rb
fc=1000; %载波频率
N_sample=100; %每符号采样点个数
N_sum=1000; %符号个数
dt=Tm/N_sample;
N=N_sample*N_sum;
t=(0:dt:N_sum*Tm-dt)';
c=sin(2*pi*fc*t); %载波信号
NFFT=2.^16;
Nloop=50;
span = 5; %截断长度
alpha = 1; %滚降系数
st2=0; %矩形脉冲信号
st3=0; %滚降脉冲信号
for i =1:Nloop
    d = 2 * randi([0, 1], N_sum, 1) - 1; %产生双极性信号 'N_su
    st_bb=rectpulse(d,N_sample); %生成矩形波
    st_2psk=st_bb.*sin(2*pi*fc*t);
    h = (rcosdesign(alpha, span, N_sample)).*5; %冲激响应
    y = filter(h, 1, upsample(d, N_sample)); %对信号进行滤波处
    st_rf=y.*sin(2*pi*fc*t);
```

Figure 15: Code section

Figure 15 illustrates part of the code for the rectangular pulse and the roll-off pulse. In this code, I set the number of sample points per symbol and the number of symbols. When calculating the power spectral density, I used a 'for 'loop to execute the code many times and take the average, which better translated the real power spectral density. Figures 16 and 17 demonstrate the results for rectangular pulse waveforms and roll-off pulse waveforms, respectively. The complete code is shown in Appendix B.

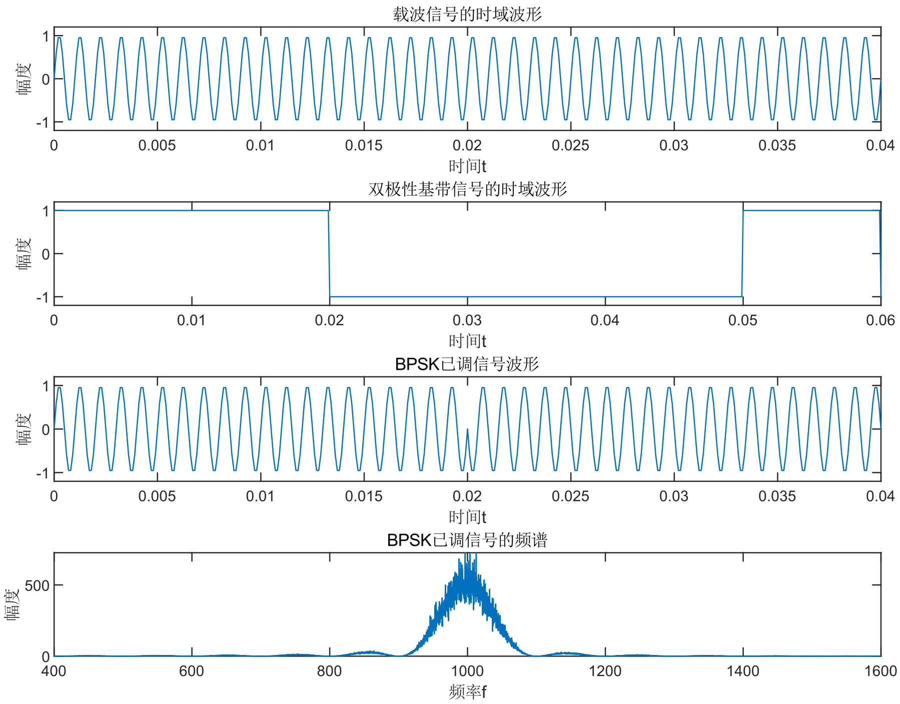


Figure 16: Rectangular pulse waveforms

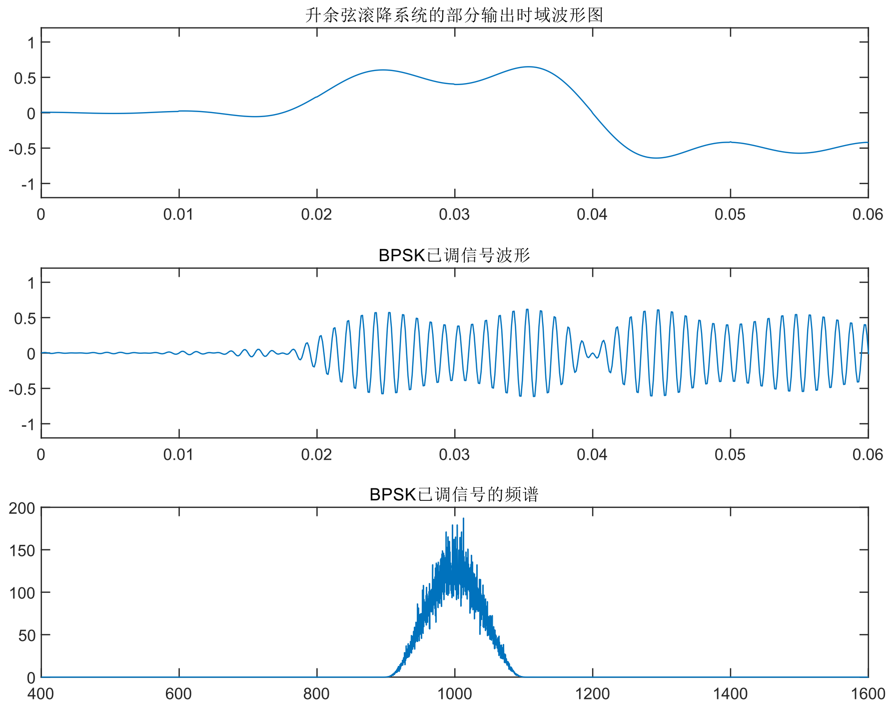


Figure 17: Roll-off pulse waveforms

4.2. BPSK Demodulation

In this experiment, the coherent demodulation of the 2PSK signal is performed assuming an absence of noise influence during the channel transmission process. After transmission through the channel, the demodulated signal initially undergoes band-pass filtering. The primary role of the band-pass filter is to selectively retain signal components within a narrow spectral range where the original information is positioned while eliminating interference from other frequencies. This filtration concentrates the signal energy within the frequency band of the original information, facilitating accurate information recovery.

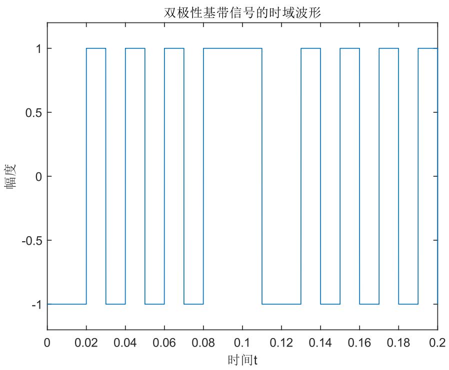
Subsequently, the signal passes through a multiplier, where the multiplied signal aligns its frequency and phase with the carrier signal. Coherent demodulation necessitates knowledge of the carrier frequency and phase employed at the transmitter. The processed signal then traverses through a low-pass filter, preserving the signal within the baseband frequency range corresponding to the original information. This step significantly contributes to signal recovery and enhances demodulation accuracy during coherent demodulation.

Finally, the output signal is obtained through sampling decisions, concluding the coherent demodulation process.

Figure 18 shows the code for the tuning section. Figure 19 shows the transmitted signal and the demodulated output signal of the rectangular pulse waveform. Figure 20 shows the transmitted signal and the demodulated output signal of the roll-off pulse waveform. Figure 21 illustrates the frequency responses of the band-pass and low-pass filters used in this experiment. The complete code is shown in Appendix C.

```
%% 在信道中进行传输
%如果进行滚降脉冲波形的分析的话，st2_2pskin=st_rf
st2_2pskin = st_2psk; %假设没有信道噪声的情况下，调制信号的输出等于解调信号的输入
wc1=[2*pi*(fc-fm)/N_sample,2*pi*(fc+fm)/N_sample];
B1 = fir1(256,wc1/pi);
st_out0=filter(B1,1,st2_2pskin); %通过带通滤波器

st_out1=st_out0.*sin(2*pi*fc*t); %与相干载波相乘
wc2 = 2*2*pi*fm/N_sample;
wc2=wc2/100;
B2 = fir1(2000,wc2/pi);
st_out2 = filter(B2,1,st_out1); %通过低通滤波器

% 进行抽样判决
threshold = 0; % 设置一个阈值，根据信号特性调整阈值
received_bits = zeros(N_sum, 1); % 初始化接收到的比特流

for i = 1:N_sum
    sample_index = (i - 1) * N_sample + 1; % 计算当前符号的起始抽样位置
    sample = st_out2(sample_index); % 对解调后的信号进行抽样

    % 根据信号的正负来判断是哪个比特值
    if sample < threshold
        received_bits((i - 1) * N_sample + 1 : i * N_sample) = ones(N_sample, 1);
    else
        received_bits((i - 1) * N_sample + 1 : i * N_sample) = zeros(N_sample, 1);
```

Figure 18: Code section



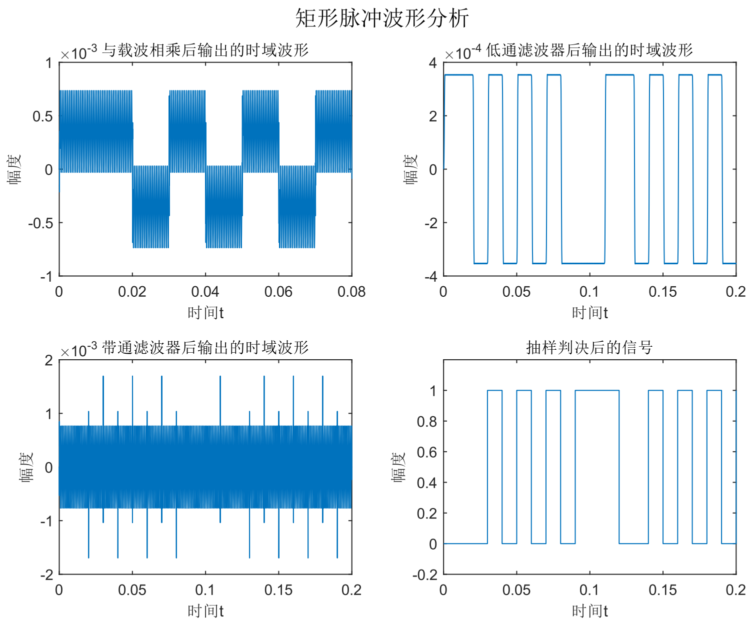
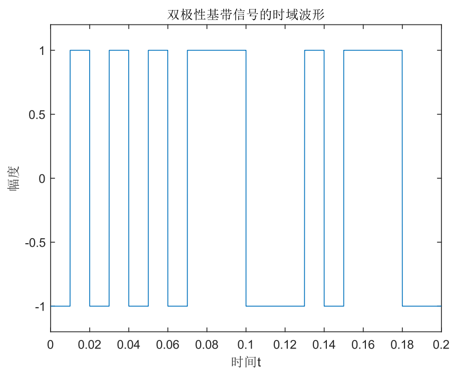


Figure 19: Rectangular pulse waveform



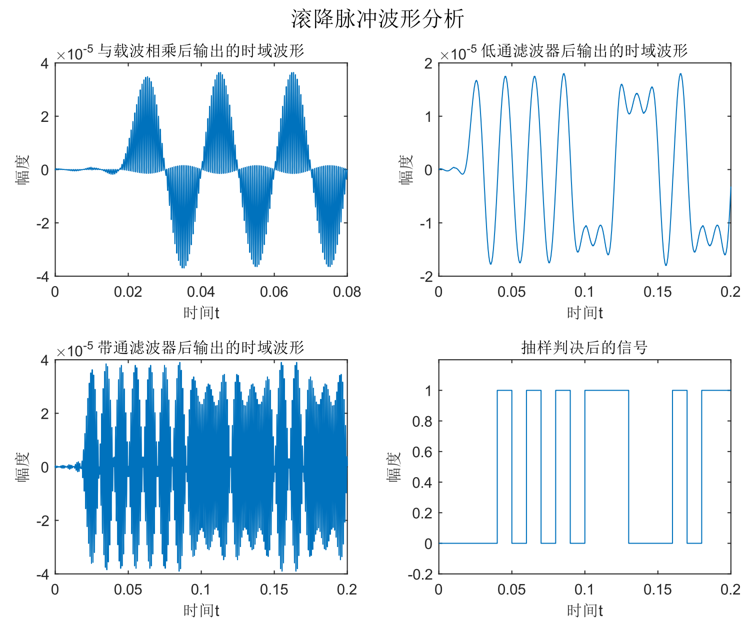


Figure 20: Roll-off pulse waveform

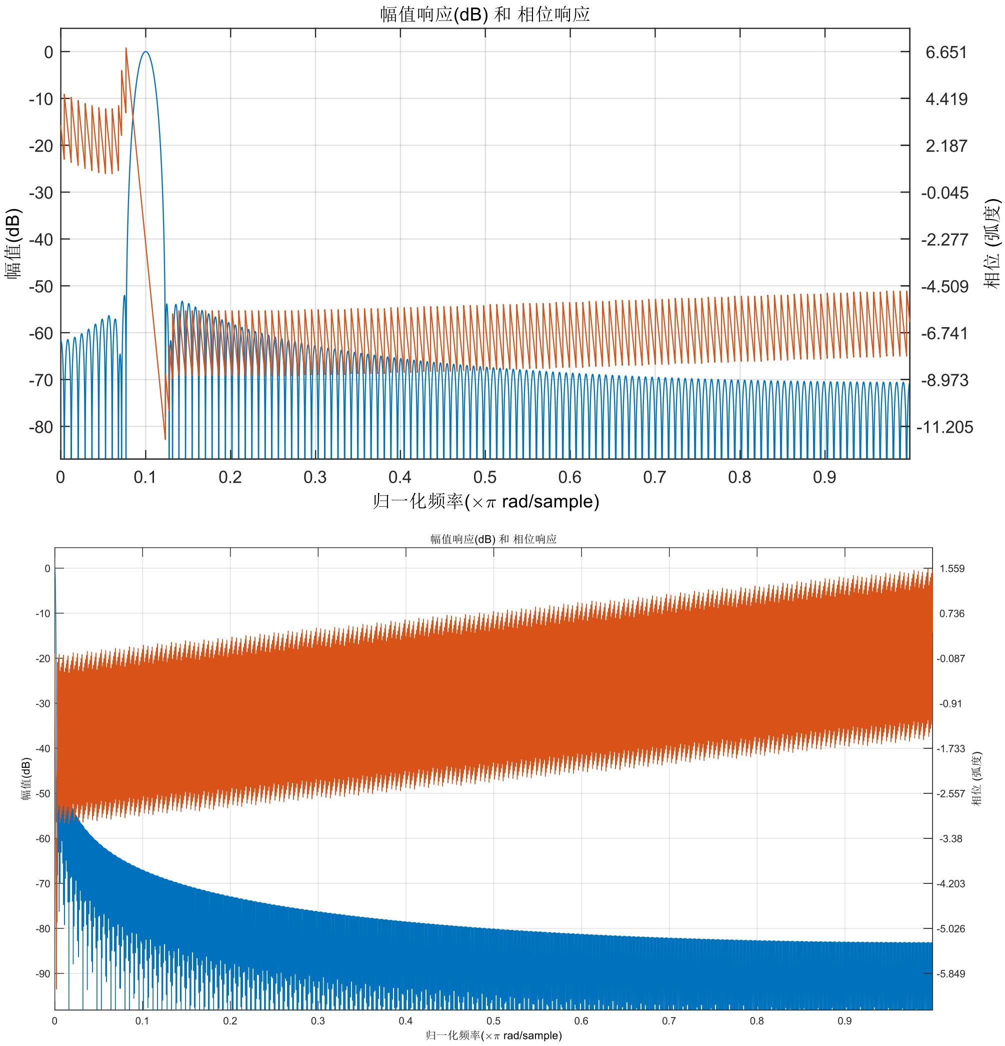


Figure 21: Bandpass and low-pass filter frequency responses

4.3. Calculate the BER performances of QPSK

```
%% 计算QPSK的误码率性能
M = 4; % QPSK
numBits = 1e6; % 生成的随机二进制位数
EbN0_dB = 0:10; % Eb/N0范围

% 生成随机二进制位
data = randi([0 1], numBits, 1);

% QPSK映射
dataEnc = bi2de(reshape(data, [], log2(M)));
symbols = pskmod(dataEnc, M);

% 计算符号功率和比特功率
Es = mean(abs(symbols).^2);
Eb = Es/log2(M);

% 初始化BER数组
berSim = zeros(size(EbN0_dB));

% 对每个Eb/N0值进行模拟
for i = 1:length(EbN0_dB)
    % 生成高斯噪声
    n0 = Eb / (10^(EbN0_dB(i)/10));
    noise = sqrt(n0/2) * (randn(size(symbols)) + 1j*randn(size(symbols)));

    % 添加噪声并进行最优决策
    received = symbols + noise;
    dataDec = pskdemod(received, M);

    % 计算误比特率（BER）
    [~, berSim(i)] = biterr(dataEnc, dataDec);
end

% 计算理论BER
berTheory = berawgn(EbN0_dB, 'psk', M, 'nondiff');
```

Figure 22: Code section

Figure 22 shows the code section of this task. This code aims to assess the performance of a QPSK modulation system by simulating and calculating the bit error rate (BER). Initially, the code generates one million random binary bits as the data source, which are then mapped to QPSK symbols for modulation. Subsequently, Gaussian noise is introduced to the QPSK signals across various signal-to-noise ratio (Eb/N0) ranges. Using optimal decision-making methods, these signals are demodulated, and the bit error rate (BER) is calculated to explore the system's performance under different Eb/N0 conditions. Additionally, theoretical BER curves for QPSK are computed using predefined functions, providing a benchmark for comparison and analysis of actual simulated results. Plotting both simulated BER and theoretical BER curves enables an assessment of system performance under various signal-to-noise ratios, offering crucial insights into the expected bit error rates in practical applications. The complete code is shown in Appendix D.

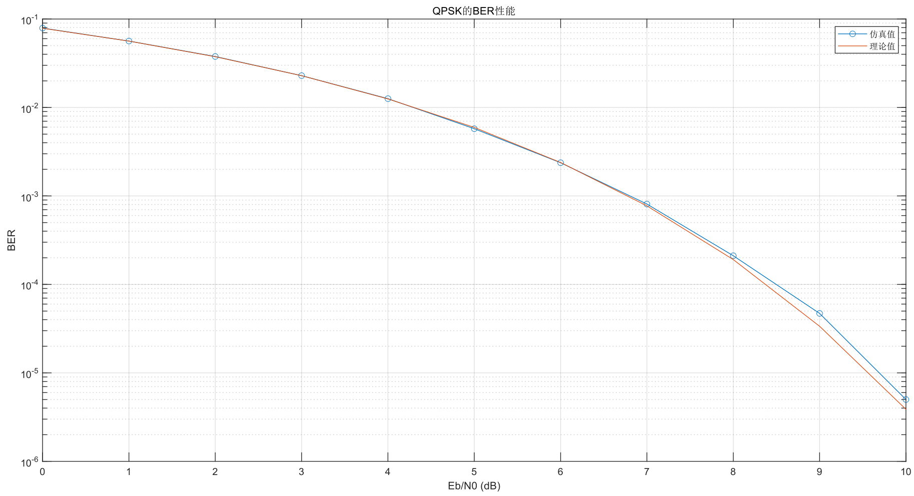


Figure 23: Bit error performance curve

Figure 23 illustrates the BER performance curves for the simulated and theoretical values. It can be seen from the figure that under the same SNR, the BER of the simulation value is always larger than that of the theoretical value. Therefore, it can be proved that the curve is drawn correctly.

4.4. Plot the received QPSK constellations for various noise powers

The physical meaning of a constellation diagram in communication is that it is a graphical representation method used for digital modulation. It can represent the amplitude and phase of signal elements, as well as the difference between the actual signal and the ideal signal, which helps to identify the type of noise in the signal, discover various modulation problems, and is beneficial for optimizing the design of communication systems.

In this experiment, to draw QPSK constellation diagrams under different SNR conditions, I first generated QPSK signals. Secondly, the noise signal is generated, and then the noise is added, and the constellation diagram is drawn. Figure 24 shows the constellation plots under different noises. The complete code is shown in Appendix E.

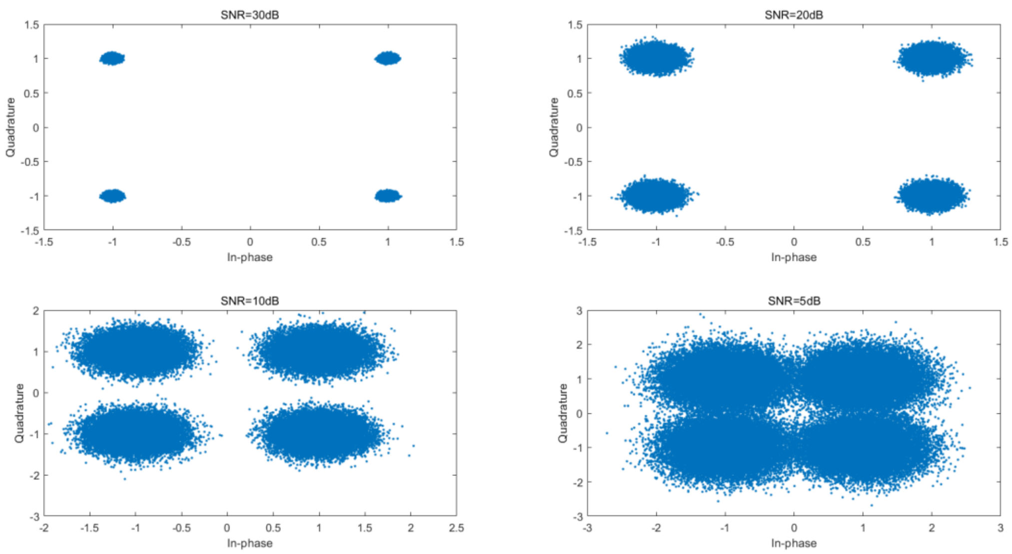


Figure 24: Constellation diagram

5. Further discussion

5.1. Analog Modulation Systems

Analog modulation systems historically played a significant role in traditional communication, notably through methods like Amplitude Modulation (AM) and Frequency Modulation (FM) in broadcasting. While their prominence has diminished in mainstream communication, analog modulation systems still find application in specialized domains such as specific audio transmissions and analog signal processing. These systems continue to hold relevance in certain audio-related transmission needs and analog signal processing scenarios.

5.2. Digital Modulation Systems

Digital modulation systems, pivotal in modern communication engineering, revolutionized data transmission by allowing information to be encoded in digital format. These systems, employing techniques like Quadrature Amplitude Modulation (QAM), Phase Shift Keying (PSK), and Frequency Shift Keying (FSK), dominate contemporary wireless communication, internet protocols, and satellite communications. Their inherent flexibility, superior error correction, and adaptability across diverse environments make digital modulation systems indispensable, powering high-speed data transfers, improving spectral efficiency, and ensuring robust communication links.

6. Conclusion

This experiment has solidified my comprehension of both the theoretical principles and mathematical models underlying analog and digital modulation systems.

The MATLAB simulations facilitated a deeper insight into the comprehensive procedures involved in modulation and demodulation processes, effectively merging theoretical concepts with practical applications. Furthermore, the exploration of engineering applications in both analog and digital modulation has provided a practical context for understanding their real-world significance.

References

[1] Jiang Qing, FAN Xinyue, Chen Shanxue. Principles of Communication [M]. Beijing: Tsinghua University Press, 2016.

Appendix

Appendix A: Code for Analog Modulation Systems

```
%% DSB 信号的调制
clear; clc;
Ac=1; % 载波振幅
Am=1; % 信号振幅
f_c=10000; % 载波频率
f_m=1000; % 信号频率
fs = 10.*f_c; % 采样频率
t=0:1/fs:0.01; % 时间范围
m=Am.*cos(2*pi.*f_m.*t); % 调制信号
c=Ac.*cos(2*pi.*f_c.*t); % 载波信号
uDSB=m.*c; % DSB 已调信号

% 计算包络线
envelope = abs(hilbert(uDSB));

% 绘制时域信号
figure(1);
subplot(3,1,1);
plot(t,m);
title('调制信号');
subplot(3,1,2);
plot(t,c);
title('载波信号');
subplot(3,1,3);
plot(t,uDSB);
hold on;
plot(t,envelope,'r--','LineWidth',1.0); % 添加上包络线
plot(t,-envelope,'r--','LineWidth',1.0); % 添加下包络线
hold off;
title('DSB 已调信号及其包络');

%% DSB 调制信号的频谱
% 定义 FFT 长度
N = 2048;

M=fftshift(fft(m,N));
C=fftshift(fft(c,N));
UDSB=fftshift(fft(uDSB,N));

f=(0:N-1)*fs/N;
```

```
f=f-fs/2;
figure(2);
subplot(3,1,1);
plot(f,abs(M));xlabel('f');xlim([0 15000]);title('调制信号的频谱图');
subplot(3,1,2);
plot(f,abs(C));xlabel('f');xlim([0 22000]);title('载波信号的频谱图');
subplot(3,1,3);
plot(f,abs(UDSB));xlabel('f');xlim([0 22000]);title('已调信号的频谱图');

%% DSB 已调信号的相干解调，与本地载波相乘
s=uDSB.*c;
S=fftshift(fft(s,N));
figure;
subplot(2,1,1);
plot(t,s);xlim([0,0.01]);xlabel('t');grid on; title('与载波相乘后的时域波形图');
subplot(2,1,2);
plot(f,abs(S));xlim([0,25000]);xlabel('f');grid on; title('与载波相乘后的频谱图');

%% 设计低通滤波器滤除高频分量，得到解调信号
wc=1.5*2*f_m/fs;
B=fir1(32,wc);
% figure;
% freqz(B,1,N,fs);

%% 再通过低通滤波器，恢复原信号
[h, w]=freqz(B, 1, N);
so=filter(B,1,s);
SO=fftshift(fft(so,N));
figure;
plot(w*fs/(2*pi),20*log10(abs(h)));grid on;
title('低通滤波器的频率响应图');
figure;
subplot(211);
plot(t,so);xlim([0 0.01]);xlabel('t');grid on; title('解调器输出信号的时域波形图');
subplot(212);
plot(f,abs(SO));xlabel('f');axis([0 22000 0 130]);grid on;title('解调器输出信号的频谱图');

%% 在解调载波中引入相位差
pd1=pi/8; pd2=pi/4; pd3=pi/3; pd4=pi/2; % Choose a phase difference in degrees
c_demod1 = uDSB.*cos(2*pi*f_c*t + pd1); % Asynchronous carrier for demodulation
c_demod2 = uDSB.*cos(2*pi*f_c*t + pd2);
c_demod3 = uDSB.*cos(2*pi*f_c*t + pd3);
c_demod4 = uDSB.*cos(2*pi*f_c*t + pd4);
% 通过低通滤波器恢复原始信号
```

```
% 在这里使用之前定义的低通滤波器 B 对解调信号进行处理
so1 = filter(B, 1, c_demod1);
so2 = filter(B, 1, c_demod2);
so3 = filter(B, 1, c_demod3);
so4 = filter(B, 1, c_demod4);

% 绘制已解调信号通过低通滤波器后的结果
figure;
subplot(2,2,1);
plot(t, so1); axis([0 0.01 -0.5 0.5]);xlabel('t'); grid on; title('Phase Diff. is $\pi/8$', 'Interpreter', 'latex');
subplot(2,2,2);
plot(t, so2); axis([0 0.01 -0.5 0.5]);xlabel('t'); grid on; title('Phase Diff. is $\pi/4$', 'Interpreter', 'latex');
subplot(2,2,3);
plot(t, so3);axis([0 0.01 -0.5 0.5]); xlabel('t'); grid on; title('Phase Diff. is $\pi/3$', 'Interpreter', 'latex');
subplot(2,2,4);
plot(t, so4);axis([0 0.01 -0.5 0.5]); xlabel('t'); grid on; title('Phase Diff. is $\pi/2$', 'Interpreter', 'latex');

%% 绘制 AM 和 SSB 调制的时域波形
A =1;
s_am=(A+m).*c;
s_ssb=filter(B,1,uDSB); %DSB 信号通过滤波器，产生 SSB 信号

% 计算包络线
envelope = abs(hilbert(s_am));
figure;
subplot(211);
plot(t,s_am);xlabel('t');grid on; title('AM 信号的时域图');
hold on;
plot(t,envelope,'r--','LineWidth',1.0); % 添加上包络线
plot(t,-envelope,'r--','LineWidth',1.0); % 添加下包络线
hold off;
subplot(212);
plot(t,s_ssb);xlim([0.005 0.01]);xlabel('t');grid on; title('下边带信号的时域图');
```

Appendix B: BPSK modulation

```
%% BPSK 信号的矩形脉冲和滚降脉冲
clc; close all;
Tm=0.01; %符号周期 Tm=1/Rb
fc=1000; %载波频率
N_sample=50; %每符号采样点个数
N_sum=200; %符号个数
dt=Tm/N_sample;
N=N_sample*N_sum;
```

```
t=(0:dt:N_sum*Tm-dt)';
c=sin(2*pi*fc*t); %载波信号
NFFT=2.^16;
Nloop=10;
a = 1;   %  滚降因子
st1=0;
st2=0; %矩形信号
for i =1:Nloop
    d = 2 * randi([0, 1], N_sum, 1) - 1; %产生双极性信号 'N_sum' 是要生成的随机整数矩阵的行数
'1' 是要生成的随机整数矩阵的列数
    st_bb=rectpulse(d,N_sample); %生成矩形波
    st_2psk=st_bb.*sin(2*pi*fc*t);

    window=boxcar(length(st_bb)); %定义矩形窗
    [pxx1,~]=periodogram(st_bb,window,NFFT,1/dt);
    [pxx2,f]=periodogram(st_2psk,window,NFFT,1/dt);
    st1 = st1+pxx1; st2=st2+pxx2;
end

st1=st1/Nloop;
st2=st2/Nloop;
st3=st3/Nloop;

figure;
subplot(2,2,1);
plot(t,c); axis([0 0.04 -1.2 1.2]);
xlabel('时间 t');ylabel('幅度');title('载波信号的时域波形')
subplot(2,2,2);
plot(t,st_bb); axis([0 0.06 -1.2 1.2]);
xlabel('时间 t'); ylabel('幅度') ;title('双极性基带信号的时域波形');
subplot(2,2,3); plot(t,st_2psk); axis([0 0.06 -1.2 1.2]);
xlabel('时间 t'); ylabel('幅度') ;title('BPSK 已调信号波形');
subplot(2,2,4); plot(f,st2.*10.^5); xlim([400 1600]);
xlabel('频率 f'); ylabel('幅度'); title('BPSK 已调信号的频谱');
figure;
stem(d, '.');
axis([0 50 -1.5 1.5]);
grid on;
title('随机双极性信号');
```

Appendix C: Demodulation of BPSK

```
%% BPSK 的解调
clear;close all;
```

```
fm=100;
Tm=1/fm;
fc=1000; %载波频率
N_sample = 20000;
N_sum=100; %符号个数
dt=Tm/N_sample;
N=N_sample*N_sum;
t=(0:dt:N_sum*Tm-dt)';
c=sin(2*pi*fc*t); %载波信号
NFFT=2.^16;
d = 2 * randi([0, 1], N_sum, 1) - 1;
span = 5; %截断长度
alpha = 1; %滚降系数

%对于矩形脉冲
st_bb=rectpulse(d,N_sample); %产生基带信号
st_2psk=st_bb.*sin(2*pi*fc*t);
% 对于滚降脉冲
h = (rcosdesign(alpha, span, N_sample)).*5; %冲激响应
y = filter(h, 1, upsample(d, N_sample)); %对信号进行滤波处理，生成滚降脉冲波形
st_rf=y.*sin(2*pi*fc*t);

%% 在信道中进行传输
%如果进行滚降脉冲波形的分析的话，st2_2pskin=st_rf
st2_2pskin = st_rf; %假设没有信道噪声的情况下，调制信号的输出等于解调信号的输入
wc1=[2*pi*(fc-fm)/N_sample,2*pi*(fc+fm)/N_sample];
B1 = fir1(256,wc1/pi);
st_out0=filter(B1,1,st2_2pskin); %通过带通滤波器

st_out1=st_out0.*sin(2*pi*fc*t); %与相干载波相乘
wc2 = 2*2*pi*fm/N_sample;
wc2=wc2/100;
B2 = fir1(2000,wc2/pi);
st_out2 = filter(B2,1,st_out1); %通过低通滤波器

% 进行抽样判决
threshold = 0; % 设置一个阈值，根据信号特性调整阈值
received_bits = zeros(N_sum, 1); % 初始化接收到的比特流

for i = 1:N_sum
    sample_index = (i - 1) * N_sample + 1; % 计算当前符号的起始抽样位置
    sample = st_out2(sample_index); % 对解调后的信号进行抽样

    % 根据信号的正负来判断是哪个比特值
```

```
    if sample < threshold
        received_bits((i - 1) * N_sample + 1 : i * N_sample) = ones(N_sample, 1);
    else
        received_bits((i - 1) * N_sample + 1 : i * N_sample) = zeros(N_sample, 1);
    end
end
%% 图像绘制
figure;
plot(t,st_bb); axis([0 0.2 -1.2 1.2]);
xlabel('时间 t'); ylabel('幅度') ;title('双极性基带信号的时域波形');
%矩形脉冲波形分析
figure('Position', [100, 100, 800, 600]);    % Adjust figure size as needed
%sgtitle('矩形脉冲波形分析');
sgtitle('滚降脉冲波形分析');
subplot(2,2,1); plot(t,st_out1);
xlabel('时间 t');ylabel('幅度');xlim([0 0.08]);
title('与载波相乘后输出的时域波形');

subplot(2,2,2); plot(t,st_out2);
xlabel('时间 t');ylabel('幅度');xlim([0 0.2]);
title('低通滤波器后输出的时域波形');

subplot(2,2,3); plot(t,st_out0);
xlabel('时间 t');ylabel('幅度');xlim([0 0.2]);
title('带通滤波器后输出的时域波形');

% 重新定义时间向量，使其长度与 received_bits 匹配
t_received_bits = (0:dt:N_sum * Tm - dt)';
subplot(2,2,4); plot(t_received_bits, received_bits);
xlabel('时间 t');ylabel('幅度'); axis([0 0.2 -0.2 1.2]);
title('抽样判决后的信号');

%fvtool(B1,1); % 带通滤波器的频率响应
fvtool(B2,1); %低通滤波器的频率响应
```

Appendix D: The BER performances of QPSK

```
%% 计算 QPSK 的误码率性能
M = 4; % QPSK
numBits = 1e6; % 生成的随机二进制位数
EbN0_dB = 0:10; % Eb/N0 范围

% 生成随机二进制位
data = randi([0 1], numBits, 1);
```

```
% QPSK 映射
dataEnc = bi2de(reshape(data, [], log2(M)));
symbols = pskmod(dataEnc, M);

% 计算符号功率和比特功率
Es = mean(abs(symbols).^2);
Eb = Es/log2(M);

% 初始化 BER 数组
berSim = zeros(size(EbN0_dB));

% 对每个 Eb/N0 值进行模拟
for i = 1:length(EbN0_dB)
    % 生成高斯噪声
    n0 = Eb / (10^(EbN0_dB(i)/10));
    noise = sqrt(n0/2) * (randn(size(symbols)) + 1j*randn(size(symbols)));

    % 添加噪声并进行最优决策
    received = symbols + noise;
    dataDec = pskdemod(received, M);

    % 计算误比特率（BER）
    [~, berSim(i)] = biterr(dataEnc, dataDec);
end

% 计算理论 BER
berTheory = berawgn(EbN0_dB, 'psk', M, 'nondiff');

% 比较模拟结果和理论 BER 曲线
semilogy(EbN0_dB, berSim, 'o-');
hold on;
semilogy(EbN0_dB, berTheory);
grid on;
legend('仿真值', '理论值');
xlabel('Eb/N0 (dB)');
ylabel('BER');
title('QPSK 的 BER 性能');
```

Appendix E: The QPSK constellations

```
clear; close all; clc;
%% 绘制不同噪声功率下的 QPSK 星座图
Qm = 2; % 每符号比特数
```

```
N_num = 10^5; % 仿真符号数

d1 = sign(randn(1, N_num));
d2 = sign(randn(1, N_num));
d = d1 + 1i * d2;

% n=generateNoise(定义对数信噪比,比特值);
n1 = generateNoise(30, d); % 调用函数生成噪声
n2 = generateNoise(20, d);
n3 = generateNoise(10, d);
n4 = generateNoise(5, d);

rt1 = d + n1; % 加噪
rt2 = d + n2; rt3 = d + n3; rt4 = d + n4;
figure;
subplot(2,2,1);
plot(rt1, '.'); xlabel('In-phase'); ylabel('Quadrature');
title('SNR=30dB');
subplot(2,2,2);
plot(rt2, '.'); xlabel('In-phase'); ylabel('Quadrature');
title('SNR=20dB');
subplot(2,2,3);
plot(rt3, '.'); xlabel('In-phase'); ylabel('Quadrature');
title('SNR=10dB');
subplot(2,2,4);
plot(rt4, '.'); xlabel('In-phase'); ylabel('Quadrature');
title('SNR=5dB');

%% 下面为定义的函数部分
% function n = generateNoise(EbN0_dB, d)
%       Qm = 2; % 每符号比特数
%       Es = mean(abs(d).^2);
%       Eb = Es / Qm;
%
%       ebn0 = 10.^(EbN0_dB/10); % 计算线性信噪比
%       sigma = sqrt(Eb / ebn0 / 2);
%       n = sigma * randn(size(d)) + 1i * sigma * randn(size(d)); % 生成噪声
% end
```