

2PSK&QPSK 数字带通信号仿真

08023214 张鼴译萱

1 实验目的

实验旨在通过构建和仿真 2PSK 和 QPSK 两种经典数字带通调制系统，掌握调制、解调、误码率（BER）与误符号率（SER）仿真及系统参数分析方法，并探究载波频偏对系统性能的影响。

2 实验原理

本实验构建了一个完整的数字带通传输系统。实验首先将本人学号（08023214）中的每位十进制数字转换为 4 位二进制数，串联形成基带二进制比特流 $b(n)$ 。该序列作为数字信源，分别经由 2PSK 和 QPSK 系统进行传输仿真。系统整体仿真框图如图 1 所示。

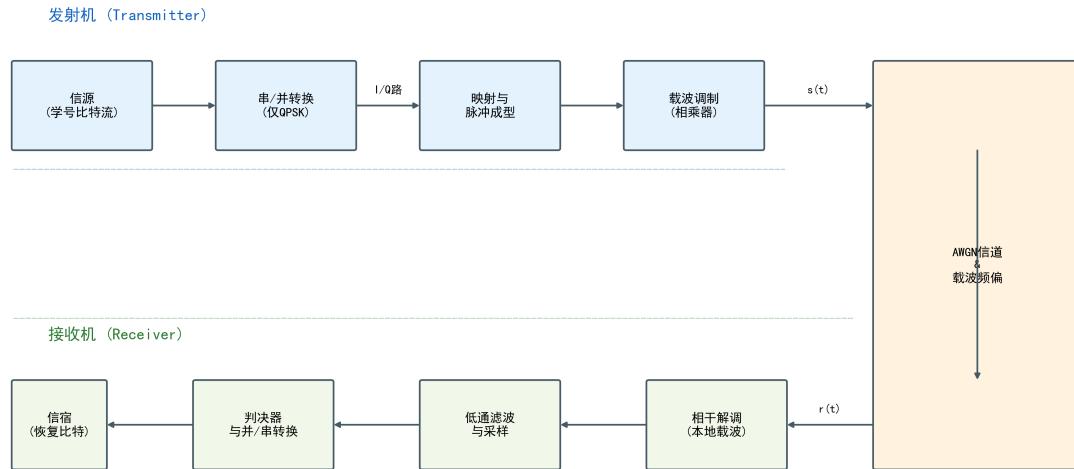


图 1: 2PSK/QPSK 数字通信系统仿真框图

2.1 2PSK（二进制相移键控）原理

2PSK (Binary Phase Shift Keying) 利用载波的两种不同相位来表征数字信息。

2.1.1 信号模型

2PSK 信号的时域表达式为:

$$s_{2\text{PSK}}(t) = \begin{cases} A \cos(2\pi f_c t + 0), & \text{当发送比特为 1} \\ A \cos(2\pi f_c t + \pi), & \text{当发送比特为 0} \end{cases} \quad (1)$$

其中, f_c 为载波频率, A 为振幅。上式可统一写为双极性形式:

$$s_{2\text{PSK}}(t) = a_n \cdot A \cos(2\pi f_c t), \quad a_n \in \{+1, -1\} \quad (2)$$

这意味着 2PSK 本质上是抑制载波的双边带调幅信号。

2.1.2 相干解调与判决

由于 2PSK 信号未携带离散载波分量, 必须采用相干解调。

1. 乘法器: 接收信号 $r(t) = s(t) + n(t)$ 与本地恢复的同频同相载波 $\cos(2\pi f_c t)$ 相乘。
2. 积分清除与低通滤波:

$$y(t) = \int_0^{T_b} r(t) \cos(2\pi f_c t) dt \approx \frac{1}{2} a_n A T_b + \text{噪声分量} \quad (3)$$

3. 抽样判决: 设置判决门限为 0。若 $y > 0$, 判决为'1'; 若 $y \leq 0$, 判决为'0'。

2.1.3 抗噪性能

在加性高斯白噪声 (AWGN) 信道下, 2PSK 的理论误比特率 (BER) 为:

$$P_{e,2\text{PSK}} = Q \left(\sqrt{\frac{2E_b}{N_0}} \right) \quad (4)$$

2.2 QPSK (正交相移键控) 原理

QPSK (Quadrature Phase Shift Keying) 利用四个相位传输信息, 每个符号包含 2 个比特。

2.2.1 正交分解与信号生成

QPSK 的时域表达式可分解为:

$$s_{\text{QPSK}}(t) = I(t) \cos(2\pi f_c t) - Q(t) \sin(2\pi f_c t) \quad (5)$$

其中, $I(t)$ 和 $Q(t)$ 的取值为 $\pm A$ 。

2.2.2 频带利用率与误码率

- 误码率: 采用格雷码映射时:

$$P_{e,\text{QPSK}} \approx Q \left(\sqrt{\frac{2E_b}{N_0}} \right) \quad (6)$$

核心参数如下：

- 符号速率 $R_s = 100$ Sym/s, 载波 $f_c = 1000$ Hz, 采样率 $f_s = 16 \times 10^3$ Hz
- 2PSK 波形 $s(t) = A \cos(2\pi f_c t)$ 或 $-A \cos(2\pi f_c t)$
- QPSK 波形 $s(t) = A_I \cos(2\pi f_c t) - A_Q \sin(2\pi f_c t)$, 其中 $A_I, A_Q \in \{+1, -1\}$

3 Matlab 仿真

3.1 主要参数说明

参数	数值
载波频率 f_c (Hz)	1000
符号速率 R_s (Sym/s)	100
采样频率 f_s (Hz)	16000
比特信噪比 E_b/N_0 (dB)	4
仿真次数	100
学号	08023214
频偏百分比	4%
系统主瓣带宽	200 Hz

表 1: 仿真参数设置表

3.2 仿真代码

```
clear; clc; close all;

%% 1. 参数设置与初始化
stu_num_str = '\08023214'; % 学号
fprintf('当前仿真学号：%s\n', stu_num_str);

% 将学号转换为二进制比特流 (BCD码)
info_bits_str = 'elec';
for i = 1:length(stu_num_str)
    digit = str2double(stu_num_str(i));
    info_bits_str = [info_bits_str, dec2bin(digit, 4)];
end
info_bits = info_bits_str - '\0'; % 转换为数值数组

% 系统参数
fc = 1000; % 载波频率 1kHz
Rs = 100; % 符号速率 100 Symbols/s
Ts = 1/Rs; % 符号周期
fs = 16*fc; % 采样率 16kHz
dt = 1/fs; % 采样时间间隔
```

```

EbNo_dB = 4; % 目标比特信噪比 (dB)
N_repeat = 100; % 蒙特卡洛仿真次数
freq_offset_pct = str2double(stu_num_str(end)); % 频偏百分比 (取学号末位)
if freq_offset_pct == 0, freq_offset_pct = 1; end
freq_offset = freq_offset_pct / 100;

fprintf(['仿真参数: fs=%dHz, fc=%dHz, Eb/N0=%ddB, 频偏=%d%%\n', ...
    fs, fc, EbNo_dB, freq_offset_pct);

%% 2. BPSK (2PSK) 系统仿真
fprintf('\n---BPSK系统仿真开始---\n');

% 2.1 BPSK 调制
bpsk_symbols = 2*info_bits - 1; % 0->-1, 1->+1
num_bits = length(info_bits);
samples_per_symbol = round(Ts/dt);

% 生成时间轴
total_samples = num_bits * samples_per_symbol;
t_bpsk = (0:total_samples-1) * dt;

% 生成调制信号
bpsk_signal = zeros(1, total_samples);
for i = 1:num_bits
    start_idx = (i-1)*samples_per_symbol + 1;
    end_idx = i*samples_per_symbol;
    % 矩形脉冲成型
    bpsk_signal(start_idx:end_idx) = bpsk_symbols(i) * cos(2*pi*fc*t_bpsk(start_idx:end_idx));
end

% 2.2 绘制波形与频谱
figure('Name', 'BPSK_Waveform_and_Spectrum', 'Color', 'w');
subplot(2,1,1);
plot(t_bpsk, bpsk_signal, 'LineWidth', 1);
xlim([0 10*Ts]); % 仅显示前10个符号
title('BPSK信号波形(部分)'), xlabel('时间(s)'), ylabel('幅度(V)');
grid on;

Nfft = 4096;
freq_axis = linspace(-fs/2, fs/2, Nfft);
bpsk_spectrum = fftshift(abs(fft(bpsk_signal, Nfft)/length(t_bpsk)));
subplot(2,1,2);
plot(freq_axis, 10*log10(bpsk_spectrum), 'LineWidth', 1);
title('BPSK信号频谱'), xlabel('频率(Hz)'), ylabel('功率谱密度(dB/Hz)');
grid on; xlim([-5000 5000]);
saveas(gcf, 'BPSK_combined.png'); % 保存组合图

% 2.3 信道与解调 (蒙特卡洛仿真)

```

```

Eb = sum(abs(bpsk_signal).^2) / (num_bits * fs); % 计算平均比特能量
NO = Eb / (10^(EbN0_dB/10));
noise_var = NO * fs / 2; % 实信号噪声方差

ber_bpsk_vec = zeros(1, N_repeat);
ber_bpsk_offset_vec = zeros(1, N_repeat);

% 预计算带频偏载波 (用于频偏仿真)
fc_offset = fc * (1 + freq_offset);
for i = 1:N_repeat
    noise = sqrt(noise_var) * randn(1, length(bpsk_signal));

    % --- 情况A: 无频偏 ---
    rx_signal = bpsk_signal + noise;
    demod_bits = zeros(1, num_bits);
    for j = 1:num_bits
        idx = ((j-1)*samples_per_symbol + 1) : (j*samples_per_symbol);
        % 相干解调
        metric = sum(rx_signal(idx) .* cos(2*pi*fc*t_bpsk(idx)));
        demod_bits(j) = metric > 0;
    end
    ber_bpsk_vec(i) = mean(info_bits ~= demod_bits);

    % --- 情况B: 有频偏 ---
    % 重新生成带频偏的发送信号
    bpsk_signal_offset = zeros(1, total_samples);
    for j = 1:num_bits
        idx = ((j-1)*samples_per_symbol + 1) : (j*samples_per_symbol);
        bpsk_signal_offset(idx) = bpsk_symbols(j) * cos(2*pi*fc_offset*t_bpsk(idx));
    end
    rx_signal_offset = bpsk_signal_offset + noise;

    demod_bits_offset = zeros(1, num_bits);
    for j = 1:num_bits
        idx = ((j-1)*samples_per_symbol + 1) : (j*samples_per_symbol);
        % 接收端仍使用标准载波解调 (导致失配)
        metric = sum(rx_signal_offset(idx) .* cos(2*pi*fc*t_bpsk(idx)));
        demod_bits_offset(j) = metric > 0;
    end
    ber_bpsk_offset_vec(i) = mean(info_bits ~= demod_bits_offset);
end

fprintf('BPSK 平均 BER (无频偏): %.6f\n', mean(ber_bpsk_vec));
fprintf('BPSK 平均 BER (有频偏): %.6f\n', mean(ber_bpsk_offset_vec));

%% 3. QPSK 系统仿真

```

```

fprintf('\\\\n---QPSK系统仿真开始---\\n');

% 3.1 QPSK 调制
% 确保比特数为偶数
if mod(length(info_bits), 2) ~= 0
    qpsk_info_bits = [info_bits, 0];
else
    qpsk_info_bits = info_bits;
end
num_symbols_qpsk = length(qpsk_info_bits) / 2;

% 串并转换
I_bits = qpsk_info_bits(1:2:end);
Q_bits = qpsk_info_bits(2:2:end);
I_sym = 2*I_bits - 1;
Q_sym = 2*Q_bits - 1;

t_qpsk = (0 : (num_symbols_qpsk*samples_per_symbol - 1)) * dt;
qpsk_signal = zeros(1, length(t_qpsk));

for i = 1:num_symbols_qpsk
    idx = ((i-1)*samples_per_symbol + 1) : (i*samples_per_symbol);
    carrier_I = cos(2*pi*fc*t_qpsk(idx));
    carrier_Q = sin(2*pi*fc*t_qpsk(idx));
    qpsk_signal(idx) = I_sym(i)*carrier_I - Q_sym(i)*carrier_Q;
end

% 3.2 绘制波形与频谱
figure('Name', 'QPSK_Waveform_and_Spectrum', 'Color', 'w');
subplot(2,1,1);
plot(t_qpsk, qpsk_signal, 'LineWidth', 1);
xlim([0 10*Ts]);
title('QPSK信号波形(部分)'); xlabel('时间(s)'); ylabel('幅度(V)');
grid on;

qpsk_spectrum = fftshift(abs(fft(qpsk_signal, Nfft)/length(t_qpsk)));
subplot(2,1,2);
plot(freq_axis, 10*log10(qpsk_spectrum), 'LineWidth', 1);
title('QPSK信号频谱'); xlabel('频率(Hz)'); ylabel('功率谱密度(dB/Hz)');
grid on; xlim([-5000 5000]);
saveas(gcf, 'QPSK_combined.png');

% 3.3 信道与解调
ber_qpsk_vec = zeros(1, N_repeat);
ber_qpsk_offset_vec = zeros(1, N_repeat);

for i = 1:N_repeat
    noise = sqrt(noise_var) * randn(1, length(qpsk_signal));

```

```
% --- 无频偏 ---
rx_signal = qpsk_signal + noise;
demod_bits_temp = [];
for j = 1:num_symbols_qpsk
    idx = ((j-1)*samples_per_symbol + 1) : (j*samples_per_symbol);
    % I路解调
    r_I = sum(rx_signal(idx) .* cos(2*pi*fc*t_qpsk(idx)));
    % Q路解调 (注意sin项前面的负号, 与调制对应)
    r_Q = sum(rx_signal(idx) .* -sin(2*pi*fc*t_qpsk(idx)));
    demod_bits_temp = [demod_bits_temp, (r_I>0), (r_Q>0)];
end
ber_qpsk_vec(i) = mean(qpsk_info_bits == demod_bits_temp);

% --- 有频偏 ---
qpsk_signal_offset = zeros(1, length(t_qpsk));
for j = 1:num_symbols_qpsk
    idx = ((j-1)*samples_per_symbol + 1) : (j*samples_per_symbol);
    carrier_I = cos(2*pi*fc_offset*t_qpsk(idx));
    carrier_Q = sin(2*pi*fc_offset*t_qpsk(idx));
    qpsk_signal_offset(idx) = I_sym(j)*carrier_I - Q_sym(j)*carrier_Q;
end
rx_signal_offset = qpsk_signal_offset + noise;

demod_bits_temp = [];
for j = 1:num_symbols_qpsk
    idx = ((j-1)*samples_per_symbol + 1) : (j*samples_per_symbol);
    r_I = sum(rx_signal_offset(idx) .* cos(2*pi*fc*t_qpsk(idx)));
    r_Q = sum(rx_signal_offset(idx) .* -sin(2*pi*fc*t_qpsk(idx)));
    demod_bits_temp = [demod_bits_temp, (r_I>0), (r_Q>0)];
end
ber_qpsk_offset_vec(i) = mean(qpsk_info_bits == demod_bits_temp);
end

fprintf('QPSK\u2225平均\u2225BER\u2225(无频偏):\u2225%.6f\u2225\n', mean(ber_qpsk_vec));
fprintf('QPSK\u2225平均\u2225BER\u2225(有频偏):\u2225%.6f\u2225\n', mean(ber_qpsk_offset_vec));
```

3.3 仿真结果截图

Case 1: 2PSK 输出波形与频谱

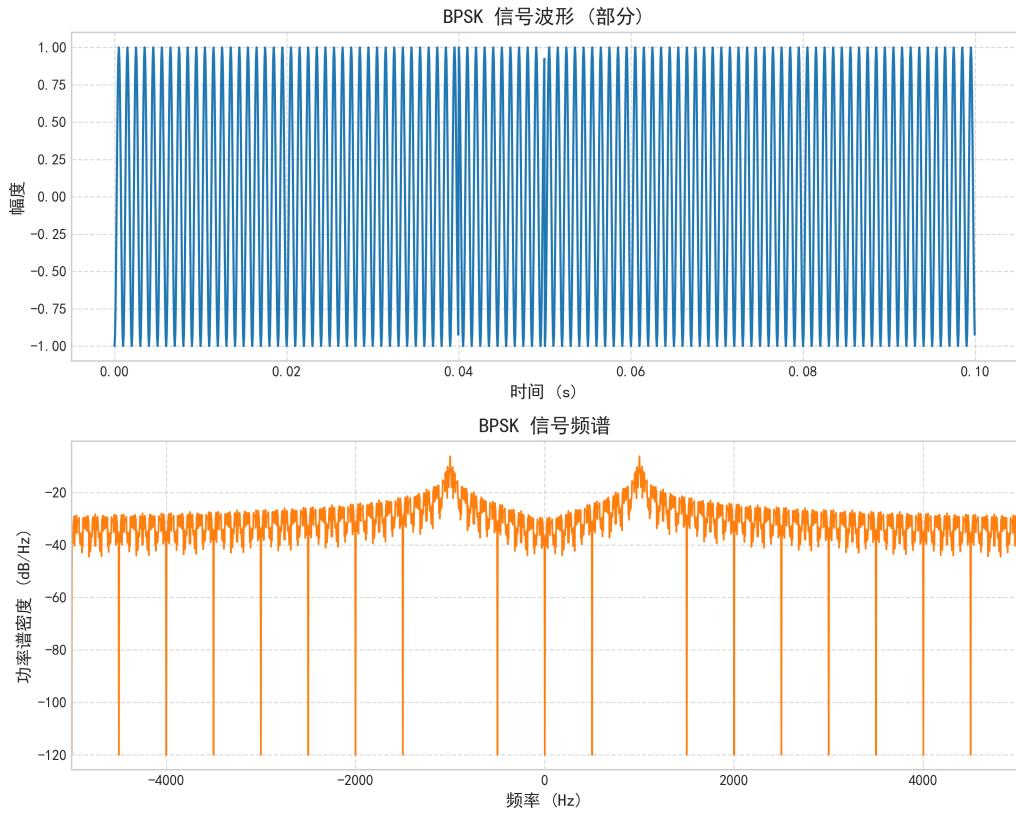


图 2: 2PSK 输出信号波形与频谱

Case 2: QPSK 输出波形与频谱

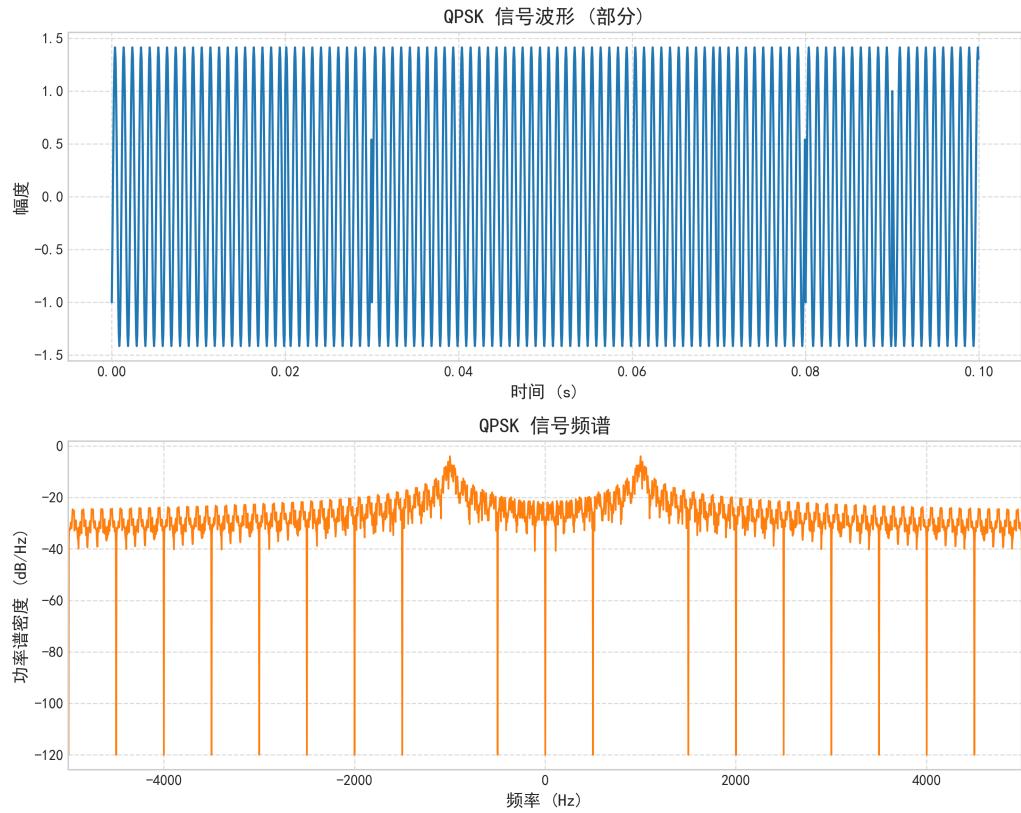


图 3: QPSK 输出信号波形与频谱

3.4 仿真数值结果

2PSK 系统性能

指标	无频偏	4% 频偏
判决后学号 (首次)	08421214	461159151013
首次 BER	0.062500	0.562500
首次 SER	0.062500	0.562500
100 次平均 BER	0.014375	0.496250
100 次平均 SER	0.014375	0.496250

表 2: 2PSK 仿真结果 ($E_b/N_0 = 4\text{dB}$)

QPSK 系统性能

指标	无频偏	4% 频偏
判决后学号 (首次)	08023214	31061414906
首次 BER	0.000000	0.468750
首次 SER	0.000000	0.687500
100 次平均 BER	0.012500	0.508437
100 次平均 SER	0.025000	0.771250

表 3: QPSK 仿真结果 ($E_b/N_0 = 4\text{dB}$)

4 仿真结果分析

对仿真结果进行分析，可以得出如下结论：

- 无频偏情况：**在 $E_b/N_0 = 4\text{dB}$ 的条件下，两种调制方式的误码率都较低。QPSK 在首次仿真中表现完美 (BER=0)，2PSK 有少量错误，这在随机噪声下是正常波动。从百次平均结果看，两者的平均 BER 性能相近，这与理论上 2PSK 和 QPSK 在相同 E_b/N_0 下有相似的比特差错性能的结论一致。QPSK 的频带利用率是 2PSK 的两倍。此外，由仿真设置可知符号速率 $R_s = 100 \text{ Baud}$ ，且采用矩形脉冲成型，理论上信号的主瓣带宽（零点到零点）为 $2R_s = 200 \text{ Hz}$ ，这一特征在频谱图中清晰可见。
- 有频偏情况：**当接收端存在 4% 的载波频偏时，系统性能急剧恶化。两种调制方式的 BER 都飙升至接近 0.5，这意味着信号几乎无法被正确解调，判决后的学号也完全错误。这是因为相干解调对载波的相位和频率高度敏感，频偏导致本地载波与接收信号的载波分量不正交，从而产生严重的码间干扰。
- 实验结果表明，载波同步在相干解调系统中至关重要。即使是百分之几的频偏，也足以导致通信失败。

在上述分析之外，我利用星座图以及误码率与信噪比的关系曲线对仿真结果作了进一步的分析。

4.1 星座图

星座图是观察数字调制信号质量的有力工具。它将解调后的同相 (I) 和正交 (Q) 分量在二维平面上进行采样展示，可以直观地看到噪声、频偏等损伤对信号判决的影响。

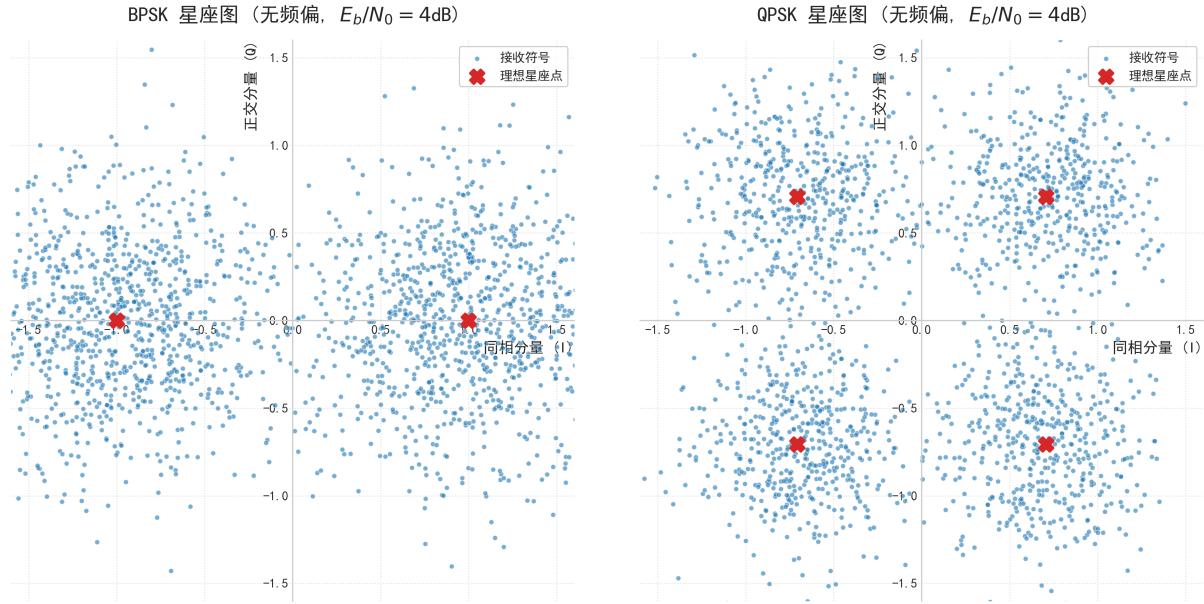


图 4: 无频偏下的 BPSK (左) 与 QPSK (右) 星座图 ($E_b/N_0 = 4\text{dB}$)

图 4 展示了在无频偏、信噪比为 4dB 时接收到的信号星座点。可以看到，接收信号点围绕着理想的星座位置（红叉）形成“云团”，这是高斯白噪声影响的典型表现。尽管存在噪声，但判决区域清晰，BPSK 的两个点集和 QPSK 的四个点集可以被明确区分，因此误码率较低。

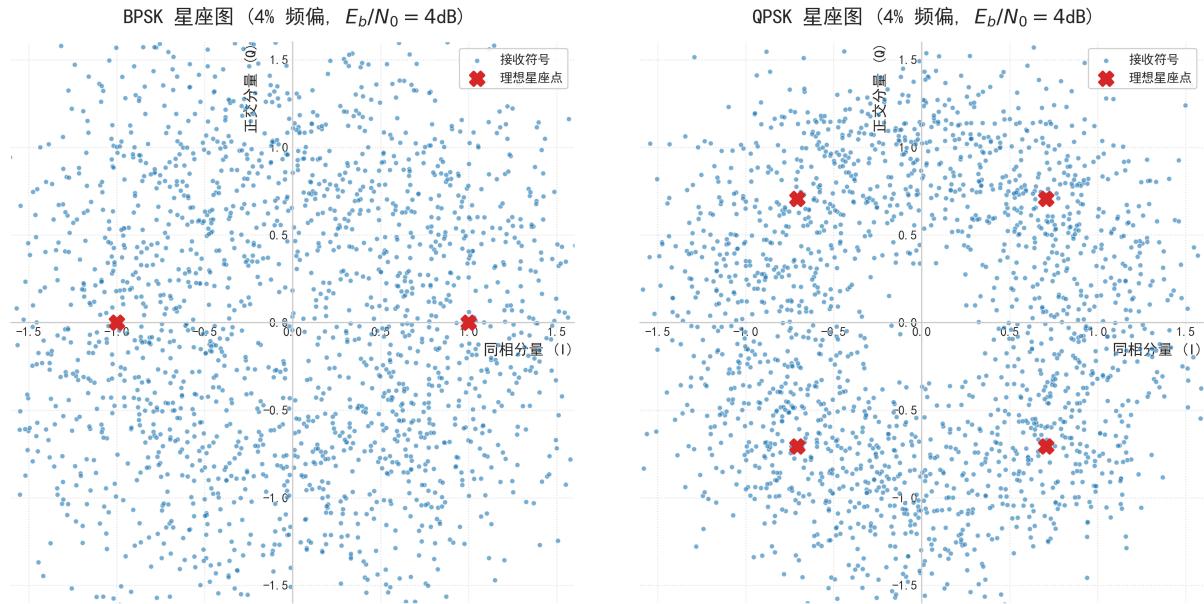


图 5: 存在 4% 频偏下的 BPSK (左) 与 QPSK (右) 星座图 ($E_b/N_0 = 4\text{dB}$)

图5展示了存在 4% 载波频偏时的灾难性影响。由于收发两端载波的频率失配，在相干解调时引入了一个随时间累积的相位误差项 $\Delta\omega \cdot t$ 。这导致接收到的星座点发生了旋转。随着符号序列的进行，旋转角度越来越大，最终形成环状轨迹。这使得判决器完全无法正确区分原始符号，导致误码率急剧上升到接近 0.5 的水平，这与之前表格中的数据完全吻合。

4.2 误码率和信噪比的关系曲线

误码率 (BER) 与信噪比 (E_b/N_0) 的关系是衡量一个通信系统可靠性的核心指标。我们通过在 0dB 到 10dB 的信噪比范围内进行仿真，绘制了系统的性能曲线。

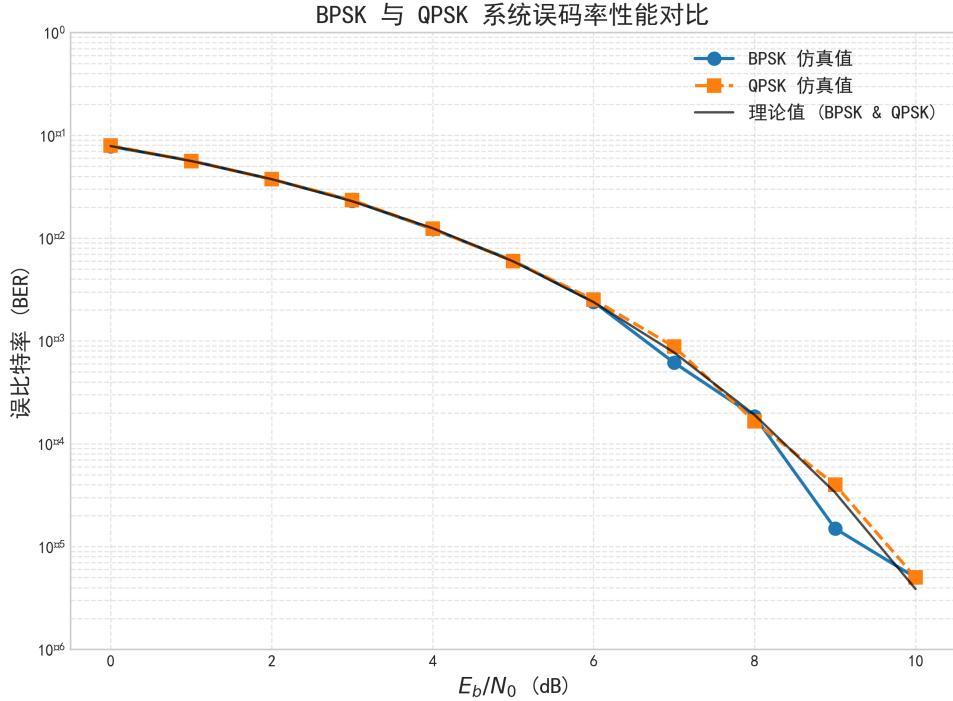


图 6: BPSK 与 QPSK 系统在 AWGN 信道下的 BER 性能曲线

从图 6 中可以得出以下关键结论：

- 仿真与理论一致性：**仿真得到的 BPSK 和 QPSK 误码率曲线与理论曲线（红色实线）高度吻合，这验证了实验的正确性。
- BPSK 与 QPSK 性能对比：**实验结果表明，在相同的比特信噪比 E_b/N_0 条件下，BPSK 和（采用格雷码的）QPSK 具有几乎完全相同的误比特率性能。这印证了通信原理中的经典结论，即 QPSK 通过牺牲相位来换取频带利用率的加倍，而其比特可靠性并未降低。
- 信噪比的影响：**曲线显示，随着 E_b/N_0 的增加，误码率呈指数级下降。例如，将信噪比从 4dB 提升到 8dB，误码率可以改善超过两个数量级。这直观地展示了信噪比对通信质量的决定性作用。

参考资料

- 《通信原理》，东南大学通信原理教学团队，人民邮电出版社