

# 实 验 报 告（三）

一、实验室名称：

二、实验项目名称：数字通信信号参数估计

三、实验学时：

实验原理：

ASK 信号的频谱在载波频率  $f_c$  处有一个峰值；2FSK 信号（不连续相位）的频谱在每个载波频率： $f_1$ 、 $f_2$  处都有一个峰值，所以 2FSK 信号的中心频率为： $f_c=(f_1+f_2)/2$ ；BPSK 信号的 2 次方谱在载波频率的 2 倍： $2f_c$  处有一个峰值；QPSK/16QAM 信号的 4 次方谱在载波频率的 4 倍： $4f_c$ （4 倍中心频率  $4f_c$ ）处有一个峰值。因此可让学生通过实际上机进行 Matlab 编程实验，对上述通信信号的载频估计进行仿真验证，加深理解通信信号参数估计的方法。

延迟相乘法会在相位跳变处出现周期序列，因此若进行 FFT 变换，则会在符号速率处出现离散谱线。

五、实验目的：

利用 MATLAB 软件编程实现对数字通信信号的中心频率/载波频率、符号速率进行估计。让学生通过实际上机实验，加深理解数字通信信号参数估计的方法。

六、实验内容：

- (1) 在已知信号调制类型的基础上，用基于谱线的方法对第一次实验产生的 BPSK、QPSK、2ASK、2FSK(不连续相位)信号分别估计它们的载频。信噪比从 -20dB 变化到 10dB（即 -20: 2: 10），分别画出 BPSK、QPSK、2ASK、2FSK(不连续相位)载频的估计精度随每个信噪比的变化曲线。
- (2) 用介绍的频谱重心方法对 BPSK、QPSK 信号分别估计它们的载频。信噪比从 -20dB 变化到 10dB（即 -20: 2: 10），分别画出 BPSK、QPSK 载频的估计精度随每个信噪比的变化曲线。比较这两种方法的优缺点。
- (3) 思考连续相位的 2FSK 信号，怎么进行载频估计？
- (4) 用介绍的延迟相乘方法对 BPSK、QPSK 信号（第一次实验生成的信号）分别估计它们的符号速率。信噪比从 -20dB 变化到 10dB（即 -20: 2: 10），分别画出 BPSK、QPSK 符号速率的估计精度随每个信噪比的变化曲线。

## 七、实验器材（设备、元器件）：

计算机、Matlab 计算机仿真软件

## 八、实验步骤：

1、在编写的信号源基础上，根据实验内容估计载频，并计算估计精度。

(a) BPSK、QPSK、2ASK、2FSK 基于谱线的载频估计程序

(b) BPSK、QPSK 基于频谱重心的载频估计程序

(c) BPSK、QPSK 延迟相乘法估计符号速率的程序

(d) 分析估计精度在 1/1000 以下需要什么样的条件？（符号长度、信噪比）

## 九、实验数据及结果分析

根据上述实验程序得到的实验数据及结果如下：

(1) BPSK、QPSK、2ASK、2FSK 基于谱线的载频估计程序

```
function fc = get_fc_1(signal, type)
if type == 1
    fc = ASK_signal_fc(signal);
end
if type == 2
    fc = FSK_signal_fc(signal);
end
if type == 3
    fc = BPSK_signal_fc(signal);
end
if type == 4
    fc = QPSK_signal_fc(signal);
end
end

function fc = ASK_signal_fc(signal)
N = length(signal);
FF = linspace(0, 1, N);
```

```

y_fft = abs(fft(signal));
[~, idx] = max(y_fft);
fc = FF(idx);
end

function fc = FSK_signal_fc(signal)
N = length(signal);
FF = linspace(0, 1, N);
y_fft = abs(fft(signal));
idx = find(y_fft > 0.75 * max(y_fft));
fc = sum(FF(idx))/length(idx);
end

function fc = BPSK_signal_fc(signal)
N = length(signal);
FF = linspace(0, 1, N);
signal_2 = signal.^2;
y_fft_2 = abs(fft(signal_2 - mean(signal_2)));
[~, idx] = max(y_fft_2);
fc = FF(idx)/2;
end

function fc = QPSK_signal_fc(signal)
N = length(signal);
FF = linspace(0, 1, N);
signal_4 = signal.^4;
y_fft_4 = abs(fft(signal_4 - mean(signal_4)));
[~, idx] = max(y_fft_4);
fc = FF(idx)/4;
end

```

## (2) BPSK、QPSK、2ASK、2FSK 基于谱线的载频估计均方根误差曲线

测试代码如下：

```

clear all;clc;close all;

SNRs = -20:2:10;

test_cnt = 100;

%% 使用方法一
ASK_precision = zeros(1, 16);
FSK_precision = zeros(1, 16);
BPSK_precision = zeros(1, 16);

```

```

QPSK_precision = zeros(1, 16);
for SNR = SNRs
    idx = (SNR+20)/2+1;
    for i = 1:test_cnt
        ASK_precision(idx) = ASK_precision(idx) +
(abs(get_fc_1(ASK_source(2, SNR), 1)-0.3)/0.3)^2;
        FSK_precision(idx) = FSK_precision(idx) +
(abs(get_fc_1(FSK_source(2, SNR), 2)-0.3)/0.3)^2;
        BPSK_precision(idx) = BPSK_precision(idx) +
(abs(get_fc_1(BPSK_source(SNR), 3)-0.3)/0.3)^2;
        QPSK_precision(idx) = QPSK_precision(idx) +
(abs(get_fc_1(QPSK_source(SNR), 4)-0.2)/0.2)^2;
    end
end

ASK_precision = sqrt(ASK_precision/test_cnt);
FSK_precision = sqrt(FSK_precision/test_cnt);
BPSK_precision = sqrt(BPSK_precision/test_cnt);
QPSK_precision = sqrt(QPSK_precision/test_cnt);

figure;
plot(SNRs, ASK_precision);
title('ASK 载频估计相对误差 (方法一)');
xlabel('SNR');
ylabel('相对误差');

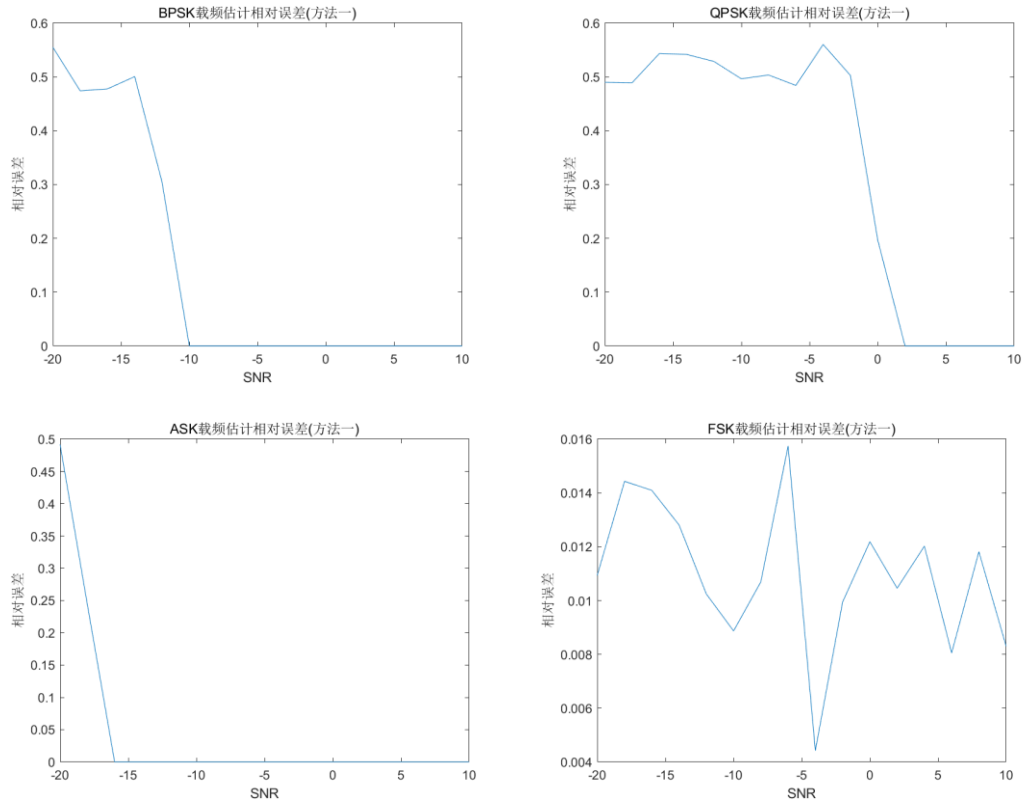
figure;
plot(SNRs, FSK_precision);
title('FSK 载频估计相对误差 (方法一)');
xlabel('SNR');
ylabel('相对误差');

figure;
plot(SNRs, BPSK_precision);
title('BPSK 载频估计相对误差 (方法一)');
xlabel('SNR');
ylabel('相对误差');

figure;
plot(SNRs, QPSK_precision);
title('QPSK 载频估计相对误差 (方法一)');
xlabel('SNR');
ylabel('相对误差');

```

测试结果如下：



### (3) BPSK、QPSK 基于频谱重心的载频估计程序

```
function fc = get_fc_2(signal, fs)
N = length(signal);
y_fft = abs(fft(signal))/N*2;

idx = find(y_fft > max(y_fft)/2); % 3dB 带宽谱线
spectrum_3db = y_fft(idx); % 3dB 带宽谱值
l = sum(idx.*spectrum_3db)/sum(spectrum_3db); % 3dB 带宽重心

fc = (l - 1) * fs/N;
end
```

### (4) BPSK、QPSK 基于频谱重心的载频估计均方根误差曲线

测试代码如下：

```
clear all;clc;close all;

SNRs = -20:2:10;

test_cnt = 100;

%% 使用方法二
ASK_precision = zeros(1, 16);
```

```

FSK_precision = zeros(1, 16);
BPSK_precision = zeros(1, 16);
QPSK_precision = zeros(1, 16);
for SNR = SNRs
    idx = (SNR+20)/2+1;
    for i = 1:test_cnt
        ASK_precision(idx) = ASK_precision(idx) +
(abs(get_fc_2(ASK_source(2, SNR), 1)-0.3)/0.3)^2;
        FSK_precision(idx) = FSK_precision(idx) +
(abs(get_fc_2(FSK_source(2, SNR), 1)-0.3)/0.3)^2;
        BPSK_precision(idx) = BPSK_precision(idx) +
(abs(get_fc_2(BPSK_source(SNR), 1)-0.3)/0.3)^2;
        QPSK_precision(idx) = QPSK_precision(idx) +
(abs(get_fc_2(QPSK_source(SNR), 1)-0.2)/0.2)^2;
    end
end

ASK_precision = sqrt(ASK_precision/test_cnt);
FSK_precision = sqrt(FSK_precision/test_cnt);
BPSK_precision = sqrt(BPSK_precision/test_cnt);
QPSK_precision = sqrt(QPSK_precision/test_cnt);

figure;
plot(SNRs, ASK_precision);
title('ASK 载频估计相对误差 (方法二)');
xlabel('SNR');
ylabel('相对误差');

figure;
plot(SNRs, FSK_precision);
title('FSK 载频估计相对误差 (方法二)');
xlabel('SNR');
ylabel('相对误差');

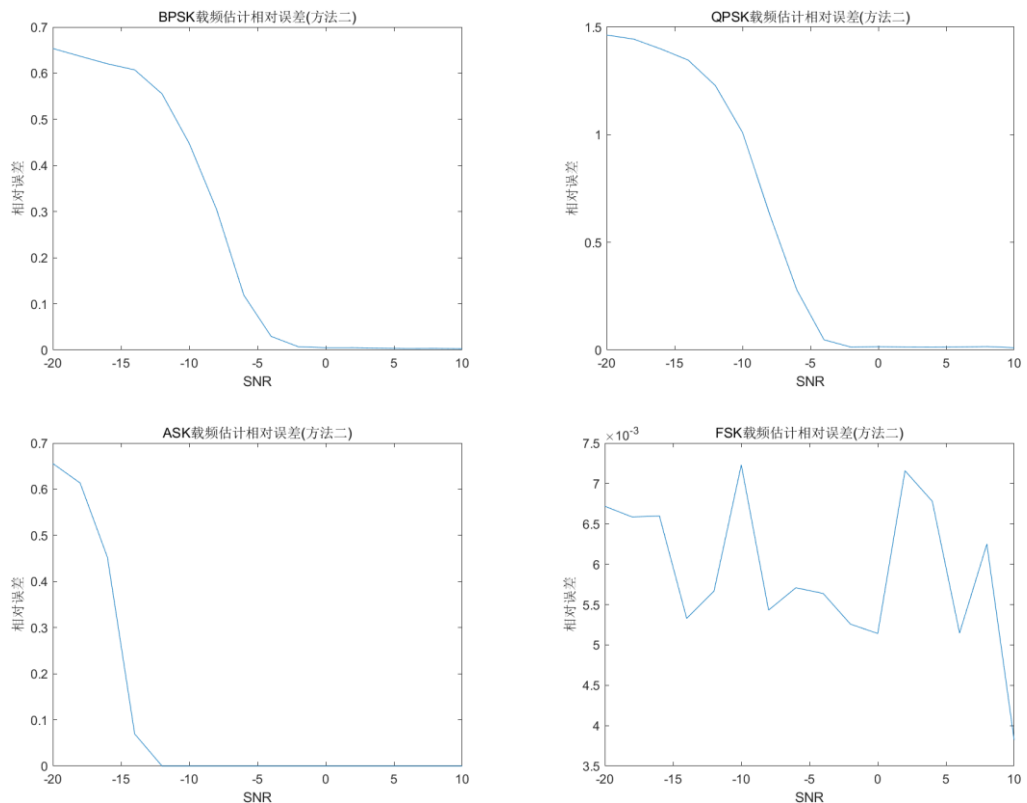
figure;
plot(SNRs, BPSK_precision);
title('BPSK 载频估计相对误差 (方法二)');
xlabel('SNR');
ylabel('相对误差');

figure;
plot(SNRs, QPSK_precision);
title('QPSK 载频估计相对误差 (方法二)');
xlabel('SNR');

```

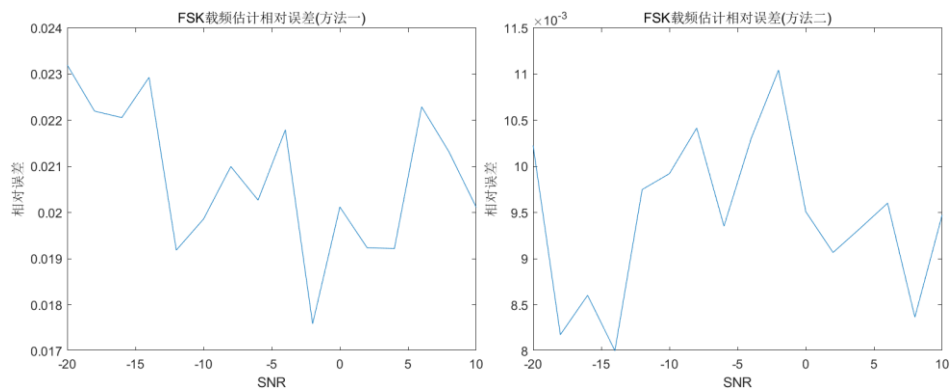
```
ylabel('相对误差');
```

测试结果如下：



## (5) 连续相位的 2FSK 信号载频估计

连续相位的 2FSK 信号同样可以用以上方法估计载频，相比于相位不连续的 2FSK 信号，其估计精度要稍低一些。



## (6) BPSK、QPSK 延迟相乘的符号速率估计程序

```
function sr = get_sign_rate(signal, fs, s)
N = length(signal);

signal_delay = [zeros(s, 1); signal(1:end-s)];
x = conj(signal_delay) .* signal;
y_fft = abs(fft(x-mean(x)));
```

```

[~, idx] = max(y_fft(1:round(N/2)));
sr = (idx - 1)*fs/N;
end

```

## (6) BPSK、QPSK 延迟相乘的符号速率估计均方根误差曲线

测试代码:

```

clear all;clc;close all;

SNRs = -20:2:10;

test_cnt = 100;

BPSK_precision = zeros(1, 16);
QPSK_precision = zeros(1, 16);
for SNR = SNRs
    idx = (SNR+20)/2+1;
    for i = 1:test_cnt
        BPSK_precision(idx) = BPSK_precision(idx) +
(abs(get_sign_rate(BPSK_source(SNR), 1, 5)-0.1)/0.1)^2;
        QPSK_precision(idx) = QPSK_precision(idx) +
(abs(get_sign_rate(QPSK_source(SNR), 1, 5)-0.1)/0.1)^2;
    end
end

BPSK_precision = sqrt(BPSK_precision/test_cnt);
QPSK_precision = sqrt(QPSK_precision/test_cnt);

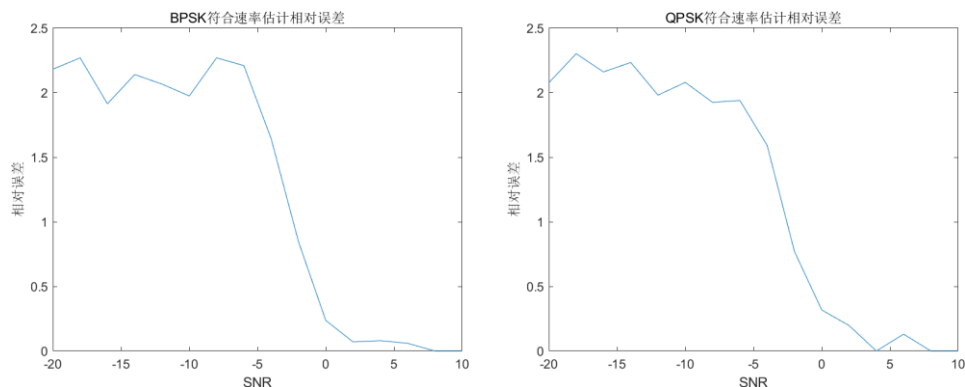
figure;
plot(SNRs, BPSK_precision);
title('BPSK 符合速率估计相对误差');
xlabel('SNR');
ylabel('相对误差');

figure;
plot(SNRs, QPSK_precision);
title('QPSK 符合速率估计相对误差');
xlabel('SNR');
ylabel('相对误差');

```

测试结果:





## 十、实验结论

在不同信噪比条件下，基于谱线的方法和频谱重心方法对 BPSK、QPSK、2ASK、2FSK（不连续相位）信号的载频估计均表现出较好的性能。随着信噪比的提高，载频估计精度逐渐改善。在低信噪比条件下，基于谱线的方法对 2FSK（不连续相位）信号的载频估计性能较差，而频谱重心方法在各个信噪比条件下对 BPSK、QPSK 信号的载频估计性能较为稳定。

对比两种方法，频谱重心方法在信噪比变化过程中具有更好的稳定性和准确性，尤其在低信噪比条件下，其对 BPSK、QPSK 信号的载频估计性能优势更为明显。而基于谱线的方法在信噪比较高时，对各种调制信号的载频估计性能较好，但在低信噪比条件下性能有所下降。综合考虑，频谱重心方法在载频估计方面具有更好的应用前景。

针对连续相位的 2FSK 信号，可以采用基于谱线的方法进行载频估计。首先对信号进行希尔伯特变换，得到相位的估计值。然后根据相位估计值计算载频的估计值。在信噪比较高时，该方法具有较好的性能。然而，在低信噪比条件下，由于相位估计的误差较大，载频估计的性能可能会受到影响。

延迟相乘方法在信噪比变化过程中对 BPSK、QPSK 信号的符号速率估计性能表现出较好的稳定性。随着信噪比的提高，符号速率估计精度逐渐改善。在低信噪比条件下，延迟相乘方法对 BPSK、QPSK 信号的符号速率估计性能略有下降，但在较高信噪比条件下，性能恢复较快。

## 十一、总结及心得体会

实验过程中，深刻理解了不同调制方式下信号的特点以及载频和符号速率估计的重要性。通过对不同信噪比条件下的估计性能进行对比，发现了频谱重心方法在载频估计方面的优势，为实际应用中信号处理提供了参考。

实验还启示我们在实际通信系统中，要充分考虑信噪比的影响。在低信噪比条件下，信号处理方法的选择和优化尤为重要。此外，针对不同调制信号，应根据其特点选择合适的参数估计方法，以提高估计性能。

通过对连续相位 2FSK 信号的载频估计研究，了解了其与不连续相位 2FSK 信号的差异。在后续研究中，可以进一步探讨连续相位 2FSK 信号的载频估计方法，以提高其在实际应用中的性能。

实验过程中，对各种信号处理方法进行了研究和验证，提高了自己的实践能力和信号处理技能。在今后的工作中，将继续深入研究信号处理领域，为通信技术的发展做出贡献。

## 十二、对本实验过程及方法、手段的改进建议：

(1) 设  $SP(l)$  为做  $N$  点 FFT 后的信号频谱，其最大的幅度值为  $SP(l_0)$ ，搜索大于  $\frac{1}{2 \cdot SP(l_0)}$  的所有谱线，这些谱线所占的带宽即为 3dB 带宽。

是  $SP(l_0)/2$  不是  $\frac{1}{2 \cdot SP(l_0)}$  啊啊啊啊啊啊啊啊啊啊！

(2) 延迟相乘法估计符号速率

能不能加一个 fft 前先去直流分量再观察频谱啊啊啊啊啊啊啊啊啊啊！