实　验　报　告（二）

一、实验室名称：

二、实验项目名称：典型数字通信信号调制识别

三、实验学时：

实验原理：

BPSK、QPSK、FSK、ASK 数字通信信号具有不同的频谱特征，如 BPSK 平方和四次方后的傅立叶变换出现单根离散谱线，QPSK 四次方后才有单根离散谱线等。因此针对这些信号特征，可通过设置特征门限区分不同通信信号，达到信号调制识别目的。因此可让学生通过实际上机 Matlab 编程实验，对上述通信信号的特征进行仿真验证，加深理解不同通信信号的调制识别方法。

五、实验目的：

利用 MATLAB 软件编程提取数字通信信号的频谱、二次方谱、四次方谱等特征，同时使用决策树的分类方法进行调制信号的识别。让学生通过实际上机实验，加深理解不同数字通信信号的特点。

六、实验内容：

（1） 把上次实验产生的 BPSK、QPSK、2ASK、2FSK(不连续相位)信号分别画出它们的频谱、二次方谱和四次方谱，描述不同信号谱特征之间的差异。

（2） 分别用介绍的两种方法编程提取这四种信号的频谱、二次方谱、四次方谱的谱峰个数（根据每种信号的特点分别提取频谱或二次方谱或四次方谱的谱峰个数）。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)用两种方法获得谱峰个数特征随每个信噪比的变化曲线，然后设定区分门限。比较这两种方法的优缺点。

（3） 选择其中一种方法提取谱峰特征，设计识别决策树，并编写程序实现。画出 BPSK、QPSK、2ASK、2FSK(不连续相位)信号的正确识别率随信噪比（-20：2：10）的变化曲线。说明在至少信噪比下能保证每种信号的正确识别率均在 90%以上。

（4） 思考连续相位的 2FSK 信号，怎么进行调制类型识别？

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，根据实验内容提取信号特征并进行调制识别。

(a) BPSK、QPSK、2ASK、2FSK 频谱特征提取程序

(b) BPSK、QPSK、2ASK、2FSK 数字通信信号调制识别程序

（c）画出设计的决策树示意图，说明你的设计理由

九、实验数据及结果分析

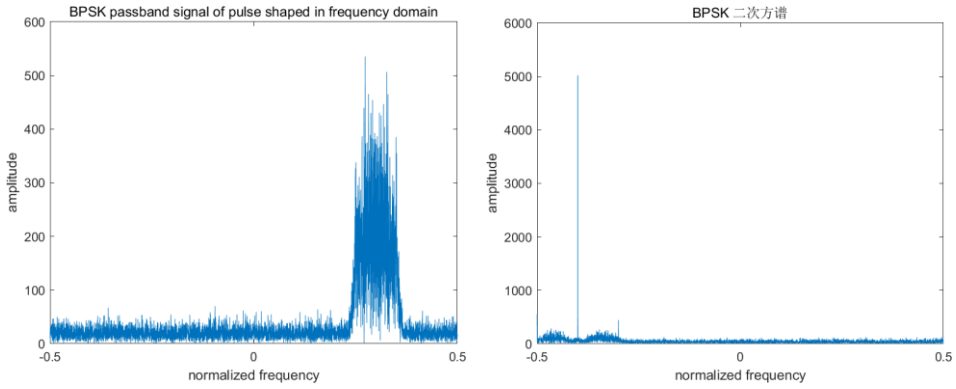
根据上述实验程序得到的实验数据及结果如下：

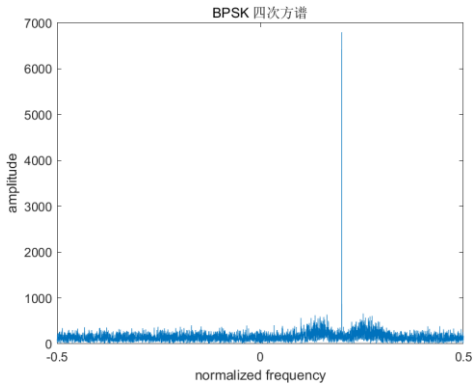
（1） BPSK、QPSK、2ASK、2FSK 频谱特征

```
clear all;clc;close all;

ASK_signal = ASK_source(2, 10);
BPSK_signal = BPSK_source(10);
FSK_signal = FSK_source(2, 10);
QPSK_signal = QPSK_source(10);

plot_spectrum(ASK_signal, 'ASK');
plot_spectrum(BPSK_signal, 'BPSK');
plot_spectrum(FSK_signal, 'FSK');
plot_spectrum(QPSK_signal, 'QPSK');

%% 功能函数

% 绘制频谱
function plot_spectrum(signal, signal_name)
N = length(signal);
FF = linspace(-0.5, 0.5, N); %% 频谱观察范围为-fs/2~fs/2 之间

[y_fft, y_fft_2, y_fft_4] = get_spectrum(signal);

figure;
plot(FF, y_fft);
title([signal_name, ' ', 'passband signal of pulse shaped in
frequency domain']);
xlabel('normalized frequency');
ylabel('amplitude');
```

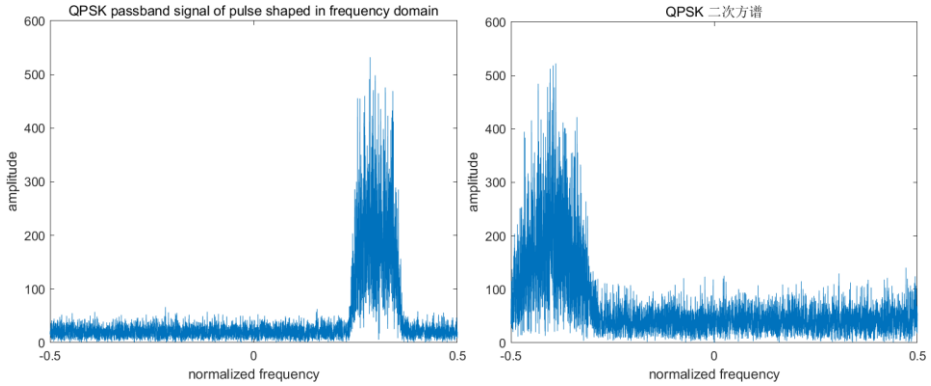
```
figure;
plot(FF, y_fft_2);
title([signal_name, ' ', '二次方谱']);
xlabel('normalized frequency');
ylabel('amplitude');

figure;
plot(FF, y_fft_4);
title([signal_name, ' ', '四次方谱']);
xlabel('normalized frequency');
ylabel('amplitude');
end

% 获取频谱
function [y_fft, y_fft_2, y_fft_4] = get_spectrum(signal)
% 频谱
y_fft = abs(fftshift(fft(signal)));

% 二次方谱
signal_2 = signal.^2;
y_fft_2 = abs(fftshift(fft(signal_2 -mean(signal_2))));

% 四次方谱
signal_4 = signal_2.^2;
y_fft_4 = abs(fftshift(fft(signal_4 -mean(signal_4))));
end
```

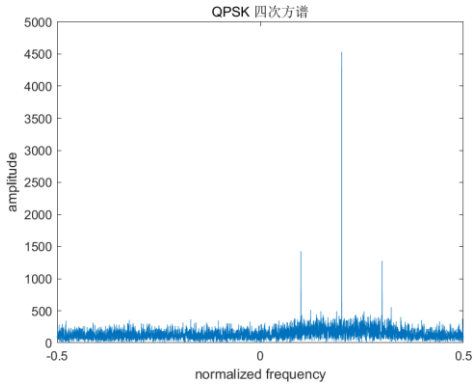
BPSK



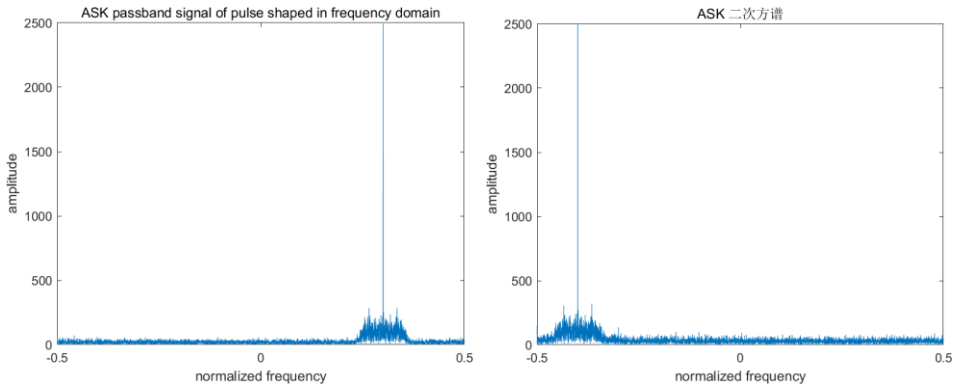

BPSK 四次方谱
amplitude
normalized frequency

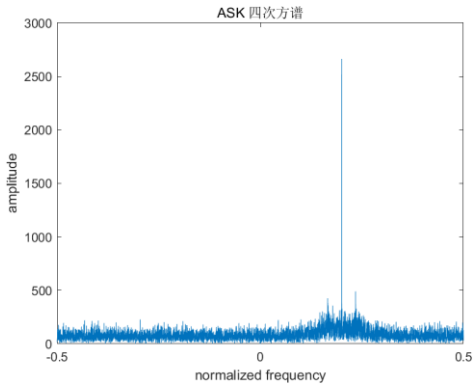
QPSK


QPSK passband signal of pulse shaped in frequency domain
amplitude
normalized frequency
QPSK 二次方谱
amplitude
normalized frequency

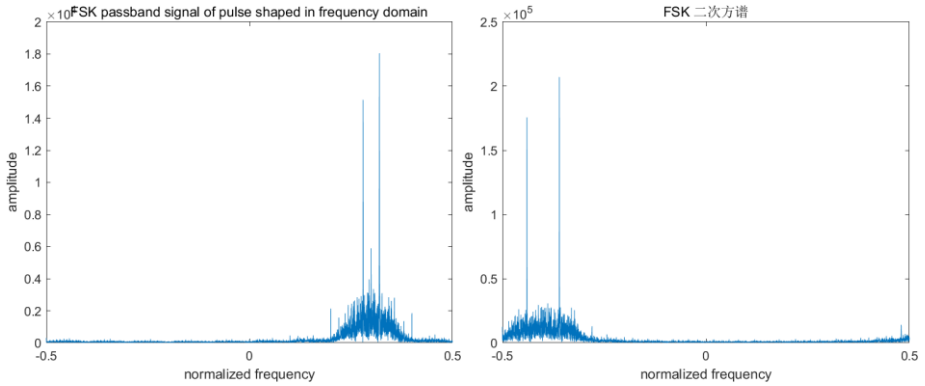

QPSK 四次方谱
amplitude
normalized frequency

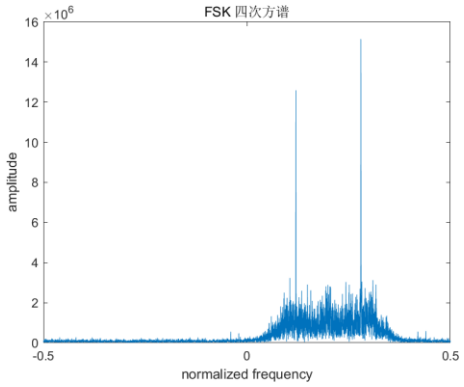
2ASK


ASK passband signal of pulse shaped in frequency domain
amplitude
normalized frequency
ASK 二次方谱
amplitude
normalized frequency



2FSK





（2） BPSK、QPSK、2ASK、2FSK 频谱特征提取程序

方法一

```
function cnt = feature_extraction_1(signal, thes_1, thes_2,
thes_4)
[y_fft, y_fft_2, y_fft_4] = get_spectrum(signal);

cnt = zeros(3, 1);
cnt(1) = length(find(y_fft > thes_1 * max(y_fft)));
cnt(2) = length(find(y_fft_2 > thes_2 * max(y_fft_2)));
cnt(3) = length(find(y_fft_4 > thes_4 * max(y_fft_4)));
end
```

方法二

```
function cnt = feature_extraction_2(signal, thes)
N = 40;

[y_fft, y_fft_2, y_fft_4] = get_spectrum(signal);
R_fft = y_fft./([zeros(N, 1);y_fft(1:end-N)] +
[y_fft(N+1:end); zeros(N, 1)]);
R_fft_2 = y_fft_2./([zeros(N, 1);y_fft_2(1:end-N)] +
[y_fft_2(N+1:end); zeros(N, 1)]);
R_fft_4 = y_fft_4./([zeros(N, 1);y_fft_4(1:end-N)] +
[y_fft_4(N+1:end); zeros(N, 1)]);

cnt = zeros(3, 1);
cnt(1) = length(find(R_fft > thes));
cnt(2) = length(find(R_fft_2 > thes));
cnt(3) = length(find(R_fft_4 > thes));
end
```

测试代码

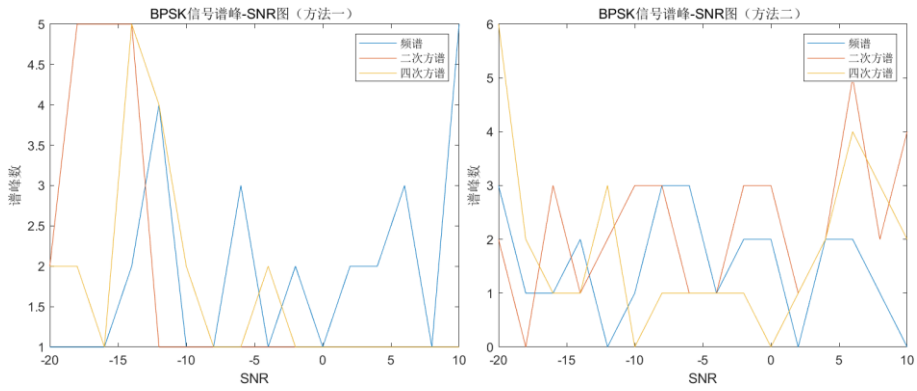
```
clear all;clc;close all;

SNRs = -20:2:10;

%% 使用方法一
ASK_cnt = zeros(3, 16);
BPSK_cnt = zeros(3, 16);
FSK_cnt = zeros(3, 16);
QPSK_cnt = zeros(3, 16);
for SNR = SNRs
    ASK_cnt(:, (SNR+20)/2+1) = feature_extraction_1(ASK_source(2, SNR), 0.75,
0.7, 0.7);
    BPSK_cnt(:, (SNR+20)/2+1) = feature_extraction_1(BPSK_source(SNR), 0.75,
0.7, 0.7);
    FSK_cnt(:, (SNR+20)/2+1) = feature_extraction_1(FSK_source(2, SNR), 0.75,
0.7, 0.7);
    QPSK_cnt(:, (SNR+20)/2+1) = feature_extraction_1(QPSK_source(SNR), 0.75,
0.7, 0.7);
end

figure;
plot(SNRs, ASK_cnt);
title('ASK 信号谱峰-SNR 图（方法一）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')
```

```
figure;
plot(SNRs, BPSK_cnt);
title('BPSK 信号谱峰-SNR 图（方法一）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')

figure;
plot(SNRs, FSK_cnt);
title('FSK 信号谱峰-SNR 图（方法一）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')

figure;
plot(SNRs, QPSK_cnt);
title('QPSK 信号谱峰-SNR 图（方法一）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')

%% 使用方法二
ASK_cnt = zeros(3, 16);
BPSK_cnt = zeros(3, 16);
FSK_cnt = zeros(3, 16);
QPSK_cnt = zeros(3, 16);
for SNR = SNRs
    ASK_cnt(:, (SNR+20)/2+1) = feature_extraction_2(ASK_source(2, SNR), 8);
    BPSK_cnt(:, (SNR+20)/2+1) = feature_extraction_2(BPSK_source(SNR), 8);
    FSK_cnt(:, (SNR+20)/2+1) = feature_extraction_2(FSK_source(2, SNR), 8);
    QPSK_cnt(:, (SNR+20)/2+1) = feature_extraction_2(QPSK_source(SNR), 8);
end

figure;
plot(SNRs, ASK_cnt);
title('ASK 信号谱峰-SNR 图（方法二）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')

figure;
plot(SNRs, BPSK_cnt);
title('BPSK 信号谱峰-SNR 图（方法二）');
xlabel('SNR');
```

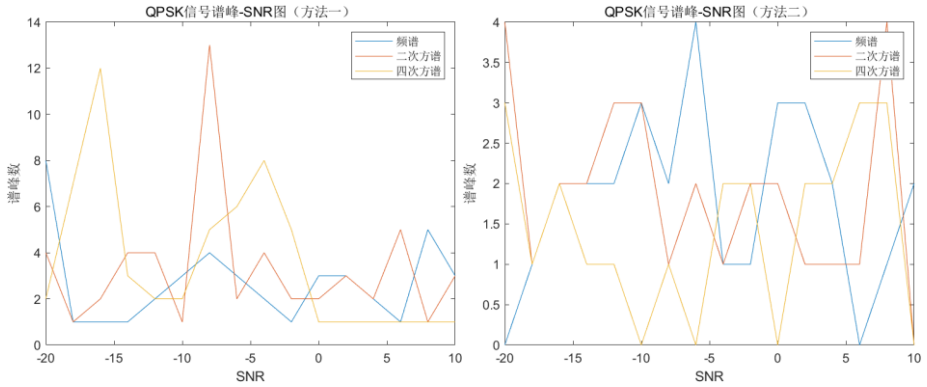
```
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')

figure;
plot(SNRs, FSK_cnt);
title('FSK 信号谱峰-SNR 图（方法二）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')

figure;
plot(SNRs, QPSK_cnt);
title('QPSK 信号谱峰-SNR 图（方法二）');
xlabel('SNR');
ylabel('谱峰数');
legend('频谱', '二次方谱', '四次方谱')
```

测试结果

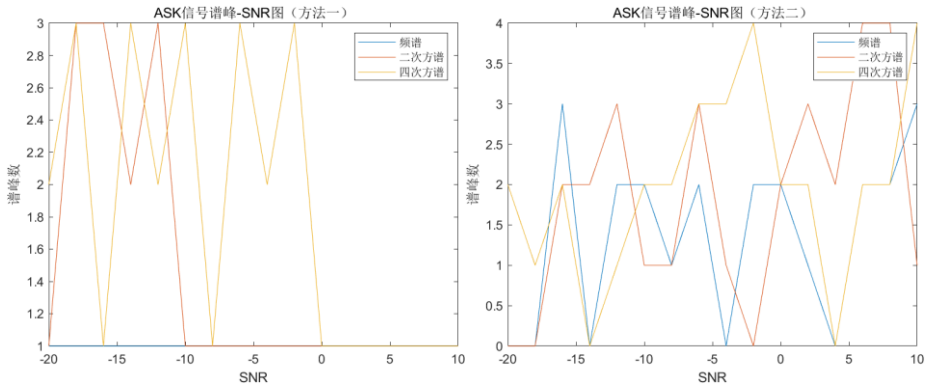
BPSK



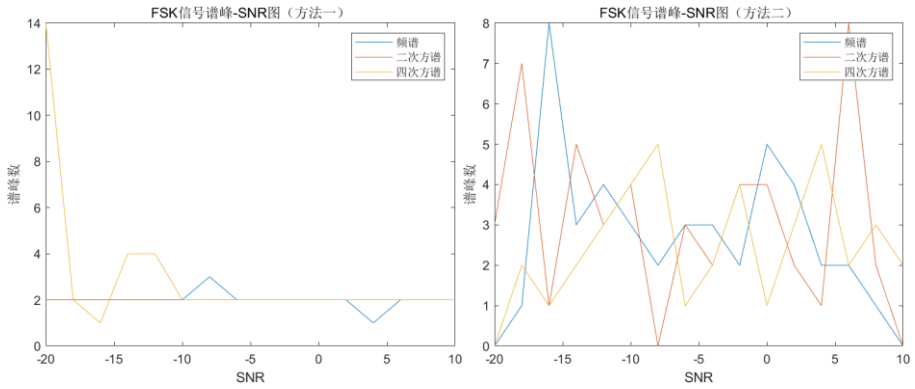
QPSK



2ASK



2FSK



（3） BPSK、QPSK、2ASK、2FSK 数字通信信号调制识别程序

选用第一种方法设计，经过反复实验比对，发现当频谱的阈值设为 0.75，二次方谱和四次方谱的阈值设为 0.7 时，识别效果较好。

```
function res = decision_tree(signal)
cnt = feature_extraction_1(signal, 0.75, 0.7, 0.7);
res = 0;
if(cnt(1) == 1)
    res = 1;% "2ASK";
else
    if(cnt(1) == 2)
        res = 2;% "2FSK";
    else
        if(cnt(2) == 1)
            res = 3;% "BPSK";
        else
            if(cnt(3) == 1)
                res = 4;% "QPSK";
            end
        end
    end
end
end
```

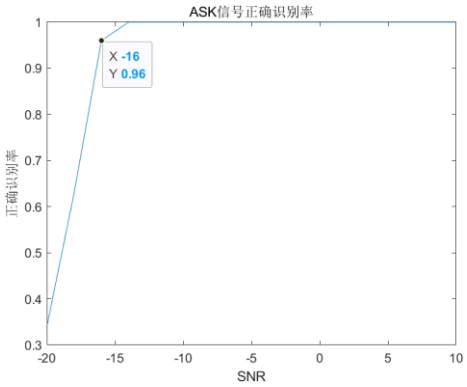
（4）BPSK、QPSK、2ASK、2FSK(不连续相位)信号的正确识别率随信噪比（-20：2：10）的变化曲线

测试代码如下，每个信噪比下每种信号测试 100 次。

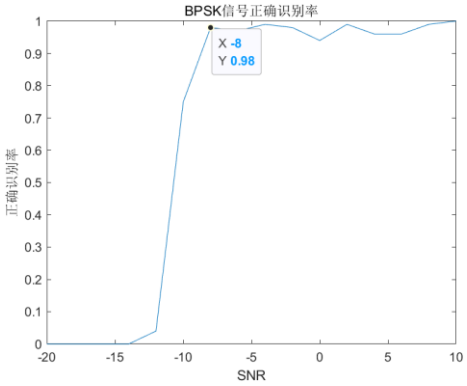
```
clear all;clc;close all;

SNRs = -20:2:10;
test_cnt = 100; % 每个信噪比下的测试次数

ASK_true_cnt = zeros(1, 16);
BPSK_true_cnt = zeros(1, 16);
FSK_true_cnt = zeros(1, 16);
QPSK_true_cnt = zeros(1, 16);
for SNR = SNRs
    for j = 1:test_cnt
        if(decision_tree(ASK_source(2, SNR)) == 1)
            ASK_true_cnt((SNR+20)/2+1) =
ASK_true_cnt((SNR+20)/2+1) + 1;
        end
        if(decision_tree(FSK_source(2, SNR)) == 2)
            FSK_true_cnt((SNR+20)/2+1) =
FSK_true_cnt((SNR+20)/2+1) + 1;
        end
        if(decision_tree(BPSK_source(SNR)) == 3)
            BPSK_true_cnt((SNR+20)/2+1) =
BPSK_true_cnt((SNR+20)/2+1) + 1;
        end
        if(decision_tree(QPSK_source(SNR)) == 4)
            QPSK_true_cnt((SNR+20)/2+1) =
QPSK_true_cnt((SNR+20)/2+1) + 1;
        end
    end
end

figure;
plot(SNRs, ASK_true_cnt/test_cnt);
title('ASK 信号正确识别率');
xlabel('SNR');
ylabel('正确识别率');

figure;
plot(SNRs, BPSK_true_cnt/test_cnt);
title('BPSK 信号正确识别率');
xlabel('SNR');
```

```
ylabel('正确识别率');

figure;
plot(SNRs, FSK_true_cnt/test_cnt);
title('FSK 信号正确识别率');
xlabel('SNR');
ylabel('正确识别率');

figure;
plot(SNRs, QPSK_true_cnt/test_cnt);
title('QPSK 信号正确识别率');
xlabel('SNR');
ylabel('正确识别率');
```

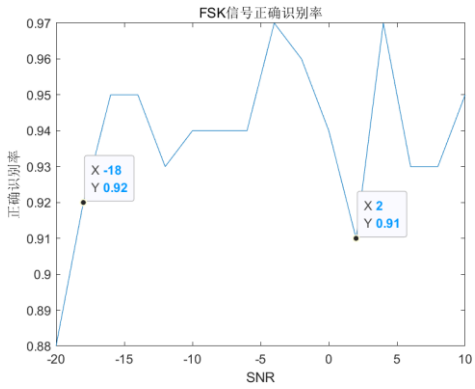
当信噪比达到 -16dB 及以上时，2ASK 信号识别正确率达到 90%以上



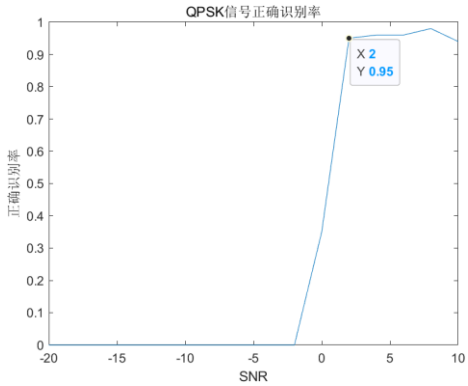
当信噪比达到 -8dB 及以上时，BPSK 信号识别正确率达到 90%以上



当信噪比达到 -18dB 及以上时，2FSK 信号识别正确率达到 90%以上



当信噪比达到 2dB 及以上时，QPSK 信号识别正确率达到 90%以上



（5） 思考连续相位的 2FSK 信号，怎么进行调制类型识别？

相位连续性分析：连续相位的 2FSK 信号具有相位连续性，在相位连续转变时不会出现跳跃。通过分析信号相位的连续性特征，可以判断信号是连续相位调制还是非连续相位调制（如 QPSK）。这种方法对于区分 CPFSK 和 QPSK 等非连续相位调制有一定效果。

十、实验结论

(1)BPSK、QPSK、2ASK、2FSK(不连续相位)信号频谱、二次方谱和四次方谱的特征:

BPSK 的二次方谱和四次方谱均为一根谱线，QPSK 的四次方谱为 1 根谱线，2ASK 的一次方谱为 1 根谱线，2FSK(不连续相位)的一次方谱为 2 根谱线。

(2）通过采用介绍的第一种方法找到最高谱线的值，设定门限分别为最高值的 0.75、0.7、0.7 倍，成功判决出信号类型。

十一、总结及心得体会

总结：

该实验通过对 BPSK、QPSK、2ASK 和 2FSK（不连续相位）信号的频谱、二次方谱和四次方谱进行分析，提取谱峰特征并设计识别决策树，实现了信号调制类型的识别任务。同时，通过比较两种方法的优缺点，选择适合的方法进行识别。在一定范围内的信噪比下，所设计的识别系统能够保证每种信号的正确识别率均在 90%以上。

心得体会：

通过这个实验，我深入了解了数字通信信号调制识别的原理和方法。我学会了如何利用频谱、二次方谱和四次方谱等谱特征来区分不同调制方式的信号，以及如何提取和分析这些特征。同时，我也了解到信噪比对信号识别的影响，并学会了设计区分门限来优化识别性能。

此外，实验中的决策树设计和编程实现过程让我更深入地理解了信号识别算法的具体步骤。通过绘制正确识别率随信噪比变化的曲线，我能够直观地了解不同信噪比下识别性能的变化趋势。

总体而言，这个实验对我加深了数字通信信号调制识别的理论和实践认识，提升了我的信号处理和算法设计能力。它也让我明白了在实际应用中，选择合适的特征提取方法和识别算法对于实现高准确率的信号识别至关重要。

十二、对本实验过程及方法、手段的改进建议：

特征提取方法的比较：在实验中，可以尝试使用更多不同的特征提取方法，并对它们进行比较和评估，以确定最适合识别任务的方法。可能存在其他有效的频谱特征或统计特征，可以进一步探索和应用。

识别算法的优化：在设计识别决策树和编写识别程序时，可以尝试使用更复杂的机器学习算法（例如基于基尼不纯度来构建决策树），以提高识别性能。这些算法可以通过对大量数据进行训练和调优，来自动发现和利用特征之间的复杂关系，从而提高识别准确率。