

实验一：数字信号的 FFT 分析

1、实验内容及要求

(1) 离散信号的频谱分析：

设信号
$$x(n) = 0.001 * \cos(0.45n\pi) + \sin(0.3n\pi) - \cos(0.302n\pi - \frac{\pi}{4})$$

此信号的 0.3π 和 0.302π 两根谱线相距很近，谱线 0.45π 的幅度很小，请选择合适的序列长度 N 和窗函数，用 DFT 分析其频谱，要求得到清楚的三根谱线。

(2) DTMF 信号频谱分析

用计算机声卡采用一段通信系统中电话双音多频（DTMF）拨号数字 0~9 的数据，采用快速傅立叶变换（FFT）分析这 10 个号码 DTMF 拨号时的频谱。

2、实验目的

通过本次实验，应该掌握：

- (a) 用傅立叶变换进行信号分析时基本参数的选择。
- (b) 经过离散时间傅立叶变换（DTFT）和有限长度离散傅立叶变换（DFT）后信号频谱上的区别，前者 DTFT 时间域是离散信号，频率域还是连续的，而 DFT 在两个域中都是离散的。
- (c) 离散傅立叶变换的基本原理、特性，以及经典的快速算法（基 2 时间抽选法），体会快速算法的效率。
- (d) 获得一个高密度频谱和高分辨率频谱的概念和方法，建立频率分辨率和时间分辨率的概念，为将来进一步进行时频分析（例如小波）的学习和研究打下基础。
- (e) 建立 DFT 从整体上可看成是由窄带相邻滤波器组成的滤波器组的概念，此概念的一个典型应用是数字音频压缩中的分析滤波器，例如 DVD AC3 和 MPEG Audio。

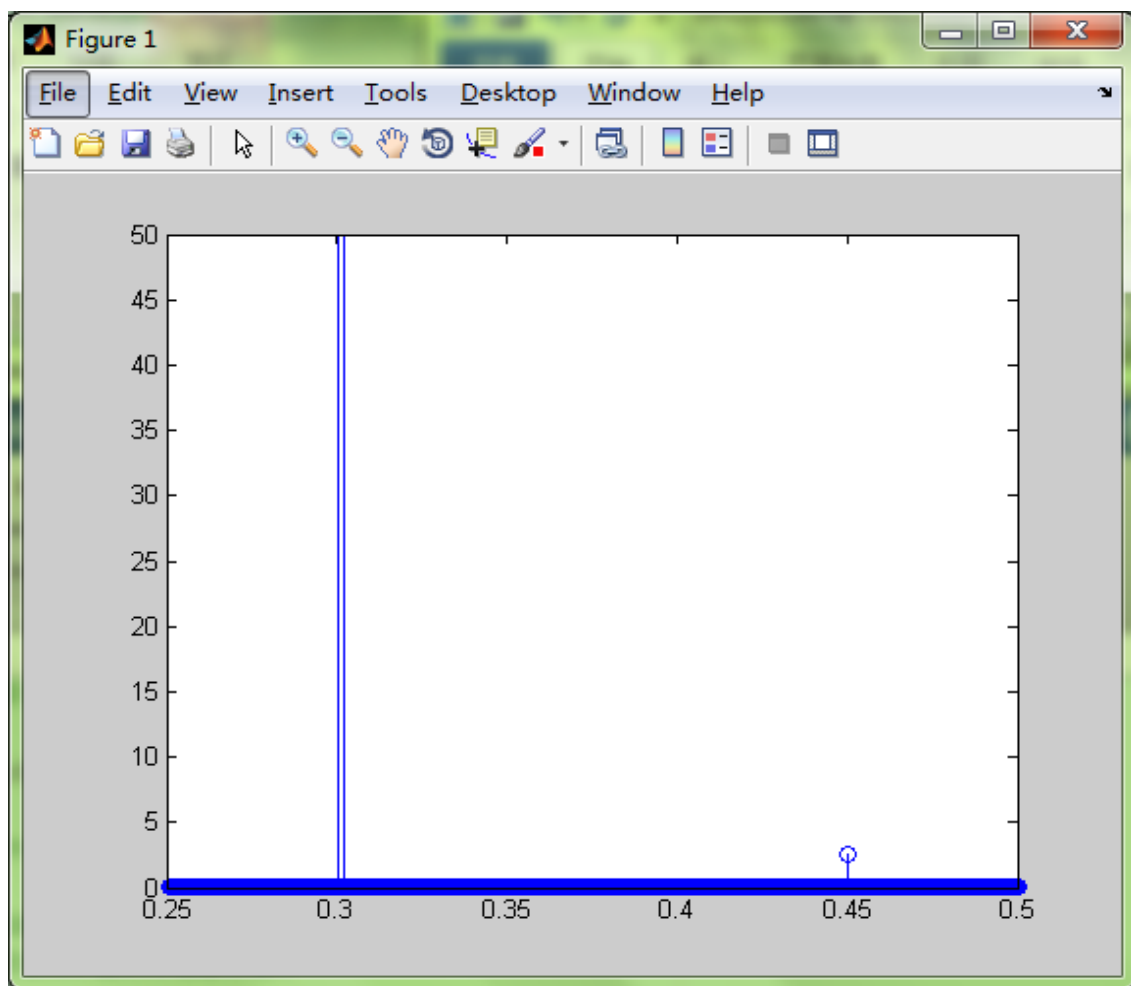
3、实验结果

(1) 离散信号的频谱分析：

【实验代码】

```
clear
close all
N=5000;                %序列长度
n=1:1:N;
x=0.001*cos(0.45*pi*n)+sin(0.3*pi*n)-cos(0.302*pi*n-pi/4);
y=fft(x,N);
magy=abs(y(1:1:N));     %fft 后的幅值
k=1:1:N;w=2*pi/N*k;
stem(w/pi,magy);        %绘脉冲图
axis([0.25,0.5,0,50])  %设定图形窗的二维坐标边界
```

【实验结果】



(2) DTMF 信号频谱分析

【实验代码】

```
clear
close all

column=[1209,1336,1477,1633];    %列频率值
line=[697,770,852,941];          %行频率值

fs=8000;                          %采样频率 8kHz
ts=1/fs;                          %采样周期

N=1024;                           %DFT 长度
n=0:N-1;
detf=fs/N;
f=0:detf:detf*(N-1);

key=zeros(16,N);                  %16*N 的全 0 矩阵，用于存放 10 数字的
                                   %DTMF 信号（长度为 N）
```

```

key(1,:)=cos(2*pi*column(1)*ts*n)+cos(2*pi*line(1)*ts*n);
key(2,:)=cos(2*pi*column(2)*ts*n)+cos(2*pi*line(1)*ts*n);
key(3,:)=cos(2*pi*column(3)*ts*n)+cos(2*pi*line(1)*ts*n);
key(4,:)=cos(2*pi*column(1)*ts*n)+cos(2*pi*line(2)*ts*n);
key(5,:)=cos(2*pi*column(2)*ts*n)+cos(2*pi*line(2)*ts*n);
key(6,:)=cos(2*pi*column(3)*ts*n)+cos(2*pi*line(2)*ts*n);
key(7,:)=cos(2*pi*column(1)*ts*n)+cos(2*pi*line(3)*ts*n);
key(8,:)=cos(2*pi*column(2)*ts*n)+cos(2*pi*line(3)*ts*n);
key(9,:)=cos(2*pi*column(3)*ts*n)+cos(2*pi*line(3)*ts*n);
key(10,:)=cos(2*pi*column(2)*ts*n)+cos(2*pi*line(4)*ts*n);

```

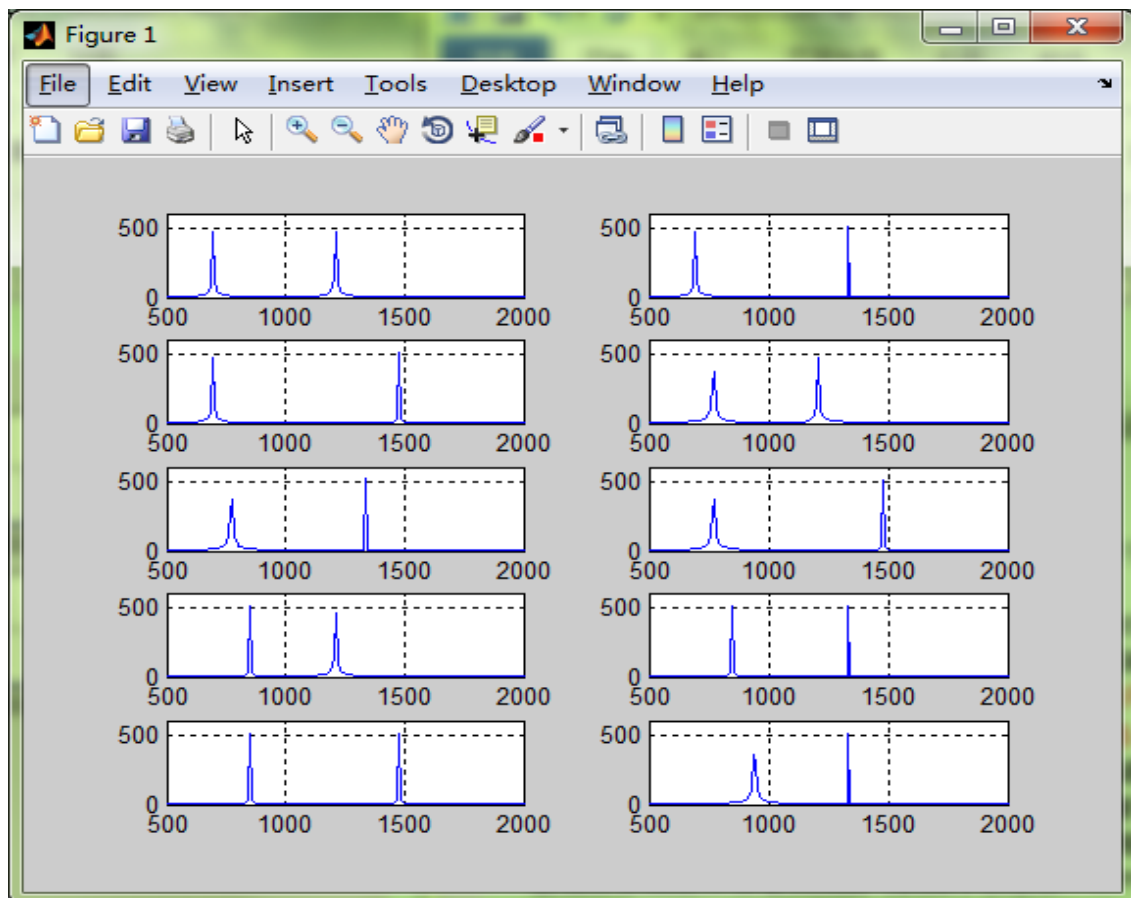
```

figure;                                %打开新的图形窗口
for i=1:10
    subplot(5,2,i)
    plot(f,abs(fft(key(i,:))));
    axis([500,2000,0,600]);           %设定图形窗的二维坐标边界
    grid;                             %打上网格
end

```

【实验结果】

图像从左到右、从上到下依次为数字 1~9 以及 0 编码后的频谱图。



实验二：DTMF 信号的编码

1、实验内容及要求

1) 把您的联系电话号码 通过 DTMF 编码生成为一个 .wav 文件。

✧ 技术指标:

✧ 根据 ITU Q.23 建议, DTMF 信号的技术指标是: 传送/接收率为每秒 10 个号码, 或每个号码 100ms。

✧ 每个号码传送过程中, 信号存在时间至少 45ms, 且不多于 55ms, 100ms 的其余时间是静音。

✧ 在每个频率点上允许有不超过 $\pm 1.5\%$ 的频率误差。任何超过给定频率 $\pm 3.5\%$ 的信号, 均被认为是无效的, 拒绝接收。

(其中关键是不同频率的正弦波的产生。可以使用查表方式模拟产生两个不同频率的正弦波。正弦表的制定要保证合成信号的频率误差在 $\pm 1.5\%$ 以内, 同时使取样点数尽量少)

2) 对所生成的 DTMF 文件进行解码。

✧ DTMF 信号解码可以采用 FFT 计算 N 点频率处的频谱值, 然后估计出所拨号码。但 FFT 计算了许多不需要的值, 计算量太大, 而且为保证频率分辨率, FFT 的点数较大, 不利于实时实现。因此, FFT 不适合于 DTMF 信号解码的应用。

✧ Xk value

✧ 由于只需要知道 8 个特定点的频谱值, 因此采用一种称为 Goertzel 算法的 IIR 滤波器可以有效地提高计算效率。其传递函数为:

✧

$$H_k(z) = \frac{1 - e^{-j2\pi k/N} z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}}$$

2、实验目的

(a)复习和巩固 IIR 数字滤波器的基本概念;

(b)掌握 IIR 数字滤波器的设计方法;

(c)掌握 IIR 数字滤波器的实现结构;

(d)能够由滤波器的实现结构分析滤波器的性能(字长效应);

(e)了解通信系统电话 DTMF 拨号的基本原理和 IIR 滤波器实现方法。

3、实验结果

【实验代码】

```
clear
```

```
close all
```

```
d=input('请键入电话号码: '); %输入电话号码(字符串)
```

```
sum=length(d); %计算电话号码长度
```

```
total_x=[];
```

```
M=500;
```

```

% 以下 for 循环完成对输入号码的编码工作
for a=1:sum %循环 sum 次
    %DTMF 表中键的 16 个 ASCII 码
    tm=[49,50,51,65;52,53,54,66;55,56,57,67;42,48,35,68];
    for p=1:4;
        for q=1:4;
            if tm(p,q)==abs(d(a));break,end %检测码相符的列号 q
            %abs(d(a))为求字符的 ASCII 码
        end
        if tm(p,q)==abs(d(a));break,end %检测码相符的行号 p
    end
    f1=[697,770,852,941]; %行频率向量
    f2=[1209,1336,1477,1633]; %列频率向量
    n=1:M;
    x=sin(2*pi*n*f1(p)/8000)+sin(2*pi*n*f2(q)/8000); %构成双频信号
    x=[x,zeros(1,M)]; %行向量前半为信号，后半为静音
    total_x=[total_x,x]; %将所编码连接起来,是一个行向量
end
wavwrite(total_x,'soundwave') %将编码的号码存为音频文件
sound(total_x); %发出声音

subplot(2,1,1);
plot(total_x); %画出编好码的号码的时域波形图
xlabel('时间');
title('DTMF 信号时域波形图');
disp('双频信号已生成并发出');

% 接收检测端的程序
k=[18 20 22 24 31 34 38 42]; %要求的 DFT 样本序号 (N=205)
N=205;
disp('接收端检测到的号码为')
for a=1:sum
    m=2*M*(a-1);
    X=goertzel(total_x(m+1:m+N),k+1); %用 Goertzel 算法计算八点 DFT 样本
    val=abs(X); %列出八点 DFT 向量的幅度值
    limit=80; %判决门限
    for r=1:4;
        if val(r)>limit, break,end %查找行号
    end
    for s=5:8;
        if val(s)>limit, break,end %查找列号
    end
    disp(char(tm(r,s-4))) %显示接收到的字符
end

```

```

subplot(2,1,2);
stem(k,val,'.');
%画出 DFT(k)幅度，k 与 val 都是长度为 8 的行向量
grid;
xlabel('k');
ylabel('|X(k)|');
disp('图上显示的是该解码信号近似基频的 DFT 幅度');
pause;
end

```

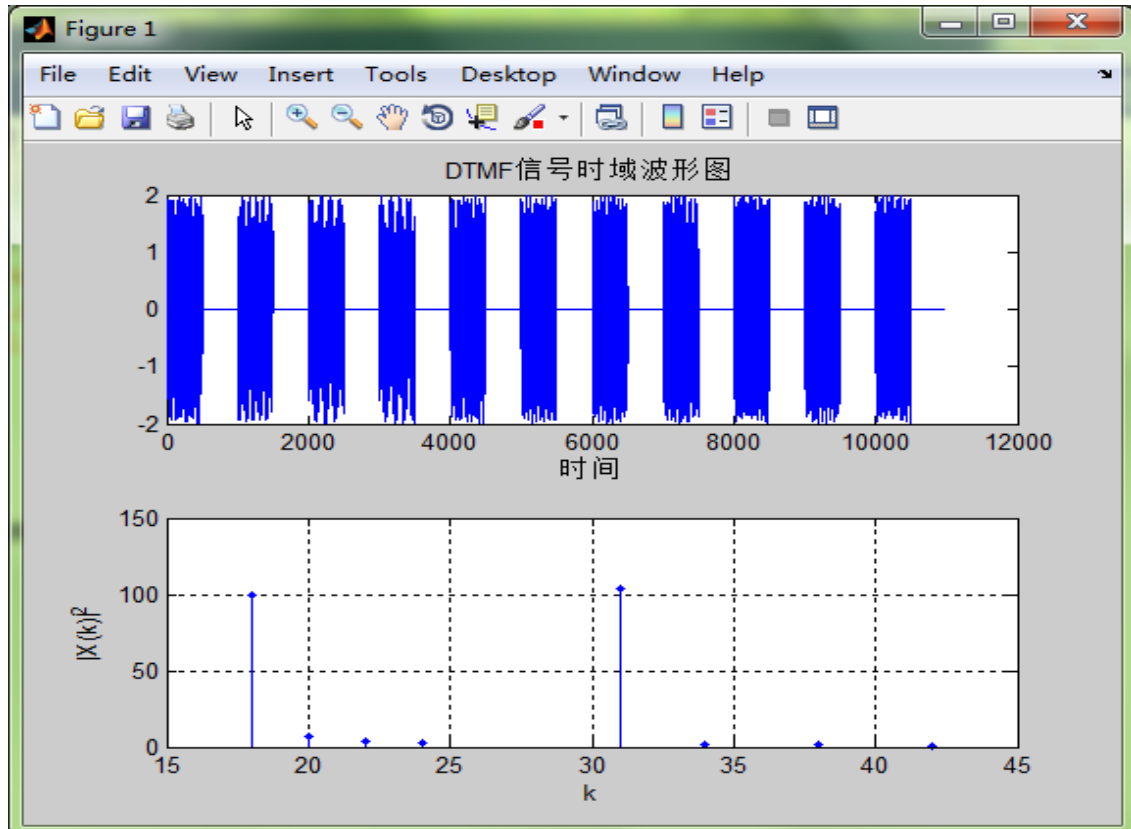
【实验结果】

输入电话号码 13261702239，则一个个解码输出并画出 $|X(k)|^2$ 与 k 的解码谱线图。

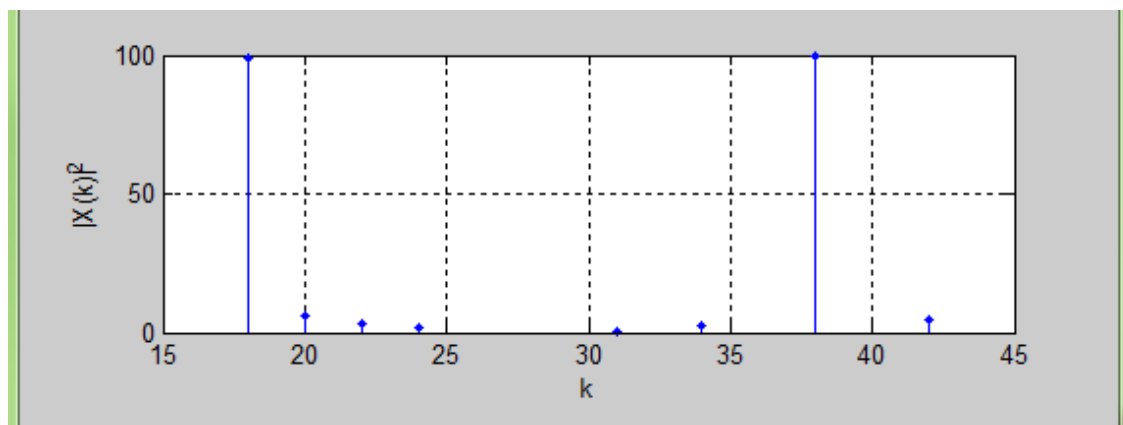
```

Command Window
>> e2dtmf_tel
请键入电话号码: 13261702239
Warning: Data clipped during write to file:soundwave
> In wavwrite>PCM_Quantize at 278
   In wavwrite>write_wavedat at 300
   In wavwrite at 137
   In e2dtmf_tel at 26
双频信号已生成并发出
接收端检测到的号码为
1
图上显示的是该解码信号近似基频的DFT幅度
3
图上显示的是该解码信号近似基频的DFT幅度

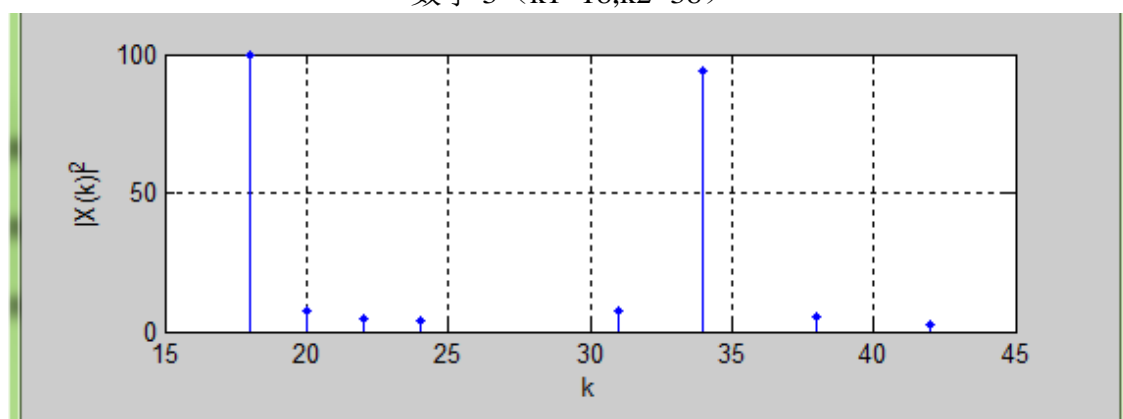
```



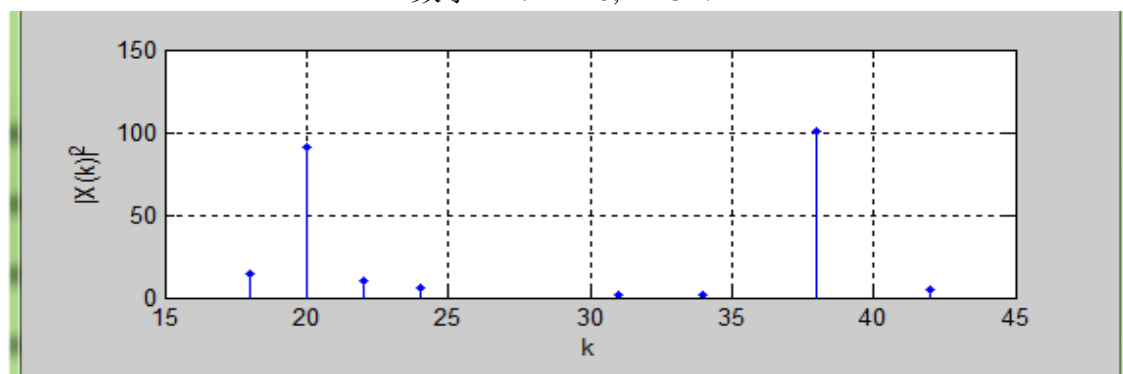
数字 1 ($k_1=18, k_2=31$)



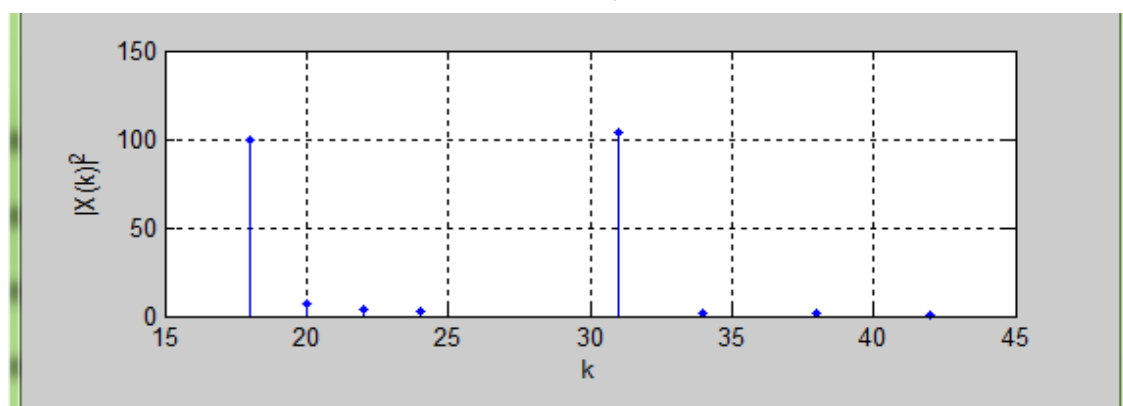
数字 3 (k1=18,k2=38)



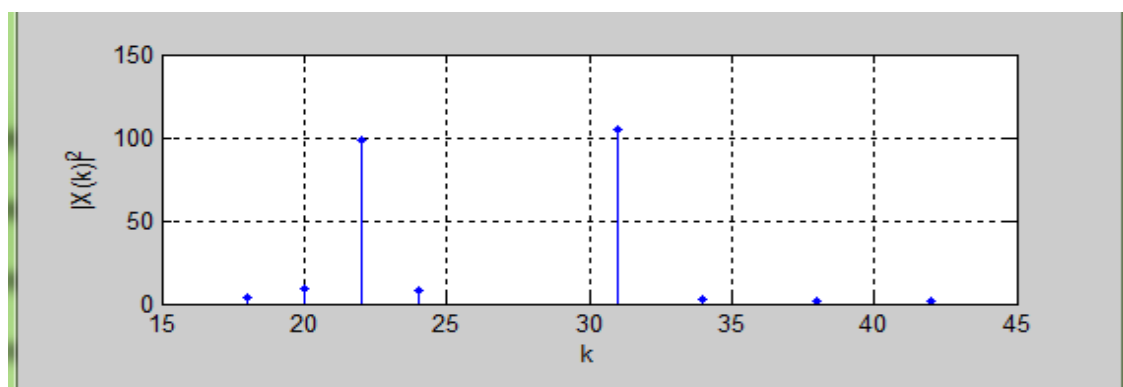
数字 2 (k1=18,k2=34)



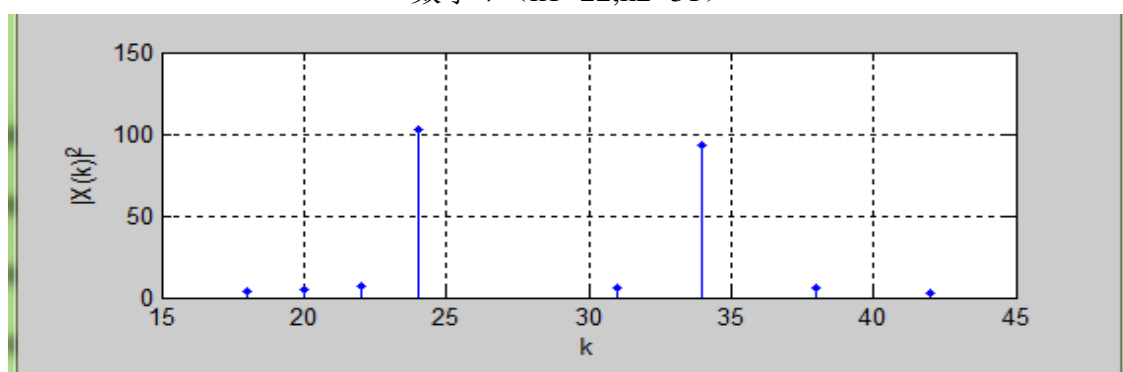
数字 6 (k1=20,k2=38)



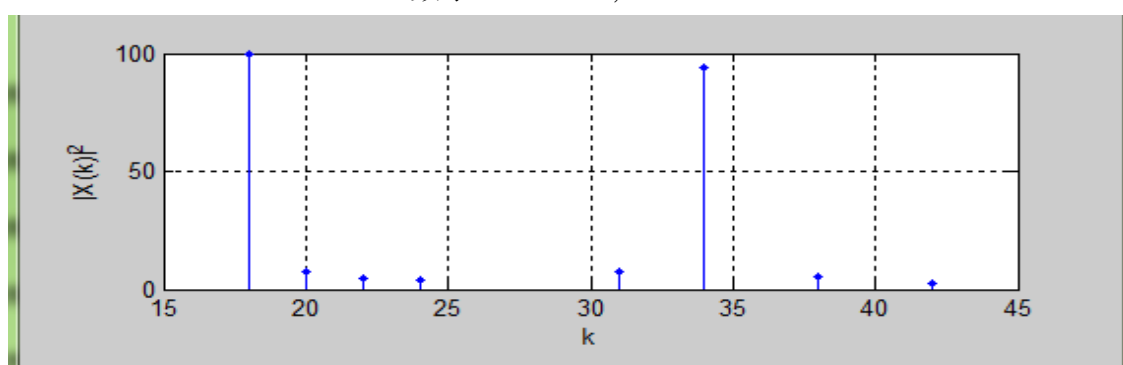
数字 1 (k1=18,k2=31)



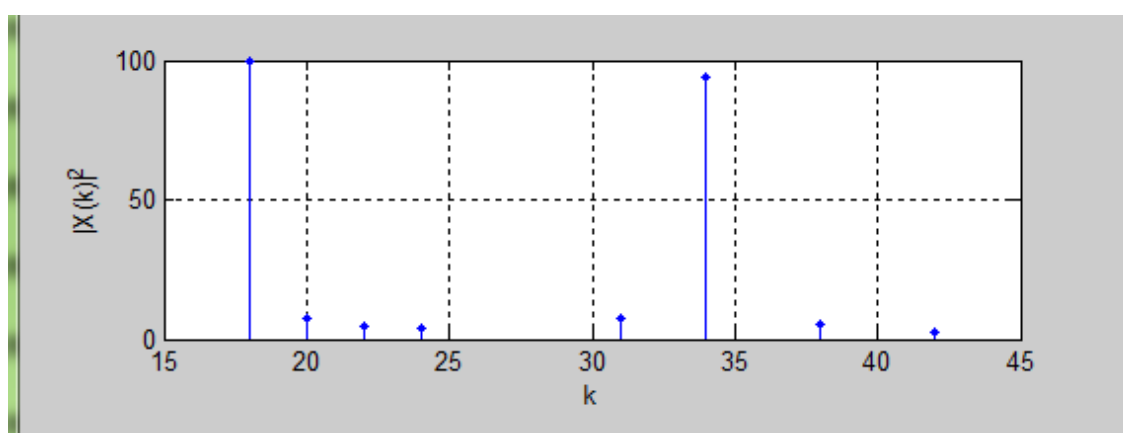
数字 7 (k1=22,k2=31)



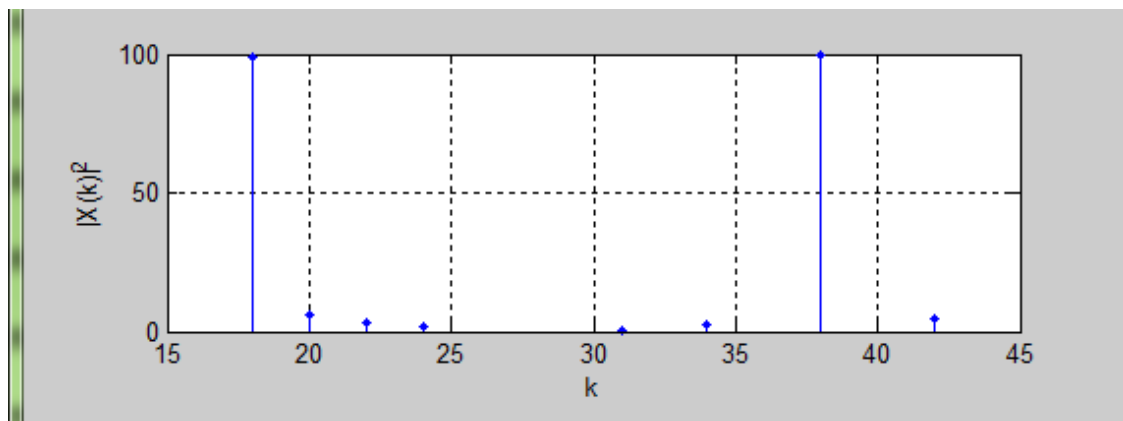
数字 0 (k1=24,k2=34)



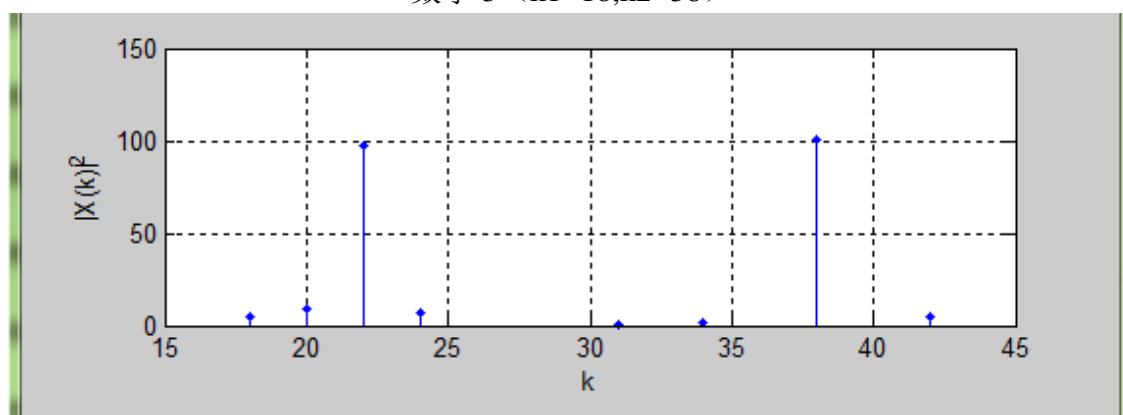
数字 2 (k1=18,k2=34)



数字 2 (k1=18,k2=34)



数字 3 ($k_1=18, k_2=38$)



数字 9 ($k_1=22, k_2=38$)

实验三：FIR 数字滤波器的设计和实现

1、实验内容及要求：

- ✧ 录制自己的一段声音，长度为 10 秒 以上，取样频率 32kHz，然后叠加一个高斯白噪声，使得信噪比为 20dB。请采用窗口法设计一个 FIR 带通滤波器，滤除噪声提高质量。

■ 提示：

- ✧ 滤波器指标参考：通带边缘频率为 4kHz，阻带边缘频率为 4.5kHz，阻带衰减大于 50dB；人类声音 300~3400
- ✧ Matlab 函数 $y = \text{awgn}(x, \text{snr}, 'measured')$ ，首先测量输入信号 x 的功率，然后对其叠加高斯白噪声；

2、实验目的

- ✧ 通过本次实验，掌握以下知识：
 - ⊕ FIR 数字滤波器窗口设计法的原理和设计步骤；
 - ⊕ Gibbs 效应发生的原因和影响；
 - ⊕ 不同类型的窗函数对滤波效果的影响，以及窗函数和长度 N 的选择。

3、实验结果

【实验代码】

```
clear
close all

fs=32000;
nbits=16;
snr=20;
[x,fs,nbits]=wavread('E:\大学课程\数字信号处理\DSP_MatLab 实验\homework\myvoice.wav');
x2=awgn(x,snr,'measured'); %叠加噪声后的信号
wavwrite(x2,fs,nbits,'E:\大学课程\数字信号处理\DSP_MatLab 实验\homework\myvoice_noise.wav');

wp=8000*pi/32000;
ws=9000*pi/32000;
deltaw=ws-wp;
wc=(wp+ws)/2;
N=ceil(11*pi/deltaw); %取整,11*pi 表明选用布莱克曼窗,N=352
b=fir1(N,wc/pi,blackman(N+1)); %选择窗函数,并归一化截止频率, N 为偶数,取 N+1

x3=filter(b,1,x2); %滤波去噪
wavwrite(x3,fs,nbits,'E:\大学课程\数字信号处理\DSP_MatLab 实验\homework\myvoice_nonoise.wav');

figure(1);
freqz(b,1,512); %画出滤波器频率响应
title('滤波器频率响应');

figure(2);
f=fs*(1:512)/1024;
y2=fft(x2,1024); %对噪声信号进行 fft 运算
y3=fft(x3,1024); %对滤噪信号进行 fft 运算
subplot(2,1,1);
plot(f,abs(y2(1:512)));
title('滤波前频谱');
xlabel('Hz');
ylabel('幅度');
subplot(2,1,2);
F2=plot(f,abs(y3(1:512)));
title('滤波后频谱');
xlabel('Hz');
ylabel('幅度');
```

```

figure(3);
t=0:1/fs:(size(x2)-1)/fs;
subplot(2,1,1);
plot(t,x2);
title('滤波前时域波形');
subplot(2,1,2);
plot(t,x3);
title('滤波后时域波形');

```

【实验结果】

