哈尔滨工业大学（深圳）

**《信号与系统》课程**

**实验报告**

（2021-2022 秋季学期）

**课程名称 ：** 信号与系统

**题 目 ：** 实验三：使用快速傅里叶变换（FFT）进行频谱分析

**班级学号** ：

**学生姓名 ：**

2021年11月10日

**一、实验源码**

主程序：

|  |
| --- |
| 实验三**典型离散信号傅里叶变换** N = 1000;  Fs = 1000;  x1 = triangle\_wave(0:N-1);  x2 = sqaure\_wave(0:N-1);  X1 = fft(x1);  X2 = fft(x2);  figure;  subplot(2, 3, 1); plot(0:9, x1(1:10)); title("三角波采样"); xlabel("n");  subplot(2, 3, 4); plot(0:9, x2(1:10)); title("矩形波采样"); xlabel("n");  subplot(2, 3, 2); plot(0:(2\*pi/Fs):2\*pi-(2\*pi/Fs), abs(X1)); title("三角波幅频特性"); xlabel("\Omega");  subplot(2, 3, 5); plot(0:(2\*pi/Fs):2\*pi-(2\*pi/Fs), abs(X2)); title("矩形波幅频特性"); xlabel("\Omega");  subplot(2, 3, 3); plot(0:(2\*pi/Fs):2\*pi-(2\*pi/Fs), angle(X1)); title("矩形波相频特性"); xlabel("\Omega");  subplot(2, 3, 6); plot(0:(2\*pi/Fs):2\*pi-(2\*pi/Fs), angle(X2)); title("矩形波相频特性"); xlabel("\Omega"); **离散序列线性卷积** N = 20000;  x1 = signal211(0:N-1);  x2 = signal212(0:N-1);  x3 = signal221(0:N-1);  x4 = signal222(0:N-1);  X1 = fft(x1);  X2 = fft(x2);  X3 = fft(x3);  X4 = fft(x4);  X12 = X1 .\* X2;  X34 = X3 .\* X4;  x12 = ifft(X12);  x34 = ifft(X34);  figure;  subplot(2, 3, 1); plot(0:19, x1(1:20)); title("x\_1序列"); xlabel("n");  subplot(2, 3, 2); plot(0:19, x2(1:20)); title("x\_2序列"); xlabel("n");  subplot(2, 3, 3); plot(0:24, x12(1:25)); title("x\_1和x\_2卷积序列"); xlabel("n");  subplot(2, 3, 4); plot(0:19, x3(1:20)); title("x\_3序列"); xlabel("n");  subplot(2, 3, 5); plot(0:19, x4(1:20)); title("x\_4序列"); xlabel("n");  subplot(2, 3, 6); plot(0:19, x34(1:20)); title("x\_3和x\_4卷积序列"); xlabel("n"); **滤波** Fs = 10000;  t = (1/Fs):(1/Fs):0.1;  x = signal3(t);  X = fft(x);  Xf = X;  Xf(20:end-20) = 0; % low pass filter  xf = ifft(Xf);  figure;  subplot(2, 2, 1); plot(t, x); title("原信号时域波形"); xlabel("Time(s)");  subplot(2, 2, 2); plot(0:(2\*pi/(0.1\*Fs)):2\*pi-(2\*pi/(0.1\*Fs)), abs(X)); title("原信号频域波形"); xlabel("\Omega");  subplot(2, 2, 3); plot(0:(2\*pi/(0.1\*Fs)):2\*pi-(2\*pi/(0.1\*Fs)), abs(Xf)); title("滤波信号频域波形"); xlabel("\Omega");  subplot(2, 2, 4); plot(t, abs(xf)); title("滤波信号时域波形"); xlabel("Time(s)"); **采样频率变化** f11 = 5;  f12 = 15;  f13 = 40;  t11 = 0:(1/f11):30;  t12 = 0:(1/f12):10;  t13 = 0:(1/f13):10;  x11 = signal4(t11);  x12 = signal4(t12);  x13 = signal4(t13);  x11\_128 = x11(1:128);  x12\_128 = x12(1:128);  x13\_128 = x13(1:128);  X11 = fft(x11);  X12 = fft(x12);  X13 = fft(x13);  figure;  subplot(2, 3, 1); stem(0:(1/f11):(127/f11), x11(1:128)); title("f\_1=5Hz,128点时域采样"); xlabel("Time(s)");  subplot(2, 3, 2); stem(0:(1/f12):(127/f12), x12(1:128)); title("f\_2=15Hz,128点时域采样"); xlabel("Time(s)");  subplot(2, 3, 3); stem(0:(1/f13):(127/f13), x13(1:128)); title("f\_3=40Hz,128点时域采样"); xlabel("Time(s)");  subplot(2, 3, 4); plot(0:(2\*pi/(128)):2\*pi-(2\*pi/(128)), abs(X11(1:128))); title("f\_1=5Hz,128点采样幅度谱"); xlabel("\Omega");  subplot(2, 3, 5); plot(0:(2\*pi/(128)):2\*pi-(2\*pi/(128)), abs(X12(1:128))); title("f\_2=15Hz,128点采样幅度谱"); xlabel("\Omega");  subplot(2, 3, 6); plot(0:(2\*pi/(128)):2\*pi-(2\*pi/(128)), abs(X13(1:128))); title("f\_3=40Hz,128点采样幅度谱"); xlabel("\Omega");  f2 = 60;  t2 = 0:(1/f2):20;  x2 = signal4(t2);  x2p = [x2(1:64), zeros([1, 64])];  X2 = fft(x2p);  figure;  subplot(2, 2, 1); stem(0:(1/f2):(127/f2), x2p); title("f=60Hz时域波形(64点补零至128点)"); xlabel("Time(s)");  subplot(2, 2, 2); plot(0:(2\*pi/(128)):2\*pi-(2\*pi/(128)), abs(X2)); title("f=60Hz频域波形(64点补零至128点)"); xlabel("Time(s)");  f3 = 60;  t3 = 0:(1/f2):20;  x3 = signal4(t2);  x3p = x3(1:128);  X3 = fft(x3p);  subplot(2, 2, 3); stem(0:(1/f3):(127/f3), x3p); title("f=60Hz时域波形(128点不补零)"); xlabel("\Omega");  subplot(2, 2, 4); plot(0:(2\*pi/(128)):2\*pi-(2\*pi/(128)), abs(X3)); title("f=60Hz频域波形(128点不补零)"); xlabel("\Omega"); **趣味小实验** Fs = 8000;  t = 0:(1/Fs):17;  tiger\_ori = zeros([1, 16\*Fs\*0.5]);  f = [261.63, 293.66, 329.63, 261.63, 261.63, 293.63, 329.63, 261.63, 329.63, 349.23, 392, 392, 329.63, 349.23, 392, 392];  for i = 1:16  tiger\_ori((i-1)\*Fs\*0.5+1:i\*Fs\*0.5) = sin(2\*pi\*f(i)\*t((i-1)\*Fs\*0.5+1:i\*Fs\*0.5));  end  Tiger\_ori = fft(tiger\_ori);  figure;  subplot(3, 1, 1); plot(linspace(0, 6.28, length(Tiger\_ori)), abs(Tiger\_ori)); title("原音乐频域"); xlabel("\Omega");  audiowrite('5-3.wav',tiger\_ori,Fs);  Tiger\_filter = Tiger\_ori;  Tiger\_filter(0.5e4:5.5e4) = 0;  tiger\_filter = ifft(Tiger\_filter);  subplot(3, 1, 2); plot(linspace(0, 6.28, length(Tiger\_filter)), abs(Tiger\_filter)); title("消除高频分量后频域"); xlabel("\Omega");  audiowrite('5-4.wav',abs(tiger\_filter),Fs);  tiger\_high = tiger\_ori;  for i = 1:16  tiger\_high((i-1)\*Fs\*0.5+1:i\*Fs\*0.5) = 0.25 \* (sin((2\*pi\*f(i)\*t((i-1)\*Fs\*0.5+1):i\*Fs\*0.5)) + ...  sin(4\*pi\*f(i)\*t((i-1)\*Fs\*0.5+1:i\*Fs\*0.5)) + ...  sin(6\*pi\*f(i)\*t((i-1)\*Fs\*0.5+1:i\*Fs\*0.5)) + ...  sin(8\*pi\*f(i)\*t((i-1)\*Fs\*0.5+1:i\*Fs\*0.5)));  end  Tiger\_high = fft(tiger\_high);  subplot(3, 1, 3); plot(linspace(0, 6.28, length(Tiger\_high)), abs(Tiger\_high)); title("加入谐波后频域"); xlabel("\Omega");  audiowrite('5-5.wav',abs(tiger\_high),Fs); |

主程序中用到若干函数，内容如下：

signal3.m

|  |
| --- |
| function output = signal3(t)  output = (sin(100.\*t)./(100.\*t)) + rand(size(t));  end |

signal4.m

|  |
| --- |
| function output = signal4(t)  output = sin(5.\*t) + sin(9.\*t);  end |

signal211.m

|  |
| --- |
| function output = signal211(n)  output = zeros(size(n));  for i = 1:length(n)  if 0 <= n(i) && n(i) <= 4  output(i) = n(i) + 1;  elseif 5 <= n(i) && n(i) <= 9  output(i) = 10 - n(i);  end  end  end |

signal212.m

|  |
| --- |
| function output = signal212(n)  output = zeros(size(n));  for i = 1:length(n)  if 0 <= n(i) && n(i) <= 5  output(i) = 2^n(i);  elseif 6 <= n(i) && n(i) <= 11  output(i) = (-1) \* (2^(n(i) - 6));  end  end  end |

signal221.m

|  |
| --- |
| function output = signal221(n)  output = zeros(size(n));  for i = 1:length(n)  if 0 <= n(i) && n(i) <= 6  output(i) = 0.8^n(i);  elseif 7 <= n(i) && n(i) <= 11  output(i) = n(i) - 3;  end  end  end |

signal222.m

|  |
| --- |
| function output = signal222(n)  output = zeros(size(n));  for i = 1:length(n)  if 0 <= n(i) && n(i) <= 4  output(i) = n(i) - 1;  elseif 5 <= n(i) && n(i) <= 12  output(i) = (-1) \* (0.6^(n(i) - 6));  end  end  end |

sqaure\_wave.m

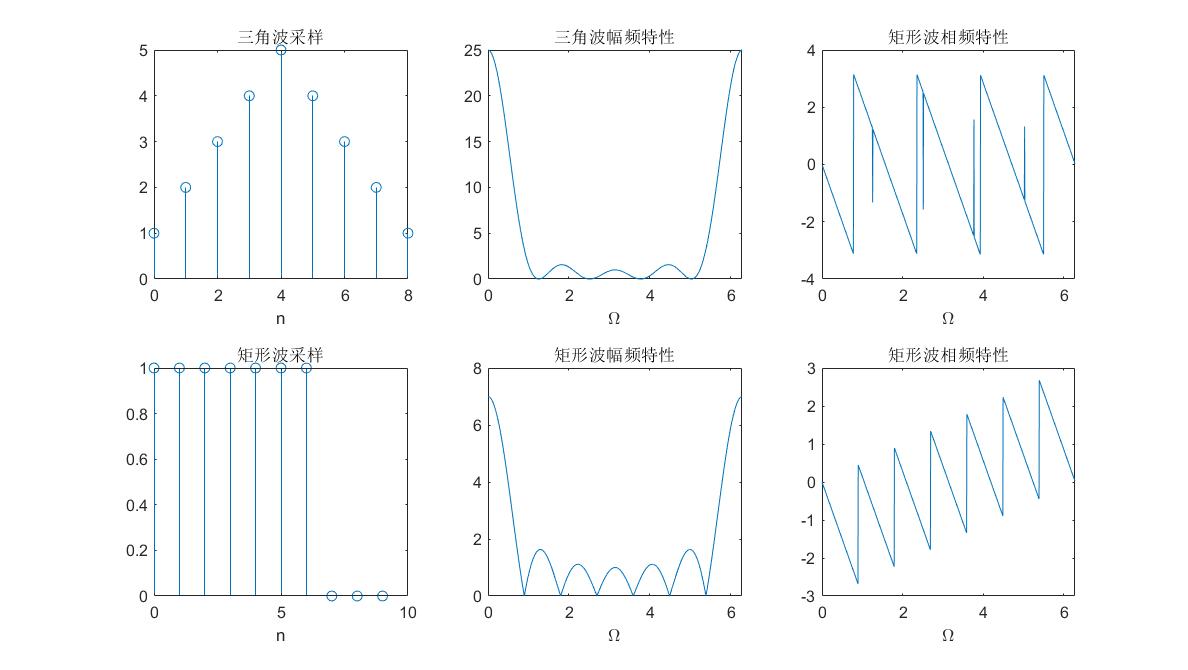
|  |
| --- |
| function output = sqaure\_wave(n)  output = zeros(size(n));  for i = 1:length(n)  if 0 <= n(i) && n(i) <= 6  output(i) = 1;  end  end  end |

triangle\_wave.m

|  |
| --- |
| function output = triangle\_wave(n)  output = zeros(size(n));  for i = 1:length(n)  if 0 <= n(i) && n(i) <=4  output(i) = n(i) + 1;  elseif 5 <= n(i) && n(i) <= 8  output(i) = 9 - n(i);  else  output(i) = 0;  end  end  end |

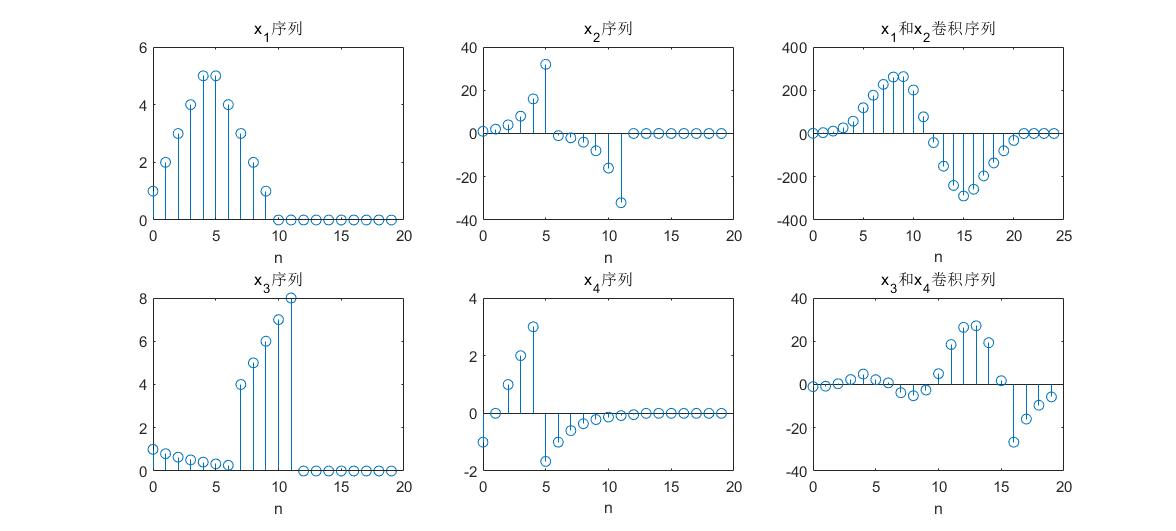
**二、实验结果及分析**

**1.典型离散信号傅里叶变换**



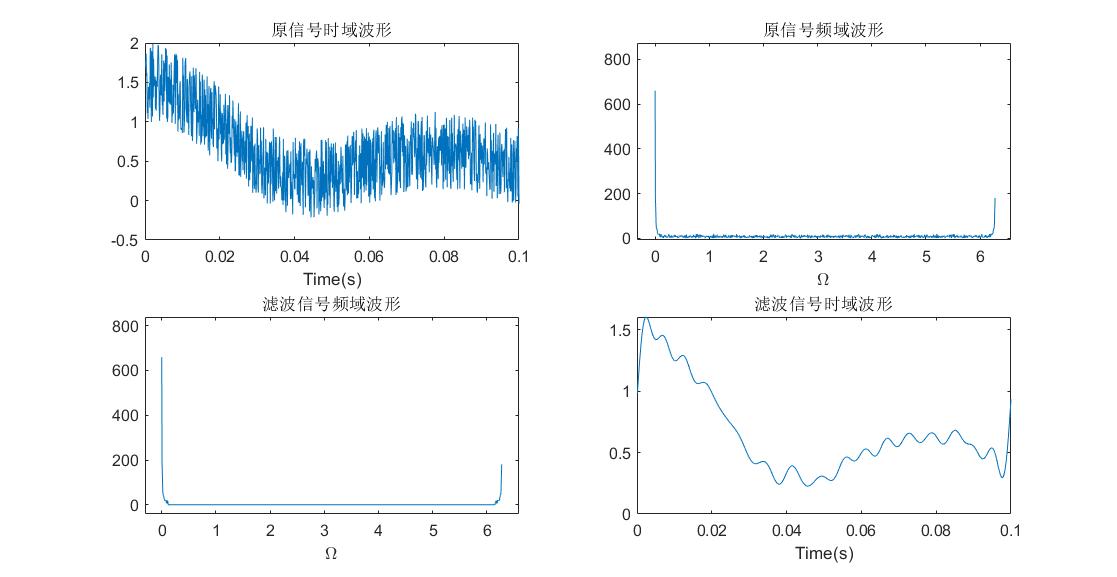
上图是三角序列和矩形序列的离散傅里叶变换的幅度谱和相位谱，在矩形波相位谱中出现了一些不连续点，这些点的横坐标与幅度谱中幅度为零的点的横坐标相同，因此这些不连续点是由于非常小的复数在求相角时的计算误差引起的。

**2.** **离散序列线性卷积**



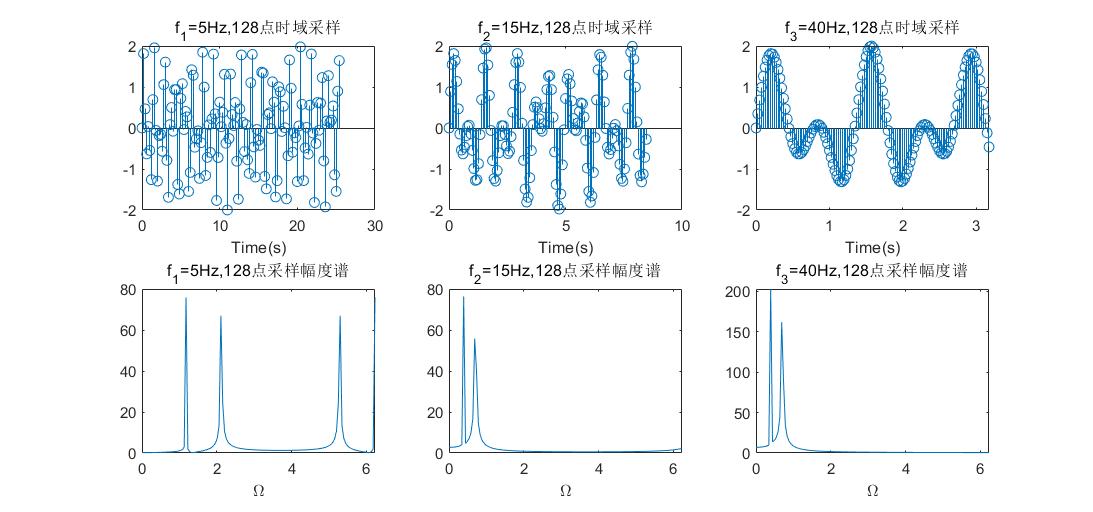
上图是利用FFT计算得到的两个离散信号的卷积结果。

**3.滤波**

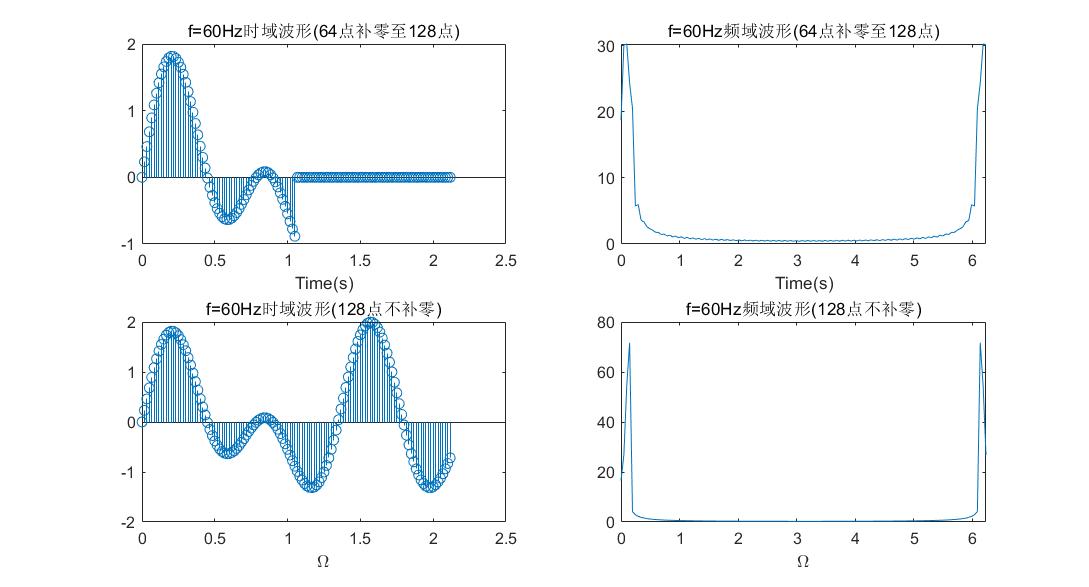


上图是使用理想低通滤波器对原信号进行滤波的结果，可以看到，在滤波后的信号中产生了振铃效应。

**4.采样频率变化**

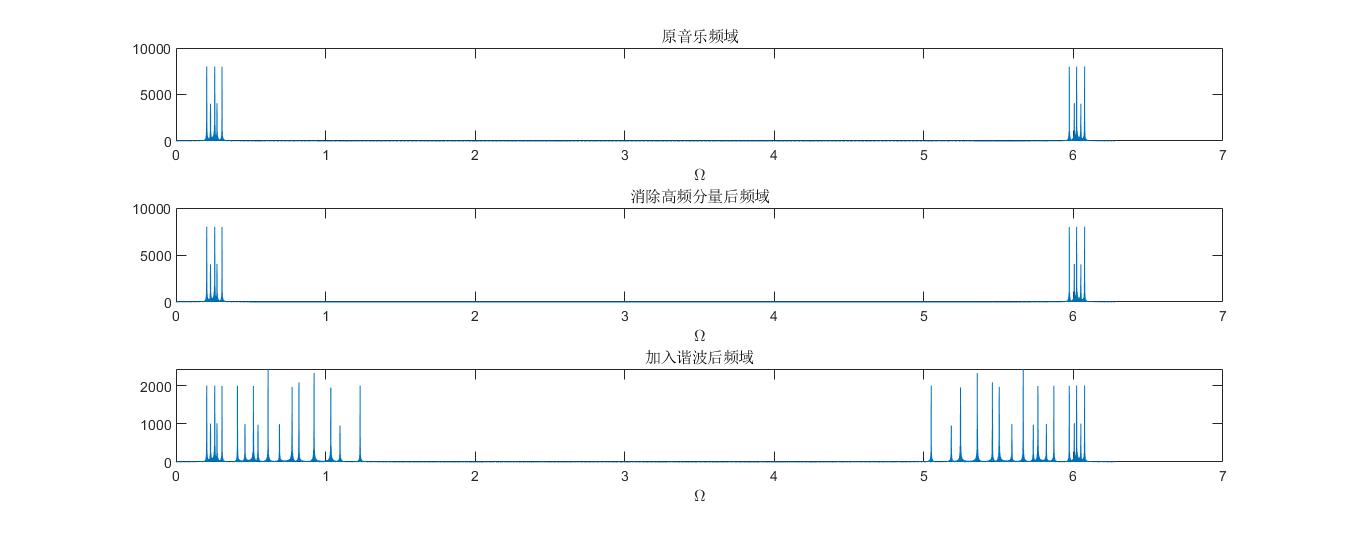


上图是使用不同频率进行采样的波形和幅度谱。



上图在60Hz采样补零和直接采样的结果和幅度谱。

**5.趣味小实验**



上图分别是原音频的幅度谱（上），经过理想低通滤波器去除高频分量之后的幅度谱（中），加入谐波之后的幅度谱（下）。

**三、思考题**

**1.理解线性卷积、圆周卷积和循环卷积的区别。用FFT计算线性卷积时，FFT的长度和N应满足什么条件？**

答： 线性卷积直接使用两个序列进行反向、移位、相乘、求和得到；圆周卷积是将两个序列放到圆上进行反褶、圆移、相乘、求和，相当于进行周期延拓之后进行线性卷积再取主值。在用FFT计算线性卷积时，FFT长度应该大于等于2N-1。

**2.实数序列****的频域幅度谱和相位谱有什么规律？虚数序列****的频域幅度谱和相位谱有什么规律？**

答：实数序列的幅度谱是偶函数，相位谱是奇函数；虚数序列的幅度谱也是偶函数，相位谱也是奇函数；并且两个仅相差一个虚数单位的实数序列和虚数序列的幅度谱相同，相位谱相差。

**3.FFT也可以滤除信号中的噪声分量，从而恢复信号，那么它与IIR和FIR滤波器的区别是什么？**

答：IIR滤波器是用两个多项式之比的有理分式来逼近频率特性，是无限冲激响应滤波器；FIR滤波器则是用的多项式来逼近要求的频率特性，没有可控制的极点；FIR滤波器设计时要反复计算，但由于h(n)为有限长序列，则可借助FFT进行快速卷积运算；而FFT滤波是指将序列通过FFT变换到频域后进行加窗截断在进行IFFT变换得到时域信号。

**4.同一连续信号离散化后有两种情况，第一种是取较长的离散序列求FFT；第二种是取较短的离散序列，结尾补零扩展成与第一种中的长度相等，再求FFT。在上述两种情况下，信号的频谱有何异同点？**

答：如两个序列都没有取到原连续信号的整周期，则第一种方法频谱更接近原连续信号的频谱，因为其采集的分量更多；第二种方法是在较少采样点的FFT变换后的频谱进行更多点的采样。如果两个序列都进行了原信号的整周期截断，则信号的频谱都与原来连续信号的频谱形状相同，但第一种的幅值更大（由实验内容4）。