

# 数字信号处理课程实验      实验报告



报告人： 李 毅 PB22051031

院 系： 信息科学技术学院

时 间： 2024 年 12 月 30 日

目录

实验 1	连续信号的离散化	3
1.1	实验目的	3
1.2	实验内容分析	3
1.2.1	实验内容 1	3
1.2.2	实验内容 2	4
1.3	实验现象总结	6
1.4	源代码	6
实验 2	栅栏效应	10
2.1	实验目的	10
2.2	实验内容	10
2.3	实验总结	10
2.4	源代码	11
实验 3	频谱泄漏	13
3.1	实验目的	13
3.2	实验内容分析	13
3.3	源代码	15
实验 4	FFT 算法	20
4.1	实验目的	20
4.2	实验内容分析	20
实验 5	信号通过 IIR 滤波器	24
5.1	实验目的	24
5.2	实验内容分析	24
5.2.1	巴特沃斯 IIR 滤波器设计	24
5.2.2	数字切比雪夫 I 型高通滤波器设计	25
5.3	源代码	26
实验 6	窗函数	30
6.1	实验目的	30
6.2	实验内容分析	30
6.2.1	实验内容 1	30
6.2.2	实验内容 2	31
6.3	实验总结	31
6.4	源代码	32

<b>实验 7    FIR 滤波器的窗函数设计方法</b>	<b>35</b>
7.1    实验目的 . . . . .	35
7.2    实验内容分析 . . . . .	35
7.2.1    矩形窗函数设计带通滤波器 . . . . .	35
7.2.2    三角形窗函数设计带通滤波器 . . . . .	35
7.2.3    汉宁窗函数设计带通滤波器 . . . . .	36
7.2.4    汉明窗函数设计带通滤波器 . . . . .	37
7.2.5    布莱克曼窗函数设计带通滤波器 . . . . .	37
7.3    实验总结 . . . . .	38
7.3.1    窗函数对滤波器性能的影响 . . . . .	38
7.3.2    阶数对滤波器性能的影响 . . . . .	38
7.3.3    Gibbs 效应 . . . . .	39
7.4    源代码 . . . . .	39
 <b>实验 8    线性相位系统</b>	 <b>44</b>
8.1    实验目的 . . . . .	44
8.2    实验内容分析 . . . . .	44
8.2.1    IIR 低通滤波器设计 . . . . .	44
8.2.2    FIR 低通滤波器设计 . . . . .	45
8.2.3    单一信号通过滤波器 . . . . .	47
8.2.4    混合信号通过滤波器 . . . . .	47
8.3    实验总结 . . . . .	48
8.4    源代码 . . . . .	48

## 实验 1 连续信号的离散化

### 1.1 实验目的

1. 探究连续时间信号与时间取样序列之间的关系，取样前后的变化。
2. 理解取样操作的本质，是希望取样序列能够承载和表达原信号的所有信息。
3. 对比原信号频谱与取样序列频谱之间的差异，判断取样序列对原信号表达的有效性。
4. 理解奈奎斯特取样准则的物理意义及初步运用，掌握如何针对实际信号选择合适的取样频率。

### 1.2 实验内容分析

#### 1.2.1 实验内容 1

设  $x_1(t) = \frac{\sin 2\pi f_h t}{t}$ ,  $f_h = 100\text{Hz}$

- (1) 依据  $x_1(t)$  的 CFT 表达式，画出幅频特性图，观察  $x_1(t)$  的幅度谱，带宽是否受限并有确定的最高频率  $f_h$ ，根据奈奎斯特取样准则确定合适的取样频率  $f_s$ 。
- (2) 在以下 5 个取样频率  $f_s$  下： $0.3f_h$ ,  $0.6f_h$ ,  $1.2f_h$ ,  $1.8f_h$ ,  $2.4f_h$ ，分别对  $x_1(t)$  进行时域取样，形成取样序列  $x_1(n)$ ，对  $x_1(n)$  进行 DTFT。依据  $x_1(n)$  的 DTFT 表达式，画出幅频特性图，观察  $x_1(n)$  的幅度谱，与  $x_1(t)$  的幅度谱进行对比和分析讨论。

$x_1(t)$  的时域波形和幅度谱如下：

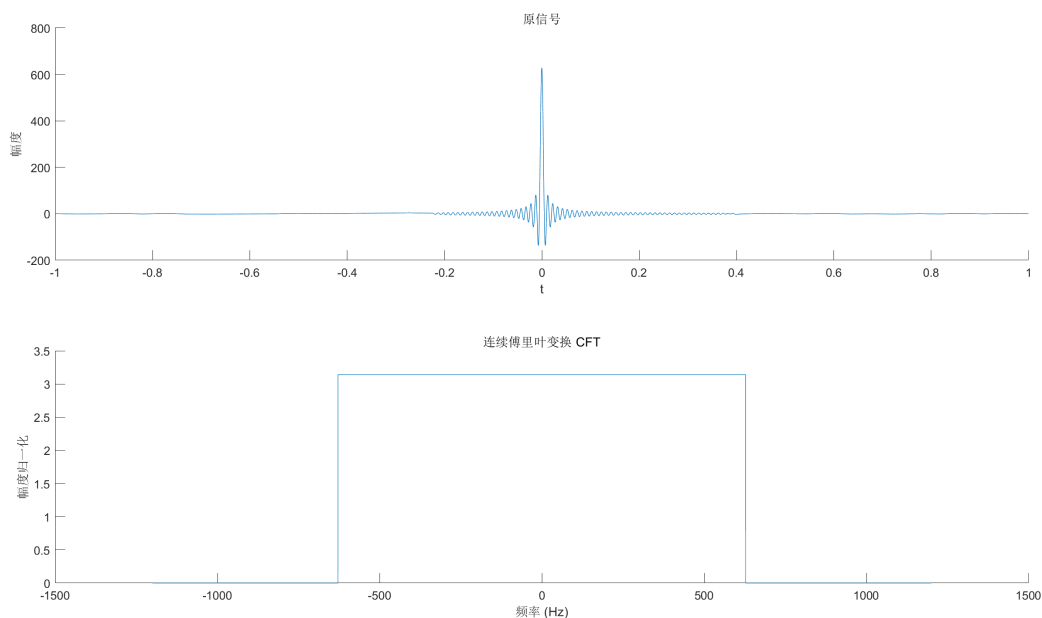


图 1:  $x_1(t)$  的时域波形和幅度谱性

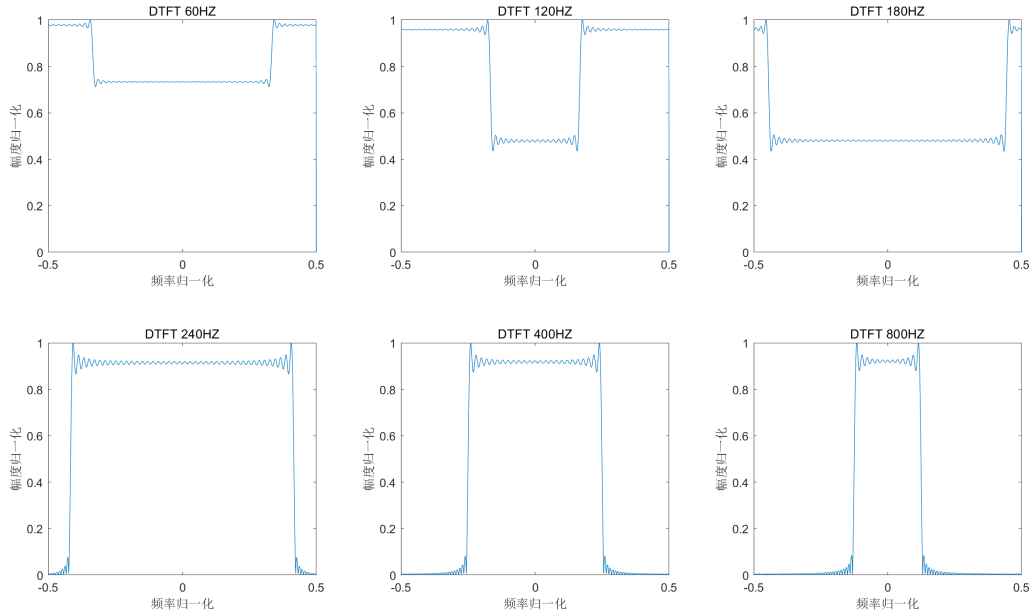


图 2: 不同取样频率的幅度谱

根据奈奎斯特采样准则, 信号  $x_1(t)$  的采样频率必须大于其最高频率的两倍, 即至少为  $2f_h = 200 \text{ Hz}$ 。如果采样频率低于该值 (例如  $f_s = 0.3f_h$  或  $f_s = 0.6f_h$ ), 则会出现严重的混叠现象, 导致信号的失真。

当  $f_s = 0.3f_h$  或  $f_s = 0.6f_h$  时, 取样后的信号出现明显的频谱混叠, 信号的频率成分发生重叠, 导致严重的失真。

随着取样频率逐渐增大, 尤其是当  $f_s = 2.4f_h$  时, 频谱变得更为清晰, 数字频率段较为紧凑, 信号的频率成分基本不再重叠, 失真减小。

### 1.2.2 实验内容 2

设  $x_2(t) = e^{-100t} \sin(2\pi f_q t)$ ,  $f_q = 100 \text{ Hz}$

- (1) 使用 MATLAB 内置的 `fourier` 函数对  $x_2(t)$  进行 CFT, 并画出幅频特性图, 观察  $x_2(t)$  的幅度谱, 观察  $x_2(t)$  带宽是否无限、频率高端幅度是否有明显衰减, 并根据频谱能量 95% 的近似原则确定合适的取样频率  $f_s$ 。
- (2)  $f_s$  取  $f_q$ ,  $4f_q$ ,  $6f_q$ ,  $10f_q$  时, 分别对  $x_2(t)$  进行时域取样, 形成不同的取样序列  $x_2(n)$ , 对  $x_2(n)$  分别进行 DTFT, 并画出幅频特性图。随着取样频率  $f_s$  的增加, 观察频谱混叠现象的变化情况, 并进行分析讨论。

$x_2(t)$  的时域波形和幅度谱如下:

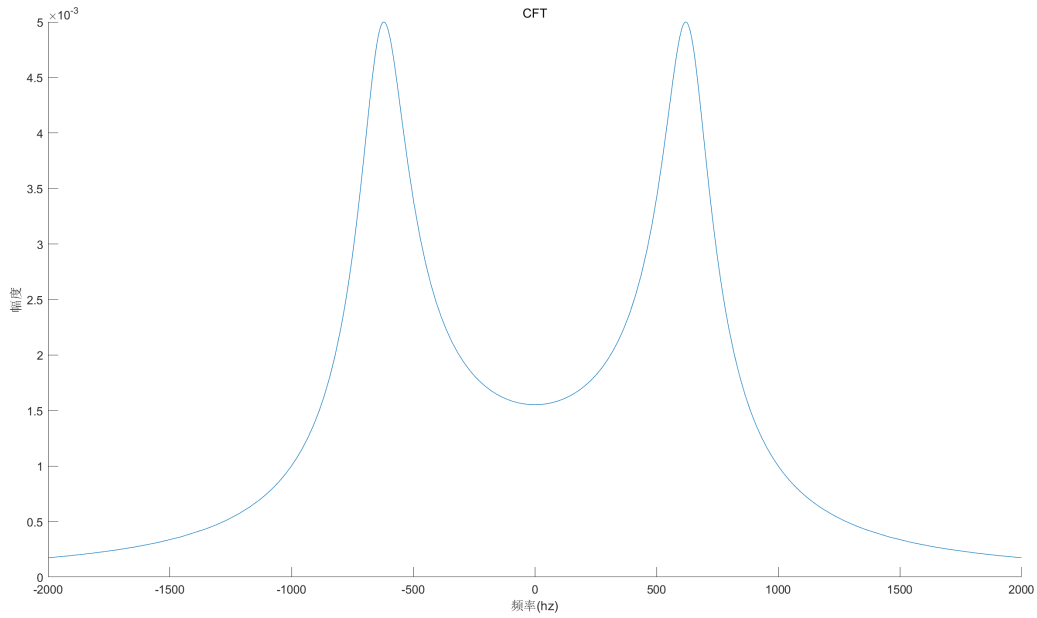


图 3:  $x_2(t)$  的时域波形和幅度谱性

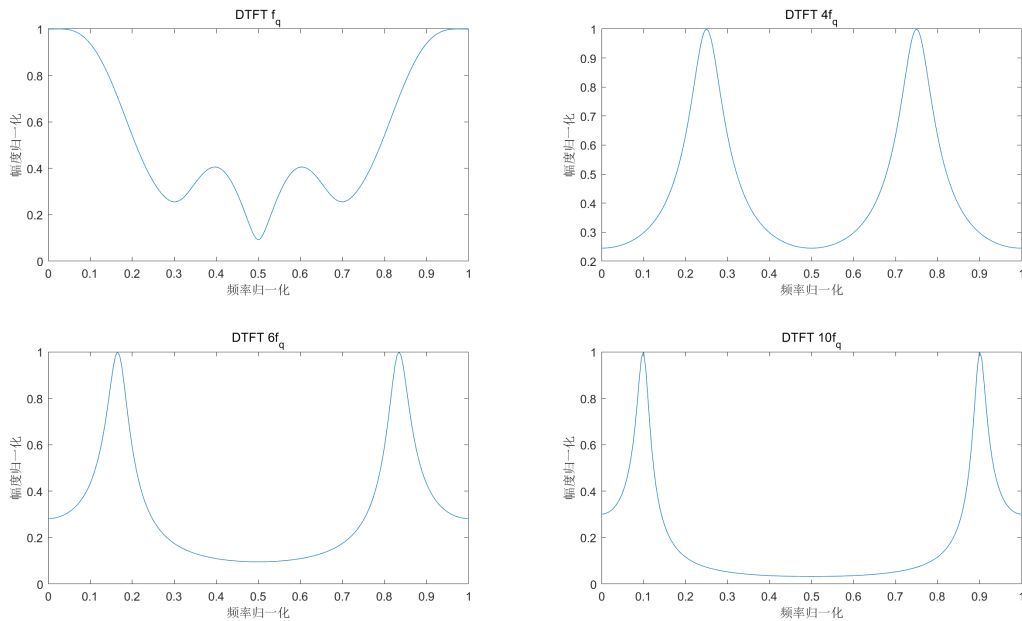


图 4: 不同取样频率的幅度谱

当采样频率  $f_s$  小于信号的两倍（如  $f_s = f_q$  或  $f_s = 4f_q$ ），出现明显的混叠现象。混叠导致低频部分的能量与高频部分发生重叠，造成频谱的扭曲。当采样频率  $f_s$  增加到  $6f_q$  或  $10f_q$  时，混叠现象逐渐

减弱，信号的频谱变得更加清晰，频谱上的能量分布更加合理。

### 1.3 实验现象总结

通过本次实验，我们观察到采样频率对信号的影响。低采样频率会导致严重的混叠现象，失真加剧；而随着采样频率的提高，信号的恢复效果逐渐改善，频谱越来越接近原始信号的频谱。特别是对于带宽有限的信号，选择合适的采样频率（至少为信号带宽的两倍）是避免混叠的关键。而对于带宽不受限的信号（如指数衰减信号），则需要更高的采样频率才能保持信号的完整性。

### 1.4 源代码

实验项目 1 源代码见附件 lab1.m。如下：

```
1      close all
2      clear all
3      syms t;
4      f_h=100;
5      f_x1 = sin(2*pi*f_h*t)/t;
6      FT_x1 = fourier(f_x1); % Fourier transform
7      figure(1);
8      subplot(2,1,1); hold on;
9      fplot(f_x1,[-1,1]) % 绘制
10     xlabel('t');
11     ylabel('幅度');
12
13     title('原信号');
14     subplot(2,1,2); hold on;
15     fplot(abs(FT_x1),[-1200,1200]);
16     xlabel('频率 (Hz)');
17     ylabel('幅度归一化');
18
19     title('连续傅里叶变换 CFT');
20     figure(2);
21     freq=[0.6,1.2,1.8,2.4,4,8];
22     for l=1:6
23         n=-50:50; % 定义采样点范围
24         f_s= freq(l)*f_h; % 采样频率
25         T=1/f_s; % 采样周期
26         f_x1n=sin(2*pi*f_h*n*T)./(n*T); % 理想采样信号序列
27
28         % 处理 n=0 时的情况，避免 0/0 的不定值问题
```

```

29     f_x1n(n==0) = 2*pi*f_h;
30
31     %DTFT
32     DTFT_fx1n=zeros(1,1001);
33     w=-pi:2*pi/1000:pi;
34     for i=1:1000
35         for k=1:101
36             DTFT_fx1n(i)=DTFT_fx1n(i)+f_x1n(k)*exp(-1i*w(i)*(k-51))
37             ;
38         end
39     end
40     max1=max(abs(DTFT_fx1n));
41     subplot(2,3,1);
42     plot(w/(2*pi),abs(DTFT_fx1n)/max1);
43     title(['DTFT ', num2str(freq(1)*f_h) 'HZ']);
44     xlabel('频率归一化');
45     ylabel('幅度归一化');
46 end

```

lab1.m

实验项目 2 源代码见附件 lab1\_2.m。如下：

```

1
2     close all
3     clear all
4     syms t;
5     f_q=100;
6     f_x2 = sin(2*pi*f_q*t)*exp(-100*t)*heaviside(t);
7     syms w;
8     FT_x2 = fourier(f_x2,w);
9     figure(1);
10    hold on;
11    fplot(abs(FT_x2),[-2000,2000]);
12    title('CFT');
13    xlabel('频率(hz)');
14    ylabel('幅度');
15
16    f_w= @(w) (abs(- 1i./(2*(w.*1i - pi*200i + 100)) + 1i./(2*(w.*1i +
    pi*200i + 100))))).^2;

```



```
17
18 E=integral(f_w,-2000,2000)/2*pi;
19 E_1=0.95*E;
20 for i=1:2000
21     if(integral(f_w,-i,i)/2*pi>=0.95*E_1)
22         break;
23     end
24 end
25
26 figure(2);
27 t_1=0:1/f_q:1;
28 t_2=0:1/(4*f_q):1;
29 t_3=0:1/(6*f_q):1;
30 t_4=0:1/(10*f_q):1;
31 f_t1 = sin(2*pi*f_q*t_1).*exp(-100*t_1);
32 f_t2 = sin(2*pi*f_q*t_2).*exp(-100*t_2);
33 f_t3 = sin(2*pi*f_q*t_3).*exp(-100*t_3);
34 f_t4 = sin(2*pi*f_q*t_4).*exp(-100*t_4);
35
36 w=0:pi/100:2*pi;
37 DTFT_w1=zeros(1,201);
38 for i=1:201
39     for k=1:101
40         DTFT_w1(i)=DTFT_w1(i)+f_t1(k)*exp(-1i*w(i)*(k-1));
41     end
42 end
43 max1=max(abs(DTFT_w1));
44 subplot(2,2,1);
45 plot(w/(2*pi),abs(DTFT_w1)/max1);
46 title('DTFT f_q');
47 xlabel('频率归一化');
48 ylabel('幅度归一化');
49
50
51 DTFT_w2=zeros(1,201);
52 for i=1:201
53     for k=1:401
54         DTFT_w2(i)=DTFT_w2(i)+f_t2(k)*exp(-1i*w(i)*(k-1));
55     end
```

```
56     end
57     max2=max(abs(DTFT_w2));
58     subplot(2,2,2);
59     plot(w/(2*pi),abs(DTFT_w2)/max2);
60     title('DTFT 4f_q');
61     xlabel('频率归一化');
62     ylabel('幅度归一化');
63
64
65     DTFT_w3=zeros(1,201);
66     for i=1:201
67         for k=1:601
68             DTFT_w3(i)=DTFT_w3(i)+f_t3(k)*exp(-1i*w(i)*(k-1));
69         end
70     end
71     max3=max(abs(DTFT_w3));
72     subplot(2,2,3);
73     plot(w/(2*pi),abs(DTFT_w3)/max3);
74     title('DTFT 6f_q');
75     xlabel('频率归一化');
76     ylabel('幅度归一化');
77
78
79     DTFT_w4=zeros(1,201);
80     for i=1:201
81         for k=1:1001
82             DTFT_w4(i)=DTFT_w4(i)+f_t4(k)*exp(-1i*w(i)*(k-1));
83         end
84     end
85     max4=max(abs(DTFT_w4));
86     subplot(2,2,4);
87     plot(w/(2*pi),abs(DTFT_w4)/max4);
88     title('DTFT 10f_q');
89     xlabel('频率归一化');
90     ylabel('幅度归一化');
```

lab1\_2.m

## 实验 2 栅栏效应

### 2.1 实验目的

1. 理解栅栏效应是连续频谱取样离散化之后呈现的一种视觉现象。
2. 理解在离散傅里叶变换（DFT）的定义下，栅栏效应呈现的不是误差

### 2.2 实验内容

1. 设置  $N$  点离散序列  $x(n) = [1, 1, 0, 1]$
2. 对  $x(n)$  分别做 DTFT 和 DFT，画出  $X(e^{j\omega})$  和  $X(k)$  的幅频特性曲线，观察描述栅栏效应现象
3. 用  $X(k)$  和内插函数重建  $X(e^{j\omega})$ ，画出  $\hat{X}(e^{j\omega})$  的幅频特性曲线，并与  $X(e^{j\omega})$  的幅频特性曲线进行比较讨论。

绘制幅频特性曲线如下：

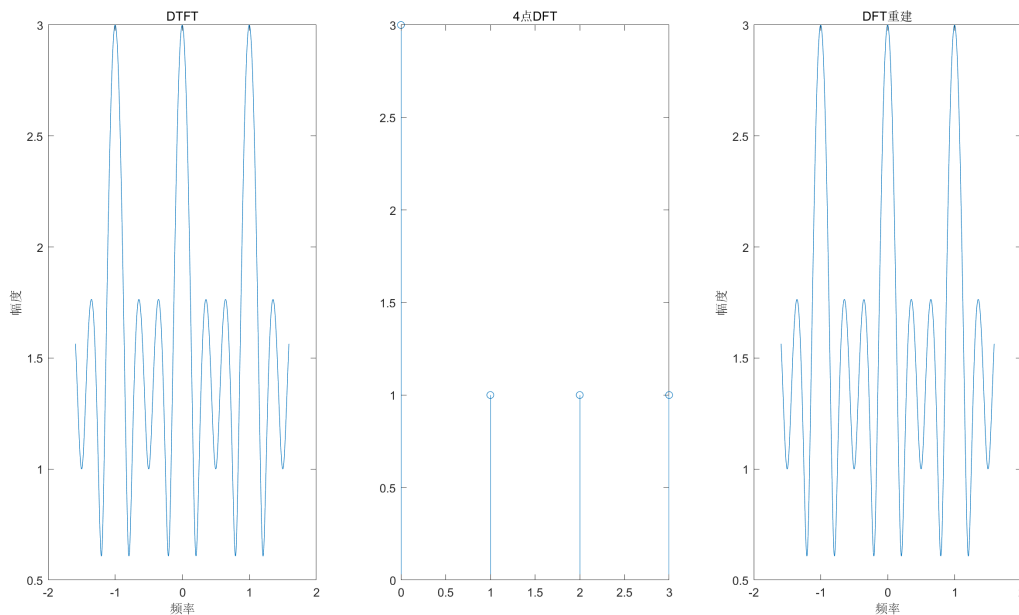


图 5: 幅频特性曲线

### 2.3 实验总结

栅栏效应是指在离散傅里叶变换（DFT）中，由于采样频率不足，导致连续时间信号的频谱在离散化后出现周期性失真，并影响信号的频谱表示。

在本实验中，我们对比了  $X(e^{j\omega})$  和  $X(k)$  的幅频特性，观察到 DFT 频谱是离散的，且在高频部分出现了明显的折叠和失真，这正是栅栏效应的表现。

为了缓解栅栏效应并恢复尽可能接近原始的频谱，我们使用了内插函数对 DFT 频谱  $X(k)$  进行重建。通过内插，我们能够在离散的频率点之间进行平滑过渡，从而恢复出更加平滑的频谱。

然而，虽然插值能够改善频谱的平滑性，但由于离散化本身的局限性，栅栏效应并不能完全消失。通过与原始的  $X(e^{j\omega})$  进行对比，我们可以看到，虽然频谱变得更加平滑，但插值方法并没有完全消除频谱的离散特性。重建后的频谱较 DFT 频谱更接近于原始的连续频谱，但仍然无法完全恢复到原始信号的频率内容。

栅栏效应并不是误差，而是离散傅里叶变换（DFT）中固有的现象。它源于频谱的离散化，频率分量的周期性重叠导致了频谱的失真。当信号的采样频率不足时，频谱中的高频部分会出现折叠，导致信息丢失或误解。因此，栅栏效应与采样频率密切相关，在采样频率较低时，频谱的失真尤为严重。

## 2.4 源代码

源代码见附件 lab2.m。如下：

```
1  x_n=[1,1,0,1];
2  %DTFT
3  w=-10:0.02:10;
4  DTFT_Xw=zeros(1,1001);
5  for i=1:1001
6      for k=1:4
7          DTFT_Xw(i)=DTFT_Xw(i)+x_n(k)*exp(-1i*w(i)*k);
8      end
9  end
10 subplot(1,3,1);
11 plot(w/(2*pi),abs(DTFT_Xw));
12 title('DTFT');
13 xlabel('频率');
14 ylabel('幅度');
15
16 n=0:3;
17 k=0:3;
18 DFT_Xw=x_n*(exp(-1i*pi/2)).^(n'*k);
19 subplot(1,3,2);
20 stem(k,abs(DFT_Xw));
21 title('4点DFT');
22
23 DTFT_rebuild=zeros(1,1001);
24 w=-10:0.02:10;
25 for i=1:1001
26     for k=0:3
```

```
27         DTFT_rebuild=DTFT_rebuild+1/4*DFT_Xw(k+1)*(1-exp(-1i*w(i)*4)
28             )/(1-exp(1i*2*pi*k/4)*exp(-i*w(i)));
29     end
30 end
31 subplot(1,3,3);
32 plot(w/(2*pi),abs(DTFT_Xw));
33 title('DFT 重建');
34 xlabel('频率');
35 ylabel('幅度');
```

lab2.m

## 实验 3 频谱泄漏

### 3.1 实验目的

1. 理解这里的误差（偏差）概念，是指处理之后相对于处理之前的变化。
2. 理解  $x(n)$  如何通过 DFT 来反映表达  $x(t)$  的频谱情况。
3. 频谱泄漏是指离散序列  $x(n)$  DFT 相对于原信号  $x(t)$  在频谱上多显示出的频率成分。
4. 产生频谱泄漏的根本原因，是对原信号  $x(t)$  的截短；信号截短之后，相对于截短之前的原信号一定发生泄漏，与时域离散化无关，与 DFT 无关。
5. 理解 DFT 对频谱泄漏的呈现形态，与 DTFT 频谱离散化的频率取样点位置方式有关

### 3.2 实验内容分析

设  $x(t) = e^{j2\pi f_1 t} + e^{j2\pi f_2 t}$ ，式中  $f_1 = 24\text{Hz}$ ， $f_2 = 60\text{Hz}$ 。

1. 对  $x(t)$  做 CFT 并进行幅度归一化，给出  $X(j\Omega)$  表达式，说明其幅频特性。
2. 判断  $x(t)$  是否为周期函数。如果是，则确定  $x(t)$  的最小周期  $T_0$  和基频  $\Omega_0$ ，对  $x(t)$  进行 FS 展开，给出  $X(m\Omega_0)$  表达式，说明其幅频特性，并与上述 CFT 的结果进行比较讨论。
3. 对  $x(t)$  用矩形窗截短，得到  $x_1(t)$ 。矩形窗宽度的选择，分为两种情况：
  - 矩形窗宽度等于  $x(t)$  最小周期  $T_0$  的整数倍（记为  $R_1$ ）；
  - 矩形窗宽度不等于  $x(t)$  最小周期  $T_0$  的整数倍（记为  $R_2$ ）。

本次实验取  $R_1 = 2T_0$  和  $R_2 = 1.6T_0$ 。在  $R_1$  和  $R_2$  两种宽度下，对  $x_1(t)$  分别做 CFT，画出  $X_1(j\Omega)$  的幅频特性曲线，并与  $X(j\Omega)$  的幅频特性进行比较讨论。

实验项目 1-3 实验结果如下：

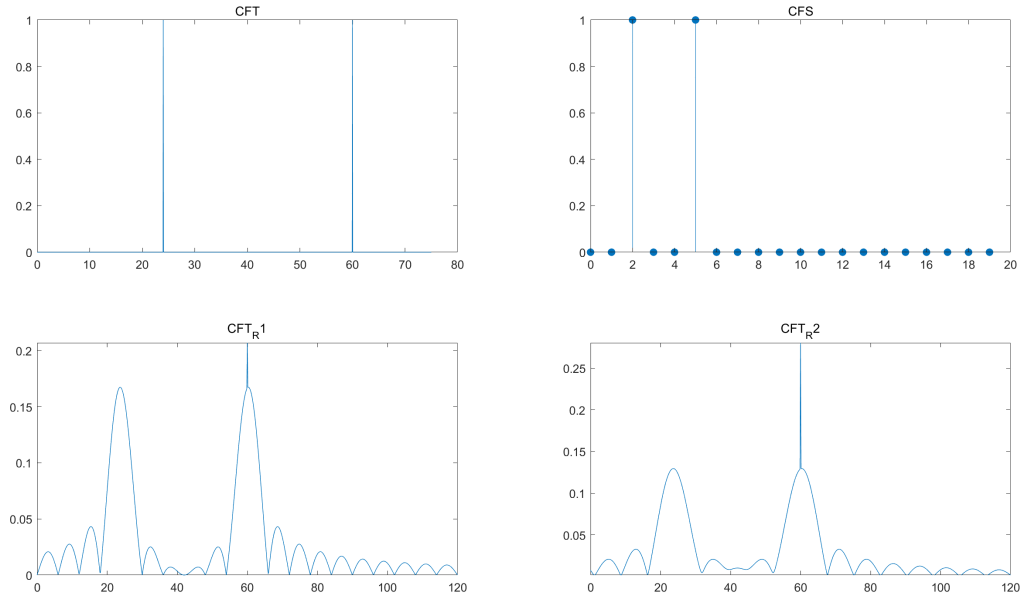


图 6: 实验项目 1-3 实验结果

实验项目 1: 对于该信号, CFT 结果应该是由两个频率成分  $f_1 = 24\text{Hz}$  和  $f_2 = 60\text{Hz}$  组成的频谱。两个频率成分的幅频响应是离散的, 且没有其他的频谱成分。幅频特性表现为在这两个频率点有一个幅值峰值, 其他频率点幅度为零。

实验项目 2: 显然信号  $x(t) = e^{j2\pi f_1 t} + e^{j2\pi f_2 t}$  为周期信号。周期为  $T_0 = \frac{1}{\text{最小公倍数}(f_1, f_2)} = \frac{1}{\text{最小公倍数}(24, 60)} = \frac{1}{120}$  秒。基频  $\Omega_0 = 2\pi f_0 = 2\pi \times 120 \text{ rad/s}$ 。信号的傅里叶级数 (FS) 展开可以写为:

$$X(m\Omega_0) = 2[\delta(m\Omega_0 - 2\pi f_1) + \delta(m\Omega_0 - 2\pi f_2)]$$

这里的傅里叶级数展开显示信号的频谱是离散的且只包含两个频率分量  $f_1$  和  $f_2$ 。与连续傅里叶变换 (CFT) 的结果相同, 频谱在这两个频率点有非零值, 其他频率点的幅度为零。

实验项目 3: 对于矩形窗截短后的信号, 频谱会发生泄漏, 特别是在矩形窗宽度  $R_2 = 1.6T_0$  时, 频谱泄漏较为明显, 表现为频谱的旁瓣增加。相比之下, 宽度为  $R_1 = 2T_0$  时, 频谱泄漏较小。此结果表明, 矩形窗的宽度越大, 泄漏现象越轻微。

4. 分别取  $f_{s1} = 600\text{Hz}$ ,  $f_{s2} = 600\text{Hz}$ , 在  $R_1$  和  $R_2$  两种宽度下, 对  $x_1(t)$  进行离散化, 对取样序列  $x_1(n)$  做 DTFT。画出  $X_1(e^{j\omega})$  幅频特性曲线 (频域表示范围取两周期), 并与  $X(j\Omega)$  和  $X_1(j\Omega)$  的幅频特性进行比较讨论。
5. 在  $R_1$  和  $R_2$  两种宽度下, 对  $x_1(n)$  做 DFT, 画出  $X_1(k)$  完整的幅频特性曲线 ( $k = 0, 1, \dots, N-1$ ), 并与  $X_1(e^{j\omega})$ 、 $X(j\Omega)$  和  $X_1(j\Omega)$  的幅频特性进行比较讨论。

6. 在  $R_1$  和  $R_2$  两种宽度下，对  $x_1(n)$  补两倍长度于自身的零值形成延长序列  $x_2(n)$ 。对  $x_2(n)$  做 DFT，画出  $X_2(k)$  完整的幅频特性曲线 ( $k = 0, 1, \dots, N-1$ )，并与  $X_1(k)$ 、 $X_1(e^{j\omega})$ 、 $X(j\Omega)$  和  $X_1(j\Omega)$  的幅频特性进行比较讨论。

实验项目 4-6 实验结果如下：

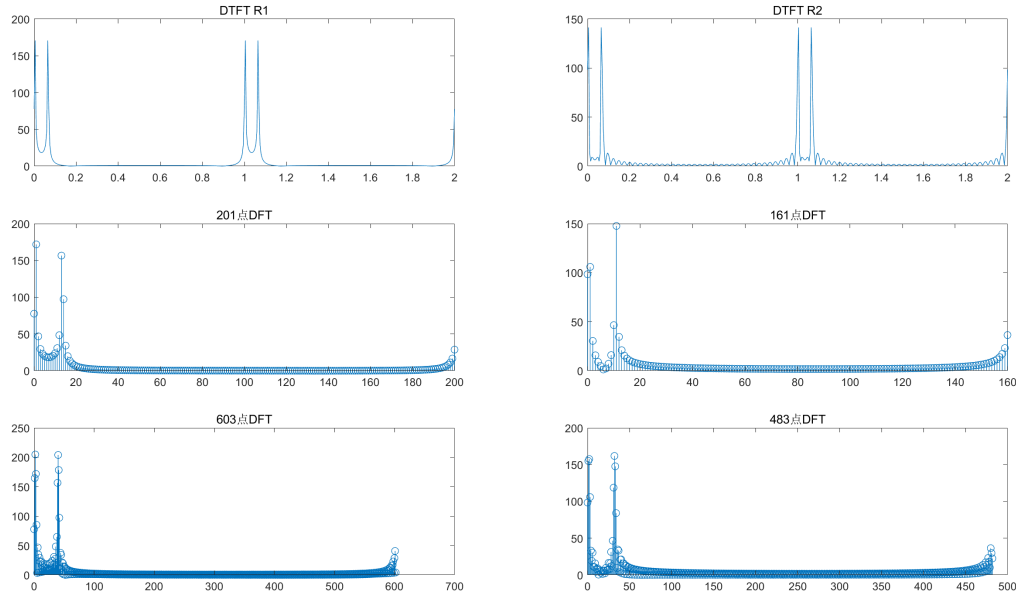


图 7: 实验项目 4-6 实验结果如下：

实验项目 4: 离散化后的信号频谱  $X_1(e^{j\omega})$  与连续信号的频谱相比，显示出明显的频谱泄漏，特别是在  $R_2$  宽度下。频谱泄漏主要表现为频率成分的旁瓣较高，且主要频率成分不再精确地集中在 24Hz 和 60Hz 上。通过对比  $X_1(e^{j\omega})$  和  $X(j\Omega)$  的幅频特性，发现离散化过程中的采样频率影响了频谱的展宽，影响了频谱泄漏现象。

实验项目 5: 与 DTFT 不同，DFT 会导致频谱在离散点上呈现，且频谱泄漏现象更加明显。通过对比  $X_1(k)$ 、 $X_1(e^{j\omega})$  和  $X(j\Omega)$ ，可以发现，DFT 的结果在某些频率点具有较高的旁瓣，尤其是在矩形窗宽度为  $R_2$  时，频谱泄漏现象尤为明显。DFT 的结果表明，采样点数（即变换的长度）和窗口宽度密切相关，较短的窗函数会引起较大的频谱泄漏。

实验项目 6: 通过对  $x_2(n)$  做 DFT，得到了频谱  $X_2(k)$ 。零填充可以增加信号的频率分辨率，尤其是在频谱显示中，频谱泄漏会相对减小，旁瓣的幅度也会有所减小。通过与原始的  $X_1(k)$  和  $X_1(e^{j\omega})$  对比，我们可以看到，虽然零填充增加了频率分辨率，但并未完全消除频谱泄漏。

### 3.3 源代码

实验项目 1-3 源代码见附件 lab3.m。实验项目 4-6 源代码见附件 lab3\_2.m。如下：



```
1     close all
2 clear all
3 f_1=24;f_2=60;
4
5 syms t;
6 f_t1=exp(1j*2*pi*f_1*t)+exp(1j*2*pi*f_2*t);
7
8 syms w;
9 FT_t1=fourier(f_t1,w);
10
11 w1 = 0:pi/10:150*pi;
12 FT_t1n = subs(FT_t1, w, w1);
13
14 idx = (FT_t1n == inf);
15 FT_t1n(idx) = 1;
16
17 subplot(2,2,1);
18 plot(w1/(2*pi), abs(FT_t1n));
19 title('CFT');
20
21 T0=1/12;
22 w_0=2*pi/T0;
23 N=20;
24 FS_1 = zeros(1, N);
25 w2 = zeros(1, N);
26
27 for m = 1: N
28     w2(m) = (m-1) * w_0;
29     FS_1(m) = int(f_t1*exp(-1j * w2(m) * t), t, 0, T0) / T0;
30 end
31
32 subplot(2,2,2);
33 stem(w2/w_0, abs(FS_1),'filled');
34 title('CFS');
35
36 %R1 矩形窗
37 R1 = heaviside(t) - heaviside(t-2*T0);
38 %R2 矩形窗
39 R2 = heaviside(t) - heaviside(t-1.6*T0);
```

```
40
41 f_r1=f_t1*R1;
42 f_r2=f_t1*R2;
43
44 syms w;
45 FT_r1 = fourier(f_r1, w);
46 subplot(2, 2, 3);
47 fplot(w/(2*pi), abs(FT_r1), [0, 2 * 2*pi*f_2]);
48 title('CFT_R1');
49
50 syms w;
51 FT_r2 = fourier(f_r2, w);
52 subplot(2, 2, 4);
53 fplot(w/(2*pi), abs(FT_r2), [0, 2 * 2*pi*f_2]);
54 title('CFT_R2');
```

lab3.m

```
1 f_1=2;f_2=40;T=1/12;
2 R_1=2*T;R_2=1.6*T;
3 f_s1=600;f_s2=600;
4 t1=-R_1:1/f_s1:R_1;
5 f_t1=exp(1i*2*pi*f_1*t1)+exp(1i*2*pi*f_2*t1);
6
7 t2=-R_2:1/f_s2:R_2;
8 f_t2=exp(1i*2*pi*f_1*t2)+exp(1i*2*pi*f_2*t2);
9 %4
10 w=0:pi/100:4*pi;
11 DTFT_w1=zeros(1,401);
12 for i=1:401
13     for k=1:201
14         DTFT_w1(i)=DTFT_w1(i)+f_t1(k)*exp(-1i*w(i)*(k-101));
15     end
16 end
17 subplot(3,2,1);
18 plot(w/(2*pi),abs(DTFT_w1));
19 title('DTFT R1');
20
21 DTFT_w2=zeros(1,401);
```

```
22 for i=1:401
23     for k=1:161
24         DTFT_w2(i)=DTFT_w2(i)+f_t2(k)*exp(-1i*w(i)*(k-81));
25     end
26 end
27 subplot(3,2,2);
28 plot(w/(2*pi),abs(DTFT_w2));
29 title('DTFT R2');
30 %5
31 n=0:200;
32 k=0:200;
33 DFT_w1=f_t1*(exp(-1i*2*pi/201)).^(n'*k);
34 subplot(3,2,3);
35 stem(k,abs(DFT_w1));
36 title('201点DFT');
37
38 n=0:160;
39 k=0:160;
40 DFT_w2=f_t2*(exp(-1i*2*pi/161)).^(n'*k);
41 subplot(3,2,4);
42 stem(k,abs(DFT_w2));
43 title('161点DFT');
44 %6
45 n=0:602;
46 k=0:602;
47 zero1=zeros(1,402);
48 f_t3=[f_t1,zero1];
49 DFT_w3=f_t3*(exp(-1i*2*pi/603)).^(n'*k);
50 subplot(3,2,5);
51 stem(k,abs(DFT_w3));
52 title('603点DFT');
53
54 n=0:482;
55 k=0:482;
56 zero2=zeros(1,322);
57 f_t4=[f_t2,zero2];
58 DFT_w4=f_t4*(exp(-1i*2*pi/483)).^(n'*k);
59 subplot(3,2,6);
60 stem(k,abs(DFT_w4));
```

61 | `title('483 点 DFT');`

lab3\_2.m

## 实验 4 FFT 算法

### 4.1 实验目的

1. 加深对快速傅里叶变换（FFT）的理解。
2. 实际编程实现 FFT 算法。

### 4.2 实验内容分析

设  $x(n) = 2\sin(\frac{\pi}{4}n) + \sin(\frac{5\pi}{8}n) + 3\sin(\frac{3\pi}{4}n)$

对序列  $x(n)$  做 DFT，使用 MATLAB 内置的 stem 函数画出幅度谱。编制按时间抽取的基 2FFT 算法程序，顺序输入、反序输出，对序列  $x(n)$  做 FFT。在命令行输出反序结果的幅度值，并将输出结果从反序转换为顺序，画出幅度谱。

FFT 算法编写思路如下：

按照如下图所示的蝶形图进行 fft 算法，将序列按奇偶索引分解成两部分。对于长度为 N 的序列，可以将其分解为两个 N/2 长度的子序列，这两个子序列在每一步递归中都能继续进行分解。将递归的子结果结合起来，即为最终的傅里叶变换结果。注意旋转因子和输出结果均为逆序。

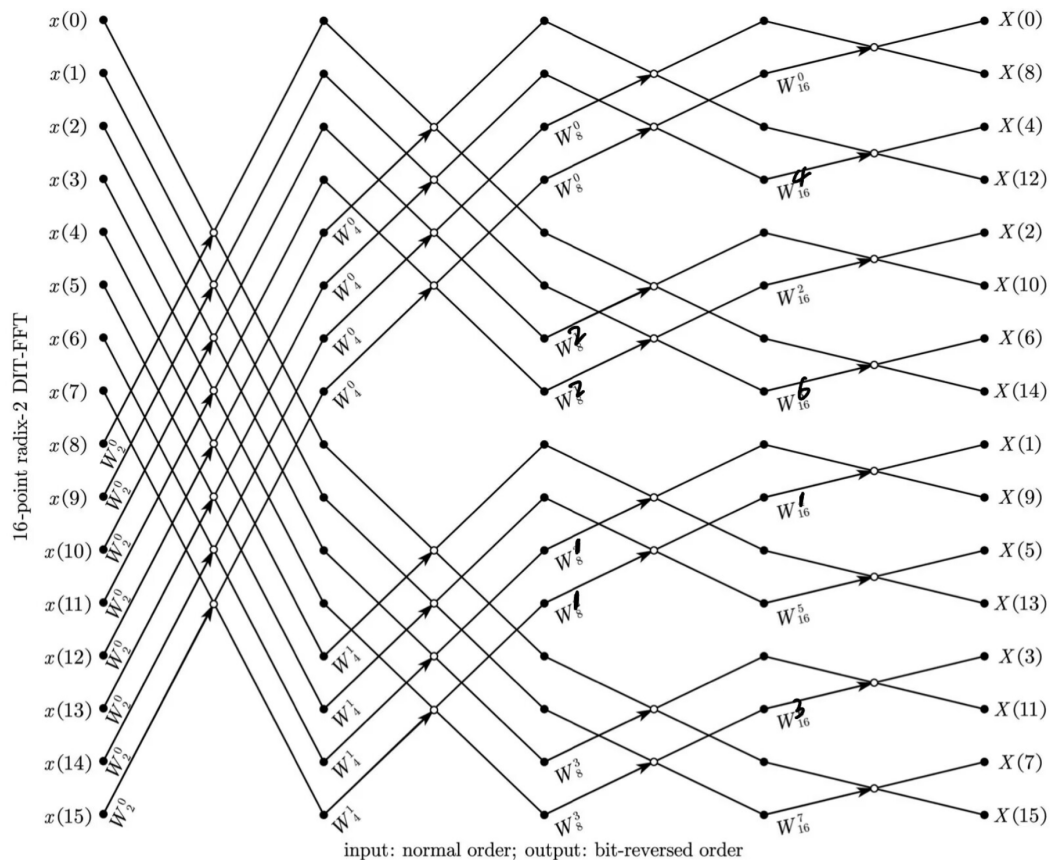


图 8: 16 点 FFT 蝶形图

第  $m$  级计算蝶形结计算公式如下:

$$X_m(k) = X_{m-1}(k) + X_{m-1}(k + \frac{N}{2^m} W_N^r)$$

$$X_m(k + \frac{N}{2^m}) = X_{m-1}(k) - X_{m-1}(k + \frac{N}{2^m} W_N^r)$$

源代码见附件 lab4.m。如下:

```
1  close all
2  clear all
3
4  % 设置序列的长度 n 和频率索引 k
5  n = 0:15;
6  k = 0:15;
7
8  % 定义信号
9  x_n = 2*sin(pi/4*n) + sin(5*pi/8*n) + 3*sin(3*pi/4*n);
10
11 % 计算DFT
12 DFT_xn = x_n * (exp(-1i * 2 * pi / 16)) .^ (n' * k);
13
14 % 绘制DFT的幅频特性
15 subplot(3,1,1);
16 stem(k, abs(DFT_xn));
17 title('16点DFT');
18
19 % 设置FFT参数, N=16
20 L = 4;
21 N = 16;
22 r = zeros(1,16);
23 x_m = [x_n; zeros(0,15); zeros(0,15); zeros(0,15); zeros(0,15)]; %
    初始化信号矩阵
24
25 for j = 1:4          % 对每一层进行迭代, 总共4层
26     signal = (bitget(k, L-j+1));
27     k2 = bitrevorder(k);          % 获取频率索引的位逆序
28     r = mod(k2, (2^(j-1)));        % 计算频率索引
29
30     % 计算新的信号矩阵
```

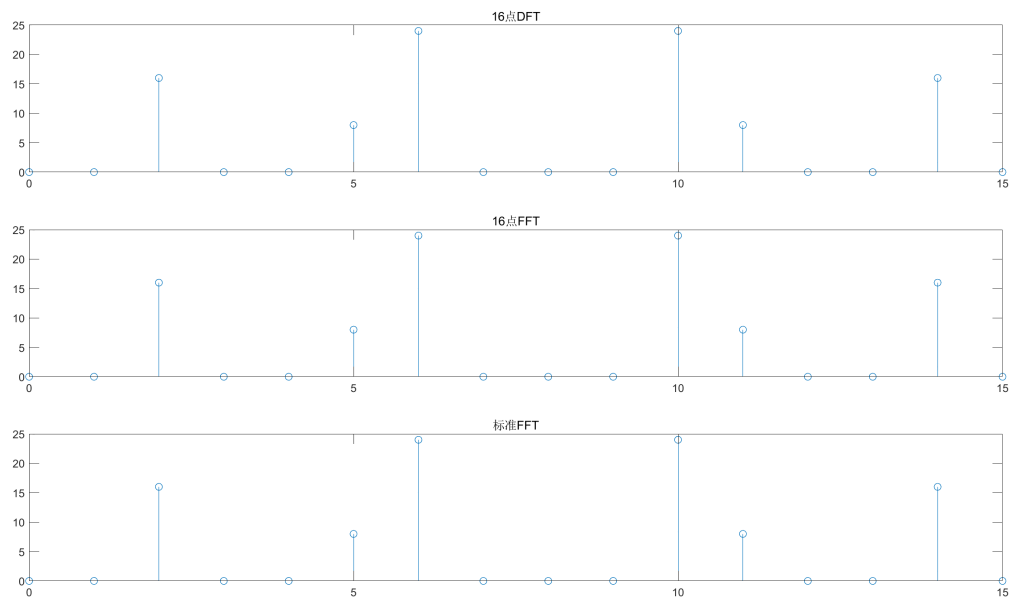
```

31     x_m(j+1,:) = (x_m(j,:) + (circshift(x_m(j,:), -2^(L-j))) .* exp
    (-1i * 2 * pi / (2^j) * r)) .* (1 - signal) + ...
32     ((circshift(x_m(j,:), 2^(L-j)) - x_m(j,:) .* exp
    (-1i * 2 * pi / (2^j) * r)) .* signal);
33 end
34
35 x_m(5,:) % 输出第5层的结果（即最终的FFT结果）
36 FFT_xn = x_m(5, k2 + 1); % 通过位反转，将结果恢复为正确顺序
37
38 % 绘制FFT的幅频特性
39 subplot(3,1,2);
40 stem(k, abs(FFT_xn));
41 title('16点FFT');
42
43 % MATLAB标准FFT结果的幅频特性
44 subplot(3,1,3);
45 stem(k, abs(fft(x_n)));
46 title('标准FFT');

```

lab4.m

实验结果如下：



ans =

列 1 至 5

-0.0000 + 0.0000i    0.0000 + 0.0000i    0.0000 + 0.0000i    0.0000 - 0.0000i    0.0000 -16.0000i

列 6 至 10

-0.0000 +24.0000i    -0.0000 -24.0000i    -0.0000 +16.0000i    -0.0000 + 0.0000i    0.0000 + 0.0000i

列 11 至 15

-0.0000 - 8.0000i    0.0000 - 0.0000i    0.0000 + 0.0000i    -0.0000 + 8.0000i    0.0000 - 0.0000i

列 16

-0.0000 - 0.0000i

图 9: FFT 幅度谱以及幅度值

观察得知，编写的 FFT 算法与 matlab 标准库的 fft 算法结果一致。



## 实验 5 信号通过 IIR 滤波器

### 5.1 实验目的

1. 理解系统对信号的作用、输入信号与输出信号的关系。
2. 任何复杂信号都可以看成是不同频率的简单正弦信号叠加的结果。

### 5.2 实验内容分析

#### 5.2.1 巴特沃斯 IIR 滤波器设计

设计一个 IIR 低通滤波器。通带起伏小于 1dB，止带衰减大于 40dB，过渡带宽小于  $0.1\pi$ ，通带截止频率为  $\omega_p(n) = r \frac{2\pi}{N}$ ，其中 r 分别取 5, 10, 15, 20, 40,  $N=100$ 。

生成一个周期为  $N=100$  的矩形信号序列  $x(n)$ ，满足：

$$x(n) = \begin{cases} 1 & 0 \leq n \leq \frac{N}{2} - 1 \\ 0 & \frac{N}{2} \leq n \leq N - 1 \end{cases}$$

取 10 个周期长度，通过低通滤波器，得到输出序列  $y(n)$ 。观察和比较滤波器取不同截止频率时， $x(n)$  和  $y(n)$  的时域波形、幅频特性的变化，特别是方波棱角的变化。

$x(n)$  和  $y(n)$  的时域波形、幅频特性如下：

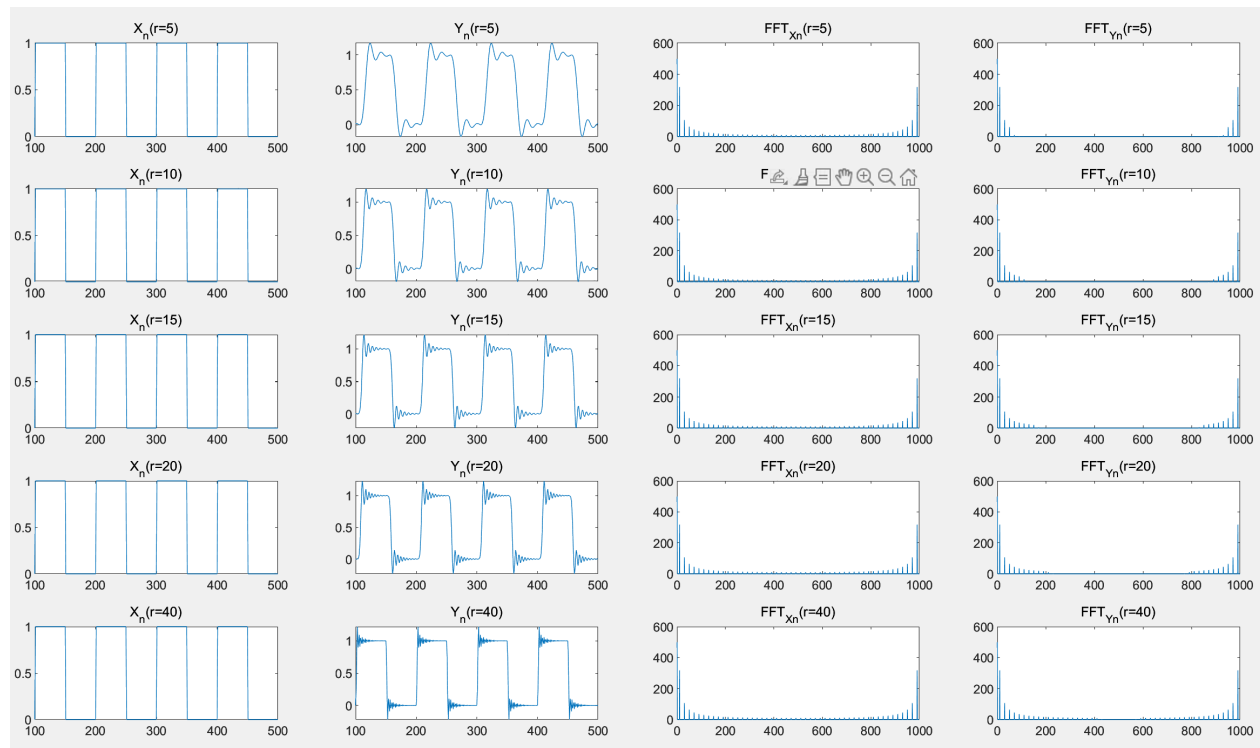


图 10: 不同  $r$  的  $x(n)$  和  $y(n)$  的时域波形、幅频特性

波形观察结果如下：

### (1) 时域波形的变化

输入信号  $x(n)$  是一个周期性矩形信号，具有明显的方波棱角。在经过低通滤波器处理后，输出信号  $y(n)$  的波形逐渐平滑。当截止频率较低时，方波的棱角部分被滤去，信号变得更加平滑。随着  $r$  值的增大即截止频率越高，输出信号的棱角越明显。

### (2) 幅频特性的变化

不同的截止频率设置影响了滤波器的幅频响应。高频分量被滤除。可以从  $y(n)$  的幅频特性上清晰的看到截止频率。

### 5.2.2 数字切比雪夫 I 型高通滤波器设计

采用双线性变换法设计一个数字切比雪夫 I 型高通滤波器。当  $\omega \leq 0.2\pi$  时，衰减大于 15dB；当  $0.3\pi \leq \omega \leq \pi$  时，衰减小于 1dB。并观察  $x(n)$  通过该高通滤波器后输出  $y_{hp}(n)$  的时域波形，并对比 (3) 中不同截止频率时输出时域波形与  $y_{hp}(n)$  叠加后的波形（即  $y(n) + y_{hp}(n)$  的时域波形）。

波形如下：

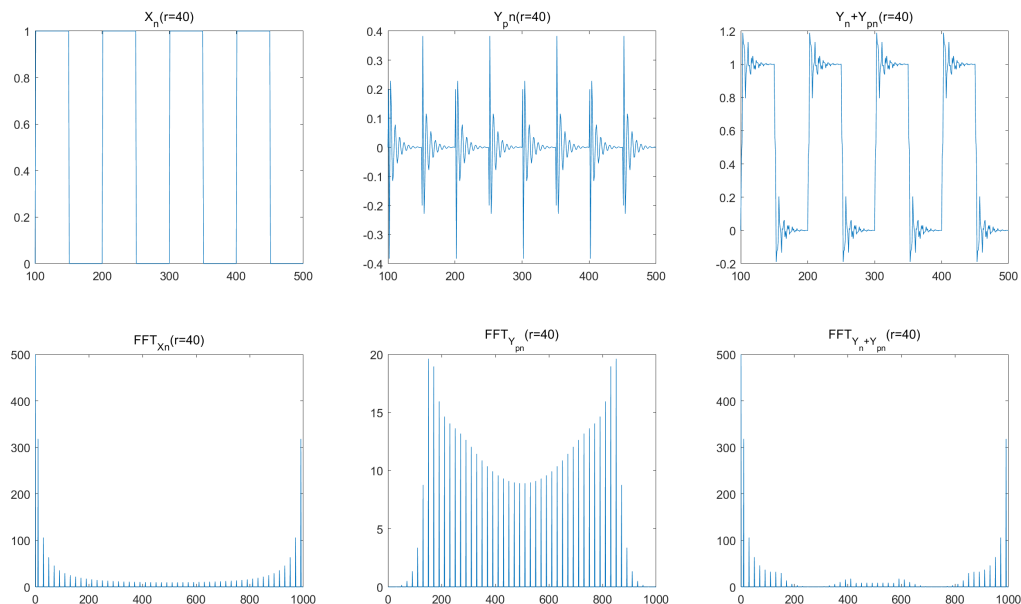


图 11:  $r=40$  时,  $x(n)$ ,  $y(n)$ ,  $y(n) + y_{hp}(n)$  的时域波形、幅频特性

- (1) 时域波形的变化：数字切比雪夫 I 型高通滤波器成功地从输入信号中滤除了低频成分，输出信号  $y_{hp}(n)$  主要保留了高频部分。特别是在频率低于  $0.2\pi$  时，信号的能量被大幅衰减，输出信号几乎为零，而在频率高于  $0.3\pi$  时，信号则保持相对完整。
- (2) 高通滤波器和低通滤波器的组合效果：通过将低通滤波器的输出信号  $y(n)$  与高通滤波器的输出信号  $y_{hp}(n)$  叠加，我们得到了一个具有更复杂频谱特征的信号。这个叠加信号展示了低频与高频成分

的组合效应，能够有效地保留输入信号中的频率特征，同时去除不需要的频率成分。

### 5.3 源代码

巴特沃斯 IIR 滤波器设计源代码见 lab5\_1.m，信号通过滤波器源代码见 lab5\_2.m。

数字切比雪夫 I 型高通滤波器设计源代码见 lab5\_3.m，信号通过滤波器源代码见 lab5\_4.m。

```
1  close all
2  clear all
3  % Matlab 使用归一化的频率参数（临界频率除以采样频率的 1/2）。
4  %这样临界频率参数的取值范围在 0 和 1 之间， 1 代表  $F_s/2$ （用角频率表示的时候对应  $\pi$ ）
5  R=[5,10,15,20,40];
6  r=R(5);
7  T_w=0.1*pi; %过渡带宽
8  N=100;
9  W_p=2*pi*r/N;
10 W_s=W_p+T_w;
11 f_s=1;
12 [n,W_n]=buttord(W_p/pi,W_s/pi,1,40);
13 [b,a]=butter(n,W_n);
14 [H,w]=freqz(b,a);
15 plot(w/pi, 20*log10(abs(H)));
16 xlabel('\omega/\pi');
17 ylabel('幅值(dB)');
```

lab5\_1.m

```
1  close all
2  clear all
3  % Matlab 使用归一化的频率参数（临界频率除以采样频率的 1/2）。
4  %这样临界频率参数的取值范围在 0 和 1 之间， 1 代表  $F_s/2$ （用角频率表示的时候对应  $\pi$ ）
5  R=[5,10,15,20,40];
6  for k=1:5
7      r=R(k);
8      T_w=0.1*pi; %过渡带宽
9      N=100;
10     W_p=2*pi*r/N;
11     W_s=W_p+T_w;
12     f_s=1;
```

```
13     [n,W_n]=buttord(W_p/pi,W_s/pi,1,40);
14     [b,a]=butter(n,W_n);
15     %freqz(b,a,512,f_s);
16
17     n=1:N*10;
18     x_n=[ones(1,N/2),zeros(1,N/2)];
19     X_n=repmat(x_n,1,10);%复制10次
20     Y_n=filter(b,a,X_n);
21     FFT_Xn=fft(X_n);
22     FFT_Yn=fft(Y_n);
23
24     subplot(5,4,4*k-3);
25     plot(1*N:5*N,X_n(1*N:5*N));
26     title(['X_n(r=' num2str(r) ')']);
27
28     subplot(5,4,4*k-2);
29     plot(1*N:5*N,Y_n(1*N:5*N));
30     title(['Y_n(r=' num2str(r) ')']);
31
32     subplot(5,4,4*k-1);
33     plot(abs(FFT_Xn));
34     title(['FFT_{Xn}(r=' num2str(r) ')']);
35
36     subplot(5,4,4*k);
37     plot(abs(FFT_Yn));
38     title(['FFT_{Yn}(r=' num2str(r) ')']);
39
40     end
```

lab5\_2.m

```
1 %双线性变换法设计 chebyshev 高通滤波器
2 wp=0.3*pi;
3 FS=1;
4 ws=0.2*pi;
5 OmegaP=2*tan(wp/2);
6 OmegaS=2*tan(ws/2);
7 [n,Wn]=cheb1ord(OmegaP,OmegaS,1,15,'s');
8 [b,a]=cheby1(n,1,Wn,'high','s');
```

```
9 %freqs(b,a) %设计模拟的
10 [bz,az]=bilinear(b,a,FS); %映射为数字的
11 [H,w]=freqz(bz,az);
12 plot(w/pi, 20*log10(abs(H)));
13 xlabel('\omega/\pi');
14 ylabel('幅值(dB)');
```

lab5\_3.m

```
1 %双线性变换法设计 chebyshev 高通滤波器
2 FS=1; T=1/FS;
3 wp=0.3*pi;
4 ws=0.2*pi;
5 OmegaP=2*tan(wp/2);
6 OmegaS=2*tan(ws/2);
7 [n1,Wn1]=cheb1ord(OmegaP,OmegaS,1,15,'s');
8 [b1,a1]=cheby1(n1,1,Wn1,'high','s');
9 %freqs(b,a) %设计模拟的
10 [bz,az]=bilinear(b,a,FS); %映射为数字的
11
12 % Matlab 使用归一化的频率参数（临界频率除以采样频率的 1/2）。
13 %这样临界频率参数的取值范围在 0 和 1 之间，1 代表 Fs/2（用角频率表示的时候对应）
14 R=[5,10,15,20,40];
15 r=R(5);
16 T_w=0.1*pi; %过渡带宽
17 N=100;
18 W_p=2*pi*r/N;
19 W_s=W_p+T_w;
20 f_s=1;
21 [n,W_n]=buttord(W_p/pi,W_s/pi,1,40);
22 [b,a]=butter(n,W_n);
23
24 n=1:N*10;
25 x_n=[ones(1,N/2),zeros(1,N/2)];
26 X_n= repmat(x_n,1,10); %复制10次
27 Y_n=filter(b,a,X_n);
28 Y_pn=filter(bz,az,X_n);
29 FFT_Xn=fft(X_n);
```

```
30 FFT_Yn=fft(Y_pn);
31 FFT_Ypn=fft(Y_pn+Y_n);
32
33 subplot(2,3,1);
34 plot(1*N:5*N,X_n(1*N:5*N));
35 title(['X_n(r=' num2str(r) ')']);
36
37 subplot(2,3,2);
38 plot(1*N:5*N,Y_pn(1*N:5*N));
39 title(['Y_pn(r=' num2str(r) ')']);
40
41 subplot(2,3,3);
42 plot(1*N:5*N,Y_n(1*N:5*N)+Y_pn(1*N:5*N));
43 title(['Y_n+Y_{pn}(r=' num2str(r) ')']);
44
45 subplot(2,3,4);
46 plot(abs(FFT_Xn));
47 title(['FFT_{Xn}(r=' num2str(r) ')']);
48
49 subplot(2,3,5);
50 plot(abs(FFT_Yn));
51 title(['FFT_{Y_{pn}}(r=' num2str(r) ')']);
52
53 subplot(2,3,6);
54 plot(abs(FFT_Ypn));
55 title(['FFT_{Y_n+Y_{pn}}(r=' num2str(r) ')']);
```

lab5\_4.m

## 实验 6 窗函数

### 6.1 实验目的

1. 理解窗函数的基本性能，掌握主瓣宽度与旁瓣电平的控制原则。
2. 探究两类相位特性对信号处理的影响

### 6.2 实验内容分析

#### 6.2.1 实验内容 1

设  $x_1(n) = 31.6e^{j\frac{3\pi}{4}n} + 0.005e^{j\frac{4\pi}{5}n}$ ,  $0 \leq n \leq 1023$ 。

分别使用矩形窗、Hamming 窗对  $x_1(n)$  做 DFT 得到  $X_1(k)$ ，画出幅度谱，用分贝 (dB) 表示，并比较分析。

绘制出结果如下：

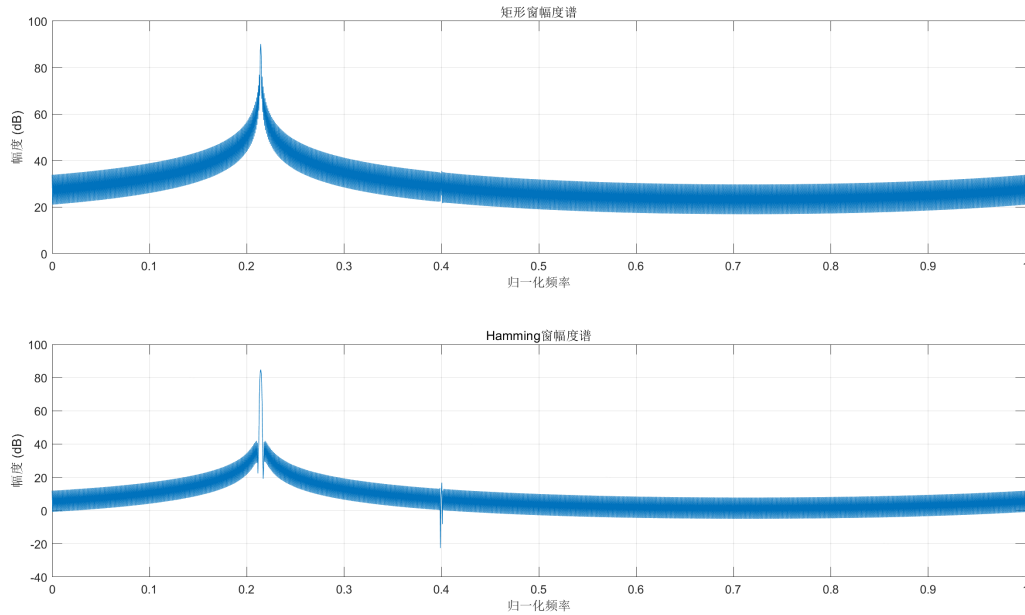


图 12:  $x_1(n)$  DFT 幅度谱

观察到矩形窗的低频旁瓣幅度超过高频分量，高频分量被淹没在低频旁瓣中。而 Hamming 窗的低频旁瓣幅度低于高频分量，可以辨认出高频分量。

矩形窗的旁瓣衰减较慢，导致低频旁瓣的幅度较高，高频分量容易被低频旁瓣淹没。因此，矩形窗适合对频谱分辨率要求较高的场景，但在处理混叠信号时效果较差。而 Hamming 窗通过对时域信号的加权处理，旁瓣衰减更快，低频旁瓣幅度显著降低（约  $-40\text{dB}$ ），从而使高频分量不容易被淹没。Hamming 窗牺牲了一定的频谱分辨率（主瓣宽度增加）来换取更低的旁瓣电平。

### 6.2.2 实验内容 2

设  $x_2(n) = 31.6e^{j\frac{3\pi}{7}n} + 10e^{j(\frac{1}{7} + \frac{1}{1024})3\pi n}$ ,  $0 \leq n \leq N-1$ 。

使用 Blackman 窗，分别在  $N=1024$  和  $N=2048$  两种情况下，对  $x_2(n)$  做 DFT 得到  $X_2(k)$ ，画出幅度谱，并比较分析。

绘制出结果如下：

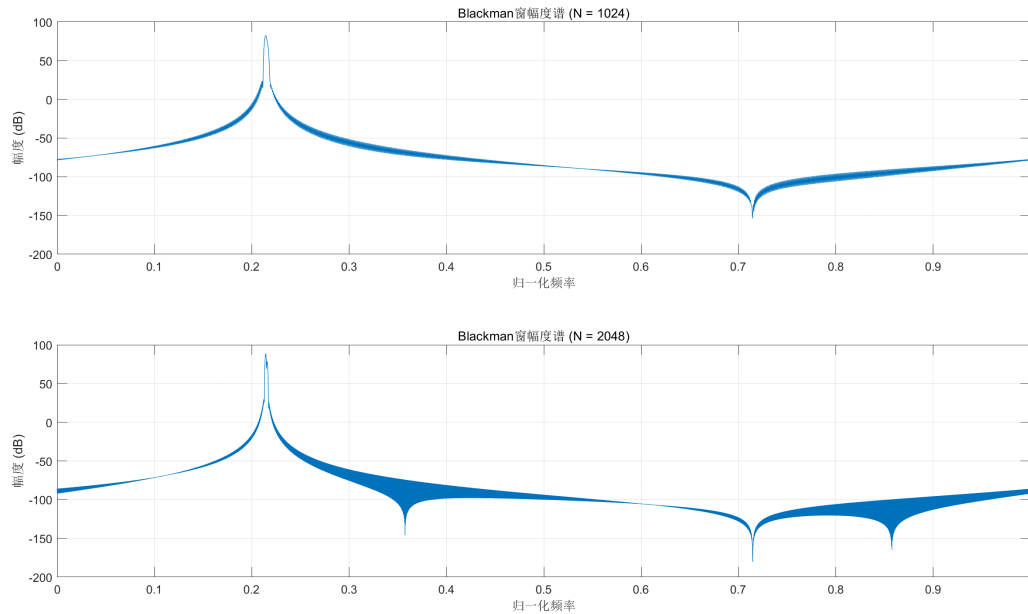


图 13:  $x_2(n)$  DFT 幅度谱

观察得到， $N=1024$  时，高频分量淹没在低频分量的主瓣中，只能看到单峰。 $N=2048$  时，低频分量的主瓣变窄，高频分量显现，可以观察到双峰。

Blackman 窗的旁瓣幅度进一步降低（约  $-60\text{dB}$ ），其主瓣宽度较矩形窗和 Hamming 窗更宽。对于包含相近频率分量的信号，Blackman 窗可能使分量的主瓣重叠，从而影响分辨率。

当窗长度（ $N$ ）增大时，主瓣宽度减小，频谱分辨率提高，高频分量逐渐显现。这说明，通过增加采样点数可以部分弥补窗函数带来的分辨率下降。

### 6.3 实验总结

- (1) 相位特性对信号的频谱表现有着决定性的影响，尤其在频率成分较为接近时，相位差异可能导致频谱重叠，影响信号的精确分析。
- (2) 窗函数的选择与相位特性密切相关，适当的窗函数可以减小因相位干扰造成的误差，尤其是在频率分量接近时。选择合适的窗函数（如矩形窗、Hamming 窗或 Blackman 窗）有助于提高信号的频谱分辨率。
- (3) 增加采样点数可以有效提高频谱分辨率，帮助区分相邻频率成分并改善相位的恢复，尤其是在信号



频率接近的情况下尤为重要。

## 6.4 源代码

源代码见附件 lab6\_1.m, lab6\_2.m。

```
1  N = 1024;
2  n = 0:N-1;
3  x1 = 31.6*exp(1j*3*pi/7*n) + 0.005*exp(1j*4*pi/5*n); % 信号
4
5  % 矩形窗
6  rect_win = boxcar(N);
7  x1_rect = x1 .* rect_win';
8  X1_rect = fft(x1_rect, 2*N);
9  X1_rect_mag = 20*log10(abs(X1_rect)); % 幅度谱 (dB)
10
11 % Hamming 窗
12 hamming_win = hamming(N);
13 x1_ham = x1 .* hamming_win';
14 X1_ham = fft(x1_ham, 2*N);
15 X1_ham_mag = 20*log10(abs(X1_ham)); % 幅度谱 (dB)
16
17 freq = (0:2*N-1)/(2*N); % 归一化频率
18
19 figure;
20 subplot(2, 1, 1);
21 plot(freq, X1_rect_mag);
22 title('矩形窗幅度谱');
23 xlabel('归一化频率');
24 ylabel('幅度 (dB)');
25 grid on;
26
27 subplot(2, 1, 2);
28 plot(freq, X1_ham_mag);
29 title('Hamming窗幅度谱');
30 xlabel('归一化频率');
31 ylabel('幅度 (dB)');
32 grid on;
```

lab6\_1.m

```
1  N1 = 1024; N2 = 2048;
2  n1 = 0:N1-1; n2 = 0:N2-1;
3  x2_N1 = 31.6*exp(1j*3*pi/7*n1) + 10*exp(1j*(1/7+1/1024)*3*pi*n1); %
    信号 N = 1024
4  x2_N2 = 31.6*exp(1j*3*pi/7*n2) + 10*exp(1j*(1/7+1/1024)*3*pi*n2); %
    信号 N = 2048
5
6  % Blackman窗
7  blackman_N1 = blackman(N1);
8  blackman_N2 = blackman(N2);
9
10 x2_N1_win = x2_N1 .* blackman_N1';
11 x2_N2_win = x2_N2 .* blackman_N2';
12
13 % FFT, 零填充到 2N 点
14 X2_N1 = fft(x2_N1_win, 2*N1);
15 X2_N2 = fft(x2_N2_win, 2*N2);
16
17 % 幅度谱 (dB)
18 X2_N1_mag = 20*log10(abs(X2_N1));
19 X2_N2_mag = 20*log10(abs(X2_N2));
20
21 freq_N1 = (0:2*N1-1)/(2*N1);
22 freq_N2 = (0:2*N2-1)/(2*N2);
23
24 figure;
25 subplot(2, 1, 1);
26 plot(freq_N1, X2_N1_mag);
27 title('Blackman窗幅度谱 (N = 1024)');
28 xlabel('归一化频率');
29 ylabel('幅度 (dB)');
30 grid on;
31
32 subplot(2, 1, 2);
33 plot(freq_N2, X2_N2_mag);
34 title('Blackman窗幅度谱 (N = 2048)');
35 xlabel('归一化频率');
36 ylabel('幅度 (dB)');
```

37 | `grid on;`

lab6\_2.m

## 实验 7 FIR 滤波器的窗函数设计方法

### 7.1 实验目的

1. 探究窗函数主瓣宽度和旁瓣电平对滤波器性能的影响。
2. 观察理解 Gibbs 效应

### 7.2 实验内容分析

滤波器性能要求：中心频率为  $f_0 = 150\text{MHz}$ ，通带宽度为  $B = 100\text{MHz}$ ，上下阻带最小衰减大于 20dB，上下过渡带宽小于 10MHz，采样频率  $f_s = 500\text{MHz}$ ，要求具有线性相位特性。

采用矩形窗、三角形窗、汉宁窗、汉明窗，布莱克曼窗设计。

改变系统阶数  $N$ （增加和减少），画出不同  $N$  下的幅频特性曲线，并分析讨论。

#### 7.2.1 矩形窗函数设计带通滤波器

$$\Delta\omega = \frac{10}{500} \times 2\pi = 0.04\pi$$

矩形窗过渡带宽  $\Delta\omega = 1.8\pi/N$ 。代入得  $N \geq 45$  时符合性能要求。

考虑到  $N$  为奇数，取  $N=11,31,45,101$  绘图如下：

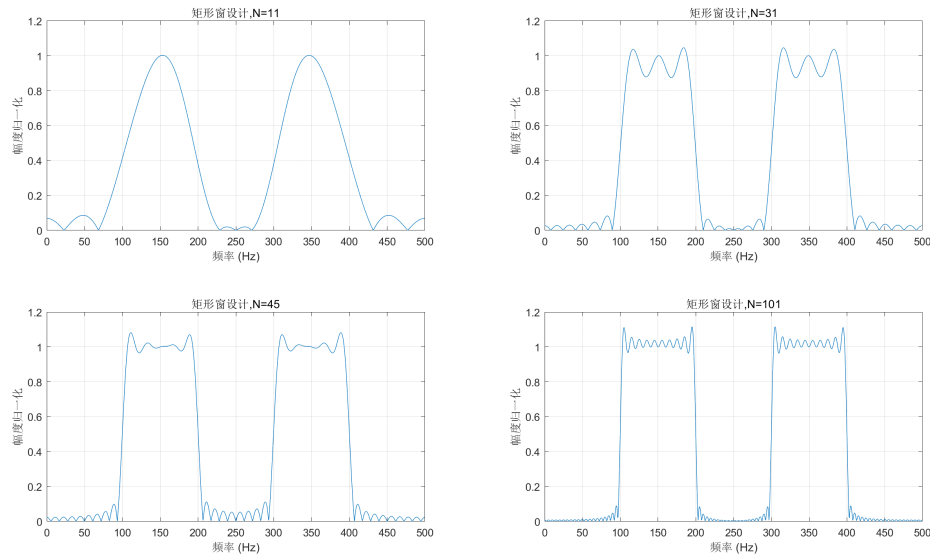


图 14: 矩形窗函数设计带通滤波器幅频特性曲线

#### 7.2.2 三角形窗函数设计带通滤波器

三角形窗过渡带宽  $\Delta\omega = 6.1\pi/N$ 。代入得  $N \geq 152.5$  时符合性能要求。

考虑到  $N$  为奇数，取  $N=51,101,153,251$  绘图如下：

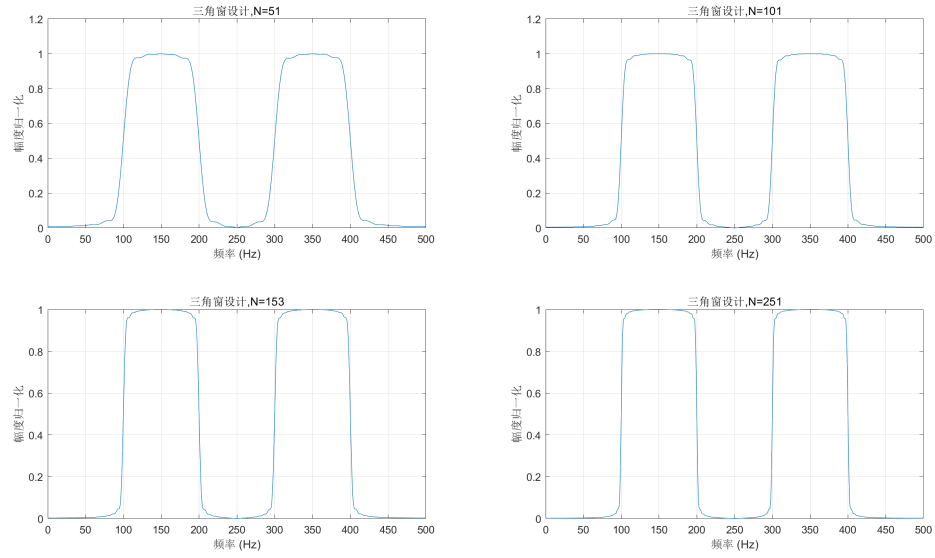


图 15: 矩形窗函数设计带通滤波器幅频特性曲线

### 7.2.3 汉宁窗函数设计带通滤波器

矩形窗过渡带宽  $\Delta\omega = 6.2\pi/N$ 。代入得  $N \geq 155$  时符合性能要求。

考虑到  $N$  为奇数，取  $N=51, 101, 155, 251$  绘图如下：

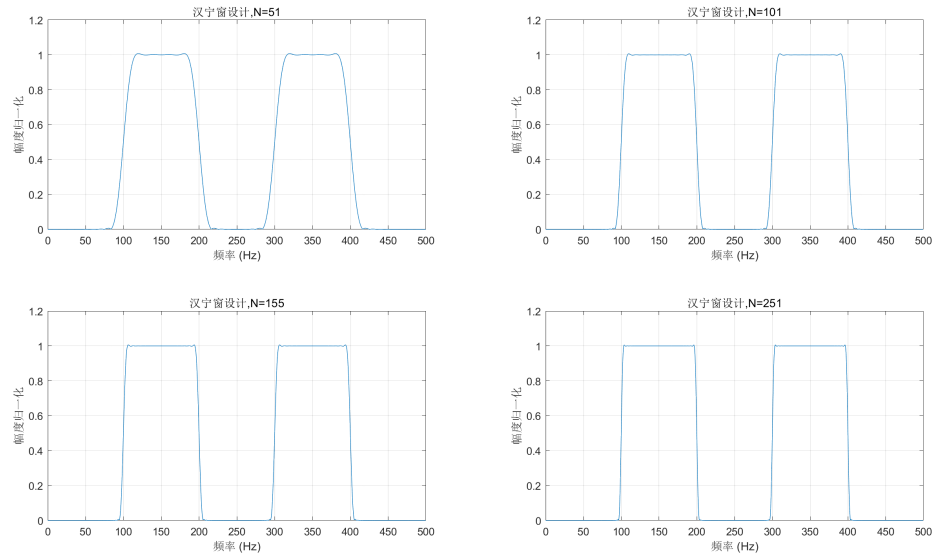


图 16: 矩形窗函数设计带通滤波器幅频特性曲线

### 7.2.4 汉明窗函数设计带通滤波器

矩形窗过渡带宽  $\Delta\omega = 6.6\pi/N$ 。代入得  $N \geq 165$  时符合性能要求。

考虑到  $N$  为奇数，取  $N=51,101,165,251$  绘图如下：

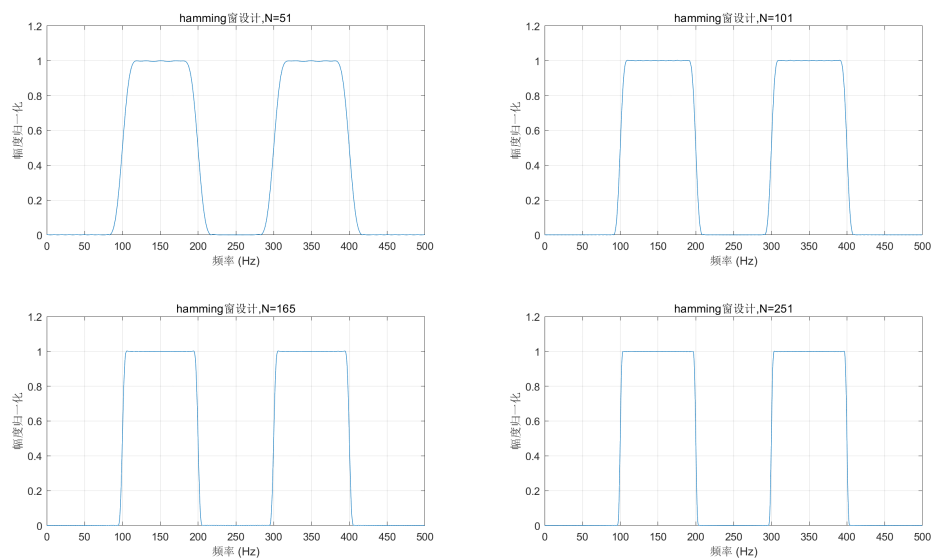


图 17: 矩形窗函数设计带通滤波器幅频特性曲线

### 7.2.5 布莱克曼窗函数设计带通滤波器

矩形窗过渡带宽  $\Delta\omega = 11\pi/N$ 。代入得  $N \geq 275$  时符合性能要求。

考虑到  $N$  为奇数，取  $N=101,201,275,401$  绘图如下：

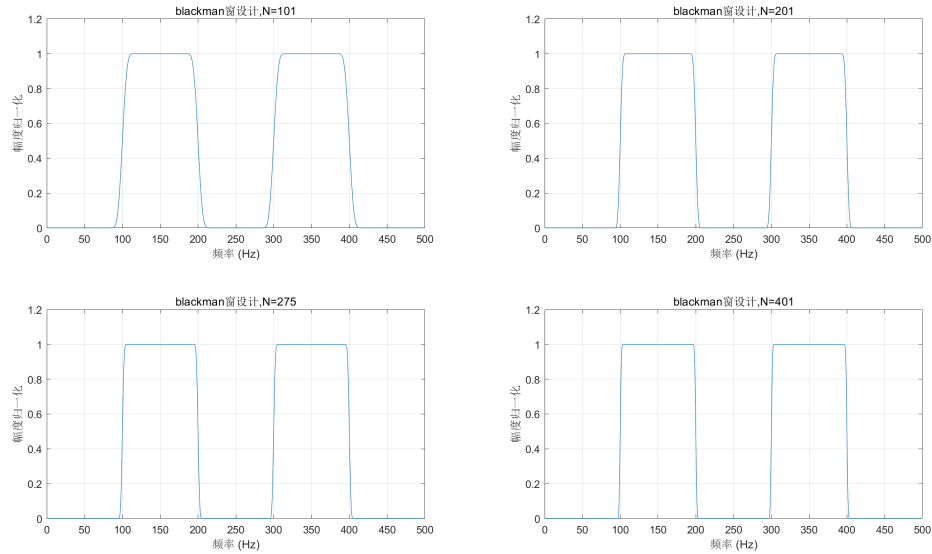


图 18: 矩形窗函数设计带通滤波器幅频特性曲线

## 7.3 实验总结

### 7.3.1 窗函数对滤波器性能的影响

窗函数的选择影响滤波器的频率响应，包括过渡带宽、通带衰减和阻带衰减等。

矩形窗的主瓣较宽，旁瓣较高，因此过渡带宽较大，导致频率响应的过渡带较宽。矩形窗能够提供较高的频谱分辨率，需要的阶数  $N$  较小。但由于其较大的旁瓣，可能在滤波器的阻带引入较大的泄漏，影响滤波器的性能。

三角形窗的旁瓣衰减较快，过渡带宽度较矩形窗小。在滤波器设计中，三角形窗通常提供较好的平衡，能有效地抑制旁瓣泄漏。但是，三角形窗的主瓣较宽，因此需要较大的阶数  $N$  才能满足过渡带宽的要求。

汉宁窗的旁瓣衰减较快，主瓣宽度比矩形窗和三角形窗更宽，因此需要更高的阶数  $N$  来满足过渡带宽要求。汉宁窗的设计提供了较好的旁瓣抑制效果，使得滤波器的频率响应更为平滑，但牺牲了部分频率分辨率。

汉明窗比汉宁窗的旁瓣衰减效果更好，提供了更为优异的阻带抑制。虽然主瓣宽度也比矩形窗和三角形窗更宽，但在实际应用中，对于要求高旁瓣衰减的场景，汉明窗表现出色。

布莱克曼窗提供最强的旁瓣抑制，但相应地，主瓣宽度也最宽，频谱分辨率相对较低，需要的阶数  $N$  最大。在需要极高阻带衰减的应用中，布莱克曼窗非常有效，尤其是在设计高性能滤波器时。

### 7.3.2 阶数对滤波器性能的影响

阶数  $N$  增大：过渡带宽逐渐减小，通带和阻带的性能逐步改善。然而，阶数过大可能导致滤波器的实现复杂度增加，因此在实际设计中，需要在性能和实现复杂度之间找到平衡。

阶数  $N$  减小：过渡带宽变宽，频谱的分辨率下降，通带和阻带的衰减效果减弱。因此，为了满足滤

波器性能要求，需要合理选择滤波器的阶数。

### 7.3.3 Gibbs 效应

无论  $N$  取的多大，滤波器肩峰的相对值不会改变。对于矩形窗，它大约等于跳变值 8.95%。

窗的形状决定了滤波器的通带波纹及阻带衰减，为了加大阻带衰减，就要选择其他形状的窗函数。由于 8.95% 的肩峰是因矩形窗函数的突变造成，故采用形状缓变的窗函数，可以减小通带最大衰减，加大阻带最小衰减。

实验中，明显看到，改变  $N$  的大小，相同滤波器肩峰的相对值不会改变。不同滤波器尖峰大小不同。

## 7.4 源代码

矩形窗设计源代码见附件 lab7\_1.m

```
1  fs = 500;           % 采样频率
2  f_low = 100;        % 通带下边界
3  f_high = 200;       % 通带上边界
4  f0 = 150;           % 中心频率
5  delta_f = 10;       % 过渡带宽
6
7  f_low_norm = f_low / (fs / 2); % 归一化下边界
8  f_high_norm = f_high / (fs / 2); % 归一化上边界
9
10 figure;
11
12 N_total=[11,31,45,101];% 滤波器阶数 (0.04pi=1.8pi/45)
13 for i=1:4
14     N = N_total(i);
15     b = fir1(N, [f_low_norm, f_high_norm], boxcar(N+1)); % 矩形窗设计
16
17     [H, f] = freqz(b, 1, 1024, 'whole', fs); % 计算幅频响应
18
19     subplot(2,2,i);
20     plot(f, abs(H));
21     title(sprintf("矩形窗设计,N=%d", N));
22     xlabel('频率 (Hz)');
23     ylabel('幅度归一化');
24     grid on;
25 end
```

lab7\_1.m



三角形窗设计源代码见附件 lab7\_2.m

```
1
2  fs = 500;           % 采样频率
3  f_low = 100;        % 通带下边界
4  f_high = 200;       % 通带上边界
5  f0 = 150;          % 中心频率
6  delta_f = 10;      % 过渡带宽
7
8  f_low_norm = f_low / (fs / 2);      % 归一化下边界
9  f_high_norm = f_high / (fs / 2);    % 归一化上边界
10
11 figure;
12 N_total=[51,101,153,251]; % 滤波器阶数 (0.04pi=6.1pi/152.5)
13 for i=1:4
14     N = N_total(i);
15     b = fir1(N, [f_low_norm, f_high_norm], triang(N+1)); % 三角窗设计
16
17     [H, f] = freqz(b, 1, 1024, 'whole', fs); % 计算幅频响应
18
19     subplot(2,2,i);
20     plot(f, abs(H));
21     title(sprintf('三角窗设计, N=%d', N));
22     xlabel('频率 (Hz)');
23     ylabel('幅度归一化');
24     grid on;
25 end
```

lab7\_2.m

汉宁窗设计源代码见附件 lab7\_3.m

```
1
2  fs = 500;           % 采样频率
3  f_low = 100;        % 通带下边界
4  f_high = 200;       % 通带上边界
5  f0 = 150;          % 中心频率
6  delta_f = 10;      % 过渡带宽
7
8  f_low_norm = f_low / (fs / 2);      % 归一化下边界
9  f_high_norm = f_high / (fs / 2);    % 归一化上边界
```

```
10
11 figure;
12
13 N_total=[51,101,155,251];           % 滤波器阶数 (0.04pi=6.2pi/155)
14 for i=1:4
15     N = N_total(i);
16     b = fir1(N, [f_low_norm, f_high_norm], hann(N+1)); % 汉宁窗设计
17
18     [H, f] = freqz(b, 1, 1024, 'whole', fs); % 计算幅频响应
19
20     subplot(2,2,i);
21     plot(f, abs(H));
22     title(sprintf(" 汉宁窗设计 ,N=%d", N));
23     xlabel('频率 (Hz)');
24     ylabel('幅度归一化');
25     grid on;
26     end
```

lab7\_3.m

汉明窗设计源代码见附件 lab7\_4.m

```
1
2 fs = 500;           % 采样频率
3 f_low = 100;        % 通带下边界
4 f_high = 200;       % 通带上边界
5 f0 = 150;           % 中心频率
6 delta_f = 10;       % 过渡带宽
7
8 f_low_norm = f_low / (fs / 2);           % 归一化下边界
9 f_high_norm = f_high / (fs / 2);         % 归一化上边界
10
11 figure;
12
13 N_total=[51,101,165,251];           % 滤波器阶数 (0.04pi=6.6pi
    /165)
14 for i=1:4
15     N = N_total(i);
16     b = fir1(N, [f_low_norm, f_high_norm], hamming(N+1)); % hamming窗设计
```

```
17
18
19 [H, f] = freqz(b, 1, 1024, 'whole', fs); % 计算幅频响应
20
21 subplot(2,2,i);
22 plot(f, abs(H));
23 title(sprintf("hamming窗设计,N=%d", N));
24 xlabel('频率 (Hz)');
25 ylabel('幅度归一化');
26 grid on;
27 end
```

lab7\_4.m

布莱克曼窗设计源代码见附件 lab7\_5.m

```
1
2 fs = 500;           % 采样频率
3 f_low = 100;        % 通带下边界
4 f_high = 200;       % 通带上边界
5 f0 = 150;           % 中心频率
6 delta_f = 10;       % 过渡带宽
7
8 f_low_norm = f_low / (fs / 2); % 归一化下边界
9 f_high_norm = f_high / (fs / 2); % 归一化上边界
10
11 figure;
12 N_total=[101,201,275,401]; % 滤波器阶数 (0.04pi=11pi
    /275)
13 for i=1:4
14     N = N_total(i);
15     b = fir1(N, [f_low_norm, f_high_norm], blackman(N+1)); % blackman窗
        设计
16
17 [H, f] = freqz(b, 1, 1024, 'whole', fs); % 计算幅频响应
18
19 subplot(2,2,i);
20 plot(f, abs(H));
21 title(sprintf("blackman窗设计,N=%d", N));
22 xlabel('频率 (Hz)');
```

```
23  ylabel('幅度归一化');  
24  grid on;  
25  end
```

lab7\_5.m

## 实验 8 线性相位系统

### 8.1 实验目的

1. 观察理解 IIR 滤波器的非线性相位特性与 FIR 滤波器的线性相位特性。
2. 探究两类相位特性对信号处理的影响

### 8.2 实验内容分析

#### 8.2.1 IIR 低通滤波器设计

设计 IIR 低通滤波器。通带截止频率  $0.8\pi$ ，通带起伏小于 1dB，过渡带宽小于  $0.1\pi$ ，止带衰减大于 40dB，幅度模型 Butterworth。

幅频特性曲线和相频特性曲线如下：

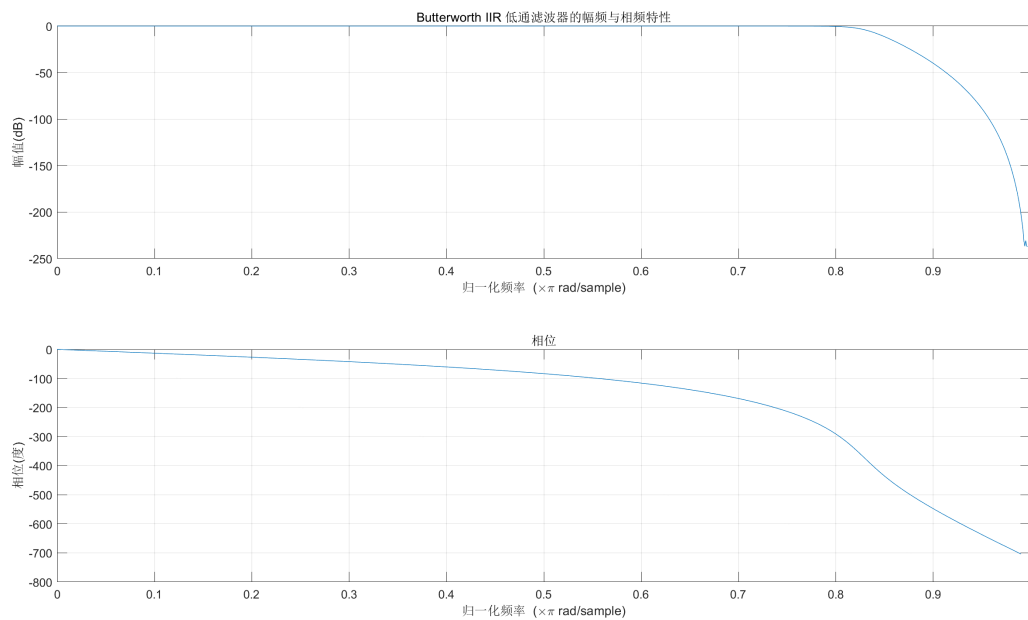


图 19: 幅频特性曲线和相频特性曲线

滤波器的群时延特性如下：

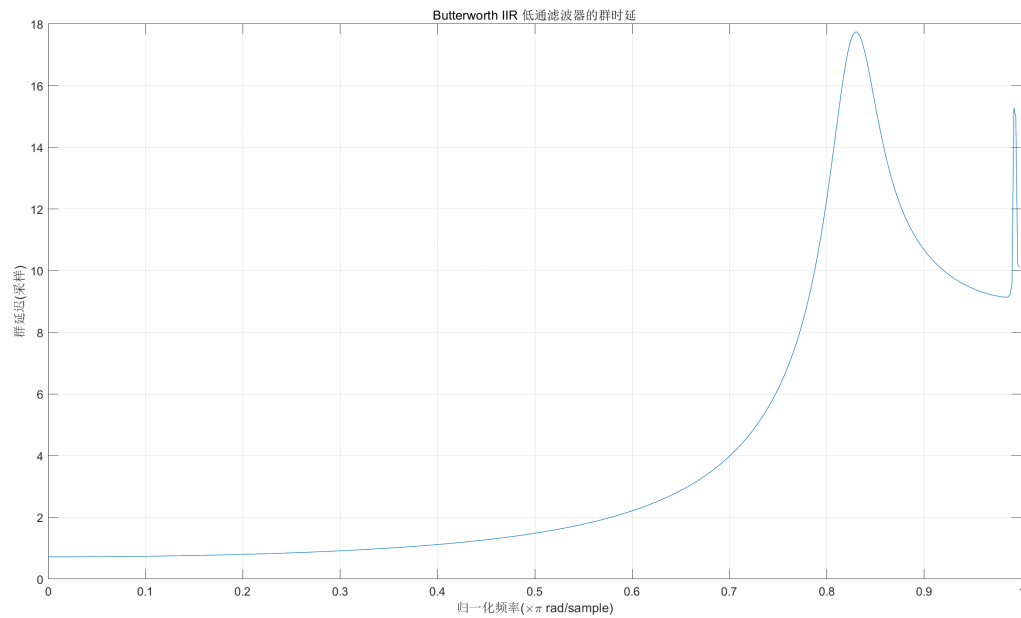


图 20: 滤波器的群时延特性

### 8.2.2 FIR 低通滤波器设计

设计 FIR 低通滤波器。通带截止频率  $0.8$ ，过渡带宽小于  $0.1\pi$ ，止带衰减大于  $40\text{dB}$ 。线性相位特性，窗函数法。默认为 Hamming 窗。

幅频特性曲线和相频特性曲线如下：

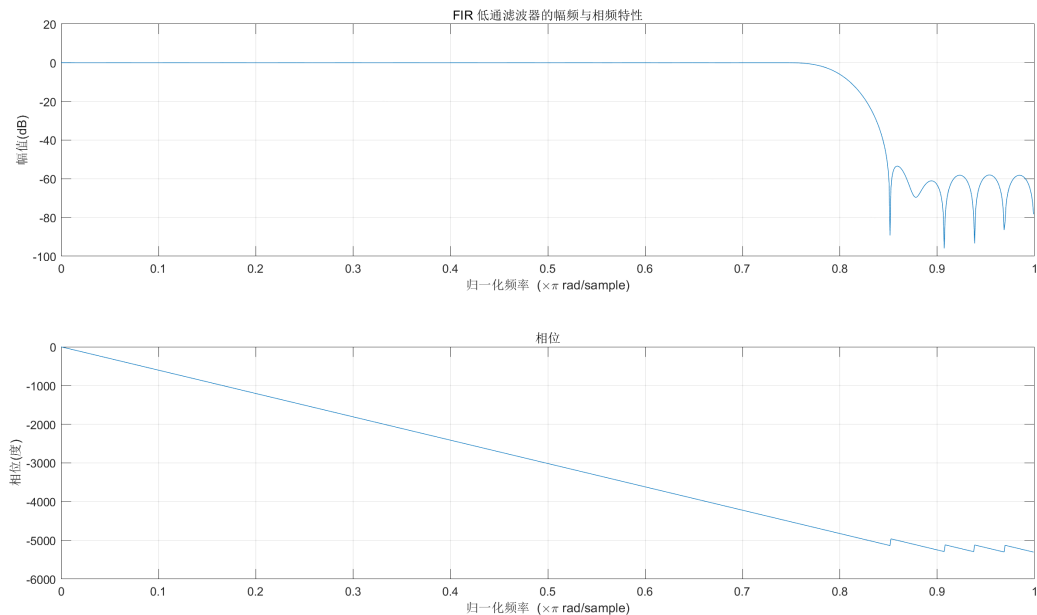


图 21: 幅频特性曲线和相频特性曲线

滤波器的群时延特性如下:

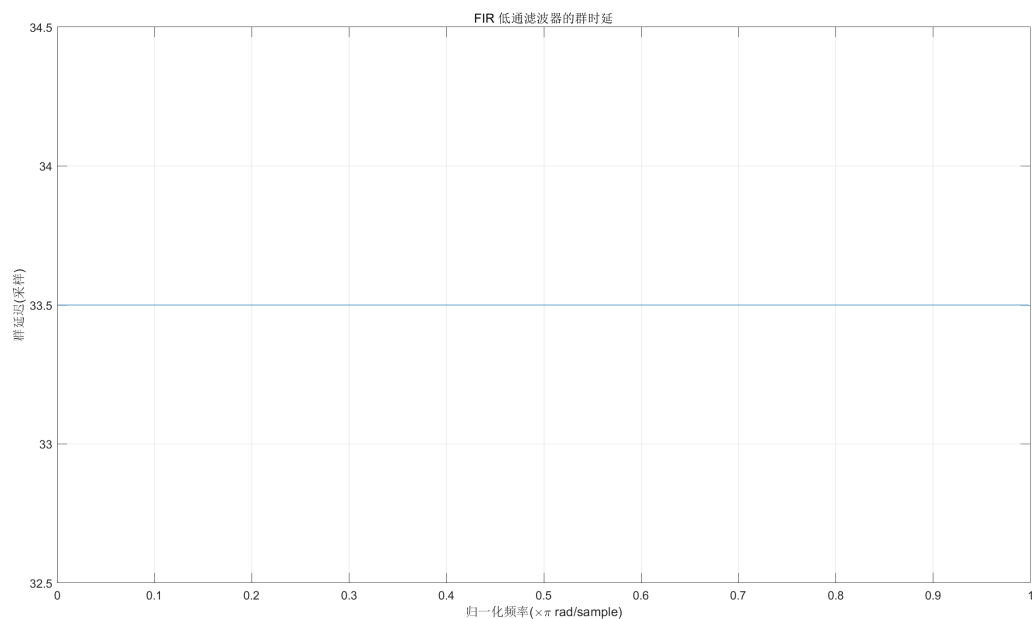


图 22: 滤波器的群时延特性

### 8.2.3 单一信号通过滤波器

$x_1(n) = \sin(\omega_1 n)$ ,  $x_2(n) = \sin(\omega_2 n)$ ,  $\omega_1 = 0.1$ ,  $\omega_2 = 0.7$ , 序列长度为  $N = 80$ , 分别输入 IIR 和 FIR 滤波器, 观察群延迟 (系统时延) 如下:

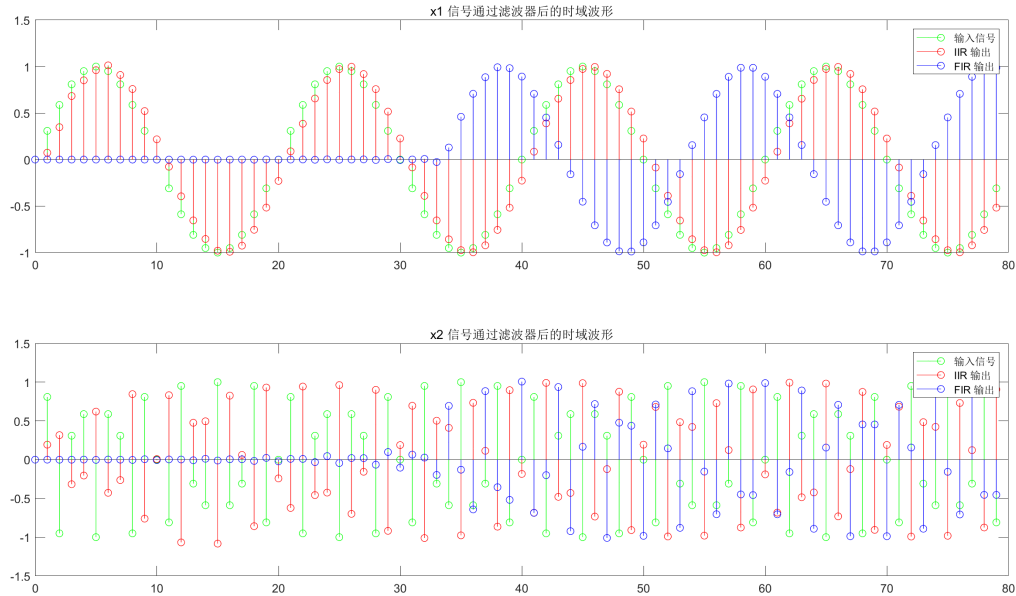


图 23: 信号通过滤波器波形

**IIR 滤波器的输出:** 由于 IIR 滤波器的非线性相位特性, 信号通过滤波器后会产生相位延迟, 频率较高的信号通过延迟较大, 频率较低的信号通过延迟较小。

**FIR 滤波器的输出:** 由于 FIR 滤波器的线性相位特性, 不同频率的信号通过滤波器的群时延相同。

### 8.2.4 混合信号通过滤波器

$x(n) = x_1(n) + x_2(n)$ , 分别输入 IIR 和 FIR 滤波器, 观察波形如下:



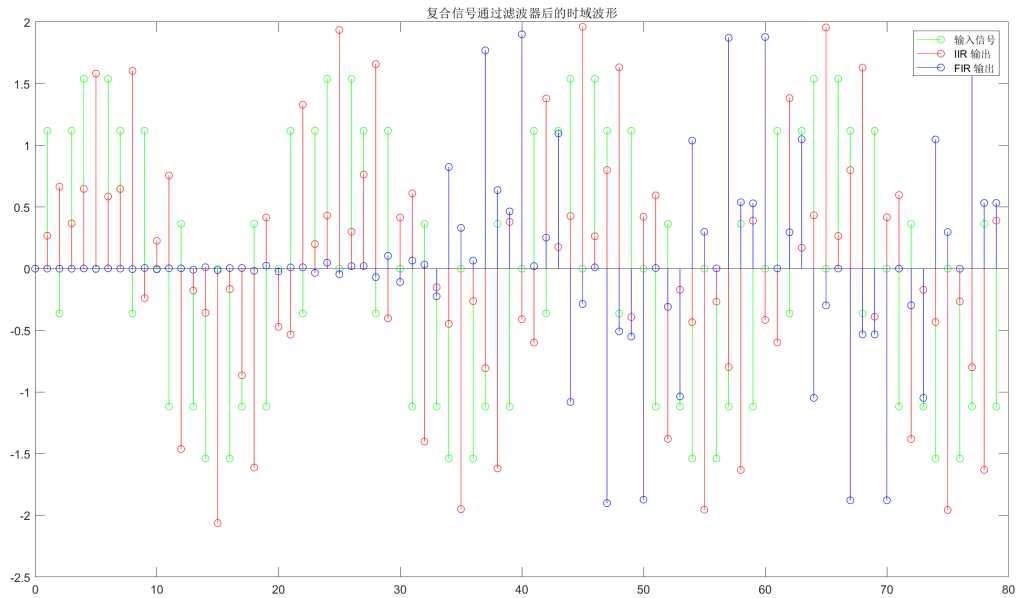


图 24: 信号通过滤波器波形

**IIR 滤波器的输出：**混合信号通过 IIR 滤波器后，信号的频率成分受到不同程度的相位失真，导致输出信号的形态变化较为明显。特别是两个频率成分之间的相位差异，使得 IIR 滤波器对信号的处理显得更加复杂，导致输出波形出现较大的失真。

**FIR 滤波器的输出：**FIR 滤波器在处理混合信号时能够保持信号的形态不变。由于其线性相位特性，FIR 滤波器有效地保留了信号的频率成分之间的相位关系，因此输出波形更加平滑，且几乎没有失真。

### 8.3 实验总结

IIR 滤波器通常具有非线性相位特性，FIR 滤波器通常具有线性相位特性。

IIR 滤波器具有较低的计算复杂度和较好的滤波性能，然而，由于其非线性相位特性，IIR 滤波器在信号通过时会引入相位失真，导致信号失真较为明显。特别是在处理高频信号时，IIR 滤波器的群时延特性会影响信号的精确度。

FIR 滤波器虽然计算复杂度较高，且其滤波性能通常需要较高的阶数才能满足设计要求，但是由于其线性相位特性，能够在信号处理中避免相位失真，特别适用于对信号形态要求较高的场合。

### 8.4 源代码

源代码见附件 lab8.m

1  
2  
3

```
% (1) Butterworth 模型设计 IIR 低通滤波器
```

```
4      fp = 0.8;                % 通带截止频率
5      fs = 0.9;                % 止带起始频率
6      Ap = 1;                  % 通带起伏
7      As = 40;                 % 止带衰减
8
9      [N, Wn] = buttord(fp, fs, Ap, As);
10     [IIR_b, IIR_a] = butter(N, Wn, 'low');    % 设计低通滤波器
11
12
13     figure(1);
14     freqz(IIR_b, IIR_a, 1024);                % 绘制幅频和相频特性
15     title('Butterworth IIR 低通滤波器的幅频与相频特性');
16
17     figure(2);
18     grpdelay(IIR_b, IIR_a);                    % 绘制群时延特性
19     title('Butterworth IIR 低通滤波器的群时延');
20
21     % (2) FIR 滤波器设计低通滤波器
22
23     fp = 0.8;                % 通带截止频率
24     delta_f = 0.1;           % 过渡带宽
25     fst = 0.9;                % 止带起始频率
26     As = 40;                 % 止带衰减
27
28     N=67;                    % (0.1pi=6.6pi/66)
29     FIR_b = fir1(N, fp, 'low');    % 默认 Hamming 窗
30
31     figure(3);
32     freqz(FIR_b, 1, 1024);        % 绘制幅频和相频特性
33     title('FIR 低通滤波器的幅频与相频特性');
34
35     figure(4);
36     grpdelay(FIR_b, 1);            % 绘制群时延特性
37     title('FIR 低通滤波器的群时延');
38
39     % (3) 输入信号 x1(n) 和 x2(n)
40
41     N = 80;
```

```
42     n = 0:N-1;
43     x1 = sin(0.1 * pi * n);
44     x2 = sin(0.7 * pi * n);
45
46     % 信号通过 IIR 滤波器
47     y1_iir = filter(IIR_b, IIR_a, x1);
48     y2_iir = filter(IIR_b, IIR_a, x2);
49
50     % 信号通过 FIR 滤波器
51     y1_fir = filter(FIR_b, 1, x1);
52     y2_fir = filter(FIR_b, 1, x2);
53
54     % 绘制时域波形
55     figure(5);
56     subplot(2, 1, 1);
57     stem(n, x1, 'g', 'DisplayName', '输入信号');
58     hold on;
59     stem(n, y1_iir, 'r', 'DisplayName', 'IIR 输出');
60     stem(n, y1_fir, 'b', 'DisplayName', 'FIR 输出');
61     legend;
62     title('x1 信号通过滤波器后的时域波形');
63
64     subplot(2, 1, 2);
65     stem(n, x2, 'g', 'DisplayName', '输入信号');
66     hold on;
67     stem(n, y2_iir, 'r', 'DisplayName', 'IIR 输出');
68     stem(n, y2_fir, 'b', 'DisplayName', 'FIR 输出');
69     legend;
70     title('x2 信号通过滤波器后的时域波形');
71
72     % (4) 复合信号  $x(n) = x1(n) + x2(n)$ 
73     x = x1 + x2;
74
75     % 信号通过 IIR 滤波器
76     y_iir = filter(IIR_b, IIR_a, x);
77
78     % 信号通过 FIR 滤波器
79     y_fir = filter(FIR_b, 1, x);
80
```

```
81 % 绘制时域波形
82 figure(6);
83
84 stem(n, x, 'g', 'DisplayName', '输入信号');
85 hold on;
86 stem(n, y_iir, 'r', 'DisplayName', 'IIR 输出');
87 stem(n, y_fir, 'b', 'DisplayName', 'FIR 输出');
88 legend;
89 title('复合信号通过滤波器后的时域波形');
```

lab8.m