



西安电子科技大学
XIDIAN UNIVERSITY

人工智能学院

数字信号处理课程设计报告

数字信号处理设计与仿真分析

杨文韬

18020100245

刘浩

19069100088

周泽熙

19069100126

2022 年 1 月 11 日

目录

1 问题描述	1
2 实验环境	1
3 基础知识回顾	2
3.1 参数概念	2
3.2 奈奎斯特采样定理	2
3.3 数字滤波器	2
3.4 信号流图	2
4 方案设计	3
5 问题求解	3
5.1 问题 1	3
5.2 问题 2	5
5.3 问题 3	5
5.4 问题 4	8
5.5 问题 5	8
6 总结	9
Appendices	10
附录 A Matlab 代码	10
附录 B Python 代码	13

数字信号处理设计与仿真分析

1 问题描述

- (1) 建立两个模拟信号的数学模型 $sa_1(t)$ 和 $sa_2(t)$ ，其中 $sa_1(t)$ 是有用信号， $sa_2(t)$ 是干扰信号。两个信号的中心频率、信号带宽等参数由学生自己选定，要求两个信号的频谱不重叠， $sa_2(t)$ 的幅度比 $sa_1(t)$ 的幅度高 20dB，两个信号时域叠加得到合成信号 $xa(t)$ ，即

$$xa(t) = sa_1(t) + sa_2(t)$$

设计计算机程序仿真产生 $sa_1(t)$ 、 $sa_2(t)$ 、 $xa(t)$ 信号，分别画出三个模拟信号的时域波形和频谱图；

- (2) 根据 $xa(t)$ 的中心频率和带宽，按照奈奎斯特采样定理选择采样频率 f_s ，分别对信号 $sa_1(t)$ 、 $sa_2(t)$ 、 $xa(t)$ 进行时域采样，得到离散信号 $s_1(n)$ 、 $s_2(n)$ 、 $x(n)$ 。利用 FFT 算法分析离散信号的频谱，分别画出三个离散信号的时域波形和频谱图；
- (3) 设计数字滤波器 $H(z)$ ，要求该滤波器对干扰信号 $s_2(n)$ 的衰减大于 40dB。提出滤波器的设计指标，并设计滤波器，给出滤波器的设计结果，绘制滤波器的幅频特性和相频特性曲线，验证滤波器的设计结果是否达到设计指标要求；
- (4) 选择实现数字滤波器 $H(z)$ 的结构，画出结构信号流图；
- (5) 将合成信号 $x(n)$ 输入数字滤波器 $H(z)$ ，按照所选择的滤波器结构，设计计算机程序计算滤波器的输出响应 $y(n)$ ，画出 $y(n)$ 的时域波形和频谱图；
- (6) 分析、总结设计结果，提交课程设计报告。

2 实验环境

- Matlab: 编写代码并编译运行。
- JupyterNotebook: 编写 Python 代码并生成 notebook 的 pdf 文件添加在附录中。
- L^AT_EX: 进行文档编写和排版。
- Visio: 绘制流程图和信号流图。

3 基础知识回顾

3.1 参数概念

1. 中心频率 (Center frequency): 周期信号对应的频率, 可以通过傅里叶级数转换为不同频率弦波的和。
2. 信号带宽 (Bandwidth): 信号所占据的频带宽度; 在被用来描述信道时, 带宽是指能够有效通过该信道的信号的最大频带宽度。对于模拟信号而言, 带宽又称为频宽, 以赫兹 (Hz) 为单位。对于数字信号而言, 带宽是指单位时间内链路能够通过的数据量。

3.2 奈奎斯特采样定理

一个有带限的连续时间信号可以被采样并从它的样本中完美恢复, 如果波形的采样频率是它的最高频率分量的两倍以上。(A bandlimited continuous-time signal can be sampled and perfectly reconstructed from its samples if the waveform is sampled over twice as fast as it's highest frequency component)

3.3 数字滤波器

数字滤波器是对数字信号进行滤波处理以得到期望的响应特性的离散时间系统。作为一种电子滤波器, 数字滤波器与完全工作在模拟信号域的模拟滤波器不同。数字滤波器工作在数字信号域, 它处理的对象是经由采样器件将模拟信号转换而得到的数字信号。线性非时变的数字滤波器包括无限长脉冲响应滤波器 (IIR 滤波器) 和有限长脉冲响应滤波器 (FIR 滤波器) 两种。这两种滤波器的系统函数可以统一以 Z 变换表示为:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}} \quad (1)$$

当 $M \geq 1$ 时, M 就是 IIR 滤波器的阶数, 表示系统中反馈环的个数。由于反馈的存在, IIR 滤波器的脉冲响应为无限长, 因此得名。若 $A(z) = 1$, 则系统的脉冲响应的长度为 $N + 1$, 故而被称作 FIR 滤波器。

3.4 信号流图

信号流图是一种表示一组联立线性代数方程的图。表明了系统中各信号的关系, 包含了结构图所包含的全部信息, 与结构图一一对应。

4 方案设计

按如图 1 所示的实验流程图进行问题求解。用 Matlab 和 Python 两种语言编写代码实现，正文中的实验图为 Matlab 运行结果，Python 运行结果与 Matlab 相似，为避免赘述放在附录以供检验对比。

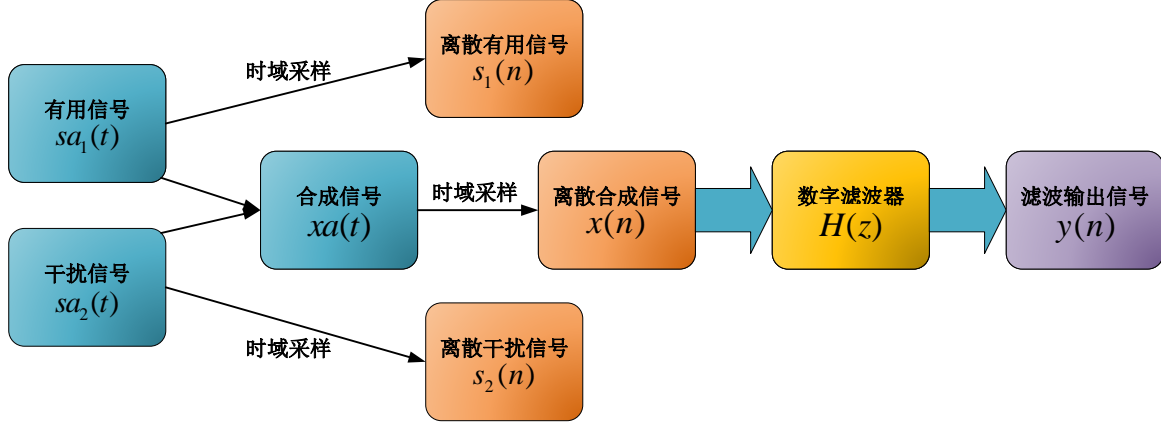


图 1: 实验流程图

5 问题求解

5.1 问题 1

假设有用信号为 $sa_1(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$ ，其中 $f_1 = 10\text{Hz}$, $f_2 = 6\text{Hz}$ ，则周期 $T_1 = 0.5\text{s}$ 。作傅里叶变换得

$$F_1(j2\pi f) = \pi [\delta(2\pi(f + f_1)) + \delta(2\pi(f - f_1)) + \delta(2\pi(f + f_2)) + \delta(2\pi(f - f_2))] \quad (2)$$

$$= \frac{1}{2} [\delta(f + 10) + \delta(f - 10) + \delta(f + 6) + \delta(f - 6)] \quad (3)$$

有用信号 $sa_1(t)$ 时域波形和频谱图如图 2 所示。

假设干扰信号 $sa_2(t)$ 的幅度是 $sa_1(t)$ 的 x 倍， $sa_1(t)$ 幅度比 $sa_2(t)$ 高 20dB，则 $20 \lg x = 20$ ，解得 $x = 10$ 。可假设干扰信号为 $sa_2(t) = 10 \cos(2\pi f_3 t) + 10 \cos(2\pi f_4 t)$ ，其中 $f_3 = 36\text{Hz}$, $f_4 = 25\text{Hz}$ ，则周期 $T_2 = 1\text{s}$ 。作傅里叶变换得

$$F_2(j2\pi f) = 10\pi [\delta(2\pi(f + f_3)) + \delta(2\pi(f - f_3)) + \delta(2\pi(f + f_4)) + \delta(2\pi(f - f_4))] \quad (4)$$

$$= 5 [\delta(f + 36) + \delta(f - 36) + \delta(f + 25) + \delta(f - 25)] \quad (5)$$

干扰信号 $sa_2(t)$ 时域波形和频谱图如图 3 所示。

两个信号时域叠加得到合成信号

$$xa(t) = sa_1(t) + sa_2(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + 10 \cos(2\pi f_3 t) + 10 \cos(2\pi f_4 t) \quad (6)$$

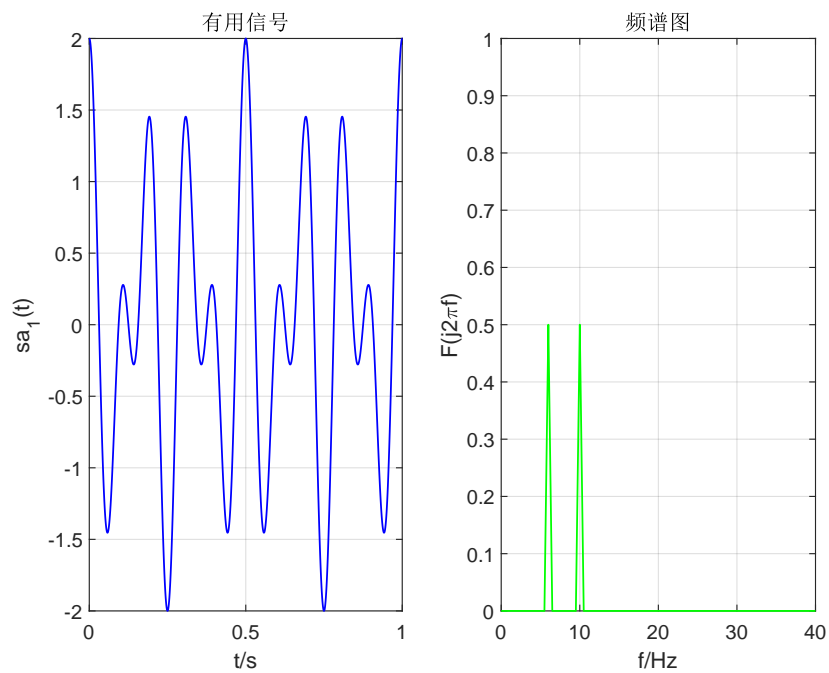


图 2: 有用信号 $sa_1(t)$ 时域波形和频谱图

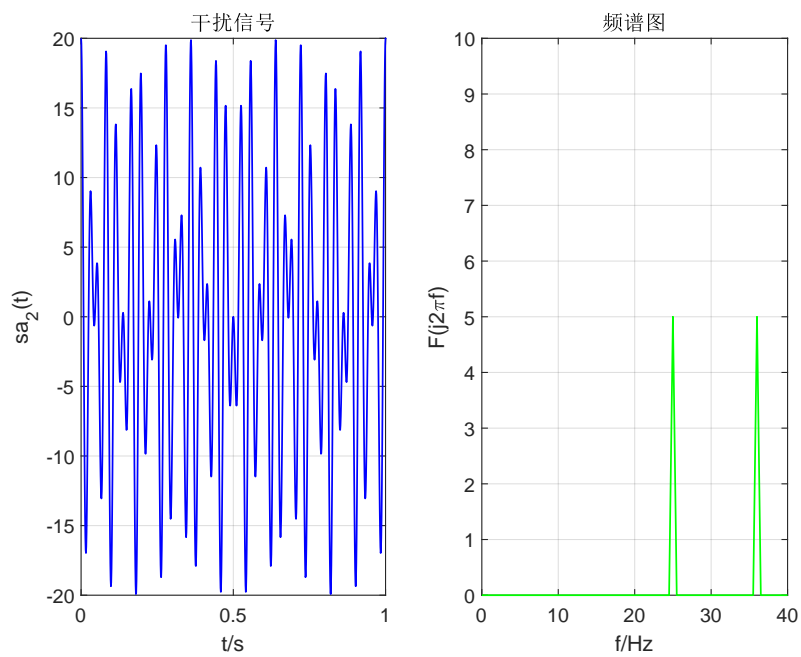


图 3: 干扰信号 $sa_2(t)$ 时域波形和频谱图

其周期 $T_3 = 1s$ 。合成信号 $xa(t)$ 时域波形和频谱图如图 4 所示。

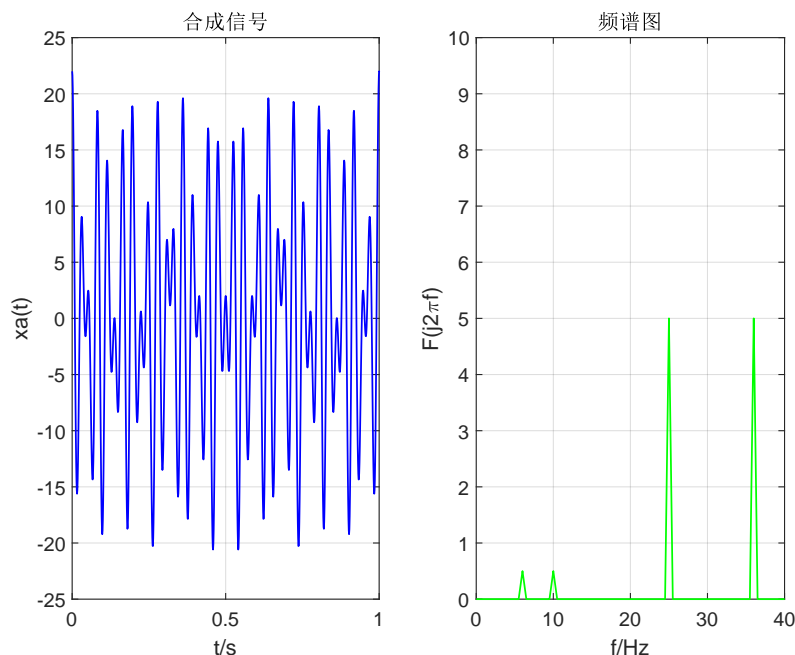


图 4: 合成信号 $xa(t)$ 时域波形和频谱图

5.2 问题 2

$x_a(t)$ 的最高频率 $f_c = 36Hz$ ，那么奈奎斯特采样频率 $f_s = 2f_c = 72Hz$ ，选择采样频率 $100Hz$ 进行采样。

对信号 $sa_1(t)$ 进行采样，并作 200 点 FFT，得到离散有用信号 $s_1(n)$ 的时域波形和频谱图如图 5 所示。

对信号 $sa_2(t)$ 进行采样，并作 200 点 FFT，得到离散干扰信号 $s_2(n)$ 的时域波形和频谱图如图 6 所示。

对信号 $xa(t)$ 进行采样，并作 200 点 FFT，得到离散合成信号 $x(n)$ 的时域波形和频谱图如图 7 所示。

5.3 问题 3

采用数字巴特沃斯低通滤波器进行滤波，设置参数通带截止频率 $W_p = 0.24\pi rad$ ，阻带截止频率 $W_s = 0.4\pi rad$ ，通带最大衰减系数 $R_p = 1dB$ ，阻带最小衰减系数 $R_s = 40dB$ 。数字巴特沃斯低通滤波器频率响应曲线如图 8 所示。 $\omega = 0.4$ 时幅度为 $-0.72dB$ ， $\omega = 0.4$ 时幅度为 $-40dB$ ，通带满足指标要求，阻带满足指标要求，过渡带比指标要求的窄。

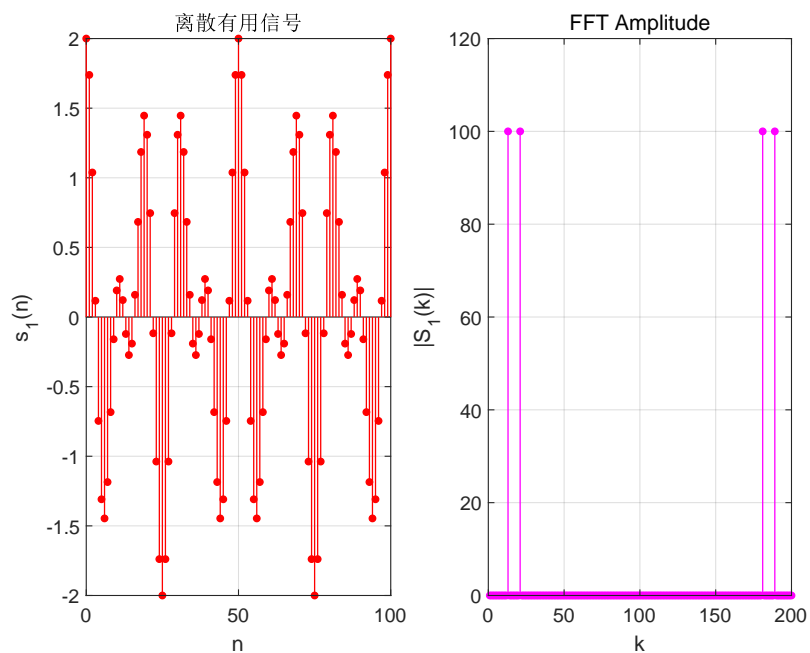


图 5: 离散有用信号 $s_1(n)$ 的时域波形和频谱图

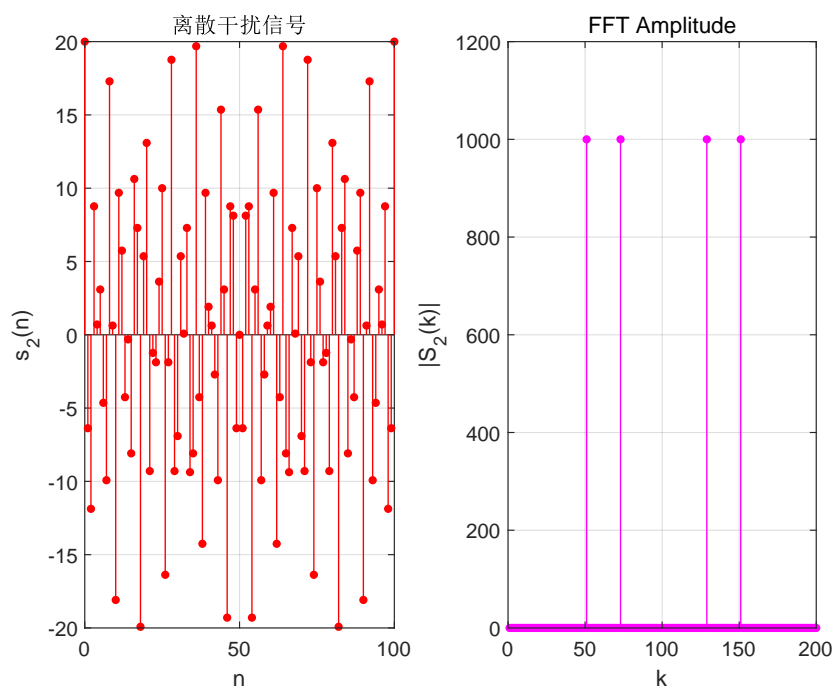


图 6: 离散干扰信号 $s_2(n)$ 的时域波形和频谱图

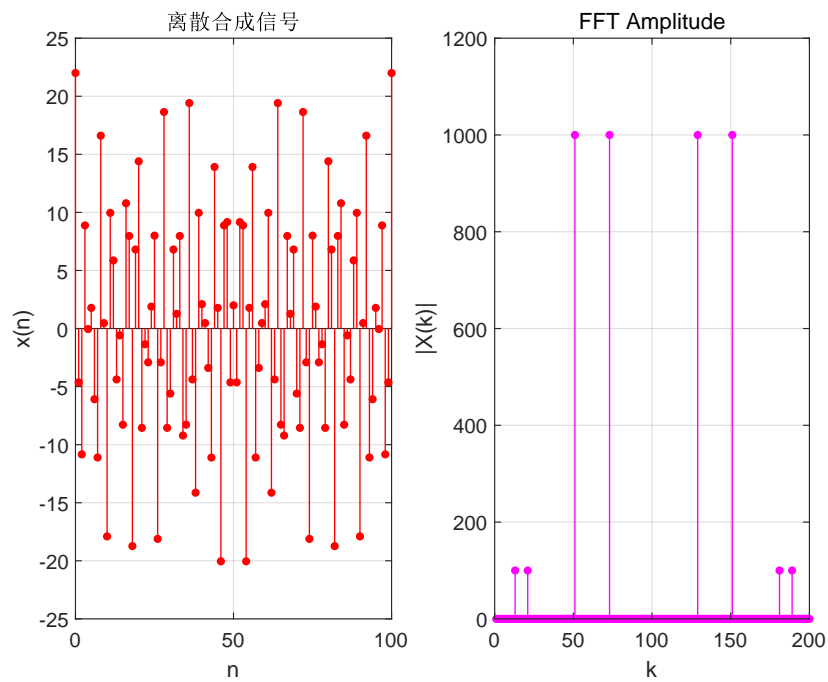


图 7: 离散合成信号 $x(n)$ 的时域波形和频谱图

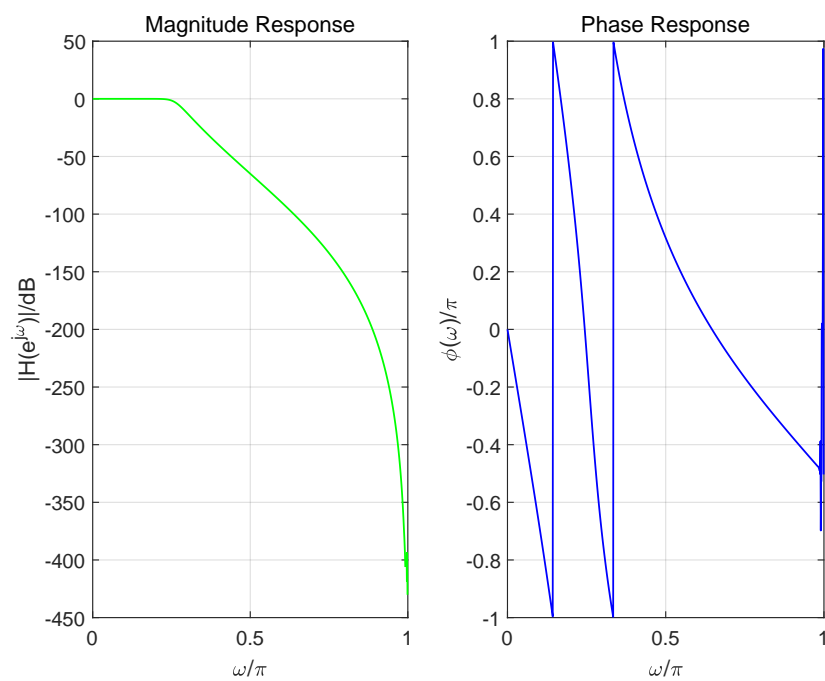


图 8: 数字低通滤波器幅频特性和相频特性曲线

5.4 问题 4

求解滤波器参数得到阶数 $N = 9$, 3dB 截止频率 $W_c = 0.2615$, 滤波器系统函数的分子、分母多项式系数向量如下:

$$B = 0.0000 \ 0.0004 \ 0.0017 \ 0.0041 \ 0.0061 \ 0.0061 \ 0.0041 \ 0.0017 \ 0.0004 \ 0.0000 \quad (7)$$

$$A = 1.0000 \ -4.2785 \ 8.8461 \ -11.2842 \ 9.6717 \ -5.7301 \ 2.3339 \ -0.6276 \ 0.1008 \ -0.0073 \quad (8)$$

根据 Matlab 官网函数说明, 得到的系统函数形式如下:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n+1)z^{-n}}{a(1) + a(2)z^{-1} + \dots + a(n+1)z^{-n}} \quad (9)$$

数字低通滤波器 $H(z)$ 结构信号流图如图 9 所示。

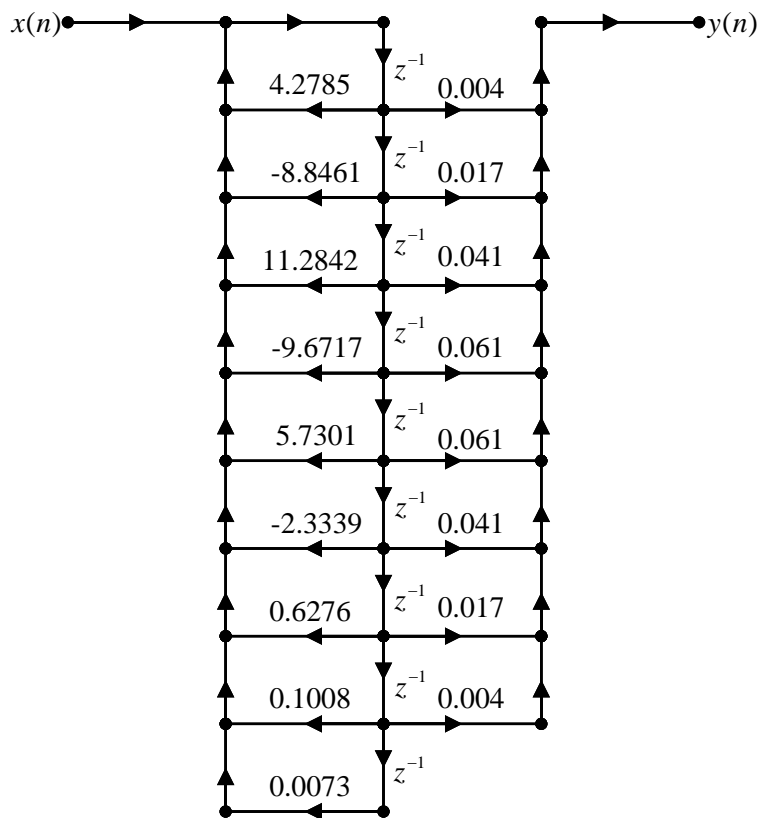


图 9: 数字低通滤波器 $H(z)$ 结构信号流图

5.5 问题 5

将合成信号 $x(n)$ 输入数字滤波器 $H(z)$, 得到输出响应 $y(n)$, 其时域波形和频谱图如图 10 所示。可见, 经过数字滤波器 $H(z)$ 滤波后, 大于截止频率的分量被滤除, 满足

低通滤波的要求。

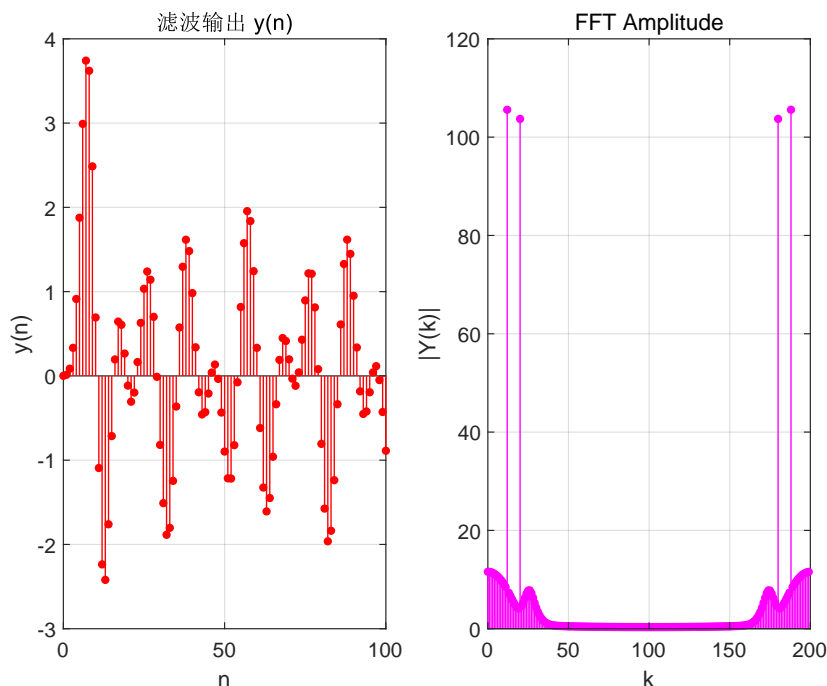


图 10: 滤波器输出响应 $y(n)$ 的时域波形和频谱图

6 总结

在本文中，我们针对数字信号处理课程设计要求，用 Visio 绘制了实验流程图如图 1 所示，针对实验流程用 Matlab 和 Python 两种语言进行编程实验与分析，绘制了 $sa_1(t)$ 、 $sa_2(t)$ 、 $xa(t)$ 三个模拟信号的时域波形和频谱图以及 $s_1(n)$ 、 $s_2(n)$ 、 $x(n)$ 三个离散信号的时域波形和频谱图，通过观测频谱特征，我们选择巴特沃斯数字低通滤波器进行滤波，并绘制滤波器的幅频特性和相频特性曲线，其指标满足设计要求，将 $x(n)$ 输入滤波器获得 $y(n)$ ，并绘制其时域波形和频谱图，最后结果符合我们的预期。本文设计的有用信号和干扰信号较为简单，但实际中的信号却都较为复杂，工业上设计滤波器流程较为复杂，需要更加深入学习相关知识并灵活运用。通过这次实验我们夯实了相关基础知识，能够较好地使用 Matlab 和 Python 进行信号处理，对 L^AT_EX 和 Visio 软件使用也更加熟练。

Appendices

附录 A Matlab 代码

```

1  % 问题 1
2  N = 2000; L = 2; k = 0:N-1;
3  dt = L / N; t = k * dt; f = k / L;
4
5  f1 = 10; f2 = 6; f3 = 36; f4 = 25;
6  sa1 = cos(2*pi*f1*t)+cos(2*pi*f2*t);
7  sa2 = 10*cos(2*pi*f3*t)+10*cos(2*pi*f4*t);
8  xa = sa1 + sa2;
9  F1 = fft(sa1,N);
10 F2 = fft(sa2,N);
11 F3 = fft(xa,N);
12
13 figure(1);
14 subplot(1,2,1)
15 plot(t,sa1,'b','LineWidth',0.8); grid on;
16 xlabel('t/s'); ylabel('sa_1(t)');
17 xlim([0 1]); title('有用信号');
18
19 subplot(1,2,2)
20 plot(f,0.5*abs(F1)/max(abs(F1)),'g','LineWidth',0.8); grid on;
21 xlabel('f/Hz'); ylabel('F(j2\pif)');
22 axis([0 40 0 1]); title('频谱图');
23
24 figure(2);
25 subplot(1,2,1)
26 plot(t,sa2,'b','LineWidth',0.8); grid on;
27 xlabel('t/s'); ylabel('sa_2(t)');
28 xlim([0 1]); title('干扰信号');
29
30 subplot(1,2,2)
31 plot(f,5*abs(F2)/max(abs(F2)),'g','LineWidth',0.8); grid on;
32 xlabel('f/Hz'); ylabel('F(j2\pif)');
33 axis([0 40 0 10]); title('频谱图');
34
35 figure(3);
36 subplot(1,2,1)
37 plot(t,xa,'b','LineWidth',0.8); grid on;
38 xlabel('t/s'); ylabel('xa(t)');
39 xlim([0 1]); title('合成信号');
40
41 subplot(1,2,2)

```

```

42 plot(f,5*abs(F3)/max(abs(F3)),'g','LineWidth',0.8); grid on;
43 xlabel('f/Hz'); ylabel('F(j2\pif)');
44 axis([0 40 0 10]); title('频谱图');
45
46 % 问题 2
47 Fs = 100; Ts = 1 / Fs; n = 0:199;
48 tn = n * Ts;
49 sn1 = cos(2*pi*f1*tn)+cos(2*pi*f2*tn);
50 sn2 = 10*cos(2*pi*f3*tn)+10*cos(2*pi*f4*tn);
51 xn = sn1 + sn2;
52 Fn1 = fft(sn1);
53 Fn2 = fft(sn2);
54 Fn3 = fft(xn);
55
56 figure(4);
57 subplot(1,2,1)
58 stem(n,sn1,'r','filled','LineWidth',0.6,'MarkerSize',3); grid on;
59 xlabel('n'); ylabel('s_1(n)');
60 xlim([0 100]); title('离散有用信号');
61
62 subplot(1,2,2)
63 stem(abs(Fn1),'m','filled','LineWidth',0.6,'MarkerSize',3); grid on;
64 xlabel('k'); ylabel('|S_1(k)|');
65 title('FFT Amplitude')
66
67 figure(5);
68 subplot(1,2,1)
69 stem(n,sn2,'r','filled','LineWidth',0.6,'MarkerSize',3); grid on;
70 xlabel('n'); ylabel('s_2(n)');
71 xlim([0 100]); title('离散干扰信号');
72
73 subplot(1,2,2)
74 stem(abs(Fn2),'m','filled','LineWidth',0.6,'MarkerSize',3); grid on;
75 xlabel('k'); ylabel('|S_2(k)|');
76 title('FFT Amplitude')
77
78 figure(6);
79 subplot(1,2,1)
80 stem(n,xn,'r','filled','LineWidth',0.6,'MarkerSize',3); grid on;
81 xlabel('n'); ylabel('x(n)');
82 xlim([0 100]); title('离散合成信号');
83
84 subplot(1,2,2)
85 stem(abs(Fn3),'m','filled','LineWidth',0.6,'MarkerSize',3); grid on;
86 xlabel('k'); ylabel('|X(k)|');
87 title('FFT Amplitude')
88

```

```

89 % 问题 3
90 fp=12; fs=20;
91 Wp=2*fp/Fs; Ws=2*fs/Fs; Rp=1; Rs=40;
92 [N,Wc] = buttord(Wp,Ws,Rp,Rs)
93 [B,A] = butter(N,Wc)
94 [H,w] = freqz(B,A,1000);
95
96 figure(7);
97 subplot(1,2,1);
98 plot(w/pi, 20*log10(abs(H)), 'g', 'LineWidth', 0.8); grid on;
99 xlabel('\omega/\pi'); ylabel('|H(e^j\omega)|/dB');
100 title('Magnitude Response');
101
102 subplot(1,2,2);
103 plot(w/pi, angle(H)/pi, 'b', 'LineWidth', 0.8); grid on;
104 xlabel('\omega/\pi'); ylabel('\phi(\omega)/\pi');
105 title('Phase Response');
106
107 % 问题 5
108 yn = filter(B,A,xn);
109 Yk = fft(yn);
110
111 n = 0:length(yn)-1;
112 figure(8);
113 subplot(1,2,1)
114 stem(n,yn, 'r', 'filled', 'LineWidth', 0.6, 'MarkerSize', 3); grid on;
115 % plot(n,yn, 'b', 'LineWidth', 1); hold on; grid on;
116 xlabel('n'); ylabel('y(n)');
117 xlim([0 100]); title('滤波输出 y(n)');
118
119 subplot(1,2,2)
120 stem(n,abs(Yk), 'm', 'filled', 'LineWidth', 0.6, 'MarkerSize', 3); grid on;
121 xlabel('k'); ylabel('|Y(k)|');
122 title('FFT Amplitude')

```

```

In [1]: import matplotlib.pyplot as plt
import numpy as np
from scipy import signal

In [2]: plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

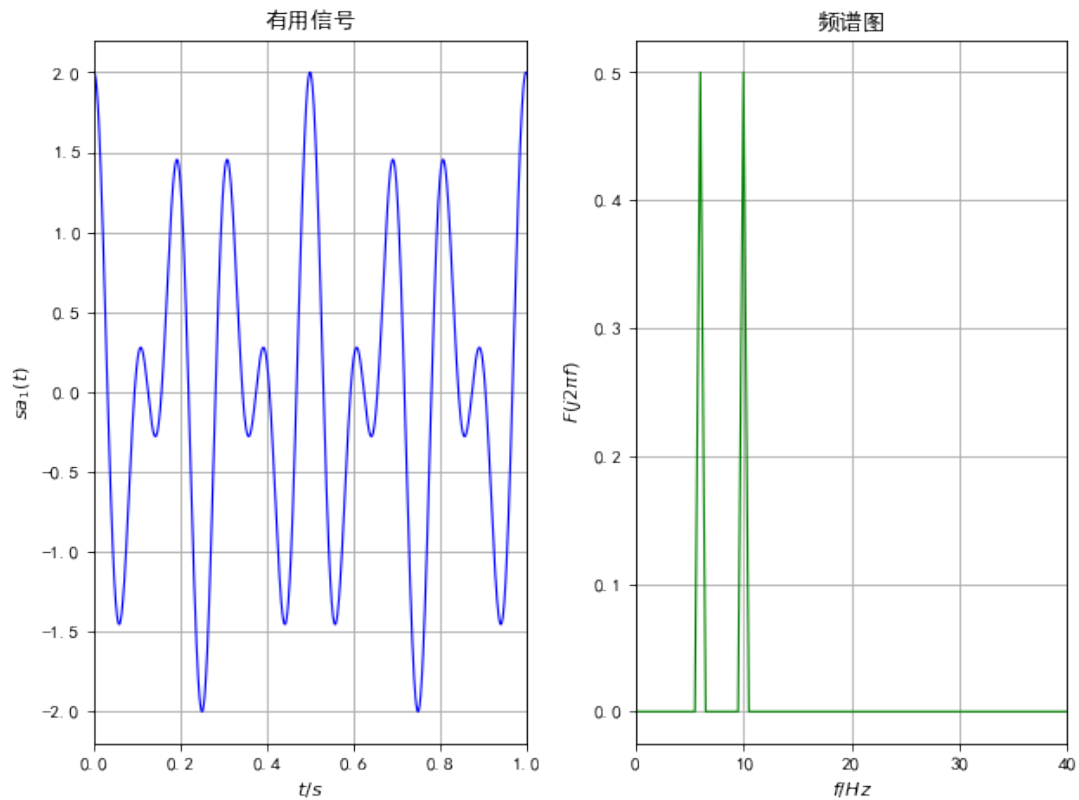
In [3]: f1 = 10; f2 = 6; f3 = 36; f4 = 25;
t = np.arange(0.0, 2.0, 0.001)
f = np.arange(0.0, 1000.0, 0.5)
sa1 = np.cos(2.0*np.pi*f1*t) + np.cos(2.0*np.pi*f2*t)
sa2 = 10*np.cos(2.0*np.pi*f3*t) + 10*np.cos(2.0*np.pi*f4*t)
xa = sa1 + sa2
F1 = np.fft.fft(sa1)
F2 = np.fft.fft(sa2)
F3 = np.fft.fft(xa)

plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
plt.plot(t, sa1, 'b', linewidth =1.0)
plt.xlim(0, 1)
plt.xlabel('$t/s$')
plt.ylabel('$sa_1(t)$')
plt.title(' [U+6709] [U+7528] [U+4FE1] [U+53F7] ')
plt.grid()

plt.subplot(122)
plt.plot(f, 0.5*abs(F1)/max(abs(F1)), 'g', linewidth =1.0)
plt.xlim(0, 40)
plt.xlabel('$f/Hz$')
plt.ylabel('$F(j2\pi f)$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

plt.tight_layout()
plt.show()

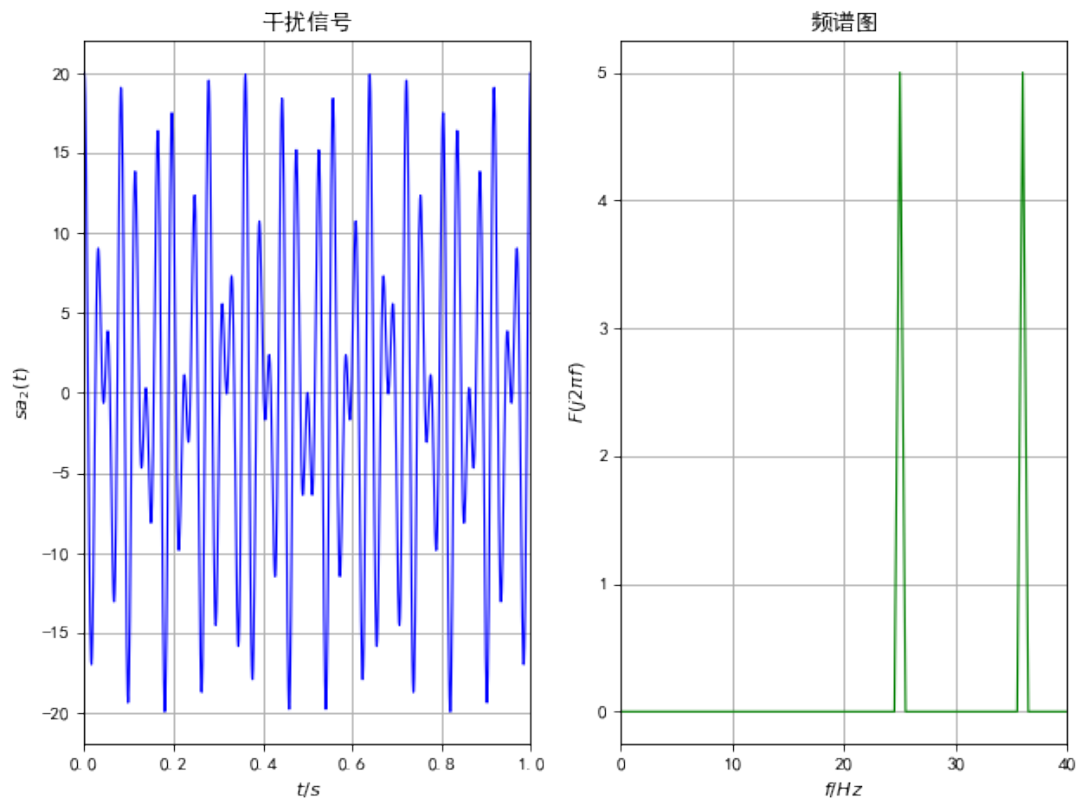
```



```
In [4]: plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
plt.plot(t, sa2, 'b', linewidth =1.0)
plt.xlim(0, 1)
plt.xlabel('$t/s$')
plt.ylabel('$sa_2(t)$')
plt.title(' [U+5E72] [U+6270] [U+4FE1] [U+53F7] ')
plt.grid()

plt.subplot(122)
plt.plot(f, 5*abs(F2)/max(abs(F2)), 'g', linewidth =1.0)
plt.xlim(0, 40)
plt.xlabel('$f/Hz$')
plt.ylabel('$F(j2\pi f)$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

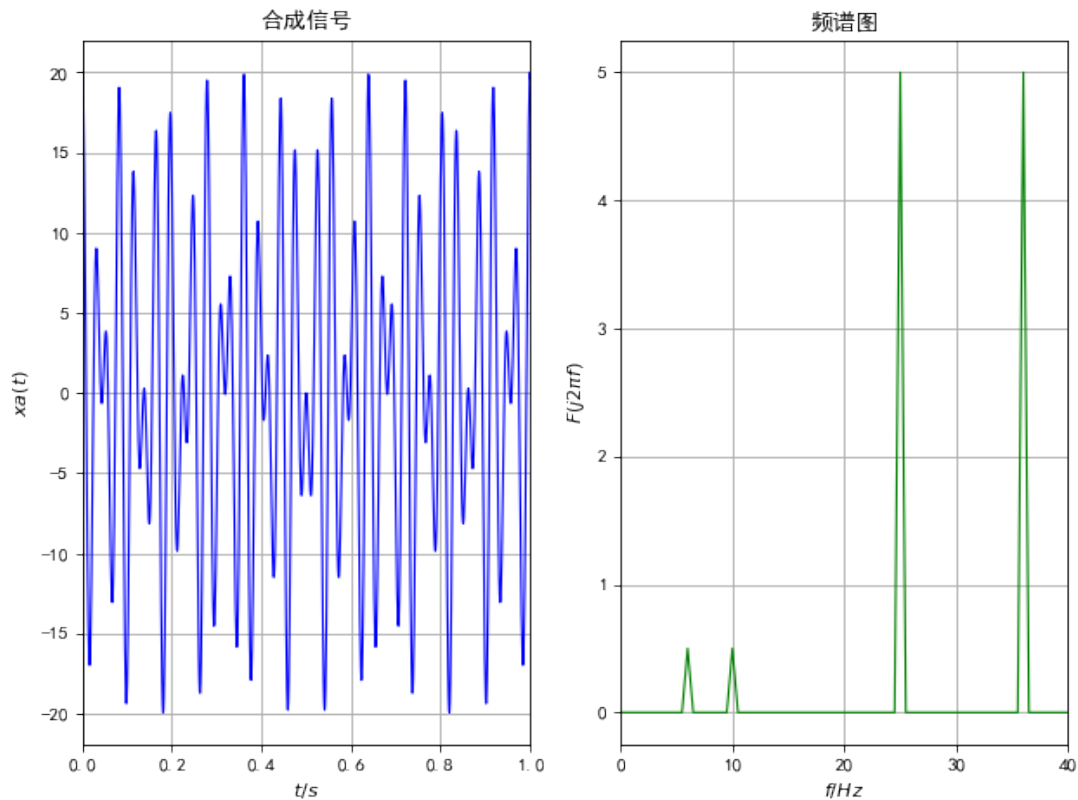
plt.tight_layout()
plt.show()
```

```
In [5]: plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
plt.plot(t, sa2, 'b', linewidth =1.0)
plt.xlim(0, 1)
plt.xlabel('$t/s$')
plt.ylabel('$sa(t)$')
plt.title(' [U+5408] [U+6210] [U+4FE1] [U+53F7] ')
plt.grid()

plt.subplot(122)
plt.plot(f, 5*abs(F3)/max(abs(F3)), 'g', linewidth =1.0)
plt.xlim(0, 40)
plt.xlabel('$f/Hz$')
plt.ylabel('$F(j2\pi f)$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

plt.tight_layout()
plt.show()
```



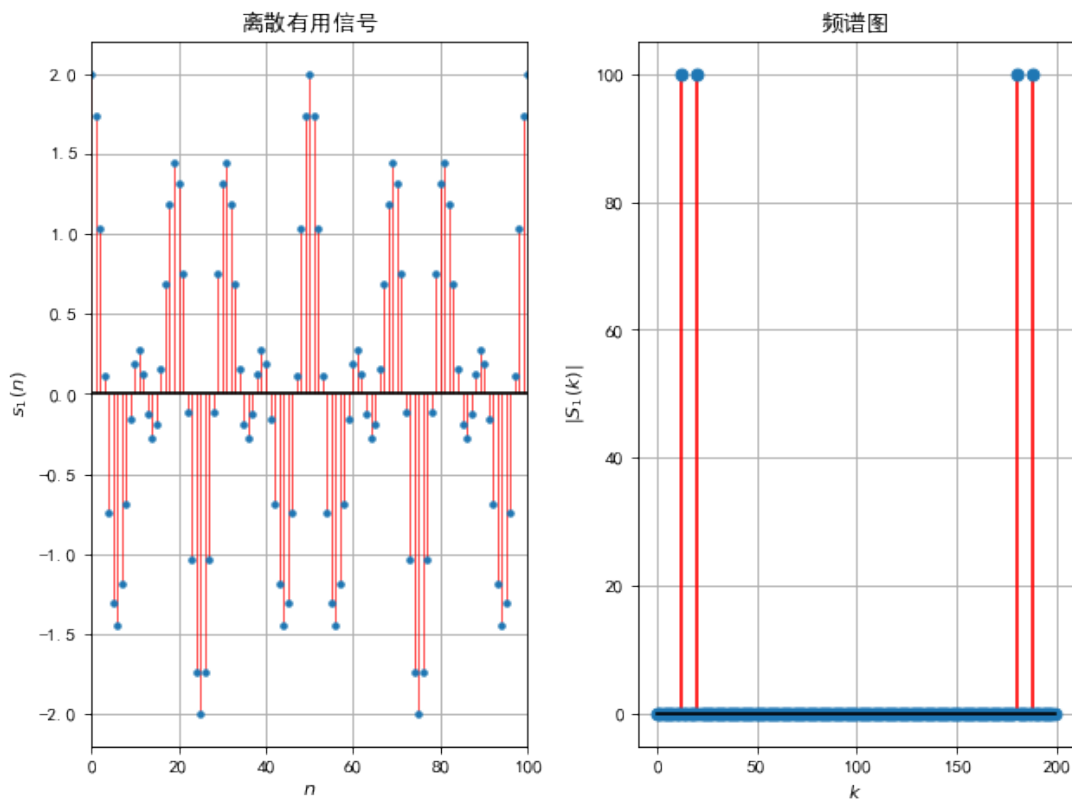
```
In [6]: nt = np.arange(0.0, 2.0, 0.01)
k = np.arange(0.0, 200.0, 1.0)
sn1 = np.cos(2.0*np.pi*f1*nt) + np.cos(2.0*np.pi*f2*nt)
sn2 = 10*np.cos(2.0*np.pi*f3*nt) + 10*np.cos(2.0*np.pi*f4*nt)
xn = sn1 + sn2
Fn1 = np.fft.fft(sn1)
Fn2 = np.fft.fft(sn2)
Fn3 = np.fft.fft(xn)

plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
markerline, stemlines, baseline = plt.stem(k, sn1, linefmt='r',
      basefmt='k-', markerfmt='C0.', use_line_collection=True)
plt.setp(stemlines, 'linewidth', 0.6)
plt.xlim(0, 100)
plt.xlabel('$n$')
plt.ylabel('$s_1(n)$')
plt.title(' [U+79BB] [U+6563] [U+6709] [U+7528] [U+4FE1] [U+53F7] ')
plt.grid()

plt.subplot(122)
```

```
plt.stem(k, abs(Fn1), linefmt='r', basefmt='k-',
        markerfmt='COo', use_line_collection=True)
plt.xlabel('$k$')
plt.ylabel('$|S_1(k)|$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

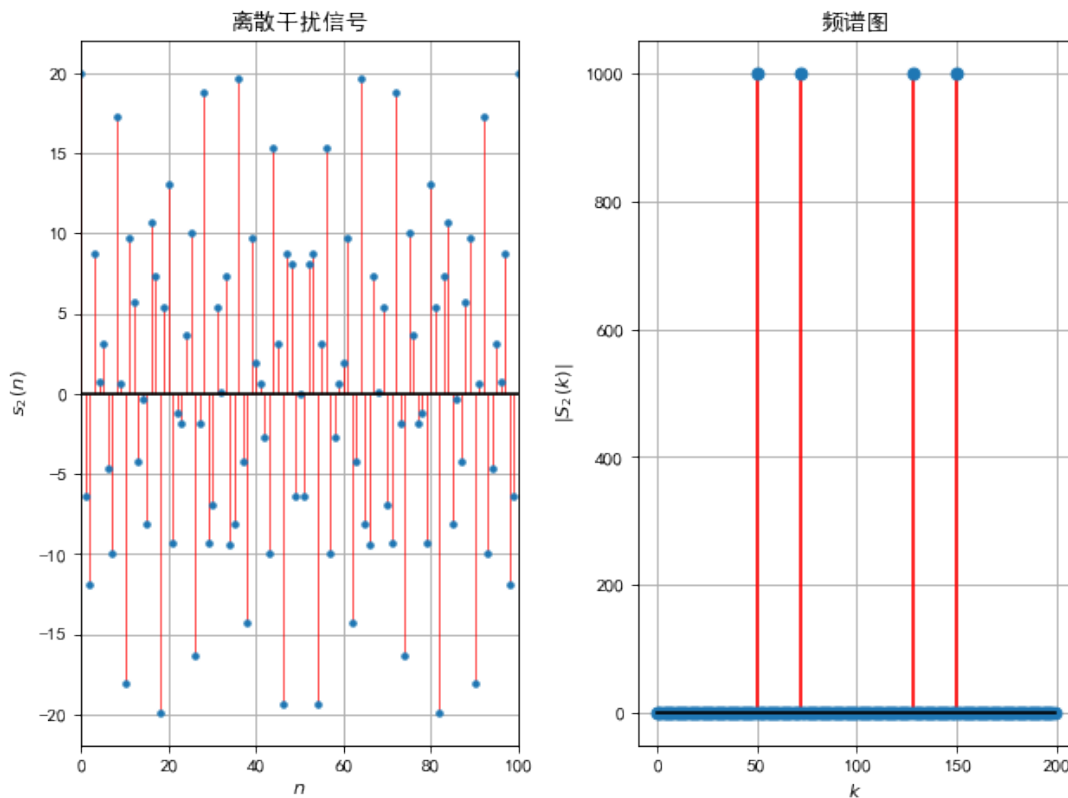
plt.tight_layout()
plt.show()
```



```
In [7]: plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
markerline, stemlines, baseline = plt.stem(k, sn2, linefmt='r',
        basefmt='k-', markerfmt='CO.', use_line_collection=True)
plt.setp(stemlines, 'linewidth', 0.6)
plt.xlim(0, 100)
plt.xlabel('$n$')
plt.ylabel('$s_2(n)$')
plt.title(' [U+79BB] [U+6563] [U+5E72] [U+6270] [U+4FE1] [U+53F7] ')
plt.grid()
```

```
plt.subplot(122)
plt.stem(k, abs(Fn2), linefmt='r', basefmt='k-',
         markerfmt='C0o', use_line_collection=True)
plt.xlabel('$k$')
plt.ylabel('$|S_2(k)|$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

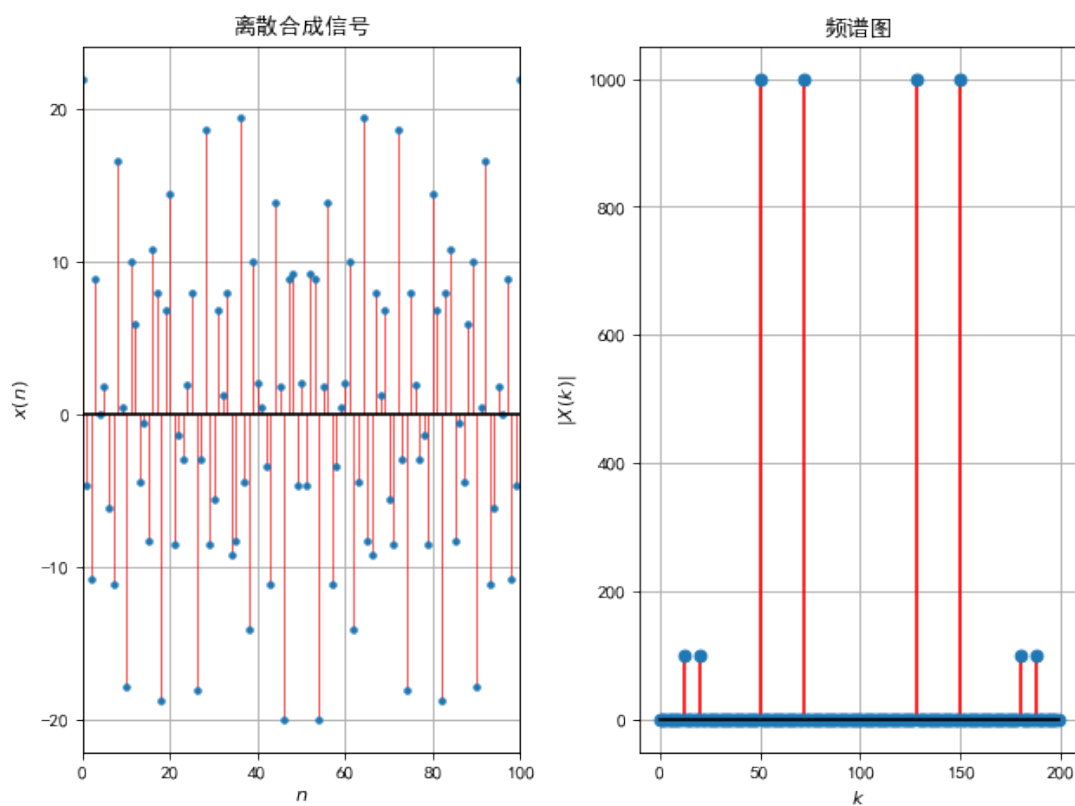
plt.tight_layout()
plt.show()
```



```
In [8]: plt.figure(figsize=(8, 6), dpi=80)
plt.subplot(121)
markerline, stemlines, baseline = plt.stem(k, xn, linefmt='r',
         basefmt='k-', markerfmt='C0.', use_line_collection=True)
plt.setp(stemlines, 'linewidth', 0.6)
plt.xlim(0, 100)
plt.xlabel('$n$')
plt.ylabel('$x(n)$')
plt.title(' [U+79BB] [U+6563] [U+5408] [U+6210] [U+4FE1] [U+53F7] ')
plt.grid()
```

```
plt.subplot(122)
plt.stem(k, abs(Fn3), linefmt='r', basefmt='k-',
        markerfmt='Co', use_line_collection=True)
plt.xlabel('$k$')
plt.ylabel('$|X(k)|$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

plt.tight_layout()
plt.show()
```



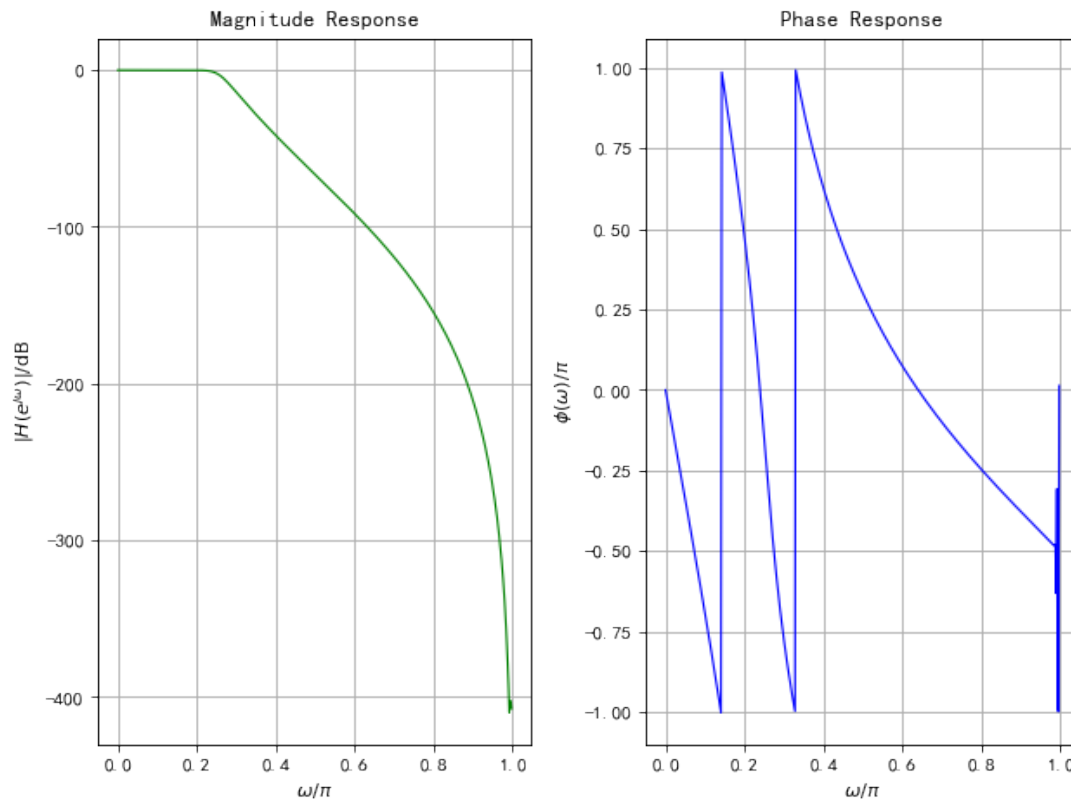
```
In [9]: fp=12; fs=20; Fs=100;
Wp=2*fp/Fs; Ws=2*fs/Fs; Rp=1; Rs=40;
N, Wc = signal.buttord(Wp, Ws, Rp, Rs)
b, a = signal.butter(N, Wc, 'low')
w, h = signal.freqz(b, a)

plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
plt.plot(w/np.pi, 20 * np.log10(abs(h)), 'g', linewidth =1.0)
```

```
plt.xlabel('$\omega/\pi$')
plt.ylabel('$|H(e^{j\omega})|/\mathrm{dB}$')
plt.title('Magnitude Response')
plt.grid()

plt.subplot(122)
plt.plot(w/np.pi, np.angle(h)/np.pi, 'b', linewidth =1.0)
plt.xlabel('$\omega/\pi$')
plt.ylabel('$\phi(\omega)/\pi$')
plt.title('Phase Response')
plt.grid()

plt.tight_layout()
plt.show()
```



```
In [10]: yn = signal.filtfilt(b, a, xn)
         Yk = np.fft.fft(yn)

plt.figure(figsize =(8, 6), dpi =80)
plt.subplot(121)
markerline, stemlines, baseline = plt.stem(k, yn, linefmt='r',
```

```

        basefmt='k-', markerfmt='C0.', use_line_collection=True)
plt.setp(stemlines, 'linewidth', 0.6)
plt.xlim(0, 100)
plt.ylim(-3, 4)
plt.xlabel('$n$')
plt.ylabel('$y(n)$')
plt.title(' [U+6EE4] [U+6CE2] [U+8F93] [U+51FA] [U+4FE1] [U+53F7] ')
plt.grid()

plt.subplot(122)
markerline, stemlines, baseline = plt.stem(k, abs(Yk), linefmt='r',
        basefmt='k-', markerfmt='C0.', use_line_collection=True)
plt.setp(stemlines, 'linewidth', 0.6)
plt.xlabel('$k$')
plt.ylabel('$|Y(k)|$')
plt.title(' [U+9891] [U+8C31] [U+56FE] ')
plt.grid()

plt.tight_layout()
plt.show()

```

