安徽大學

数字信号处理实验报告

年级专业 22级 电子信息工程

学号 P12214061

姓名 李想

实验7　IIR 数字滤波器的设计实验

一．实验目的

1. 掌握设计 IIR 数字滤波器的原理和方法。

2. 熟悉 IIR 数字滤波器特性。

二．实验设备

PC 兼容机一台，操作系统为 WindowsXP，安装 CCS 软件。

三．实验原理

1. 利用模拟滤波器设计 IIR 数字滤波器的基础理论。（请参考教材《数字信号处理教程》）

2．根据要求采用双线性变换法设计 IIR 数字低通滤波器：

要求：数字低通巴特沃斯滤波器在其通带边缘 $f_p = 1\text{kHz}$ 处的增益为-3dB，在 $f_{st} = 12\,\text{kHz}$ 处的阻带衰减为 30dB，采样频率 $f_s = 25\text{kHz}$ 。

(1) 利用 $\omega = \Omega \cdot T_s = 2\pi f / f_s$ 关系把由 Hz 为单位的技术指标转换成以弧度为单位的数字频率，得到数字滤波器的通带截止频率 ω_p 和阻带截止频率 ω_{st} 。

$\omega_p = 2\pi f_p / f_s = 2\pi \cdot 1000/25000 = 0.08\pi$ 弧度

$\omega_{st} = 2\pi f_{st} / f_s = 2\pi \cdot 12000/25000 = 0.96\pi$ 弧度

(2) 由于采用双线性变换法，故需考虑预畸变，将数字域指标转变为模拟域指标。即由 $\Omega = \frac{2}{T_s}\tan\frac{\omega}{2} = 2f_s \cdot \tan\frac{\omega}{2}$ 求得模拟低通滤波器的通带截止频率 Ω_p 和阻带截止频率 Ω_{st} 。

$\Omega_p = 2f_s \tan\frac{\omega_p}{2} = 6316.5$ 弧度/秒； $\Omega_{st} = 2f_s \tan\frac{\omega_{st}}{2} = 794727.2$ 弧度/秒

$\delta_1 = 3\,\text{dB}$， $\delta_2 = 30\,\text{dB}$ 。

(3) 计算所需滤波器的阶数：

$$\lambda_{sp} = \Omega_{st} / \Omega_p \qquad k_{sp} = \sqrt{\frac{10^{0.1\delta_1} - 1}{10^{0.1\delta_2} - 1}}$$

$$\therefore N \geq -\frac{\lg k_{sp}}{\lg \lambda_{sp}} = 0.714 \quad \text{故取 } N = 1 \text{。}$$

因此，一阶巴特沃斯滤波器就足以满足要求。

(4) 求一阶模拟低通巴特沃斯滤波器的系统函数为：

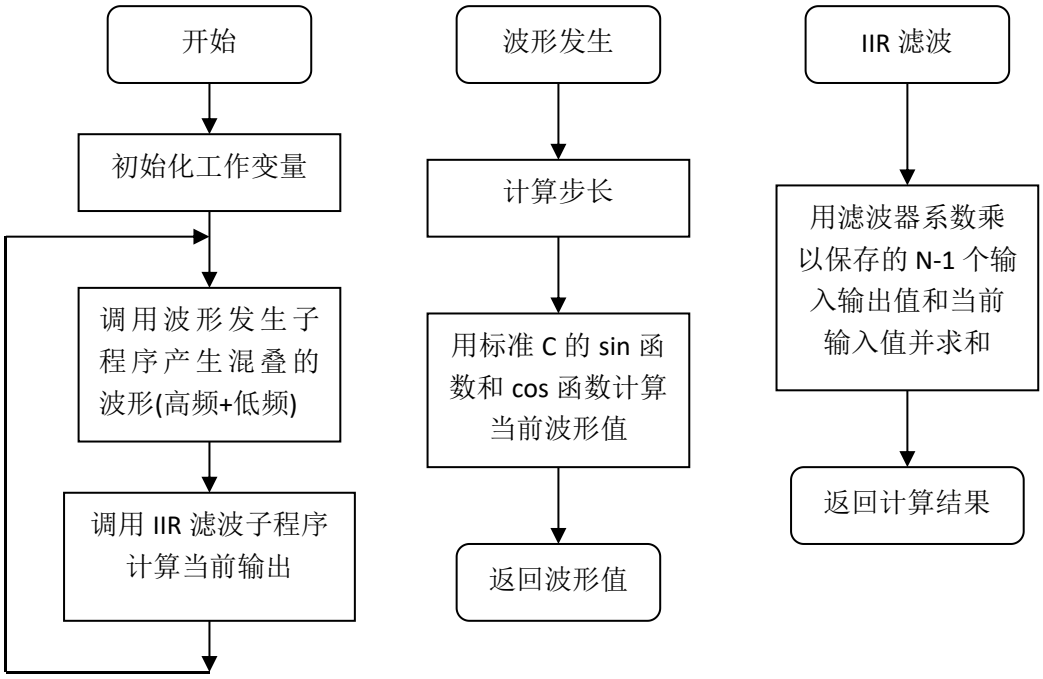
$$H(s) = \frac{\Omega_c}{s + \Omega_c} = \frac{6316.5}{s + 6316.5} \qquad (\Omega_c = \Omega_p)$$

(5) 由双线性变换法代入：$s = \frac{2}{T_s}\frac{z-1}{z+1} = 2f_s\frac{z-1}{z+1}$，可得到数字滤波器的系统函数为：

$$H(z) = \frac{6316.5}{50000\dfrac{z-1}{z+1} + 6316.5} = \frac{0.1122(1+z^{-1})}{1-0.7757z^{-1}}$$

因此，差分方程为：$y(n) = 0.7757y(n-1) + 0.1122x(n) + 0.1122x(n-1)$

3．程序流程图：



四．实验步骤

1. 启动 CCS，并设置软件仿真工作模式。（参看实验 5 的第四部分的第 1 步）

2. 导入工程文件：（参看实验 2 的第四部分的第 3 步）

工程目录为 C:\ICETEK\ICETEK-VC5509-AF v2.1\Lab0502-IIR

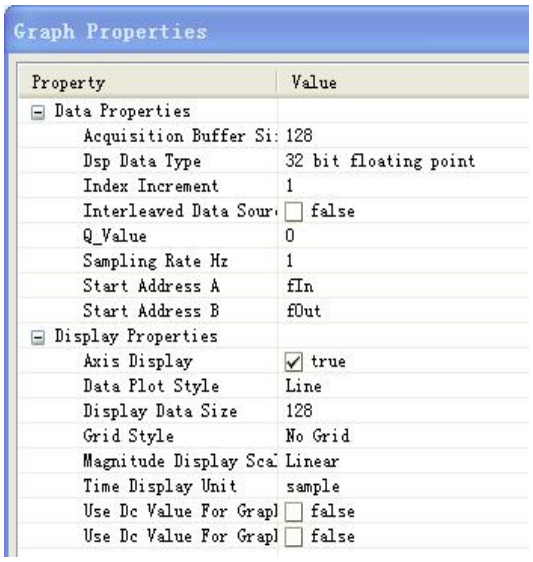
浏览 iir.c 文件的内容，理解各语句作用。

```
float IIR()
{
 float fSum;
 fSum=0.0;
 for ( i=0;i<IIRNUMBER;i++ )
 {
     fSum+=(fXn[i]*fAn[i]);
     fSum+=(fYn[i]*fBn[i]);
 }
 return(fSum);
}
```

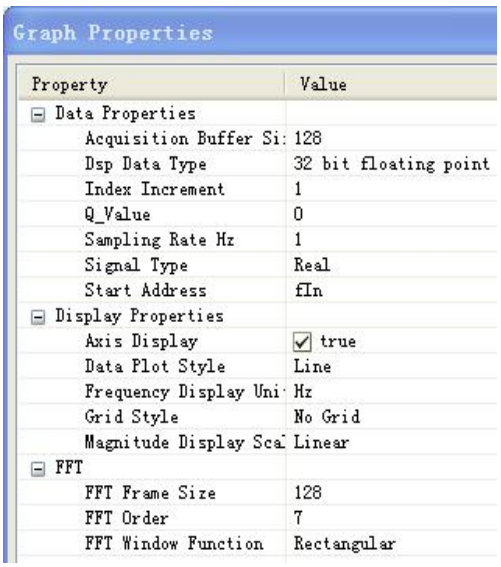
3. 点击图标  ，CCS 会自动连接，编译和下载程序。

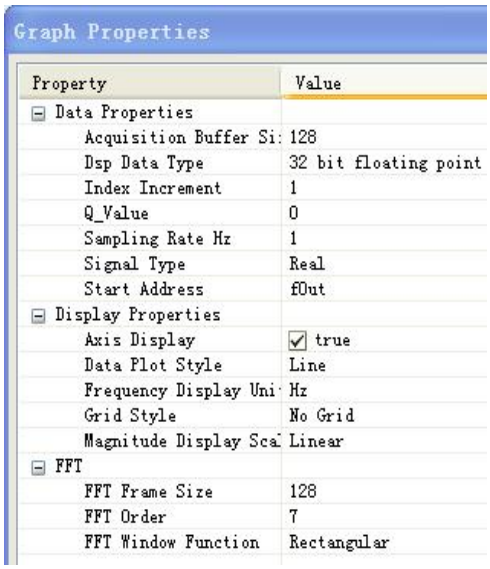
4. 打开观察窗口：

(1) 选择菜单 Tools->Graph->Dual Time 进行如下设置：



(2) 选择菜单 Tools->Graph->FFT Magnitude，新建 2 个观察窗口，分别进行如下设置：





5．设置断点：在程序中有注释“break point”的语句上设置软件断点。

使用菜单的 View->Break points 打开断点观察窗口，在刚才设置的断点上右键->Breadk point properties 调出断点的属性设置界面，设置 Action 为 Refresh All windows。则程序每次运行到断点，所有的观察窗口值都会被刷新。

6．运行程序，观察结果：

按 F8 键运行程序，观察各个窗口中的波形。

7．修改程序：在**float IIR**(.)子程序中的程序行**return**(fSum)；之前加入语句：

fYn[0]=fSum；然后再重新编译程序并下载、运行，观察新的结果；并分析原因。

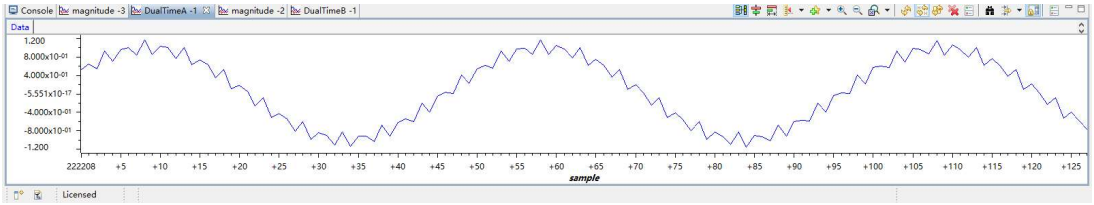
8．退出 CCS。

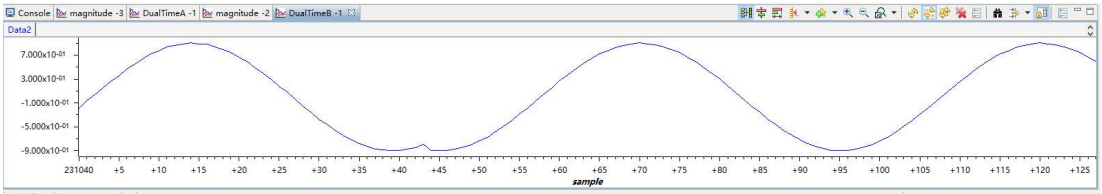
五．实验报告要求

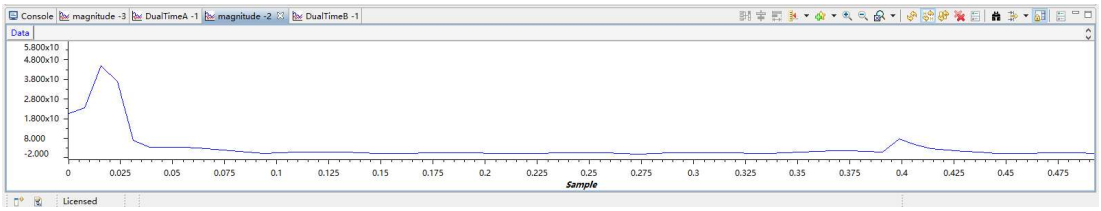
1．记录实验结果图，并分别从时域和频率解释观察到的现象。

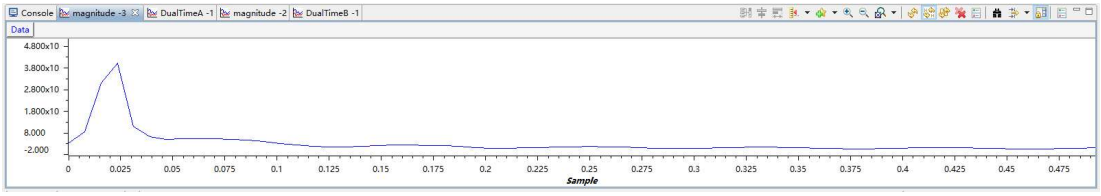
2．逐渐调整 fStepSignal2 的值，记录并分析实验结果。

3．根据实验原理中的设计要求，试用 MATLAB 编程实现 IIR 数字滤波器的设计，并完成与实验结果相同的滤波功能。


Console
magnitude -3
DualTimeA -1
magnitude -2
DualTimeB -1
Data
1.200
8.000x10-01
4.000x10-01
-5.551x10-17
-4.000x10-01
-8.000x10-01
-1.200
222208
+5
+10
+15
+20
+25
+30
+35
+40
+45
+50
+55
+60
+65
+70
+75
+80
+85
+90
+95
+100
+105
+110
+115
+120
+125
sample
Licensed


Console
magnitude -3
DualTimeA -1
magnitude -2
DualTimeB -1
Data2
7.000x10-01
3.000x10-01
-1.000x10-01
-5.000x10-01
-9.000x10-01
231040
+5
+10
+15
+20
+25
+30
+35
+40
+45
+50
+55
+60
+65
+70
+75
+80
+85
+90
+95
+100
+105
+110
+115
+120
+125
sample
Licensed


Console
magnitude -3
DualTimeA -1
magnitude -2
DualTimeB -1
Data
5.800x10
4.800x10
3.800x10
2.800x10
1.800x10
8.000
-2.000
0
0.025
0.05
0.075
0.1
0.125
0.15
0.175
0.2
0.225
0.25
0.275
0.3
0.325
0.35
0.375
0.4
0.425
0.45
0.475
Sample
Licensed


Console
magnitude -3
DualTimeA -1
magnitude -2
DualTimeB -1
Data
4.800x10
3.800x10
2.800x10
1.800x10
8.000
-2.000
0
0.025
0.05
0.075
0.1
0.125
0.15
0.175
0.2
0.225
0.25
0.275
0.3
0.325
0.35
0.375
0.4
0.425
0.45
0.475
Sample

线性变换法设计 IIR 数字低通滤波器

```
clc
clear
close all
```

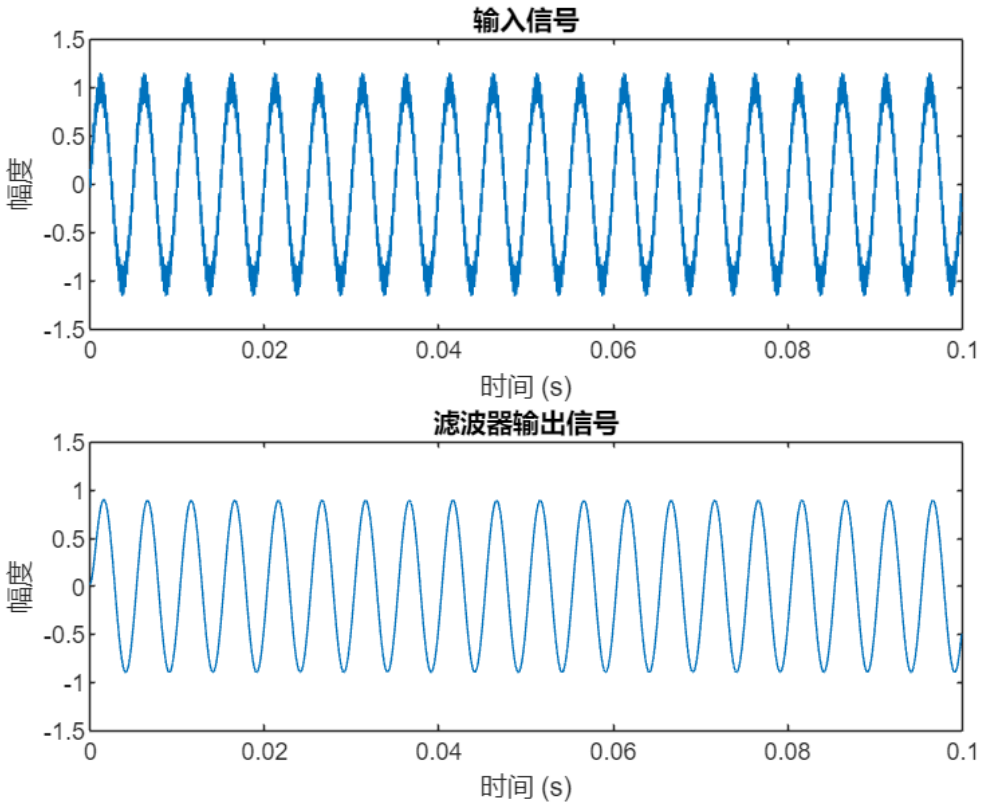
```
% 参数设置
IIRNUMBER = 2;          % 滤波器阶数
SAMPLEF = 10000;        % 采样频率（Hz）
PI = pi;

% 滤波器系数
fBn = [0.0, 0.7757];    % 反馈系数
fAn = [0.1122, 0.1122];% 前馈系数

% 滤波器状态（延迟线）
fXn = zeros(1, IIRNUMBER); % 输入信号历史值
fYn = zeros(1, IIRNUMBER); % 输出信号历史值

% 信号生成参数
fSignal1 = 0.0;                 % 信号1初始相位
fSignal2 = PI * 0.1;            % 信号2初始相位
fStepSignal1 = 2 * PI / 50;     % 信号1相位步长（对应200 Hz）
% fStepSignal2 = 2 * PI / 5;      % 信号2相位步长（对应2000 Hz）
fStepSignal2 = [2*PI/2,2*PI/3, 2*PI/4,2*PI/5, 2*PI/6]; %更改fStepSignal2的值

% 缓冲区设置
BUFFER_SIZE = 256;      % 缓冲区大小
fIn = zeros(1, BUFFER_SIZE);  % 输入信号缓冲区
fOut = zeros(1, BUFFER_SIZE); % 输出信号缓冲区
nIn = 1; nOut = 1;      % 缓冲区索引

% 模拟输入和滤波处理
NUM_ITER = 1000;        % 总采样点数
inputSignal = zeros(1, NUM_ITER); % 用于记录输入信号
outputSignal = zeros(1, NUM_ITER);% 用于记录输出信号

for m = 1:length(fStepSignal2)
    disp("fStepSignal2 = " + num2str(fStepSignal2(m)))
    for k = 1:NUM_ITER
        % 生成输入信号
        fXn(1) = sin(fSignal1) + cos(fSignal2) / 6.0;

        % 更新信号相位
        fSignal1 = fSignal1 + fStepSignal1;
        if fSignal1 >= 2 * PI
            fSignal1 = fSignal1 - 2 * PI;
        end
        fSignal2 = fSignal2 + fStepSignal2(m);
        if fSignal2 >= 2 * PI
            fSignal2 = fSignal2 - 2 * PI;
        end

        % 保存到缓冲区
```

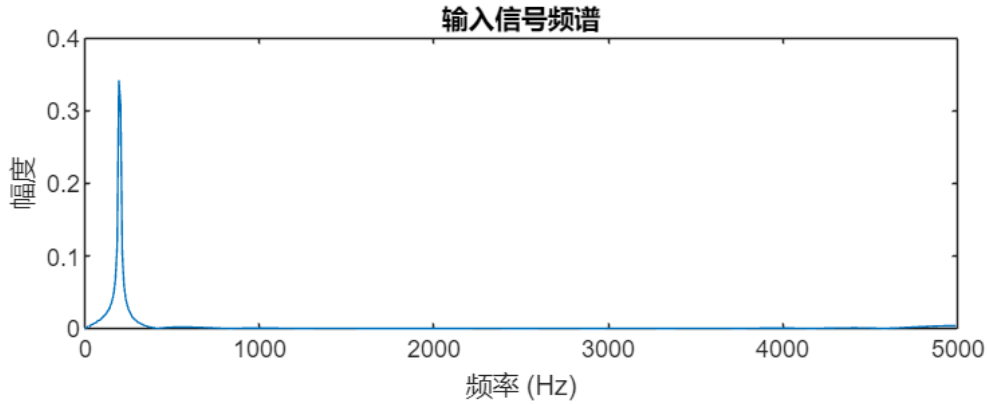
```
        fIn(nIn) = fXn(1);
        nIn = mod(nIn, BUFFER_SIZE) + 1; % 循环缓冲区索引

        % IIR 滤波器计算
        fSum = 0.0;
        for i = 1:IIRNUMBER
            fSum = fSum + fXn(i) * fAn(i) + fYn(i) * fBn(i);
        end
        fYn(1) = fSum; % 滤波器当前输出

        % 更新延迟线
        for i = IIRNUMBER:-1:2
            fXn(i) = fXn(i-1);
            fYn(i) = fYn(i-1);
        end

        % 保存滤波器输出到缓冲区
        fOut(nOut) = fSum;
        nOut = mod(nOut, BUFFER_SIZE) + 1; % 循环缓冲区索引

        % 保存结果到数组
        inputSignal(k) = fXn(1);
        outputSignal(k) = fSum;
    end

    % 绘图
    t = (0:NUM_ITER-1) / SAMPLEF; % 时间轴
    figure;
    subplot(2, 1, 1);
    plot(t, inputSignal);
    title('输入信号');
    xlabel('时间 (s)');
    ylabel('幅度');
    ylim([-1.5,1.5]);

    subplot(2, 1, 2);
    plot(t, outputSignal);
    title('滤波器输出信号');
    xlabel('时间 (s)');
    ylabel('幅度');
    ylim([-1.5,1.5]);

    % 计算频谱
    NFFT = 2^nextpow2(NUM_ITER); % 为提高频谱分辨率，对点数扩展到最近的2的幂
    freqAxis = SAMPLEF * (0:(NFFT/2)-1) / NFFT; % 频率轴

    % 输入信号频谱
    inputSpectrum = abs(fft(inputSignal, NFFT) / NUM_ITER);
    inputSpectrum = inputSpectrum(1:NFFT/2); % 取一半频谱（正频率部分）

    % 输出信号频谱
    outputSpectrum = abs(fft(outputSignal, NFFT) / NUM_ITER);
    outputSpectrum = outputSpectrum(1:NFFT/2); % 取一半频谱（正频率部分）

    % 绘制频谱
    figure;
    subplot(2, 1, 1);
    plot(freqAxis, inputSpectrum);
```

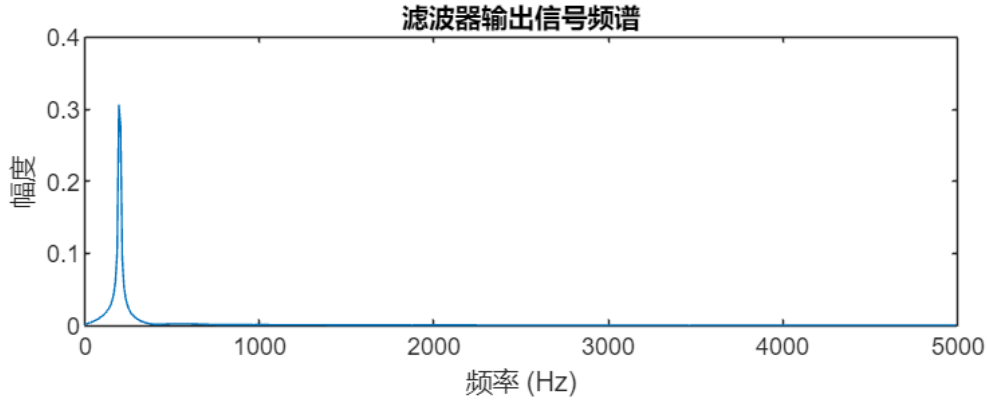
```
    title('输入信号频谱');
    xlabel('频率 (Hz)');
    ylabel('幅度');
    ylim([0,0.4]);

    subplot(2, 1, 2);
    plot(freqAxis, outputSpectrum);
    title('滤波器输出信号频谱');
    xlabel('频率 (Hz)');
    ylabel('幅度');
    ylim([0,0.4]);
end
```

Output

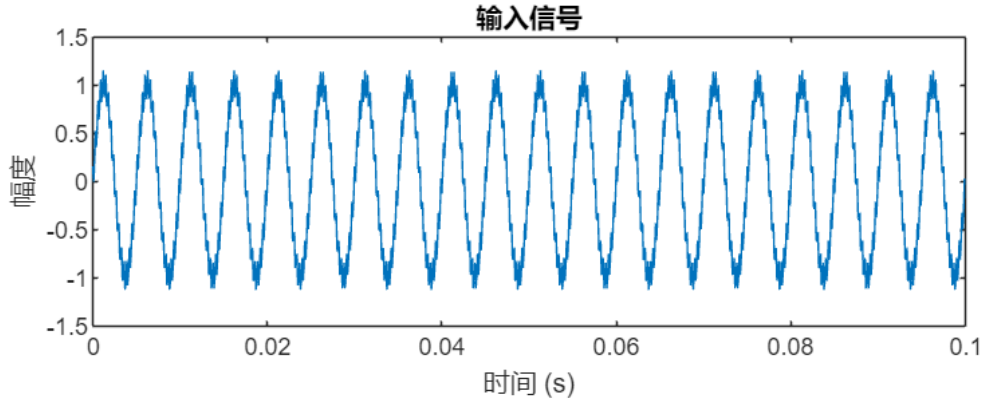
```
fStepSignal2 = 3.1416
```

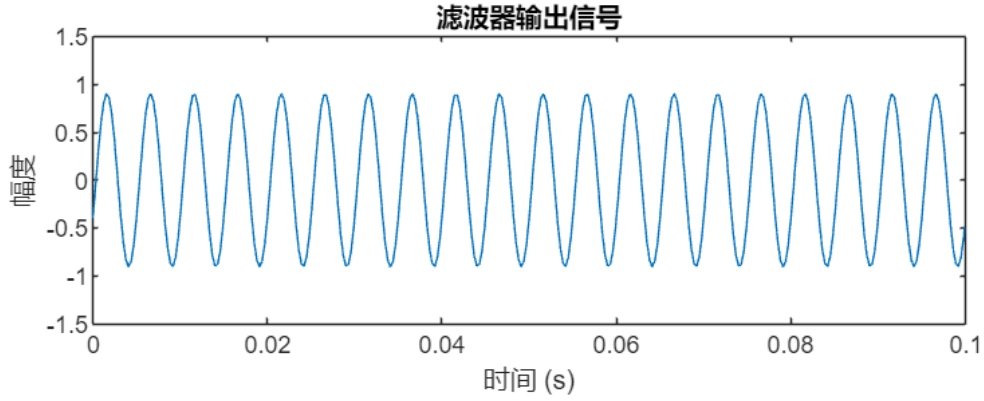


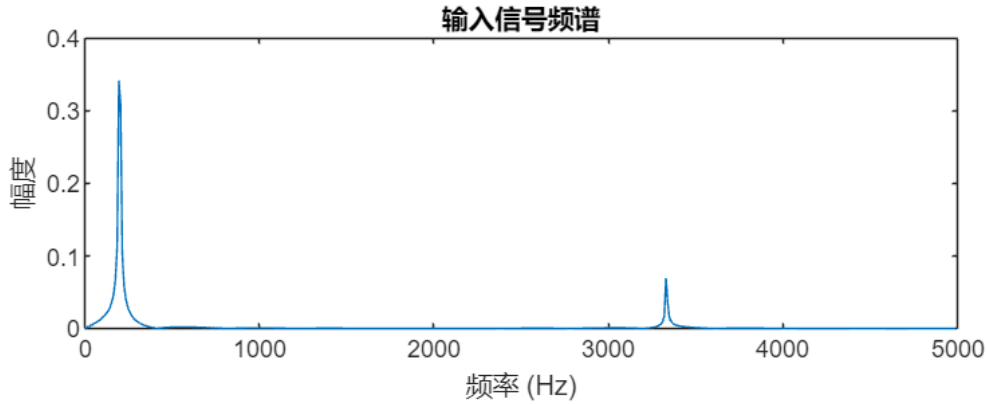


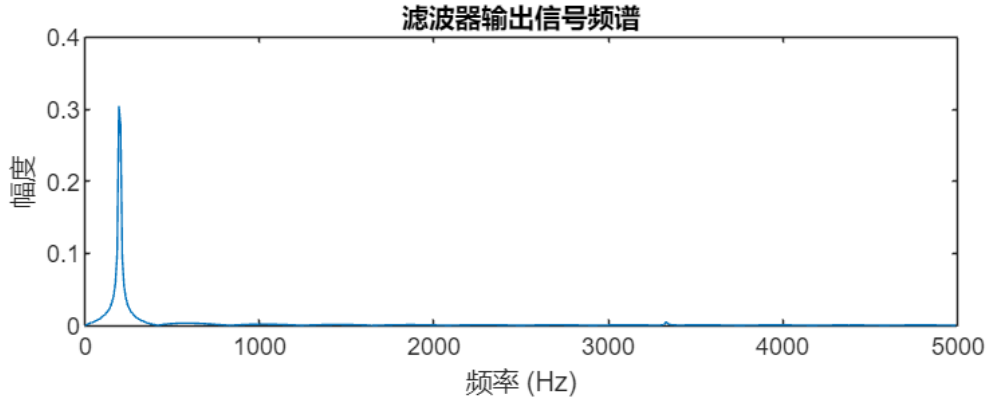



```
fStepSignal2 = 2.0944
```

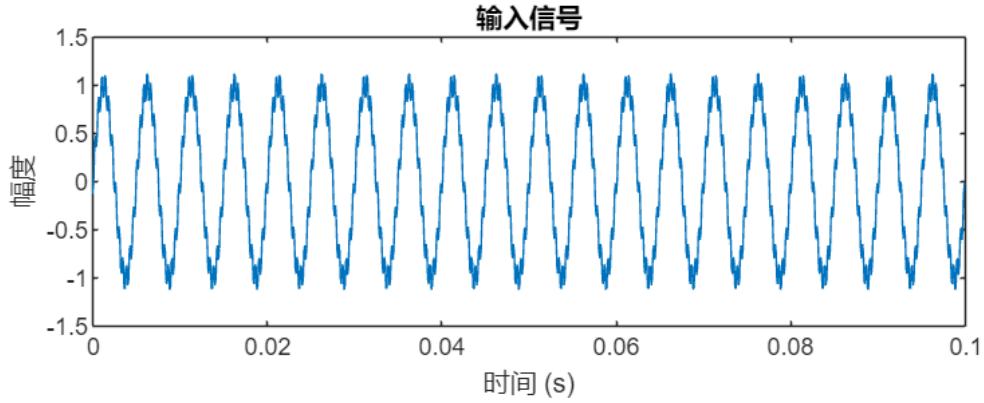


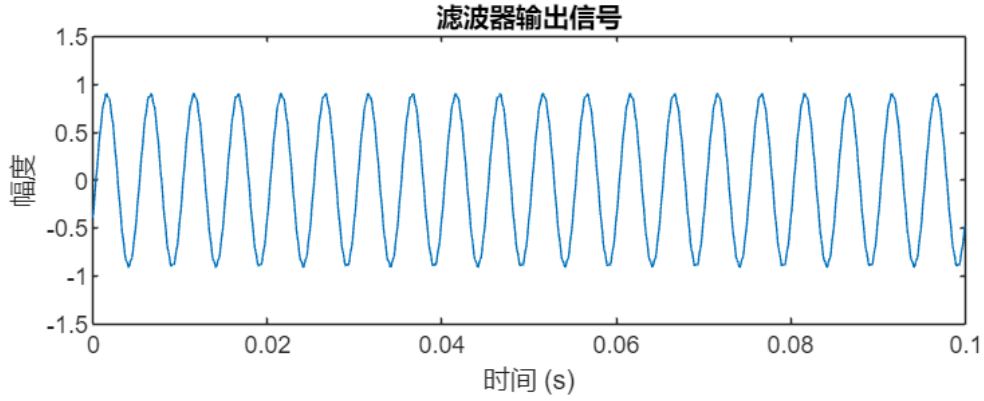


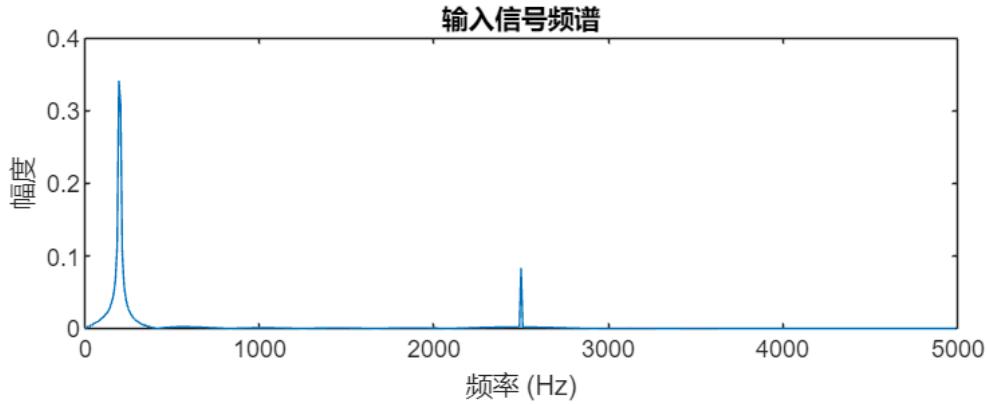


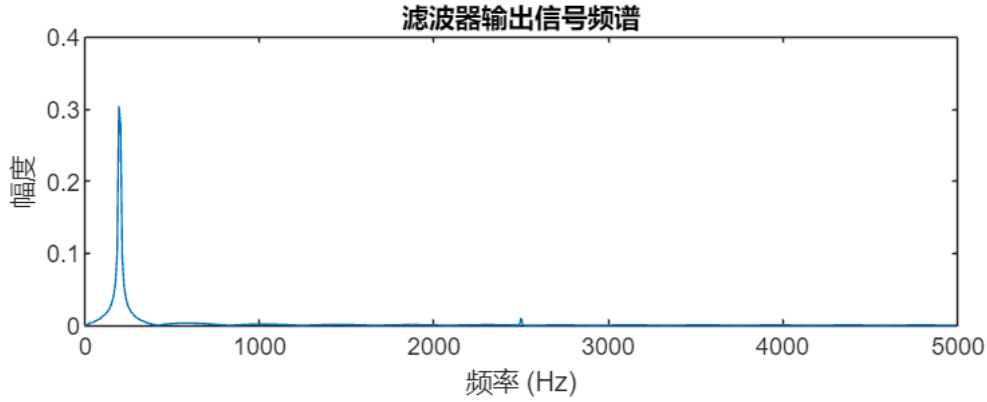



```
fStepSignal2 = 1.5708
```

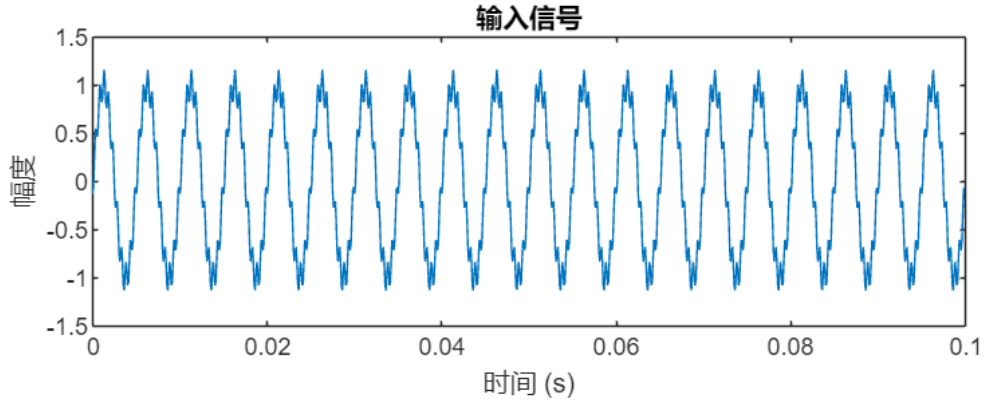


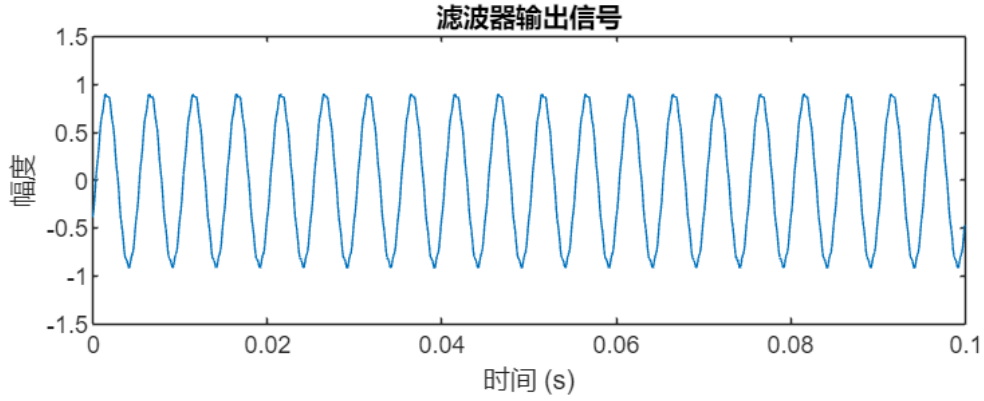


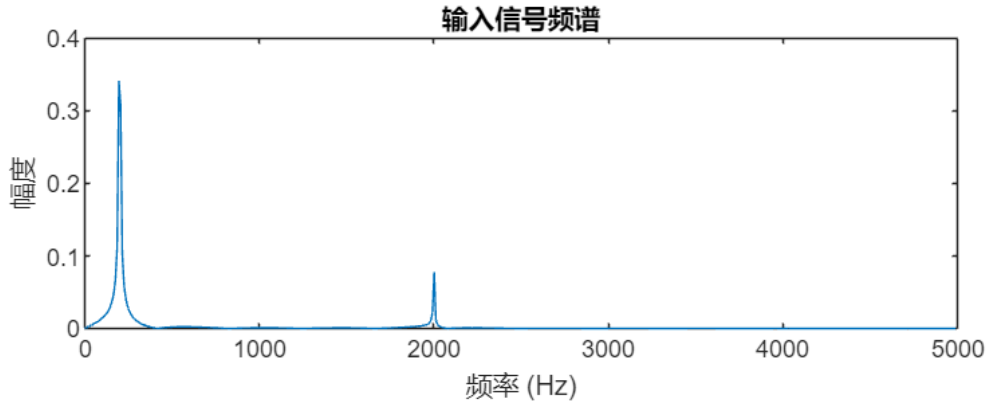


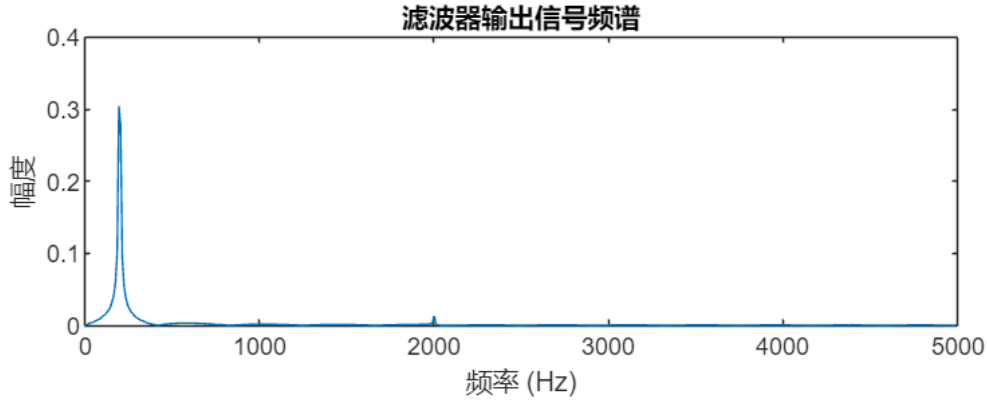



```
fStepSignal2 = 1.2566
```

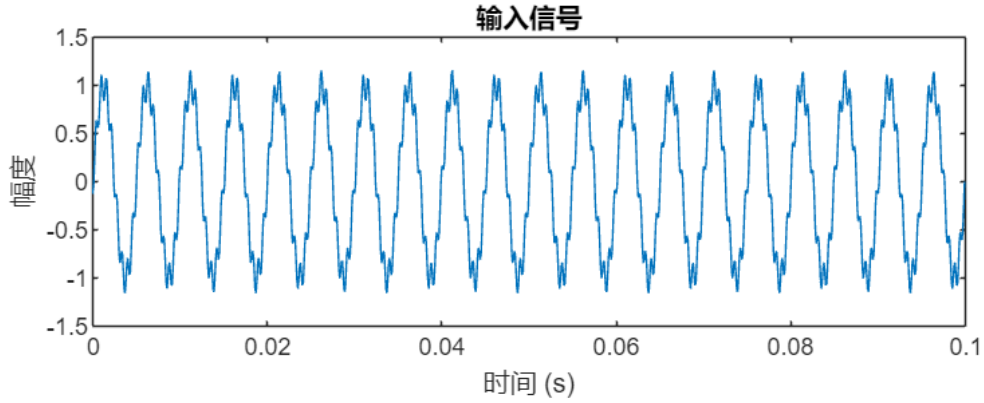


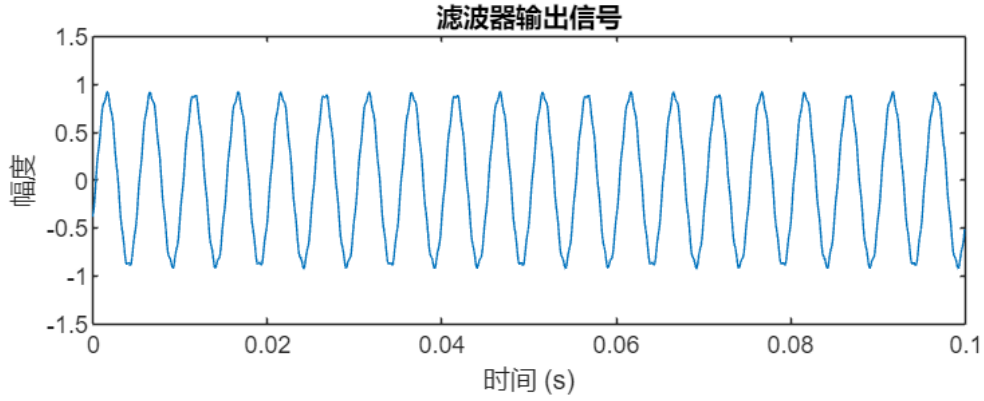


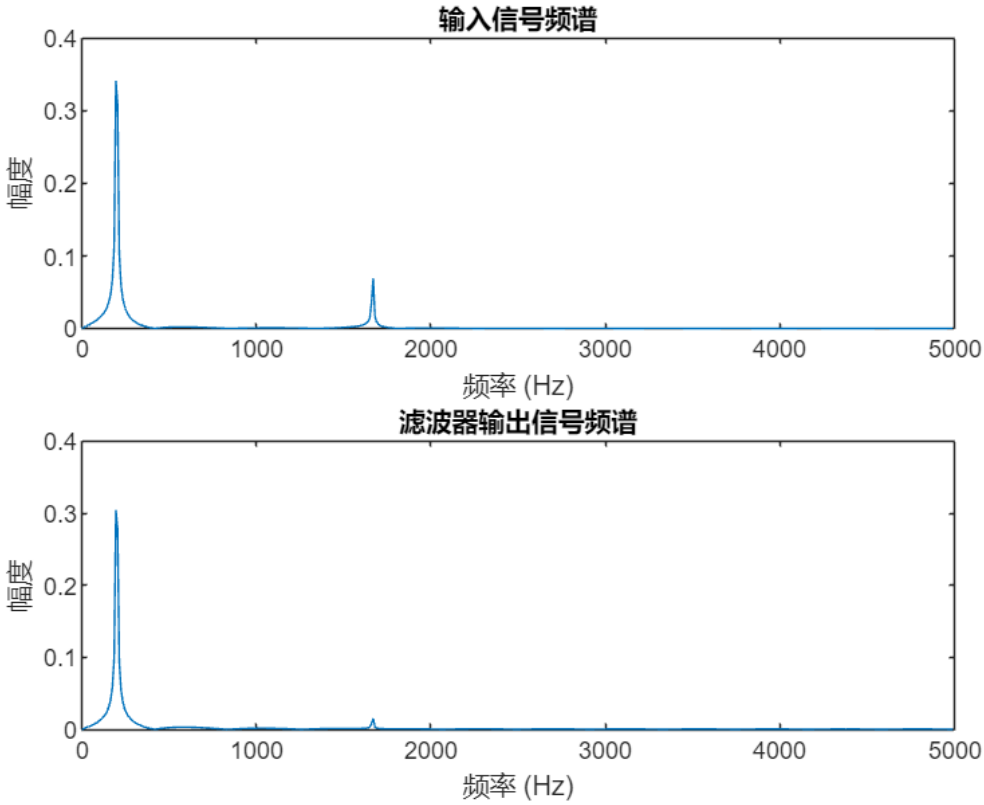





```
fStepSignal2 = 1.0472
```






输入信号频谱
0.4
0.3
0.2
0.1
0
幅度
0
1000
2000
3000
4000
5000
频率 (Hz)
滤波器输出信号频谱
0.4
0.3
0.2
0.1
0
幅度
0
1000
2000
3000
4000
5000
频率 (Hz)