

# DSP实验报告(一)

18373038 钱思远

## 一、实验要求

一、有一个模拟频率为  $f_0 = 3.5\text{kHz}$  的单频正弦信号， $x(t) = \cos(2\pi f_0 t)$ 。现以采样率为  $f_s = 8125\text{Hz}$  进行采样，得到

$$x[n] = x(t)|_{nT}, \quad n = 0, 1, \dots, 80000$$

对上面采样的信号隔点抽取，形成一新的序列，

$$y[n] = x[2n]$$

现以采样率为2倍的  $f_s$  进行采样，得到

$$x[n] = x(t)|_{nT}, \quad n = 0, 1, \dots, 80000$$

1.使用MATLAB自带的播放器进行播放。(注：sound(x,fs))，根据你所听到的三种不同的声音，你是如何解释的。

2.根据内插公式（选择合适的  $M$ ）：

$$x(t) = \sum_{n=-M}^M y[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}$$

对  $x(t)$  重新以  $f_s = 8\text{kHz}$  采样得到  $z[n], n = 0, 1, \dots, 80000$ 。使用MATLAB自带的播放器进行播放，听听该声音。并统计误差

$$E = \frac{1}{80000} \sum_{n=0}^{79999} (z[n] - x[n])^2$$

二、有一个模拟频率为  $f_0 = 3.5\text{kHz}$  和  $f_1 = 3.8\text{kHz}$  的双频正弦信号， $x(t) = \cos(2\pi f_0 t) + \cos(2\pi f_1 t)$ 。现以采样率为  $f_s = 8125\text{Hz}$  进行采样，得到

$$x[n] = x(t)|_{nT}, \quad n = 0, 1, \dots, 80000$$

设计一滤波器，提取出频率为  $f_0 = 3.5\text{kHz}$  的正弦信号。要求对  $f_1 = 3.8\text{kHz}$  的

正弦信号具有不低于  $50dB$  的抑制。将  $x[n]$  通过该滤波器获得输出  $y[n]$ 。

使用MATLAB自带的播放器进行播放，听听  $x[n]$  和  $y[n]$  的声音。并统计  $y[n]$ （考虑延迟）与  $x_d[n] = \cos(2\pi f_0 nT)$  的误差。

## 二、实验结果及分析

### (1)实验要求一

#### 1. 生成单频正弦信号并采样得到三个新序列

我们通过Matlab生成了模拟频率为  $f_0 = 3.5kHz$  的单频正弦信号，分别对单频正弦信号进行采样频率为8125Hz、4062.5Hz、16250Hz的采样。这一部分的代码如下：

```
f0=3500;%信号频率
fs=8125;%采样频率
n = 1:1:80001;
x_n = cos(2*pi*3500*(n-1)*1/fs);%针对x(t)取得的离散信号
y_n = zeros(1,40001);
for i = 0:1:40000
    y_n(i+1) = x_n(i*2+1);%对40001个数进行更新
end
x_n_2 = cos(2*pi*3500*(n-1)*0.5/fs);
```

其中 $x\_n$ 为8125Hz采样频率的序列， $y\_n$ 为 $x\_n$ 基础上隔点抽取得到的序列， $x\_n\_2$ 为两倍采样频率得到的序列。

#### 2.对三段序列进行声音播放

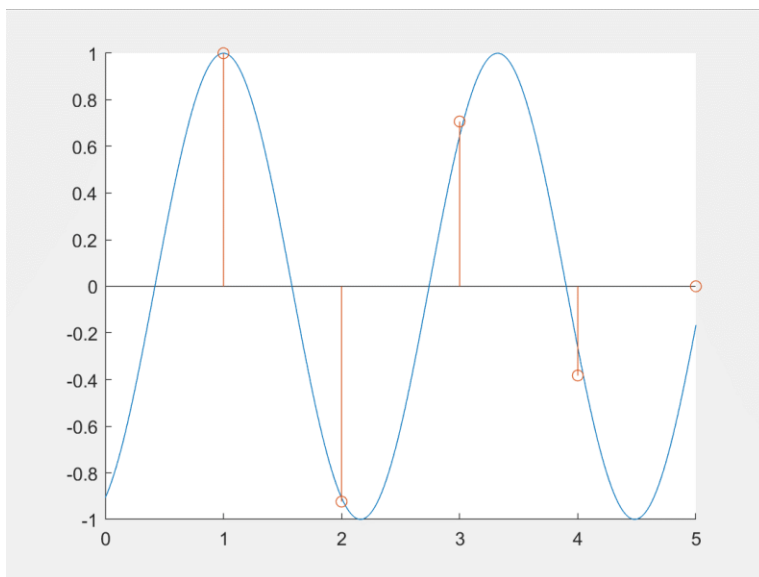
```
sound(x_n,fs); %x[n]声音播放
sound(y_n,fs/2); %y[n]声音播放
sound(x_n_2,2*fs); %2倍采样频率的x[n]声音播放
```

可以听出采样频率越高，声音越尖锐。这也符合我们对声音音调与频率关系的认知。

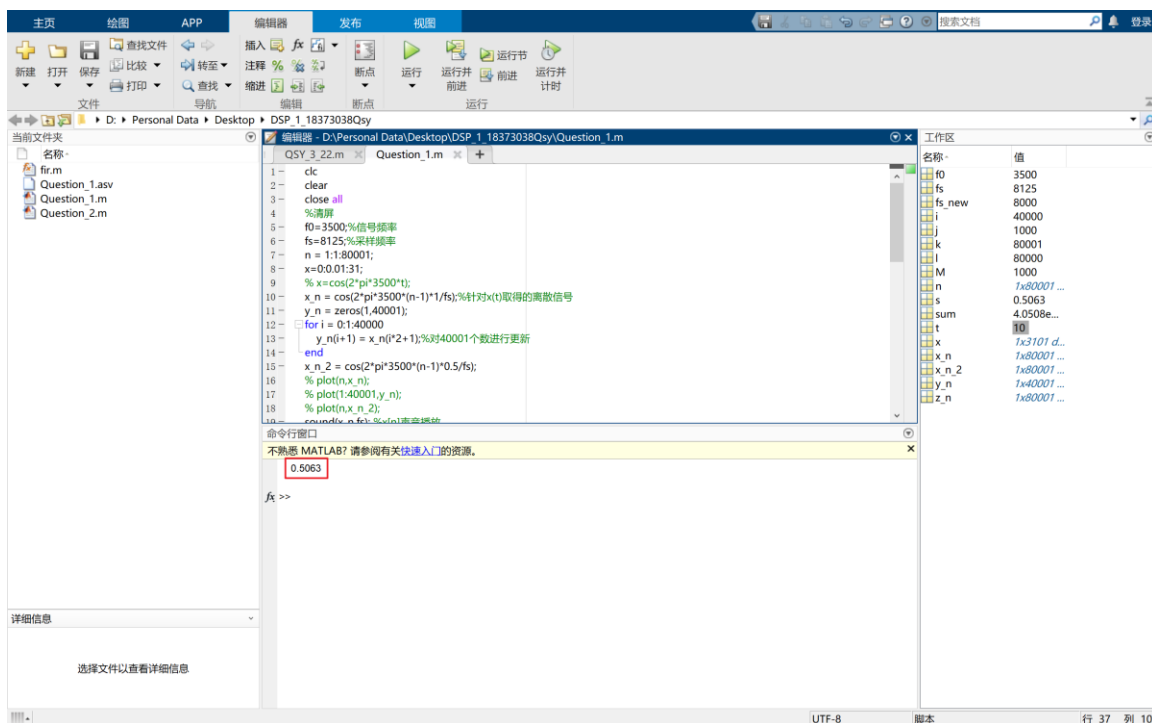
#### 3.通过内插公式重构并采样得到新序列 $z$ ，并统计与原序列相比的误差

而通过内插公式  $x(t) = \sum_{n=-M}^M y[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}$  对 $x(t)$ 进行重构，重构

后的波形如图所示：



可以看出 $x(t)$ 重构的效果较为理想。对重新采样取得的 $z$ 序列进行播放，并通过数据统计计算 $z$ 序列与 $x$ 序列的误差  $E = \frac{1}{80000} \sum_{n=0}^{79999} (z[n] - x[n])^2$ ，最终得到的误差值为：0.5063。



## (2)实验要求二

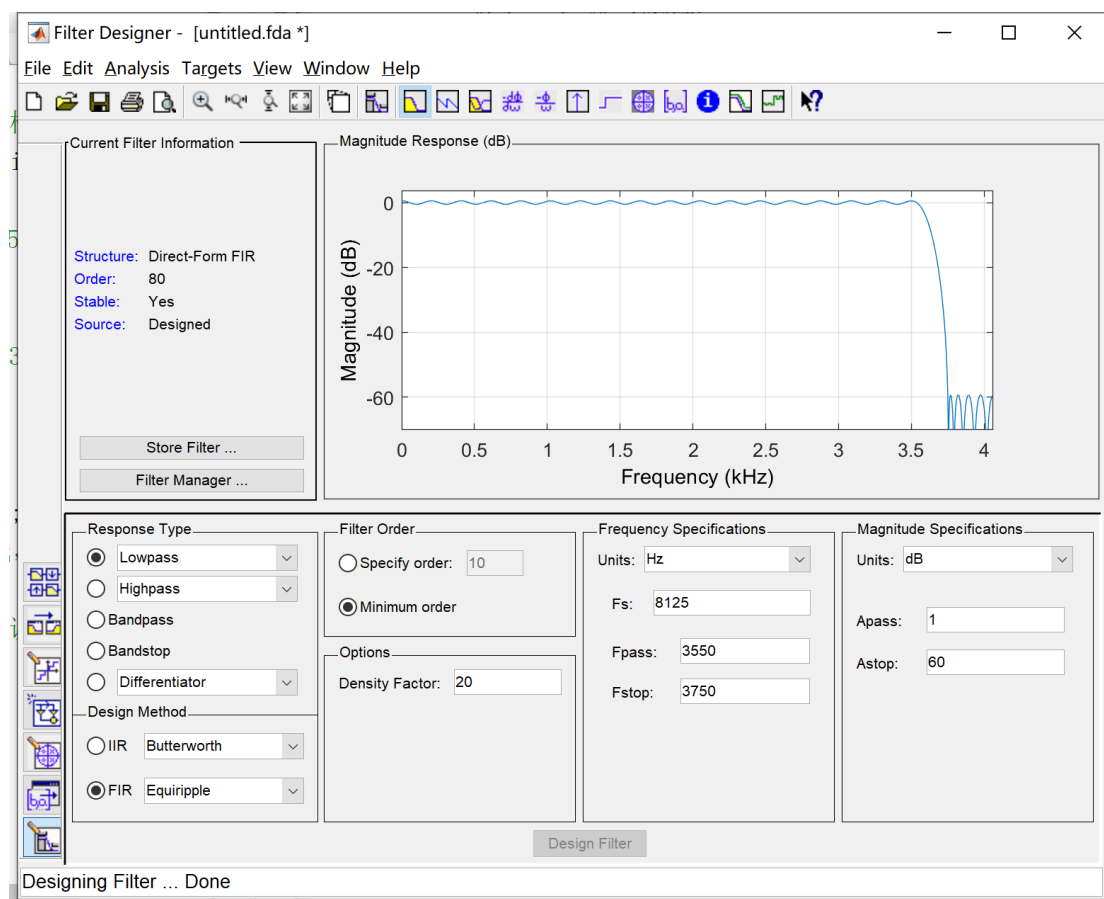
### 1.构建双频信号

在程序中设定两不同频率变量，通过两不同频率构建正弦信号并相加得到题目要求的双频正弦信号。

```
f0 = 3500;  
f1 = 3800;  
t = 0:0.001:1; %时间范围  
x_t = cos(2*pi*f0*t)+cos(2*pi*f1*t); %双频正弦信号表达式
```

### 2.设计滤波器

在命令行输入fdatool命令打开滤波器设计界面。采用低通椭圆滤波器，采样频率Fs设置为8125Hz，Fpass设置为3550Hz，Fstop设置为3750Hz，滤波器幅度响应如图。



将其导出生成.m文件，在主程序中调用滤波器。代码截图如下：

```

1  function Hd = fir
2      %FIR 返回离散时间滤波器对象。
3
4      % MATLAB Code
5      % Generated by MATLAB(R) 9.8 and Signal Processing Toolbox 8.4.
6      % Generated on: 20-Dec-2020 13:43:19
7
8      % FIR Window Lowpass filter designed using the FIR1 function.
9
10     % All frequency values are in Hz.
11     Fs = 8125; % Sampling Frequency
12
13     Fpass = 3700;      % Passband Frequency
14     Fstop = 3800;      % Stopband Frequency
15     Dpass = 0.057501127785; % Passband Ripple
16     Dstop = 0.001;     % Stopband Attenuation
17     flag = 'scale';    % Sampling Flag
18
19     % Calculate the order from the parameters using KAISERORD.
20     [N,Wn,BETA,TYPE] = kaiserord([Fpass Fstop]/(Fs/2), [1 0], [Dstop Dpass]);
21
22     % Calculate the coefficients using the FIR1 function.
23     b = fir1(N, Wn, TYPE, kaiser(N+1, BETA), flag);
24     Hd = dfilt.dfir(b);
25
26     % [EOF]
27

```

### 3.播放声音

针对模拟信号 $x(t)$ 采样后得到离散序列 $x_n$ ，将 $x_n$ 通过滤波器处理得到新的序列 $y_n$ ，播放两序列的声音。代码如下：

```

fs=8125;      %采样频率:8125Hz
N=80000;     %定义采样点数
dt=1/fs;     %采样间隔
T=(0:N)*dt;  %定义采样的每个时间点
x_n = cos(2*pi*f0*T)+cos(2*pi*f1*T);
x_n_1 = cos(2*pi*f0*T);
T_delay_1=(1:N)*dt; %考虑延迟1个dt
x_n_2 = cos(2*pi*f0*T_delay_1);
sound(x_n,fs); %x_n声音

Hd=fir;      %调用滤波器:Kaiser窗滤波器
y_n=filter(Hd,x_n); %y_n为对x_n的滤波结果
sound(y_n,fs); %y_n声音

```

### 4.分别考虑延迟与不考虑延迟两种情况统计序列误差

这部分的代码设计如下：

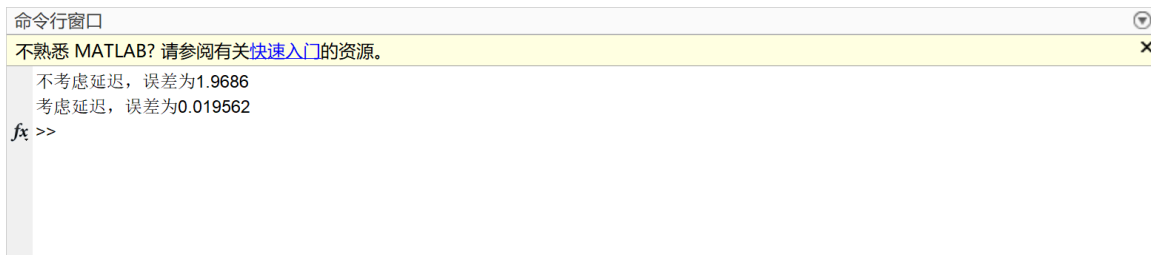
```

%不考虑延迟的情况
sum = 0;
for i = 1:N
    sum = sum+(x_n_1(i)-y_n(i))^2;
end
s = sum/N;
disp(strcat(['不考虑延迟, 误差为' num2str(s)])); %展示计算出来的误差

%考虑延迟为1个采样周期计算误差
sum = 0;
for i = 1:N
    sum = sum+(x_n_2(i)-y_n(i))^2;
end
s = sum/N; %统计y[n]与xd[n]之间的误差
disp(strcat(['考虑延迟, 误差为' num2str(s)])); %展示计算出来的误差

```

运行程序，两种情况下的误差皆可以通过命令行查看：



```

命令窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。
不考虑延迟, 误差为1.9686
考虑延迟, 误差为0.019562
fx >>

```

得到结论：考虑延迟后的误差较小，我们的滤波器设计合理，且针对延迟理论的分析是正确的。

### 三、实验总结

本次实验中，我通过对Matlab官方文件的阅读基本掌握了Matlab的使用方式。以此为前提，我针对实验要求进行了实验内容的实现，在此过程中，我学会了如何针对模拟信号进行采样，并针对采样序列进行声音播放。对内插公式的应用和滤波器的设计也让我进一步掌握了课堂知识的理论与其应用场景，从而更为深刻的理解了信号的采样与通过采样信号对原信号进行恢复的这一过程。