数字信号处理课程
小论文

班　　级：NULL

学　　号：NULL

姓　　名：GT_Shenmi

指导教师：NULL

2020 年　12 月　　1　日

基于 MATLAB 的语音信号采集处理及滤波器的设计

摘要：

随着数字信号处理技术的发展，语音信号的采集与处理系统越来越受到人们的关注。本设计是一种基于 MALAB 的，借助数字信号处理的分析方法，建立了一套语音信号采集与处理系统，能够完成对语音信号的采集与滤波处理。

关键词：FFT;频谱分析;音频信号;滤波器；

目录

1. 引言

随着数字信号处理技术的发展，语音信号的采集与处理系统越来越受到人们的关注。语音信号的采集和处理对人类来说是一个巨大的进步，相比于模拟信号，数字语音信号的处理更加灵活、方便[1]。本设计运用数字信号处理的分析方法，设计了一套语音信号处理系统，能够对采集到的语音信号频谱进行分析，并进行加噪和滤波处理。

2. 语音信号的采集

将录制的音频信号通过 Matlab 的 audioread 函数进行采样，得到采样序列 ft 以及采样频率 Fs，具体实现代码如图 2.1 所示。

```
133     function [ft,Fs] = SignalSampling(filename)
134 —       starttime = 5;
135 —       endtime = 25;
136 —       [Y,Fs] = audioread(filename);
137 —       ft = Y((Fs * starttime + 1):Fs * endtime,1);
138 —   end
```

通过 matplot 绘制音频采样信号的时域波形，如图 2.2 所示。

图 2.1 语音信号采集 MATLAB 代码

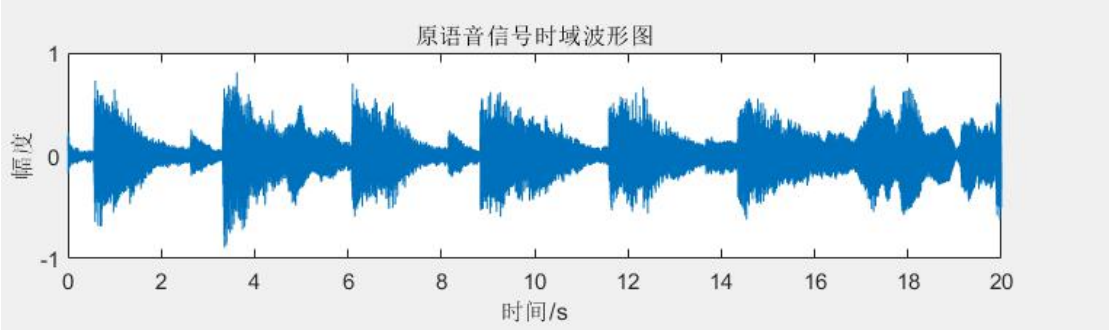

图 2.2 原语音信号时域波形图

改变采样频率后，发现当采样频率是原采样频率时，听到的声音是未失真的声音；当降低采样频率时，采样信号的频谱中包含大量低频分量，听到的声音显得比原语音信号要低沉；当增加采样频率时，采样信号的频谱中包含大量高频分量，声音会变得更加尖锐。

3. 语音信号的频谱分析

采样序列 ft 输入到 fft 函数中进行快速傅里叶变换，得到频域序列 Fejw。

```
function [Fejw,f] = SpectrumAnalysis(ft,Fs)
    N = length(ft);
    f = (0:N - 1) * Fs/N;
    Fejw = fft(ft,N);
end
```

通过 matplot 函数绘制音频采样信号的频域波形，如图 3.2 所示。

图 3.1 快速傅里叶变换函数

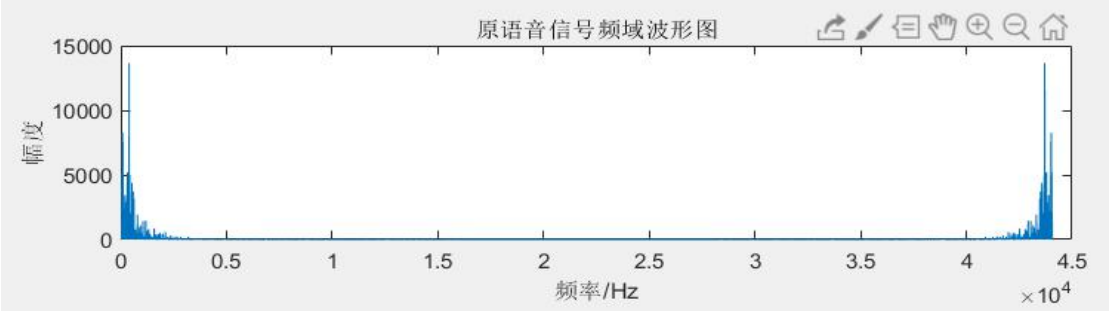

图 3.2 原语音信号频谱图

由图 3.2 可知，原语音信号的频率分量主要集中在 0 ~ 2kHz，2kHz 以上的频率分量近乎为 0。

4. 语音信号的加噪处理

4.1 余弦噪声

对音频信号在时域加入幅值为 0.05，频率为 2.5kHz 的余弦噪声：

```
function Fejw = AddCosNoise(ft,Fs)
    f = 2500;
    A = 0.05;
    t = [0:0.0001:0.1];

    N = length(ft);
    T = (0:N-1) / Fs;
    noise1 = A * cos(2 * pi * f * T);
    A = 0.05;
    noise2 = A * cos(2 * pi * f * T);
    noise1 = noise1';
    noise2 = noise2';
    noise1 = [noise1, noise2];
    Fejw = ft + noise1 ;
end
```

图 4.1 叠加余弦噪声函数

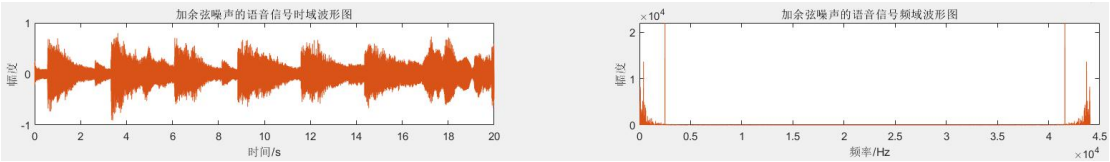

图 4.2 加余弦噪声后的语音信号时域和频域波形图

由图 4.2 可知，加入余弦噪声的语音信号频谱在频率为 2.5k（余弦噪声的频率）处出现了脉冲分量。播放加噪后的音频，可以听到很尖锐的声音，几乎听不到原来的语音。分析其原因，是因为加入噪声的频率分量为 2.5kHz，而原语音信号的频率分量分布在 2KHz 以下，小于 2.5kHz,所以会出现明显的尖锐的声音，但几乎听不到原语音是因为余弦噪声频率分量的幅值远高于原语音信号各频率分量的幅值。结合原语音信号的频谱特征，可以运用低通滤波器滤除余弦噪声的频谱分量。

4.2 随机白噪声

对采样音频信号加入幅值为 0.35 随机白噪声：

```
163     function Fejw = AddRandNoise(ft)
164 —       noise = 0.35 * randn(1,length(ft));
165 —       noise = noise';
166 —       noise = [noise,noise];
167 —       Fejw = ft + noise;
168 —   end
```

图 4.3 叠加随机白噪声代码

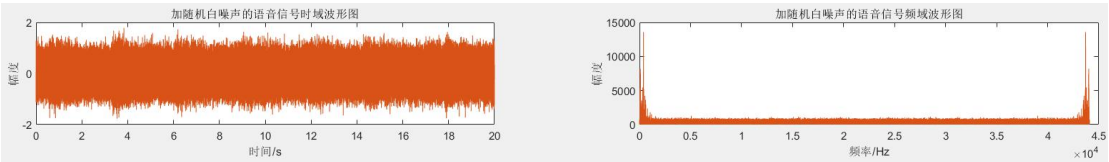

图 4.4 加白噪声后的语音信号时域和频域波形图

由图 4.4 可知，加入白噪声的语音信号频谱，在原频谱的基础上，均匀叠加了幅度相同的各个频率分量。播放加噪后的音频，可以听到“滋滋”的声音，也能听到原来的语音。分析其原因，是因为加入噪声为白噪声，其频谱中含有幅值相当的各个频率分量，所以会出现“滋滋”的声音，但由于其幅值不高，叠加在原语音信号上并不能完全覆盖原语音，所以还能听到原语音。结合原语音信号的频谱特征，仍可以运用低通滤波器滤除白噪声高于 2KHz 的频率分量。

5 加噪语音信号的噪声滤除

5.1 滤波器的设计

5.1.1 IIR 滤波器的设计

用双线性变换法设计 IIR 低通滤波器：通带波动 3db，阻带衰减 20db。

```
function [D,C] = IIR_LowPassFilter(fp,fs,Fs)
    Wp = 2 * pi * fp / Fs;
    Ws = 2 * pi * fs/ Fs;
    Ap = 3;
    As = 20;
    Fs = Fs/Fs;
    wap = tan(Wp/2);
    was = tan(Ws/2);

    [N,Wn] = buttord(wap,was,Ap,As,'s');
    [B,A] = butter(N,Wn,'s');

    [D,C] = bilinear(B,A,Fs);

end
```

图 5.1 IIR 滤波器设计代码

其幅频特性和相频特性如下：

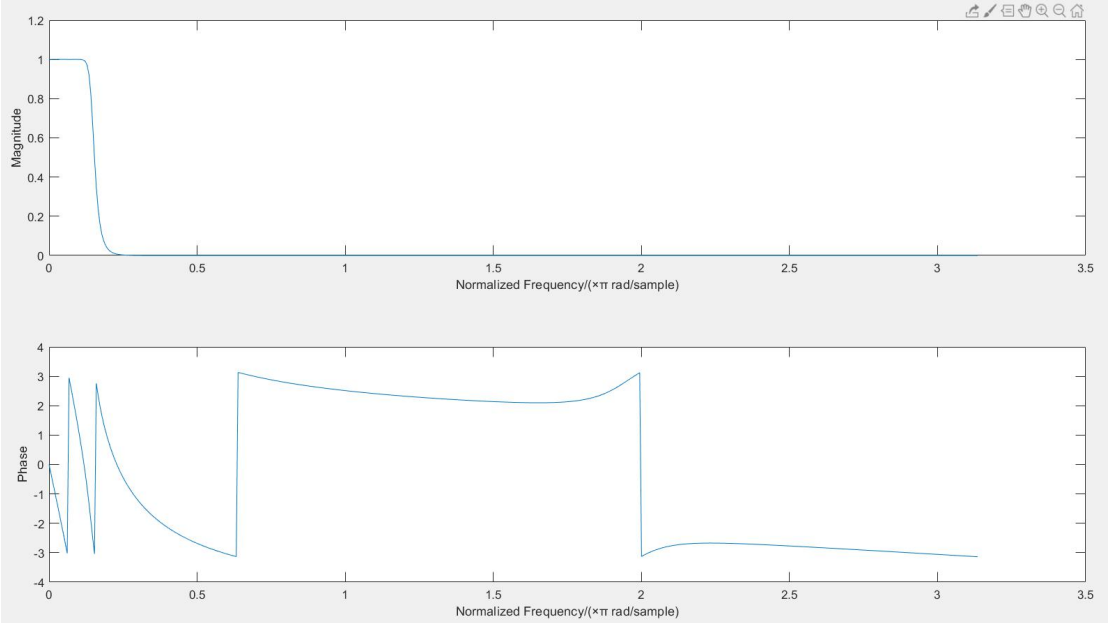

图 5.2 IIR 滤波器的幅频特性和相频特性(归一化)

5.1.2 FIR 滤波器

用汉宁窗设计 40 阶 FIR 低通滤波器：

```
186     function [H] = FIR_LowPassFilter(fp,fs,Fs)
187         %n = 40;
188 —       Wp =  2 * pi * fp;
189 —       Ws =  2 * pi * fs;
190 —       n = ceil(6.6 * pi * (Ws - Wp));
191 —       Wn = 0.5 * (Wp + Ws)/(Fs * 2 * pi);
192 —       H = fir1(n,Wn,'low',hann(n+1));
193         %freqz(H,1,512);
194 —   end
```

图 5.3 FIR 滤波器的设计代码

其幅频特性和相频特性如下：

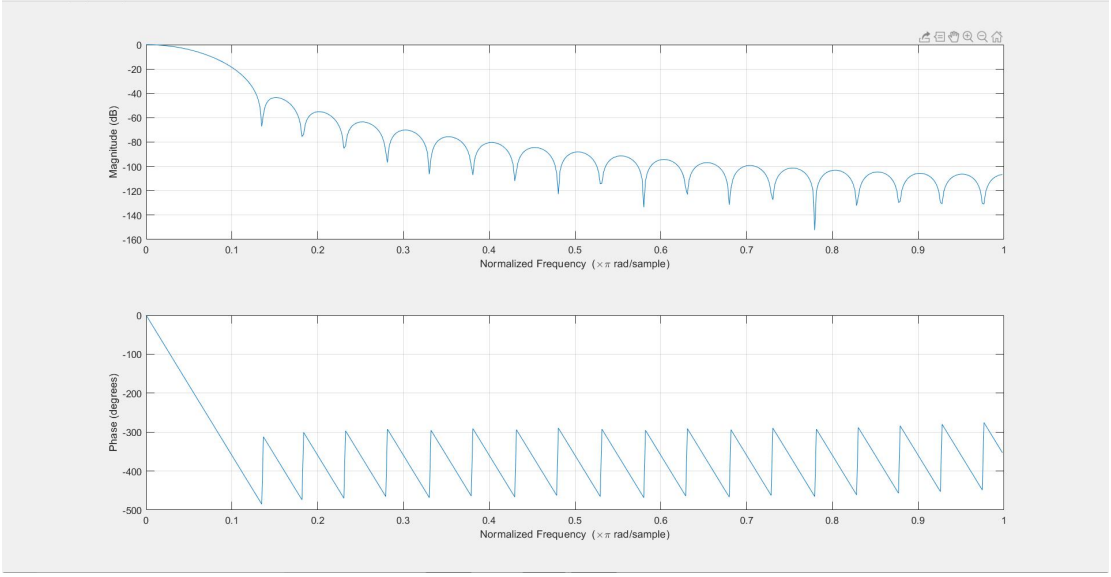

图 5.4 FIR 滤波器的幅频特性和相频特性(归一化)

5.2 加噪语音信号的滤波

5.2.1 滤除余弦噪声

用 5.1 设计的 IIR_LowPassilter 和 FIR_LowPassFilter 函数生成通带上限频率 fp = 2KHz，阻带下限频率为 2500Hz 的滤波器，滤除余弦噪声。

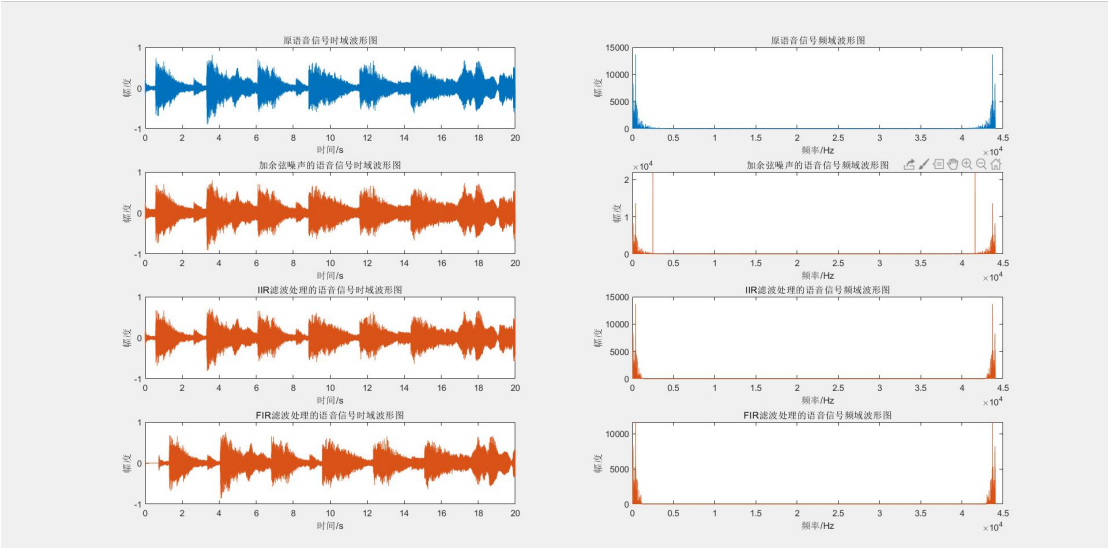

图 5.5 滤除余弦噪声的语音信号和原语音信号的时域频域对比

如图 5.5 所示，加余弦噪声的语音信号，频谱上出现了 2.5kHz 的频谱分量，时域波形开始出现变形，经过 FIR 和 IIR 滤波器后，该频率分量被滤除，其时域波形恢复。播放恢复后的语音信号，发现刺耳的声音被消除，原因是频率为 2.5KHz 的余弦噪声被滤除，剩下的都是原信号的频率分量，所以恢复成了原语音信号。

5.2.2 滤除随机白噪声

用 5.1 设计的 IIR_LowPassilter 和 FIR_LowPassFilter 函数生成通带上限频率 fp = 2KHz，阻带下限频率为 2500Hz 的滤波器，滤除随机白噪声。

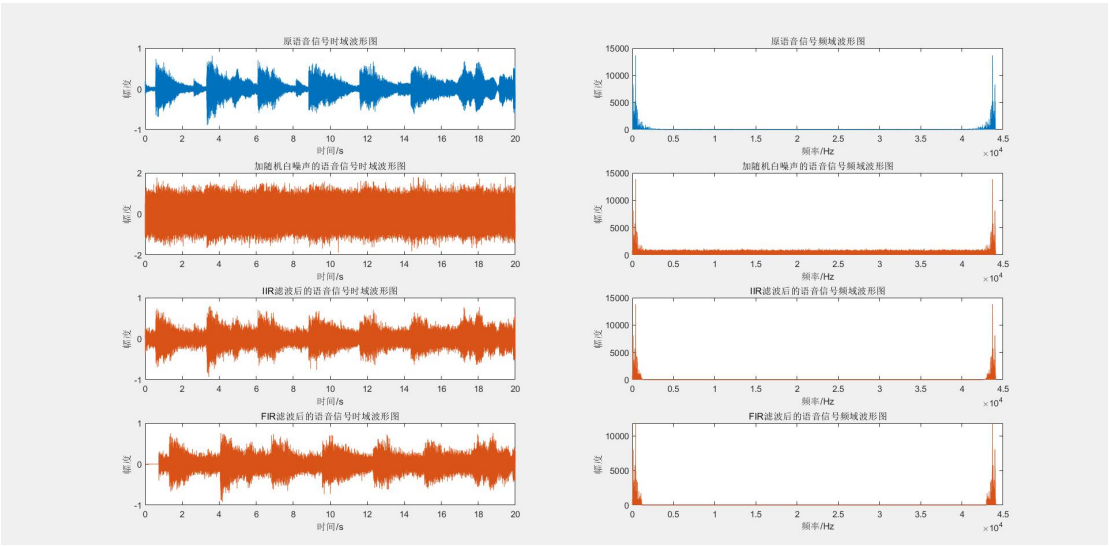

图 5.6 滤除白噪声的语音信号和原语音信号的时域频域对比

如图 5.6 所示，加白噪声的语音信号频谱，在原语音信号的频谱上均匀叠加了各个频率分量，时域波形开始出现变形，经过 FIR 和 IIR 滤波器后，该频谱高于 2kHz 的频率分量被滤除，其时域波形近似恢复。播放恢复后的语音信号，发现“滋滋”的声音减轻，并没有完全消除，原因是低通滤波器可以滤除高于 2kHz 的频率分量，但无法滤除掉 0-2KHz 的白噪声频谱分量，所以还会有轻微的“滋滋”声。

6.结论

对加入余弦噪声和白噪声的语音信号，所设计的 IIR 滤波器和 FIR 滤波器能达到指标要求，能够基本恢复原语音信号，滤波效果较好。

总的来说，IIR 滤波器的滤波效果比较好，FIR 滤波器的效果与 IIR 相比，还有差距，可以通过增大滤波器的阶数来改善滤波效果。

参考文献

[1] 李静.基于MATLAB的语音信号采集和处理系统的设计[J].山西大同大学学报(自然科学版),2016,32(02):30-33.

附录

源程序代码：

```
Fs = 10000;

Duration = 20;

n = Duration * Fs;

t = (1:n)/Fs;

samples = [1,n];

filename = "1.mp3";

[ft1,Fs1] = SignalSampling(filename);

figure(1);

subplot(4,2,1);

nft = length(ft1);


time  = (0:(nft-1))/Fs1;


plot(time,ft1);

title("原语音信号时域波形图");

xlabel('时间/s');

ylabel('幅度');


[Fejw2,Fs2] = SpectrumAnalysis(ft1,Fs1);


subplot(4,2,2);

plot(Fs2,abs(Fejw2));

title("原语音信号频域波形图");

xlabel('频率/Hz');

ylabel('幅度');


ftn1 = AddCosNoise(ft1,Fs1);
```

```
[Fejw3,Fs3] = SpectrumAnalysis(ftn1,Fs1);
subplot(4,2,3);
plot(time,ftn1);
title("加余弦噪声的语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,4);
plot(Fs3,abs(Fejw3));
title("加余弦噪声的语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

[D,C] = IIR_LowPassFilter(2000,2500,Fs1);
y = filter(D,C,ftn1);
[Fejw4,Fs4] = SpectrumAnalysis(y,Fs1);

subplot(4,2,5);
plot(time,y);
title("IIR 滤波处理的语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,6);
plot(Fs4,abs(Fejw4));
title("IIR 滤波处理的语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

H = FIR_LowPassFilter(2000,2500,Fs1);
```

```
y = filter(H,1,ftn1);
[Fejw5,Fs5] = SpectrumAnalysis(y,Fs1);

subplot(4,2,7);
plot(time,y);
title("FIR 滤波处理的语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,8);
plot(Fs5,abs(Fejw5));
title("FIR 滤波处理的语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

figure(2);
subplot(4,2,1);
plot(time,ft1);
title("原语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,2);
plot(Fs2,abs(Fejw2));
title("原语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

ftn2 = AddRandNoise(ft1);
[Fejw6,Fs6] = SpectrumAnalysis(ftn2,Fs1);
```

```
subplot(4,2,3);
plot(time,ftn2)
title("加随机白噪声的语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,4);
plot(Fs6,abs(Fejw6));
title("加随机白噪声的语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

[D,C] = IIR_LowPassFilter(2000,2500,Fs1);
y = filter(D,C,ftn2);
[Fejw7,Fs7] = SpectrumAnalysis(y,Fs1);
subplot(4,2,5);
plot(time,y);
title("IIR 滤波后的语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,6);
plot(Fs7,abs(Fejw7));
title("IIR 滤波后的语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

H = FIR_LowPassFilter(2000,2500,Fs1);
y = filter(H,1,ftn2);
[Fejw8,Fs8] = SpectrumAnalysis(y,Fs1);
```

```
subplot(4,2,7);
plot(time,y);
title("FIR 滤波后的语音信号时域波形图");
xlabel('时间/s');
ylabel('幅度');

subplot(4,2,8);
plot(Fs8,abs(Fejw8));
title("FIR 滤波后的语音信号频域波形图");
xlabel('频率/Hz');
ylabel('幅度');

function [ft,Fs] = SignalSampling(filename)
    starttime = 5;
    endtime = 25;
    [Y,Fs] = audioread(filename);
    ft = Y((Fs * starttime + 1):Fs * endtime,1);
end

function [Fejw,f] = SpectrumAnalysis(ft,Fs)
    N = length(ft);
    f = (0:N - 1) * Fs/N;
    Fejw = fft(ft,N);
end

function Fejw = AddCosNoise(ft,Fs)
    f = 2500;
    A = 0.05;
    t = [0:0.0001:0.1];
```

```
    N = length(ft);
    T = (0:N-1) / Fs;
    noise1 = A * cos(2 * pi * f * T);
    A = 0.05;
    noise2 = A * cos(2 * pi * f * T);
    noise1 = noise1';
    noise2 = noise2';
    noise1 = [noise1, noise2];
    Fejw = ft + noise1 ;
end

function Fejw = AddRandNoise(ft)
    noise = 0.35 * randn(1,length(ft));
    noise = noise';
    noise = [noise,noise];
    Fejw = ft + noise;
end

function [D,C] = IIR_LowPassFilter(fp,fs,Fs)
    Wp = 2 * pi * fp / Fs;
    Ws = 2 * pi * fs/ Fs;
    Ap = 3;
    As = 20;
    Fs = Fs/Fs;
    wap = tan(Wp/2);
    was = tan(Ws/2);

    [N,Wn] = buttord(wap,was,Ap,As,'s');
    [B,A] = butter(N,Wn,'s');
```

```
    [D,C] = bilinear(B,A,Fs);
%     [B,A] = freqz(D,C);
%     figure(3);
%     subplot(2,1,1);
%     plot(A,abs(B));
%     xlabel('Normalized Frequency/(×π rad/sample)');
%     ylabel('Magnitude');
%     subplot(2,1,2);
%     plot(A,angle(B));
%     xlabel('Normalized Frequency/(×π rad/sample)');
%     ylabel('Phase');

end

function [H] = FIR_LowPassFilter(fp,fs,Fs)
    Wp =  2 * pi * fp;
    Ws =  2 * pi * fs;
     %n = ceil(6.6 * pi * (Ws - Wp));
     n = 40;
    Wn = 0.5 * (Wp + Ws)/(Fs * 2 * pi);
    H = fir1(n,Wn,'low',hann(n+1));
    %freqz(H,1,512);
end
```