

数字信号处理 第一周作业

范云潜 18373486

微电子学院 184111 班

日期: 2020 年 12 月 24 日

目录

1 实验目的	1
2 实验原理	1
3 实验步骤	1
3.1 窗函数长度对信号 DFT 的影响	1
3.2 频域采样对信号 DFT 的影响	2
3.3 语音信号频谱分析及滤波处理	2
4 实验代码	2
4.1 窗函数长度对信号 DFT 的影响	2
4.2 频域采样对信号 DFT 的影响	3
4.3 语音信号频谱分析及滤波处理	4
5 实验结果	6
5.1 窗函数长度对信号 DFT 的影响	6
5.2 频域采样对信号 DFT 的影响	7
5.3 语音信号频谱分析及滤波处理	8
6 结果分析	8
6.1 窗函数长度对信号 DFT 的影响	8
6.2 频域采样对信号 DFT 的影响	8
6.3 语音信号频谱分析及滤波处理	9

List of Figures

1	32 点窗函数对应的 DFT 频谱	6
2	64 点窗函数对应的 DFT 频谱	7
3	128 点窗函数对应的 DFT 频谱	7

4	相同窗函数不同采样点数对应的 DFT 频谱	8
5	原始音频 FFT 频谱	9
6	原始音频降采样 FFT 频谱	9
7	带阻滤波器的单位冲激响应	10
8	带阻滤波器的频率响应	10
9	处理后音频 FFT 频谱	11
10	处理后音频降采样 FFT 频谱	11

1 实验目的

通过本实验，加深对 DFT 的连续时域样本长度、样本采样数目等参数，加窗等处理对 DFT 效果影响的认识，并通过语音信号的处理，了解如何针对特定应用设计需要的滤波器的方法。

2 实验原理

DFT 的连续时域样本越大，对应的分辨率会降低；DFT 的连续时域样本的采样长度固定时，需要一定量的采样点完成整体的频谱分析，但是当采样点过多后之后只会造成计算量的上升，而不会有精度的益处。

FFT 的频谱和模拟频谱是对应的，可以通过下式计算：

$$\Omega = \frac{\omega}{T} = \frac{2\pi k}{NT}$$

]

3 实验步骤

3.1 窗函数长度对信号 DFT 的影响

首先，由于采样的总长度在变化，不妨设总采样点为一个特定值，此值需要大于 L_{max} 即可，因为后续用到了该长度的窗函数。

接下来对长度 L 和窗函数类型进行依次遍历，在不同长度的窗函数下需要对窗函数的序列进行补零，来保证可以进行时域的加窗。之后进行 L 点的 FFT 进行 DFT 分析，画图。

3.2 频域采样对信号 DFT 的影响

此时，采样的总长度是固定的，但是采样的点数不同，因此进行加窗时，需要补零到最多点数仍可进行正常加窗。之后进行不同点数的 FFT 进行 DFT 分析，画图。

3.3 语音信号频谱分析及滤波处理

将音频读取，得到其序列与相应的采样率，对输入序列进行 FFT 得到其频谱表示，观察其干扰分量位置，并且找到对应的采样点数，计算得到对应的模拟频率。根据模拟频率设计一个带阻滤波器，将其去除。

4 实验代码

4.1 窗函数长度对信号 DFT 的影响

Listing 1: 窗函数长度与形状对 DFT 频谱的影响 task1.m

```
L = [32 64 128];
rec = 1;
ham = 2;
win_type = [rec, ham];
N = 256;

for len = 1:length(L)
    l_now = L(len);
    figure(len);
    for wt = 1:length(win_type)
        w_now = win_type(wt);

        % build the series
        n = 0:(l_now/N):l_now;
        n = n(1:length(n)-1);
        x = sin(3 * pi * n / 10) + 0.5 * sin(2 * pi * n / 5);

        % choose the window
        if w_now == rec
            win = rectwin(l_now);
            win_name = sprintf('rectangle window of %d', l_now);
        else
            win = hamming(l_now);
            win_name = sprintf('hamming window of %d', l_now);
        endif
        win = [win', zeros(1, length(n) - length(win))];
        % add the window to the signal
        y = x .* win;

        % do fft and show it
        Y = fft(y, l_now);
        Y = fftshift(Y);
        subplot(2,1,wt);
        stem(abs(Y), '.');
        title(win_name);
    end
end
```

```

    endfor
endfor

input('Press Enter to Continue...');
saveas(1,'./l3p1.png');
saveas(2,'./l3p2.png');
saveas(3,'./l3p3.png');
close all;

```

4.2 频域采样对信号 DFT 的影响

Listing 2: 频域采样点数对 DFT 频谱的影响 task2.m

```

L = 32;
N = [32 64 128 256 512];

figure(4);
for i = 1:length(N)
    subplot(length(N), 1, i);
    % figure(i);
    win_name = sprintf('hamming window and DFT with %d points', N(i));

    % build the series
    n = 0:(L/1024):L;
    n = n(1:length(n)-1);
    x = sin(3 * pi * n / 10) + 0.5 * sin(2 * pi * n / 5);

    % choose the window
    win = hamming(L);
    win = [win', zeros(1, length(n) - length(win))];
    % add the window to the signal
    y = x .* win;

    % do fft and show it
    Y = fft(y, N(i));
    Y = fftshift(Y);
    stem(abs(Y), '.');
    title(win_name);
endfor

input('Press Enter to Continue...');
saveas(4,'./l3p4.png');
close all;

```

4.3 语音信号频谱分析及滤波处理

Listing 3: 语音信号频谱分析及滤波器设计与处理 task3.m

```
[in_t, fs] = audioread ('./Test_music.wav');
smpls = length(in_t);
% fft the input audio
in_f = fft(in_t, smpls);
in_f = fftshift(in_f);
% in_f = fftshift(in_f);
f = ((1:length(in_f))-(length(in_f) / 2))/(length(in_f) / 2)*pi;

% plot the input audio's frequency
figure(5);
subplot(2,1,1);
plot(f, abs(in_f), '.-');
title("The abs of the audio\'s frequency");
subplot(2,1,2);
plot(f, angle(in_f), '.');
title("The angle of the audio\'s frequency");

% compute the white noise
in_f = fftshift(in_f);
check_f = abs(in_f(1:ceil(length(in_f)/2)));
[max_v, max_i] = max(check_f)
freq_white = 2 * pi * max_i / smpls / (1/fs) / 2 / pi;
sprintf("the original frequency of the audio\'s white noise: %f Hz\n", freq_
    white)

smpls = 10000;
% fft the input audio
in_f = fft(in_t, smpls);
% in_f = fftshift(in_f);
f = ((1:length(in_f))-(length(in_f) / 2))/(length(in_f) / 2)*pi;

% plot the input audio's frequency after down sampling
in_f = fftshift(in_f);
figure(6);
subplot(2,1,1);
plot(f, abs(in_f), '.-');
title("The abs of the audio\'s frequency: down sampling");
subplot(2,1,2);
plot(f, angle(in_f), '.-');
title("The angle of the audio\'s frequency: down sampling");

f_bands = [6.7, 7.1, 7.6, 8] * 1e3;
a = [1, 0, 1];
dev = [0.02, 0.1, 0.02];
```

```

fs = fs;
[N, Wn, beta, ftype] = kaiserord(f_bands, a, dev, fs);

win = kaiser(N+1, beta);
fcoeff = fir1(N, Wn, ftype, win);

figure(7);
stem(fcoeff);
title("Bandpass Filter: unit impulse response")

figure(8)
freqz(fcoeff, 1);
subplot(2,1,1)
title(sprintf('n = %d order Bandpass Filter: amplitude-frequency response',N))
subplot(2,1,2)
title(sprintf('n = %d order Bandpass Filter: phase-frequency response',N))

figure(9)
out_t = conv(in_t, fcoeff);
out_f = fft(out_t, length(out_t));
out_f = fftshift(out_f);
% out_f = fftshift(out_f);

% plot the output audio's frequency
f = ((1:length(out_f))-(length(out_f) / 2))/(length(out_f) / 2)*pi;
subplot(2,1,1);
plot(f, abs(out_f), '-.');
title("The abs of the processed audio\'s frequency");
subplot(2,1,2);
plot(f, angle(out_f), '-.');
title("The angle of the processed audio\'s frequency");

figure(10)
smpls = 10000;
% fft the input audio
out_f = fft(out_t, smpls);
out_f = fftshift(out_f);
% in_f = fftshift(in_f);
f = ((1:length(out_f))-(length(out_f) / 2))/(length(out_f) / 2)*pi;

subplot(2,1,1);
plot(f, abs(out_f), '-.');
title("The abs of the processed audio\'s frequency: down sampling");
subplot(2,1,2);
plot(f, angle(out_f), '-.');
title("The angle of the processed audio\'s frequency: down sampling");

```

```

input('Press Enter to Save Figures and Continue...');
saveas(5, './l3p5.png');
saveas(6, './l3p6.png');
saveas(7, './l3p7.png');
saveas(8, './l3p8.png');
saveas(9, './l3p9.png');
saveas(10, './l3p10.png');

close all;

```

5 实验结果

5.1 窗函数长度对信号 DFT 的影响

运行 task1.m，不同点数与形状对 DFT 的影响如图 1，图 2，图 3。

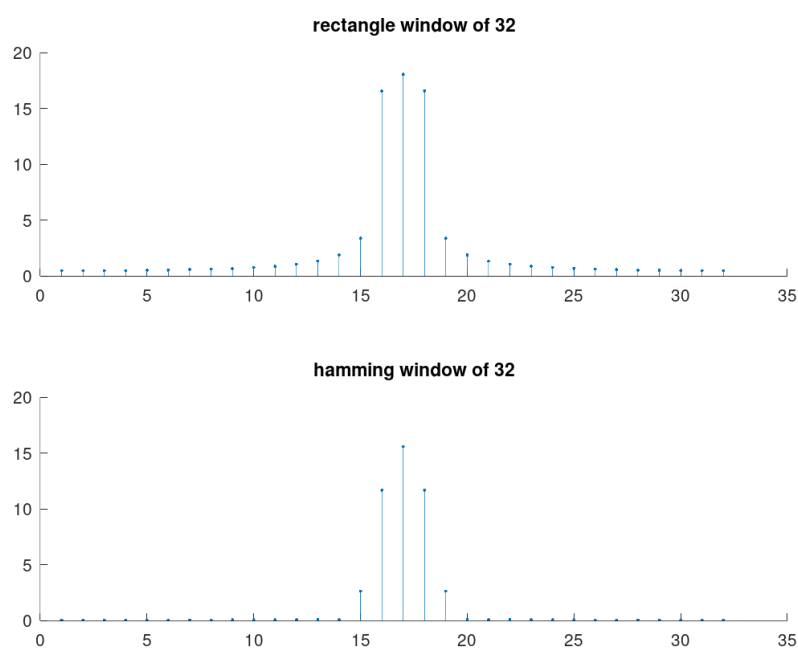


图 1: 32 点窗函数对应的 DFT 频谱

5.2 频域采样对信号 DFT 的影响

运行 task2.m，相同长度、形状窗函数下，不同采样点数对 DFT 的影响如图 4。

5.3 语音信号频谱分析及滤波处理

运行 task3.m，原始音频在序列长度的 FFT 下的频谱如图 5，计算其单频噪声在序列中的位置，推导出其模拟频率，计算输出为 the original frequency of the audio's white

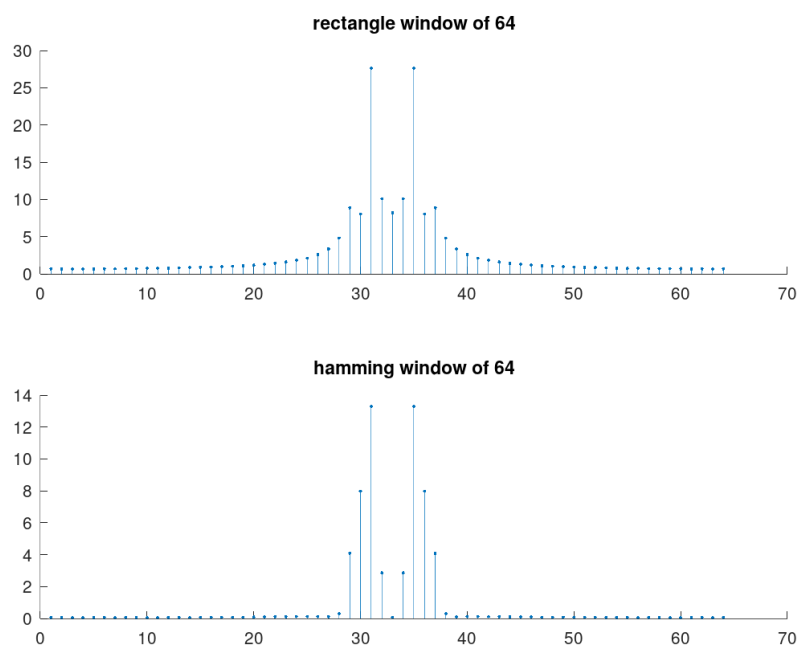


图 2: 64 点窗函数对应的 DFT 频谱

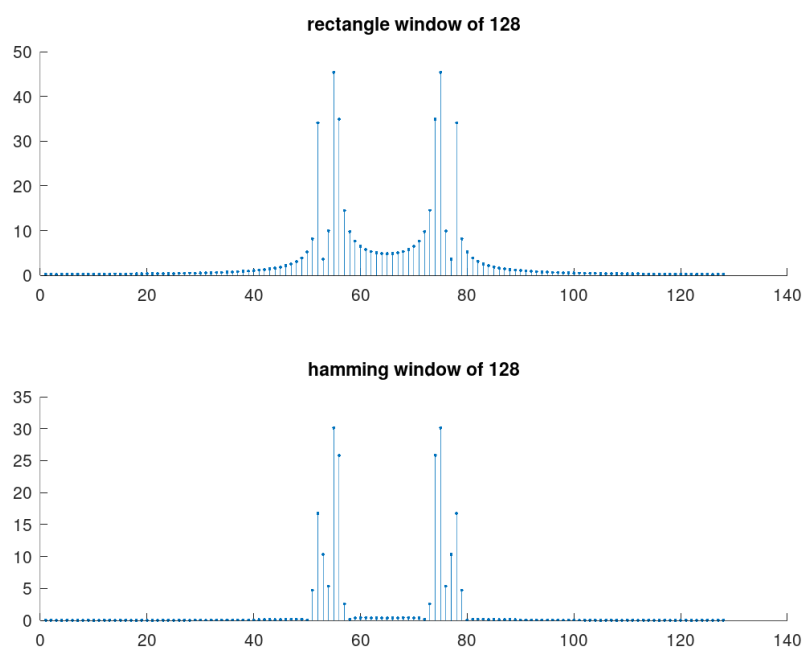


图 3: 128 点窗函数对应的 DFT 频谱

noise: 7350.200000 Hz。由于其幅频难以观察，降采样 FFT 后的频谱更易观察，如 图 6。通过设计带阻滤波器滤除此分量，本基于 Kaiser 窗的带阻滤波器的单位冲激响应如 图 7，其频率响应如 图 8，阶数为 80 阶。处理后的信号如 图 9，经过同样的降采样后，如 图 10，便于观察相位。在降采样中，计算得到该频点的幅度下降约 $46.986 = 33.4 \text{ db}$ 。

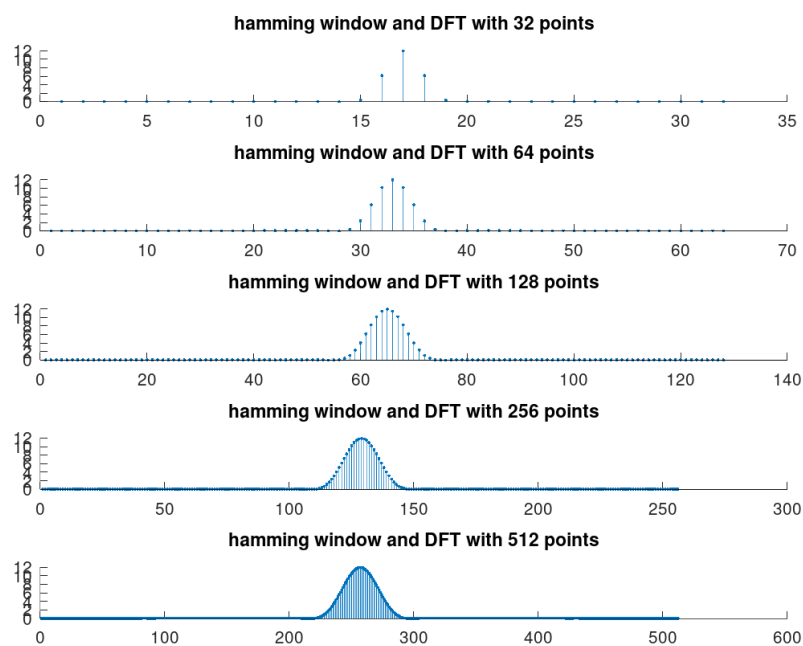


图 4: 相同窗函数不同采样点数对应的 DFT 频谱

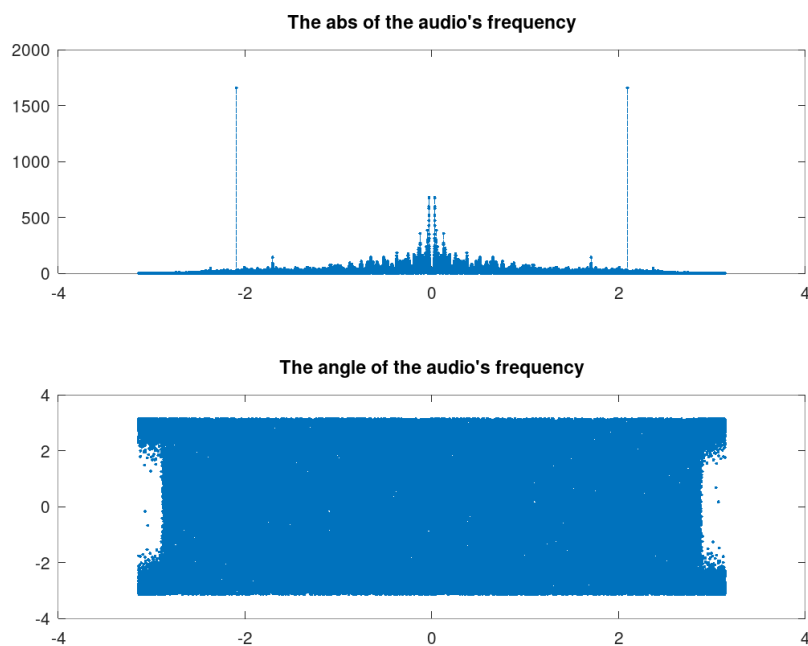


图 5: 原始音频 FFT 频谱

6 结果分析

6.1 窗函数长度对信号 DFT 的影响

经过对比图 1、图 2、图 3，可以看到随着采样点的增加，频谱的包络越来越明显，也就是越来越趋近其模拟频谱；而各张图片内部不同窗函数相互对比，可以看出不同窗函数对与次

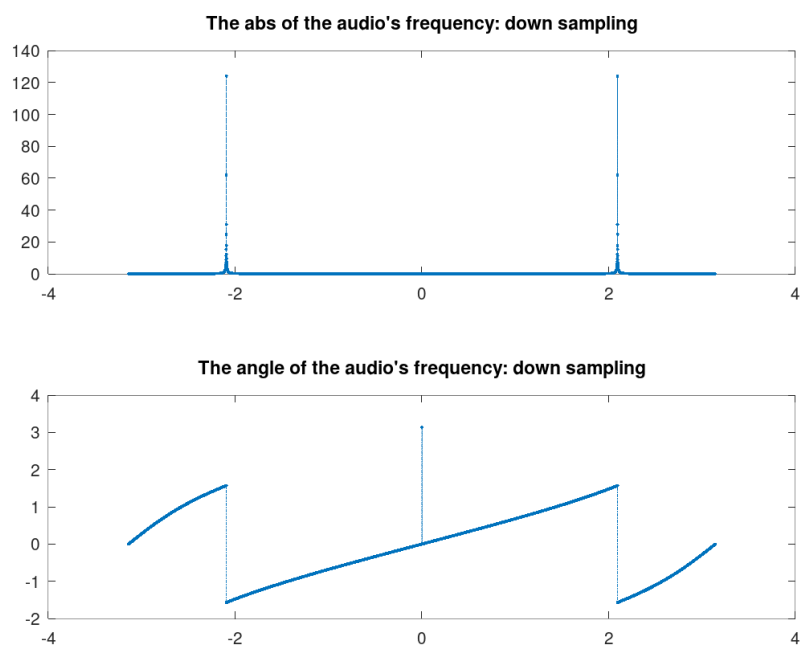


图 6: 原始音频降采样 FFT 频谱

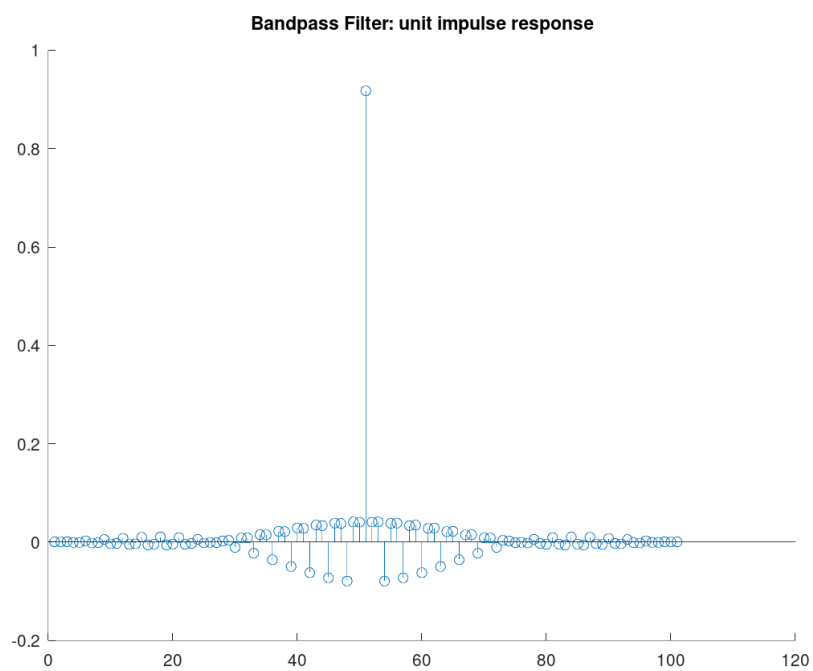


图 7: 带阻滤波器的单位冲激响应

瓣频谱的泄漏程度不同，Hamming 比较明显优于矩形窗。

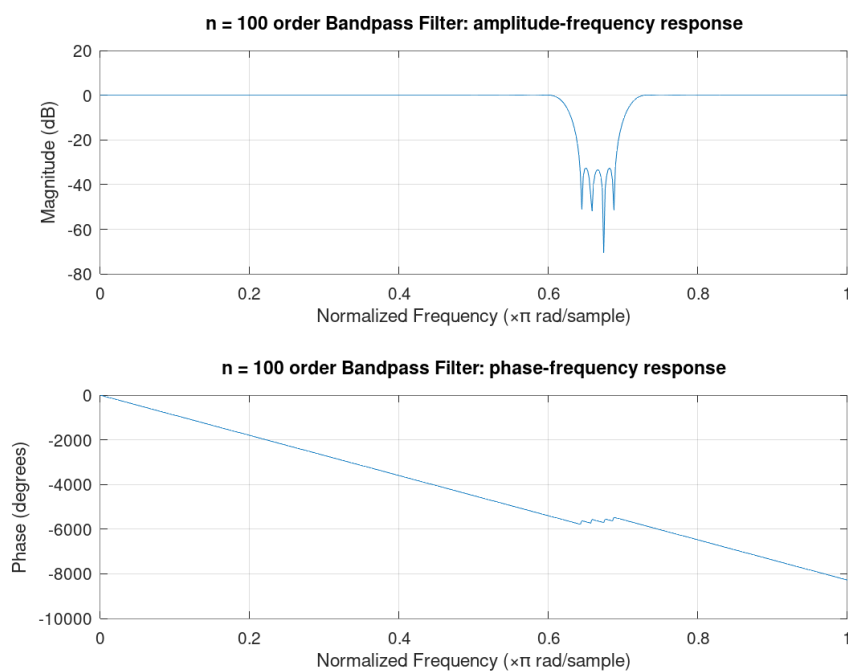


图 8: 带阻滤波器的频率响应

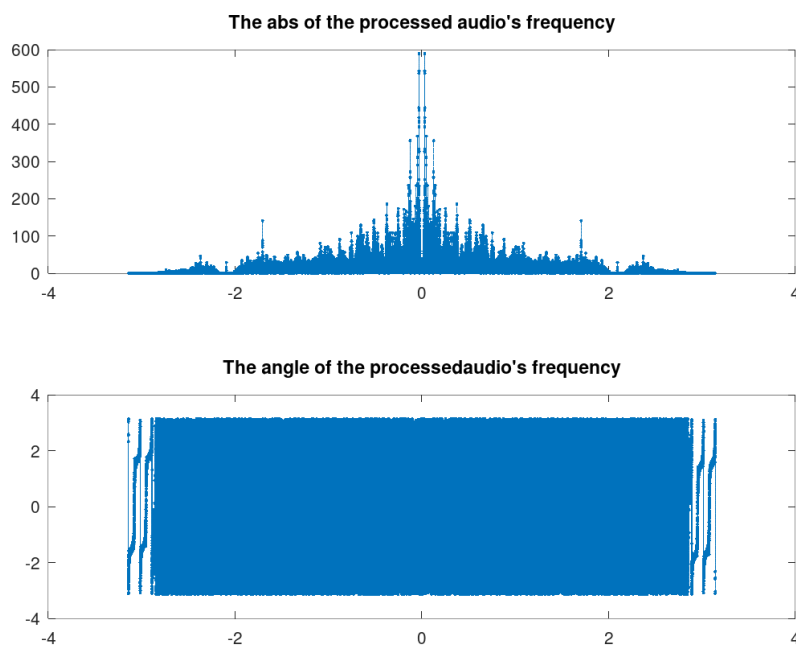


图 9: 处理后音频 FFT 频谱

6.2 频域采样对信号 DFT 的影响

如 图 4，更密集的频域采样完全没有改变频谱的形状，对于分析模拟谱的特性没有更大的帮助，仅仅增大了数字频谱的步进精度。其中 256 点乃至 128 点采样的对于分析应当已经足够。

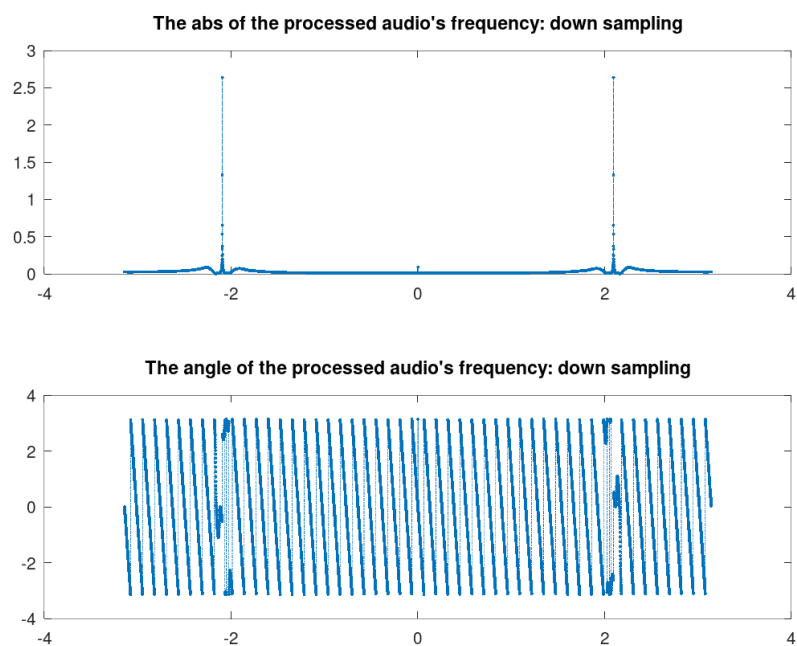


图 10: 处理后音频降采样 FFT 频谱

6.3 语音信号频谱分析及滤波处理

对于特定频率的噪音，可以通过设计滤波器的方式进行消除，同样地，我们也可以对特定频率的信号进行增强，但是滤波器设计中不可避免的出现在需要的频段出现增益或衰减的不精确，这种不精确可以达到几十个分贝，因此在一定程度上会影响整个系统的精度。