实验一 离散时间信号与系统

1. 实验目的
   1. 熟悉应用 MATLAB 表示离散时间信号。
   2. 掌握线性卷积求解系统输出的基本方法。
   3. 掌握求解离散时间系统输出的方法。
   4. 理解采样率变化对信号离散化产生的影响。

2. 实验内容
   1. 画出幅度按指数衰减的有限长复指数序列$x(n)=\left(0.9e^{-0.2\pi j}\right)^n R_{30}(n)$的实部、虚部、幅度和相位。提示：可以调用的函数有 exp()、stem()、real()、imag()、abs()、angle()等

1.1.m

```
close all
clear
n = 0:29;
xn = (0.9 .* exp(-0.2 * pi * i)).^n;
s1 = subplot(2, 2, 1); stem(n, real(xn));
title('实部')
s2 = subplot(2, 2, 2); stem(n, imag(xn));
title('虚部')
s3 = subplot(2, 2, 3); stem(n, abs(xn));
title('幅频')
s4 = subplot(2, 2, 4); stem(n, angle(xn));
title('相频')
```

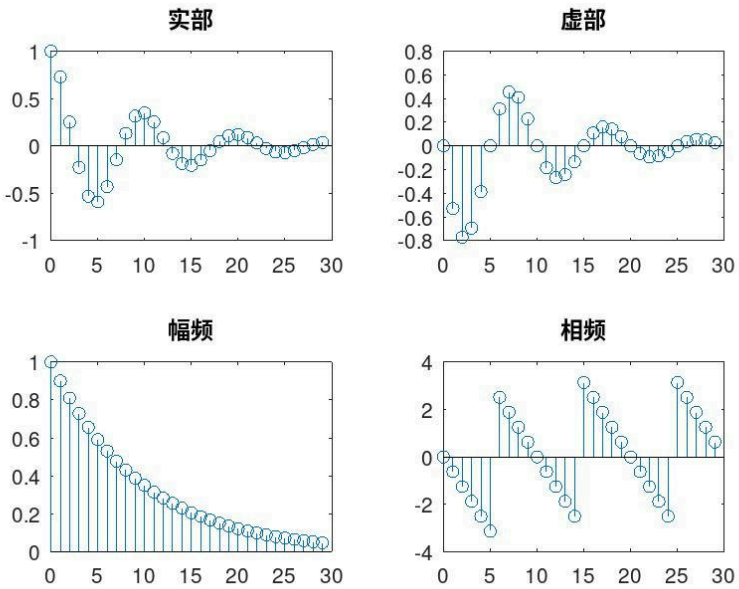


图 **1**　实验结果

   2. 11 阶滑动平均系统的输入/输出关系是$y(n)=\frac{1}{11}\sum_{k=0}^{10}x(n-k)$，输入信号是$x(n)=10\cos(0.08\pi n)+w(n)$，其中$w(n)$是一个在$[-5,5]$之间均匀分布的随机序列。试求：
      1. 用 plot 函数在 0≤n≤100 之间画出输入信号$x(n)$和输出信号$y(n)$
      2. 画出$x(n)$的 2 阶差分信号$v(n)=x(n)-2x(n-1)+x(n-2)$
      3. 画出$v(n)$与$w(n)$的相关序列
      4. 再产生一个随机序列，画出它与$v(n)$的相关序列。

1.2.m

```
close all
clear

% 11 阶滑动平均系统
n = 0:100;
wn = floor(rand(1,101) * 11) - 5;
xn = 10*cos(0.08*pi.*n)+wn;
yn = filter(1/11*ones(1,11),1,xn);

s1 = subplot(5,1,1);plot(n,xn);
title('x(n)');
s2 = subplot(5,1,2);plot(n,yn);
title('y(n)');

% x(n) 的 2 阶差分信号
vn = filter([1,-2,1],1,xn);
s3 = subplot(5,1,3);plot(n,vn);
title('v(n)');

% v 与 w 的相关序列
s4 = subplot(5,1,4);plot(-100:100,xcorr(vn,wn));
title('v(n), w(n) 相关序列');

% v 与随机的相关序列
wn = floor(rand(1,101) * 11) - 5;
s5 = subplot(5,1,5);plot(-100:100,xcorr(vn,wn));
title('v(n) 与其他随机的相关序列');
```

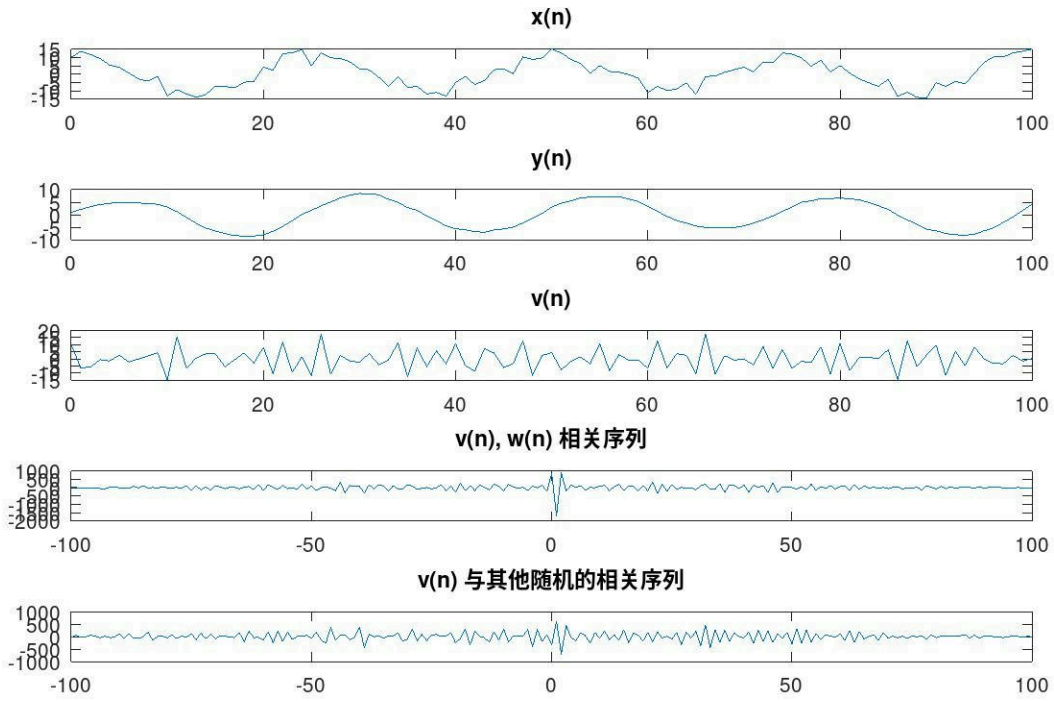


图 **2**　实验结果

- 实验分析与讨论（请手写）：

  1. 试分析 11 阶滑动平均系统的滤波特性。 过渡带特性，整体输出更平滑。

2. 试分析 2 阶差分系统的滤波特性。 高通特性
3. 由 3.和 4.的实验结果，你得出了什么结论？ 相关序列越杂乱，相关系数越小。3.中的 v(n)有 w(n)参与组成，因此相关序列在 0 处幅值较大。而 4.中的随机序列与 v(n)不相关，相关序列的值分布比较均匀。

3. 下面的差分方程可以产生声音的混响效果，请为音频文件 good.wav 合成混响的效果，并保存在 new_good.wav 文件中。用耳机欣赏混响前的音乐与混响后的音乐有何区别。

$y(n) = x(n) + \alpha x(n - R)$ 其中, α <1(比如：α=0.3,R=5000)

1.3.m

```
close all
clear
[x, fs] = audioread('../audio/good.wav');
subplot(2, 1, 1); plot(x); title('原始音频');
f = size(x)(1);
r = 4000;
y = zeros(f, 1);

for n = r + 1:f
    y(n) = 0.5 * x(n - r);
end

y = x + y;
subplot(2, 1, 2); plot(y); title('混响后音频');
audiowrite('../audio/new_good.wav', y, fs);
```

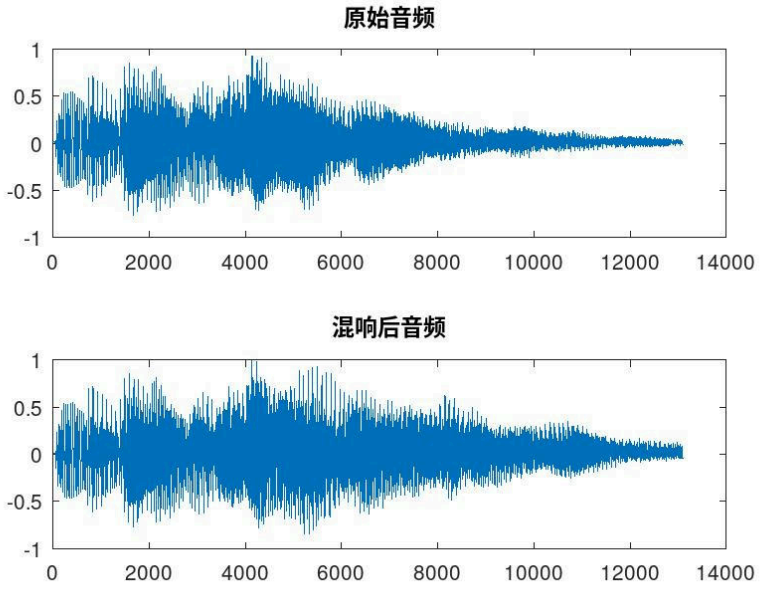


图 **3** 实验结果

- 实验分析与讨论（请手写）：
  1. 请问该系统是 IIR 系统还是 FIR 系统？ FIR 系统
  2. 请分析系统的因果性和稳定性在不同的 α 值和 R 值（R ≠ ∞）下，系统的因果性和稳定性是否会有变化？ $R \geq 0$时因果，$R < 0$时非因果；对于任意α均稳定。
  3. 请用文字简要描述不同 α 值和 R 值下的混响效果的区别。 α控制音量。α值越大，混响音量越大；R 控制延迟，R 越大，延迟越大。

4. 请编程实现实际音频信号经抽取系统$T[x(2n)]$、$T[x(4n)]$、$T[x(8n)]$后的音效。待处理的音频文件分别为钢琴乐曲卡农片段（canon.wav）和语音片段（dsp.wav）。展示你的程序设计方法并按照要求进行分析。

1.4.m

```
close all
clear

[x, fs] = audioread('../audio/canon.wav');

function y = ex(input, fs, i)
    y = downsample(input, i);
    filename = strcat('../audio/ex', int2str(i), '.wav')
    audiowrite(filename, y, fs);
end

subplot(4, 1, 1); plot(x); title('原始音频');
subplot(4, 1, 2); plot(ex(x, fs, 2)); title('T[x(2n)]');
subplot(4, 1, 3); plot(ex(x, fs, 4)); title('T[x(4n)]');
subplot(4, 1, 4); plot(ex(x, fs, 8)); title('T[x(8n)]');
```

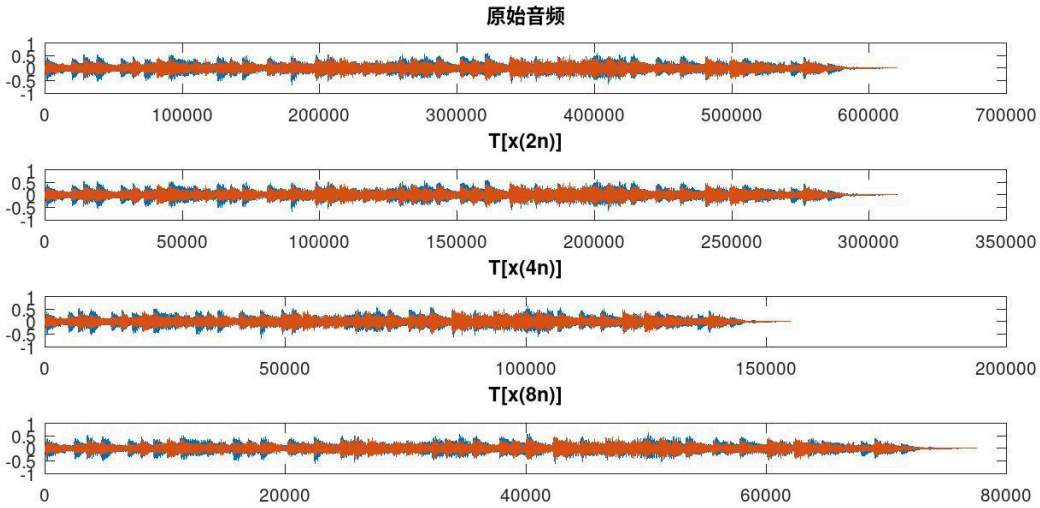


图 **4**　实验结果

- 实验分析与讨论（请手写）：
  1. 请比较分析两段乐曲在不同采样率下是否存在失真情况？ 存在。采样率越低，失真越严重。
  2. 请问随着采样率的不断下降，钢琴音频和语音哪一个失真效果更显著？请解释你的结论。 钢琴音频失真更显著。钢琴的平均频率更高，包含的信息更多。

实验二 离散傅立叶变换与分析

1. 实验目的
   1. 熟悉应用 MATLAB 求解信号频谱的方法。
   2. 掌握应用 FFT 的方法求解系统输出的步骤。
   3. 对比分析利用线性卷积求解系统输出和利用 FFT 方法求解系统输出这两种方法的不同之处。
   4. 掌握系统分析方法和简单滤波器的设计方法。

2. 实验内容
   1. 设输入信号$x(n)=\sin(0.1\pi n)+\cos(0.5\pi n), 0\le n\le 199$，某 LSI 系统的单位脉冲响应为$h(n)=\frac{1}{4}[\delta(n)+\delta(n-1)+\delta(n-2)+\delta(n-3)]$，求：
      1. 利用线性卷积求输入信号$x(n)$通过系统后的输出$y_1(n)$。
      2. 利用 FFT 的方法，先求解输入信号$x(n)$的频谱$X(k)$以及单位脉冲响应$h(n)$的频谱$H(k)$，通过计算IFFT$[X(k)\cdot H(k)]$求解系统的输出。

2.1.m

```
close all
clear
nx = 0:199;
xn = sin(0.1 * pi * nx) + cos(0.5 * pi * nx);
hn = 0.25 * ones(1, 4);
nh = 0:3;
yn = conv(xn, hn);
ny = nx(1) + nh(1):nx(end) + nh(end);
subplot(4, 1, 1); stem(ny, yn); title('y');

xk = fft(xn, 203);
subplot(4, 1, 2); plot(abs(xk)); title('Xk203');

hk = fft(hn, 203);
subplot(4, 1, 3); plot(abs(hk)); title('Hk203');

yk = xk .* hk;
yn = ifft(yk);
subplot(4, 1, 4); stem(yn); title('ynk203');
```

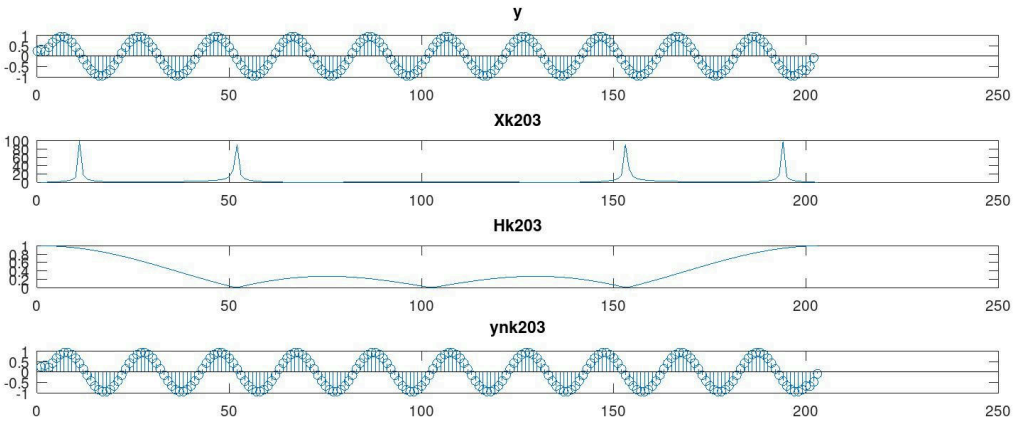


图 5　实验结果

- 实验分析与讨论（请手写）：

1. 请比较两种不同方法求得的输出及其它们的频谱$Y_1(k)$及$Y(k)$； 两种方法求得的输出$y_1(n)$和$y(n)$相同。
2. 试分析该系统的滤波特性，并结合输出信号的频率成分进行分析。 该系统的滤波特性为带阻滤波器。将$[0.5\pi, 1.5\pi]$的频率成分降到了 0.2 左右， 而通过了$[0, 0.5\pi]$和$[1.5\pi, 2\pi]$的频率成分。

2. 已知某系统的系统函数为$H(z) = \frac{1+z^{-1}+z^{-2}}{1+0.9z^{-1}+0.81z^{-2}}$，且系统稳定，试求：
   1. 求系统的零极点；（提示：可以用 tf2zp()函数）
   2. 画出系统的零极点图；（提示：可以用 zplane()函数）
   3. 画出系统的幅频响应、相频响应、群延迟。（提示：可以用 freqz()、grpdelay()函数）

2.2.m
```
n = 0:199;
a = [1 1 1];
b = [1 0.9 0.81];
[z, p, k] = tf2zp(b, a);
subplot(2, 2, 1); zplane(z, p); title('z and p');
[Fh, w] = freqz(b, a);
[Gd, w] = grpdelay(b, a);
subplot(2, 2, 2); plot(w / pi, abs(Fh)); title('|H(w)|');
subplot(2, 2, 3); plot(w / pi, angle(Fh)); title('ang|H(w)|');
subplot(2, 2, 4); plot(w / pi, Gd); title('grd|H(w)|');
```

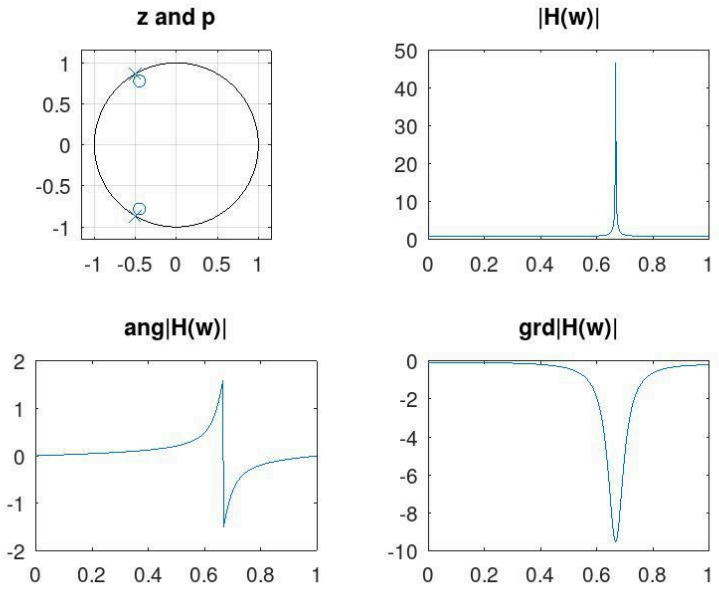


图 **6** 实验结果

- 实验分析与讨论（请手写）：
  1. 试求该系统的 ROC，并说明系统的因果性； $r = \sqrt{0.45^2 + 0.78^2} = 0.9, |z| > 0.9$，因果。
  2. 试分析该系统的滤波特性； 数字陷波器，主要滤除$\frac{2}{3}\pi$与$\frac{4}{3}\pi$的频率成分。
  3. 该滤波器是 IIR 滤波器还是 FIR 滤波器？该滤波器具有线性相位吗？ IIR 滤波器，不具有线性相位。

3. 一个 LSI 系统由下面的差分方程描述：

$$y(n) + 0.8y(n-1) - 0.64y(n-2) + 0.3125x(n)$$

1. 用 filter 函数计算并画出在$0 \leq n \leq 100$内的系统单位脉冲响应，由画出的单位脉冲响应判断系统的稳定性。
2. 画出系统零极点图及系统的幅频和相频响应曲线。
3. 如果这个系统的输入是$x(n) = \left[5 + 3\cos\left(\frac{\pi}{3}n\right)\right]u(n)$，利用 filter 函数求在$0 \leq n \leq 200$内的系统输出。分析输出信号，观察$x(n)$中的直流分量和$\frac{\pi}{3}$频率成份分量的通过情况。
4. 如果希望将$x(n)$中的直流分量完全滤除，而$\frac{\pi}{3}$频率成份分量仍然保留，应该怎样修改该系统的差分方程，用实验的方法验证你的结论。

2.3.m

```
close all
clear

function mydraw(n, b, a, x)
    y = filter(b, a, x);
    figure;
    subplot(2, 2, 1); plot(n, y); title('y');
    [z, p, k] = tf2zp(b, a);
    subplot(2, 2, 2); zplane(z, p); title('z and p');
    [Fh, w] = freqz(b, a);
    subplot(2, 2, 3); plot(w / pi, abs(Fh)); title('|H(w)|');
    subplot(2, 2, 4); plot(w / pi, angle(Fh)); title('ang|H(w)|');
end

% 1, 2
mydraw(0:100, [0.3125], [1 -0.8 0.64], [1, zeros(1, 100)])
% 3
mydraw(0:200, [0.3125], [1 -0.8 0.64], 5 + 3 .* cos(pi / 3 .* n))
% 4
mydraw(0:200, [1 -2 1], [1 -1.9 0.9025], 5 + 3 .* cos(pi / 3 .* n))
```

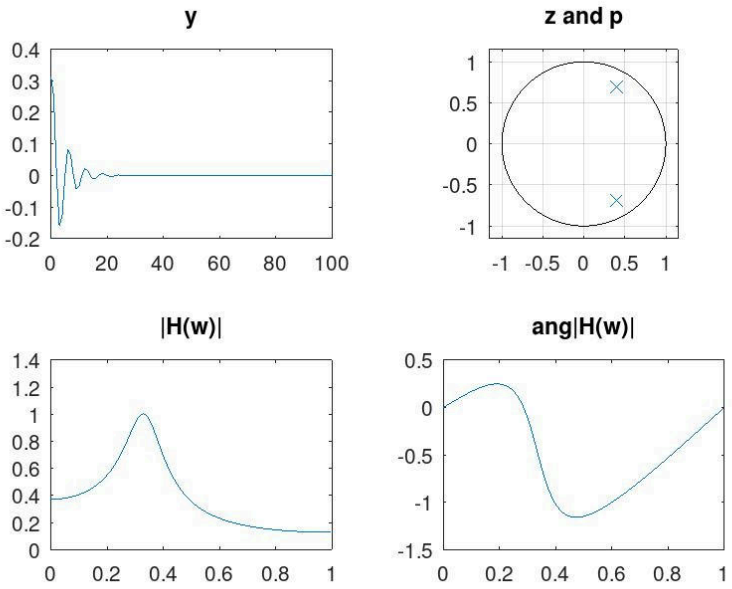


图 7　第 1-2 题

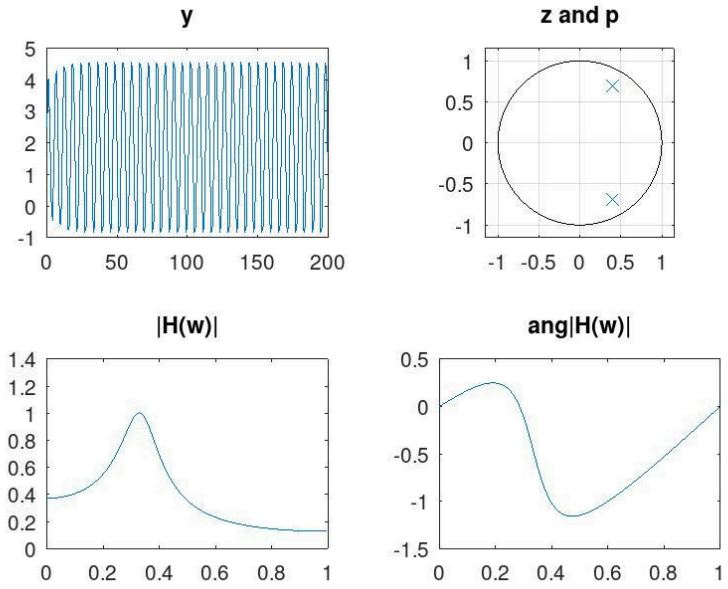


图 **8**　第 **3** 题

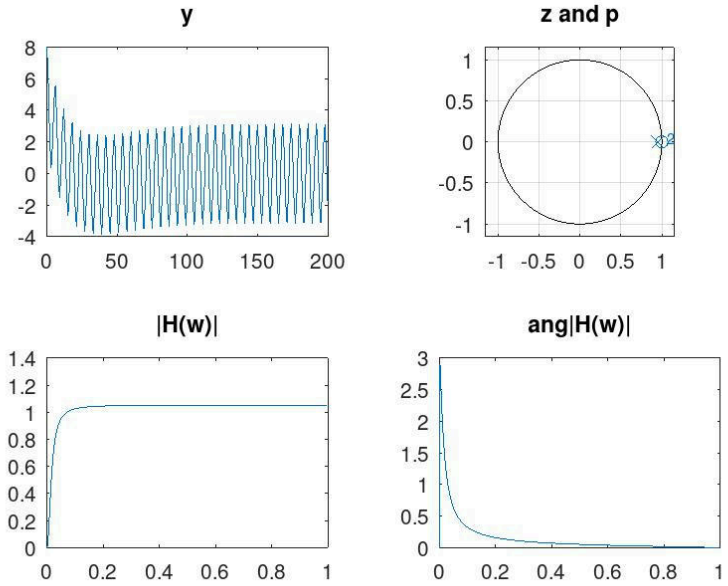


图 **9**　第 **4** 题

- 实验分析与讨论（请手写）：
  1. 请通过实验分析该系统的稳定性； 稳定，因为极点都在单位圆内。
  2. 请分析解释第 3 问中输入信号不同频率成份的通过情况，并解释原因； 直流分量被抑制，$\frac{\pi}{3}$频率成分几乎未被滤除。该系统函数对这两处的滤除作用不大。
  3. 请描述第 4 问的设计思路。

$$H(z) = \frac{(z - e^{j0})(z - e^{-j0})}{(z - 0.95e^{j0}(z - 0.95e^{-j0}))}$$

$$= \frac{z^2 - 2z + 1}{z^2 - 1.9z + 0.9025}$$

$$= \frac{1 - 2z^{-1} + z^{-2}}{1 - 1.9z^{-1} + 0.9025z^{-2}}$$

4. 音频文件 test1.wav test4.wav 中录制了钢琴上的一些按键音，其中，请用 FFT 的方法识别出每段音频文件中含有那几个音符。（注： (la)=220Hz (ci)=246.94Hz (do)=261.63Hz 2(rui)=293.66Hz 3 (mi)=329.63Hz 4(fa)=349.23Hz 5(so)=392Hz ）
   1. 请读入 test1.wav test4.wav 音频文件，画出音频信号的时域图。
   2. 请识别 test1.wav 和 test2.wav 两个音频文件中弹奏的是哪个音符（请对音频信号中的有效音符区间进行谱分析，建议尝试使用不同的窗函数进行谱分析，以缓解频谱泄漏现象）。
   3. 请识别 test3.wav 和 test4.wav 两个音频文件中弹奏了哪些音符；
   4. （拓展部分）test3.wav 和 test4.wav 两个音频文件有多个音符，且在最后有三音和弦，请尝试切分这些音符，并对其中的单音或和弦进行识别。

2.4.m

```
close all
clear
global index
index = 0;
keyset = {220, 246.94, 261.63, 293.66, 329.63, 349.23, 392}
valueset = {'la-', 'ci-', 'do', 're', 'mi', 'fa', 'so'}

global myMap
myMap = containers.Map(keyset, valueset);

function closestKey = fin_closestKey(targetNumber)
    % ...
end

function detectNotes(y, fs, filename)
    global index
    global myMap
    N = length(y);
    f = (0:N - 1) * (fs / N);
    Y = fft(y);
    N = length(f(f < 800));
    f = f(1:N);
    Y = abs(Y(1:N));

    figure(2);
    subplot(4, 1, index);
    plot(f, Y);
    title(['频谱图 - ' filename]); xlabel('频率 (Hz)'); ylabel('幅度');

    [peaky, peakx] = findpeaks(Y);
    _filter = peaky > 0.9 * max(Y);
    peakx_filtered = peakx(_filter);
    peaky_filtered = peaky(_filter);
    peakx_filtered = f(peakx_filtered);

    % disp(peakx_filtered)
    % disp(peaky_filtered)

    % disp(['在 ' filename ' 中识别到的音符: ']);
    % for i = 1:length(peakx_filtered)
```

```
    %     detected_notes = fin_closestKey(peakx_filtered(i));
    %     disp(myMap(detected_notes));
    % end

end

function process(file)
    global index
    index = index + 1;
    [x, fs] = audioread(file);

    figure(1);
    subplot(4, 1, index);
    plot(x);
    title(['test' int2str(index)]);

    [~, name, ~] = fileparts(file);
    detectNotes(x, fs, name);
end

process('../audio/test1.wav');
process('../audio/test2.wav');
process('../audio/test3.wav');
process('../audio/test4.wav');
```

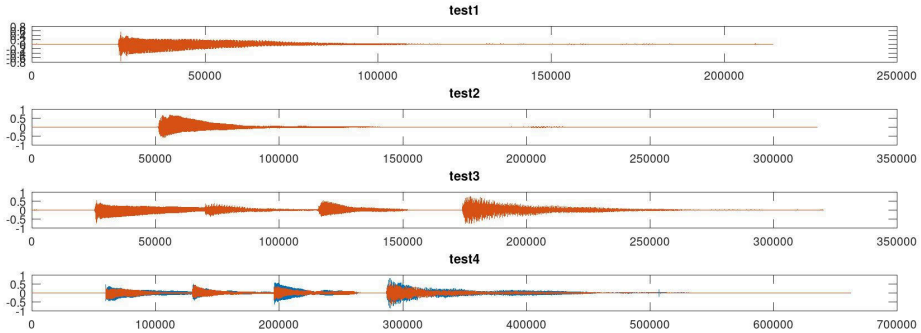


图 **10**　时域图

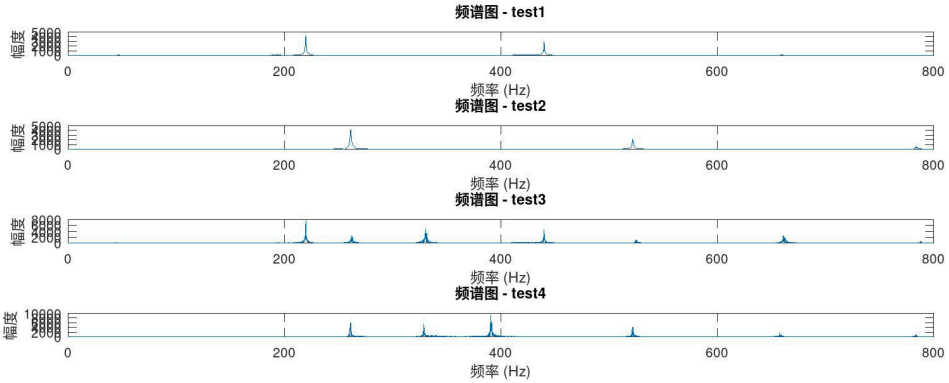


图 **11**　频谱图

3-4 题尚未完成。

- 实验分析与讨论（请手写）：

1. 请分析使用不同窗函数时谱分析的结果； 矩形窗旁瓣衰减较慢。汉宁窗与哈明窗的能量集中在主瓣。
2. 根据你的程序实验结果，请回答 test1.wav test4.wav 音频文件中包含了哪些音符？

实验三 IIR 数字滤波器的设计

1. 实验目的
   1. 掌握应用 MATLAB 进行 IIR 数字滤波器设计的基本方法。
   2. 灵活运用 MATLAB 提供的函数设计各种类型的 IIR 数字滤波器。
2. 实验内容
   1. 已知数字带通滤波器的通带指标为：$\omega_{p1} = 0.3\pi, \omega_{p3} = 0.7\pi, \alpha_p = 0.5$dB阻带指标为：$\omega_{s1} = 0.2\pi, \omega_{s2} = 0.8\pi, \alpha_s = 50$dB,分别设计满足以上指标的巴特沃斯、切比雪夫 I 型、切比雪夫 II 型、以及椭圆数字滤波器，并画出滤波器的频率响应。

3.1.m

```
wp = [0.3, 0.7];
ws = [0.2, 0.8];
rp = 0.5; rs = 50;
fs = 100;

[n, Wn] = buttord(wp, ws, rp, rs);
[b, a] = butter(n, Wn);
[H, w] = freqz(b, a);
subplot(2, 2, 1);
plot(w * fs / (2 * pi), abs(H)); grid; title('butter');

[n, Wn] = cheb1ord(wp, ws, rp, rs);
[b, a] = cheby1(n, rp, Wn);
[H, w] = freqz(b, a);
subplot(2, 2, 2);
plot(w * fs / (2 * pi), abs(H)); grid; title('cheby1');

[n, Wn] = cheb2ord(wp, ws, rp, rs);
[b, a] = cheby2(n, rs, Wn);
[H, w] = freqz(b, a);
subplot(2, 2, 3);
plot(w * fs / (2 * pi), abs(H)); grid; title('cheby2');

[n, Wn] = ellipord(wp, ws, rp, rs);
[b, a] = ellip(n, rp, rs, Wn);
[H, w] = freqz(b, a);
subplot(2, 2, 4);
plot(w * fs / (2 * pi), abs(H)); grid; title('ellip');
```

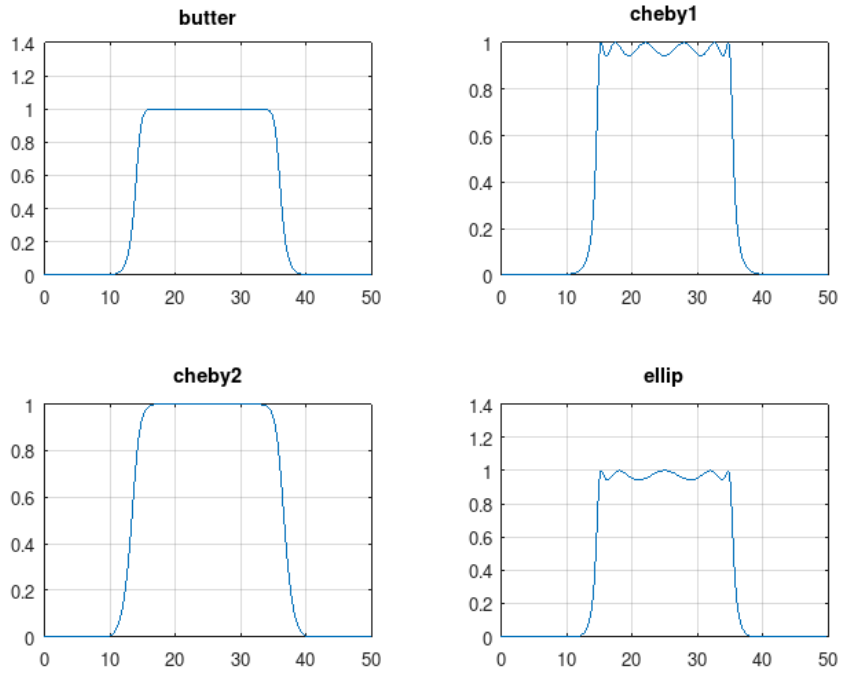


图 **12**　实验结果

- 实验分析与讨论（请手写）：请比较和分析不同类型滤波器在相同技术指标情况下设计出的滤波器阶数、以及滤波器幅频响应的特点。椭圆数字滤波器阶数最低，过渡带最短；椭圆数字滤波器和切比雪夫 II 型滤波器通带波动较大。巴特沃斯滤波器与切比雪夫 II 型滤波器的通带更平滑。

2. 设计一个数字高通滤波器 H(z)，它用在下面的结构中，

$$x(t) \to A/D \to H(z) \to D/A \to y(t)$$

满足下列要求：

1. 采样频率为 10kHz；
2. 阻带边缘频率为 1.5kHz，；
3. 通带边缘频率为 2kHz，；
4. 滤波器的幅频响应具有单调的通带和阻带特性。

画出该数字高通滤波器的频率响应。请自行设计一个输入信号（至少应含有两种及以上的频率成分）通过该滤波器，观察并解释滤波器的输出结果。

3.2.m

```
fs = 10000;
wp = 1500 * 2 / fs;
ws = 2000 * 2 / fs;
rp = 3; rs = 40;

[n, Wn] = buttord(wp, ws, rp, rs);
[b, a] = butter(n, Wn, 'high');
[H, w] = freqz(b, a);
subplot(3, 1, 1); plot(w * fs / (2 * pi), abs(H)); grid; title('butter');

t = 0:1 / fs:1;
f1 = 900;
f2 = 4500;
x = sin(2 * pi * f1 * t) + 0.5 * sin(2 * pi * f2 * t);
y = filter(b, a, x);
```

```
subplot(3, 1, 2);
plot(t(1:100), x(1:100));
title('Original Signal'); xlabel('Time (s)'); ylabel('Amplitude');

subplot(3, 1, 3);
plot(t(1:100), y(1:100));
title('Filtered Signal'); xlabel('Time (s)'); ylabel('Amplitude');
```

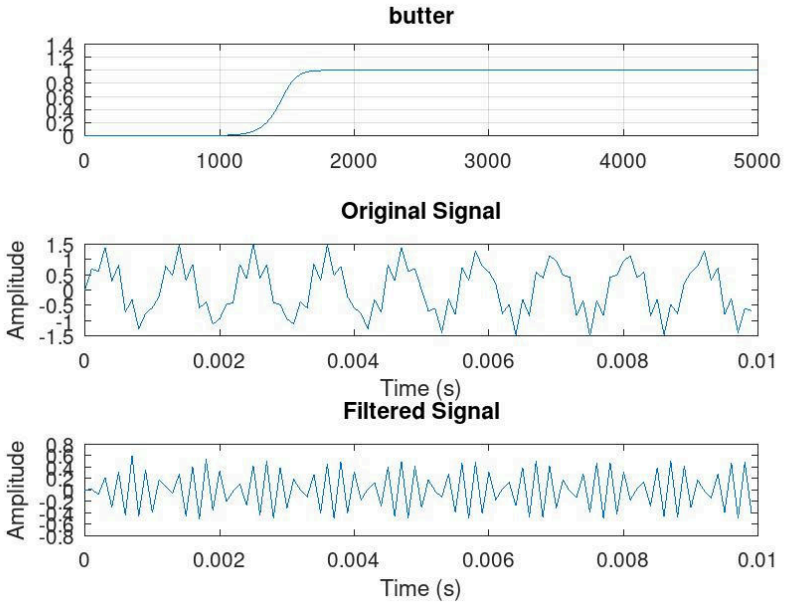
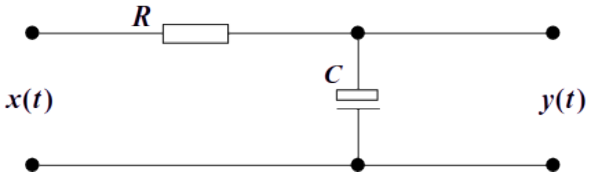


图 **13**　实验结果

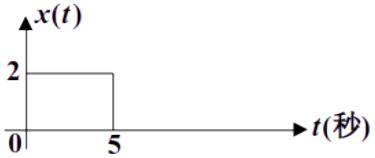
- 实验分析与讨论（请手写）：请写出该滤波器的技术指标，说明设计的输入信号以及实验获得的滤波结果。请预测一下，如果该滤波器用椭圆滤波器来设计，所得到的滤波器阶数应该会变多还是变少？为什么？

  $\Omega_s = 2\pi f_s = 3k\pi$ rad/s, $\Omega_p = 2\pi f_p = 4k\pi$ rad/s, $N \geq \frac{(\lg k_{sp})}{\lg \lambda_{sp}} = 16.02$，取$H = 17$。实验结果是成功率除了$X_n1$的频率成分。如果换成椭圆滤波器，阶数会减少。

3. 已知某系统如图所示



系统的输入 x(t)为：



若现在用数字滤波器模拟上述系统的输入输出效果，试求：

1. 请选择系统的采样频率 fs，画出 x(n)；

2. 画出系统的幅频响应$|H(e^{j\omega})|$；
3. 画出系统的输出 y(n)。

3.3.m

```
close all
clear
fs = 1/0.02;
r = 6;
x = 0:1 / fs:r;
xn = 2 * (x >= 0 & x < 5);
value = xn(xn > 0);
b = 0.02; a = [1 -0.9802];
figure 1; stem(xn); title('x(n)');
figure 2; freqz(0.02, [1 -0.9802]);
yn = filter(b, a, value);
figure 3; plot(yn); title('yn');
```

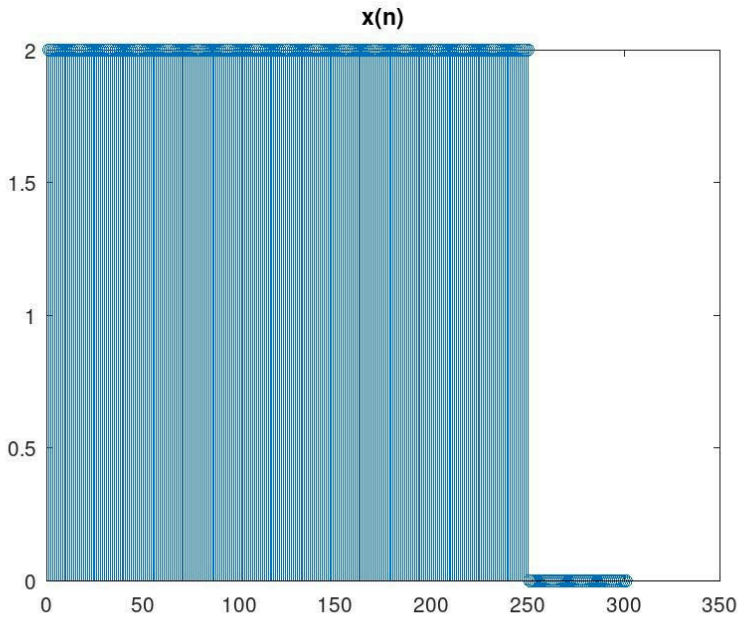


图 16　x(n)

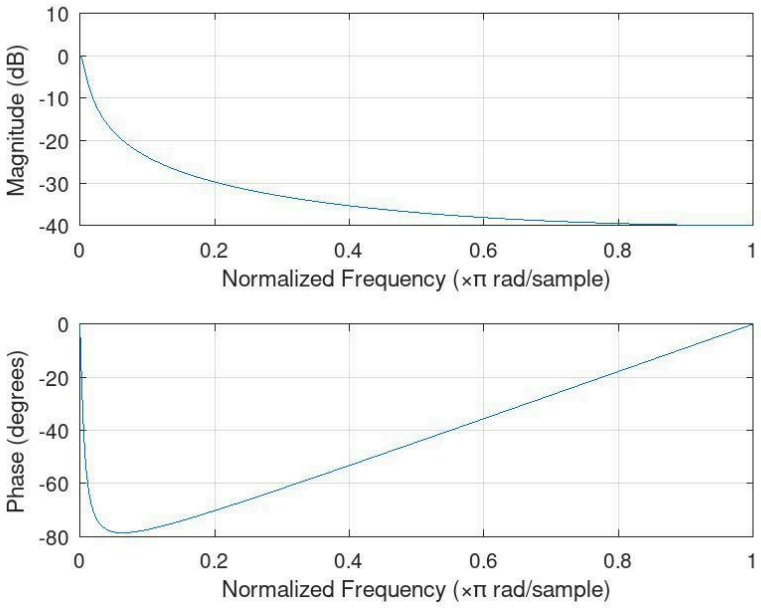


图 **17**　幅频响应与相频响应

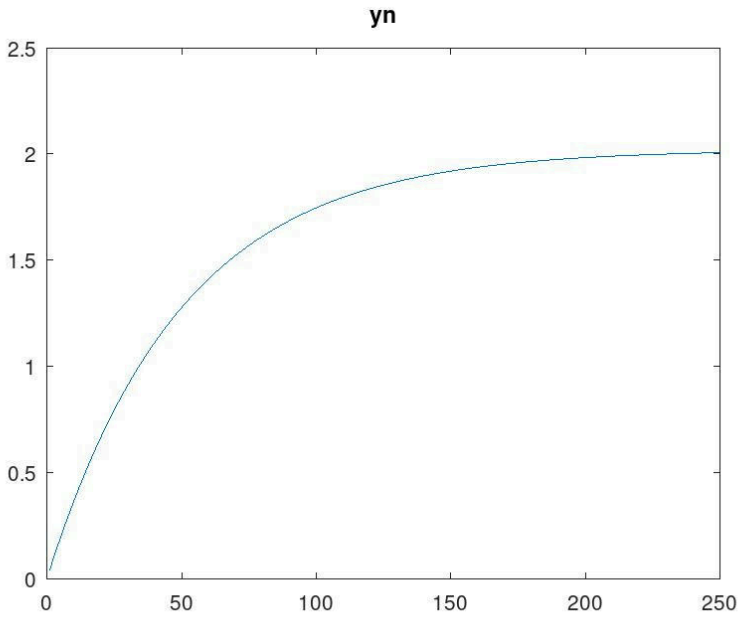


图 **18**　**y(n)**

- 实验分析与讨论（请手写）：请写出必要的系统分析步骤，分析该滤波器的滤波特性，并解释滤波结果。
  $x(t) = R * i(t) + y(t) = R\frac{C(dy(t))}{d(t)} + y(t), X(s) = SY(s) + X(s), H(s) = \frac{Y(s)}{X(s)} = \frac{1}{s+1}$。低通，$\Omega_{st} = 100\ \mathrm{rad/s}, f_{st} = 16\ \mathrm{Hz}$。滤波器确实滤除了低频成分，但过渡带较宽，效果不佳。

实验四 FIR 数字滤波器的设计

1. 实验目的
   1. 掌握应用 MATLAB 实现 FIR 数字滤波器的窗函数设计方法。
   2. 掌握应用 MATLAB 实现 FIR 数字滤波器的频率采样设计方法。
   3. 理解 FIR 数字滤波器的线性相位约束条件。

2. 实验内容
   1. 选择适当的窗函数设计一个数字带阻滤波器，通带指标为：$\omega_{p1} = 0.3\pi, \omega_{p2} = 0.7\pi, \alpha_p = 0.5$ dB,阻带指标为：$\omega_{s1} = 0.4\pi, \omega_{s2} = 0.6\pi, \alpha_s = 50$ dB,请画出设计的滤波器的脉冲响应和幅频响应。

4.1.m

```
close all
clear
wp = [0.3 * pi, 0.7 * pi];
ws = [0.4 * pi, ws2 = 0.6 * pi];
tr_width = 0.1 * pi;
M = ceil(6.6 * pi / tr_width);
wc = (wp + ws) ./ 2;
h = fir1(M, wc / pi, 'stop', hamming(M + 1));
[H, w] = freqz(h, 1);
subplot(2, 1, 1); stem(0:M, h); title('脉冲响应');
subplot(2, 1, 2); plot(w / pi, 20 * log10(abs(H))); title('幅频响应');
grid on;
```

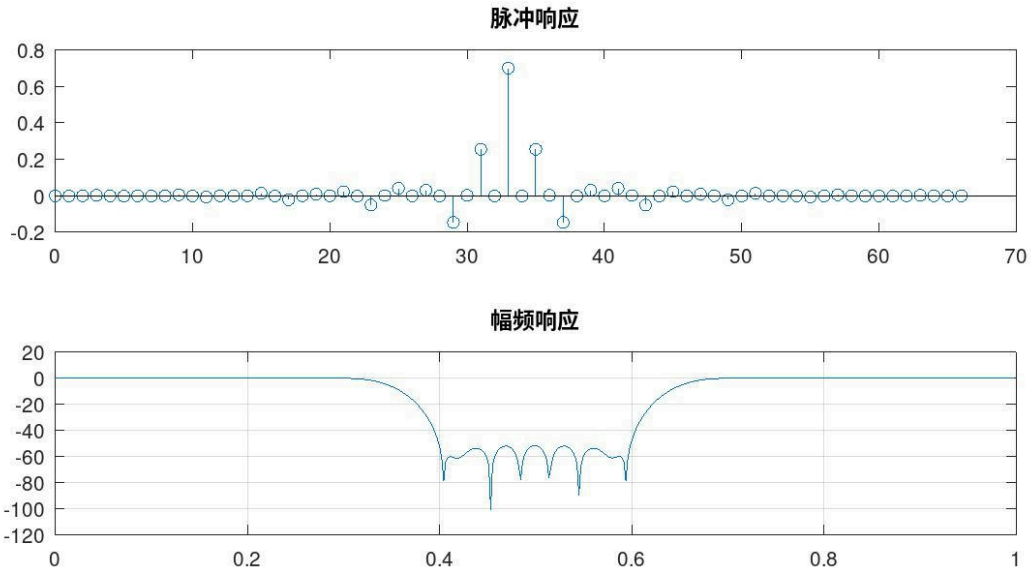


图 **19** 实验结果

- 实验分析与讨论（请手写）：请写出此题选窗的依据，以及滤波器长度的计算方法。另外，此题中对滤波器单位脉冲响应长度的奇偶性是否有要求？为什么？ $\alpha_s = 50$ db，与哈明窗的-53dB 相近。使用第一类滤波器实现带阻滤波器，N 只能取奇数。

1. 用频率采样法设计题 1 中的带阻滤波器，选择适当的滤波器阶数，为确保滤波效果，要求过渡带中有两个样本（按照经验值，过渡带两个样本的幅度可以取为 0.109 和 0.594）。请画出设计的滤波器的脉冲响应和幅频响应。

4.2.m

```
close all
clear

M = 61;
alpha = (M - 1) / 2;
k = 0:M - 1;
wk = (2 * pi / M) * k;
Hrs = [ones(1, 10), 0.594, 0.109, zeros(1, 7), 0.109, 0.594, ones(1, 20),
0.594, 0.109, zeros(1, 7), 0.109, 0.594, ones(1, 9)];
k1 = 0:floor((M - 1) / 2);
k2 = floor((M - 1) / 2) + 1:M - 1;
angH = [-alpha * 2 * pi / M * k1, alpha * 2 * pi / M * (M - k2)];
H = Hrs .* exp(1j * angH);
h = real(ifft(H, M));
[Ha, w] = freqz(h, 1);

subplot(2, 1, 1);
stem(k, h);
axis([-1, M, -0.1, 0.3]);
title('脉冲响应');
grid;

subplot(2, 1, 2);
plot(w / pi, 20 * log10(abs(Ha)));
axis([0, 1, -60, 10]);
title('幅频响应');
grid;
```

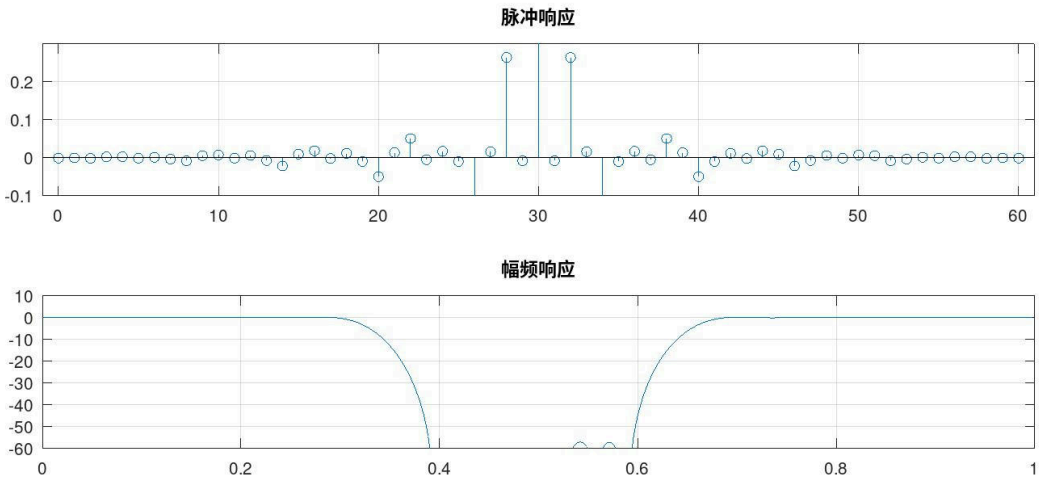


图 **20**　实验结果

- 实验分析与讨论（请手写）：请写出应用频域采样法设计该滤波器的方法。带阻滤波器只能使用一类滤波器：偶对称，N 为奇数。取 M=61，$\alpha = \frac{M-1}{2} = 30$

2. 用等波纹最佳逼近法设计 1 中的滤波器（注：可以使用 remezord()函数和 remez()函数，参考第七章讲义例程）。

4.3.m

```
close all
clear
% octave 没有 remezord
f = [0.3 0.4 0.6 0.7]; m = [1 0 1];
```

```
rp = 0.5; rs = 60;
dat1 = (10^(rp / 20) - 1) / (10^(rp / 20) + 1); dat2 = 10^(-rs / 20);
rip = [dat1 dat2 dat1];
[M, fo, mo, wo] = remezord(f, m, rip);
hn = remez(M, fo, mo, wo);
[H, w] = freqz(hn, 1);
subplot(2, 1, 1); stem(0:M, hn); title('脉冲响应');
subplot(2, 1, 2); plot(w / pi, 20 * log10(abs(H))); title('幅频响应');
```

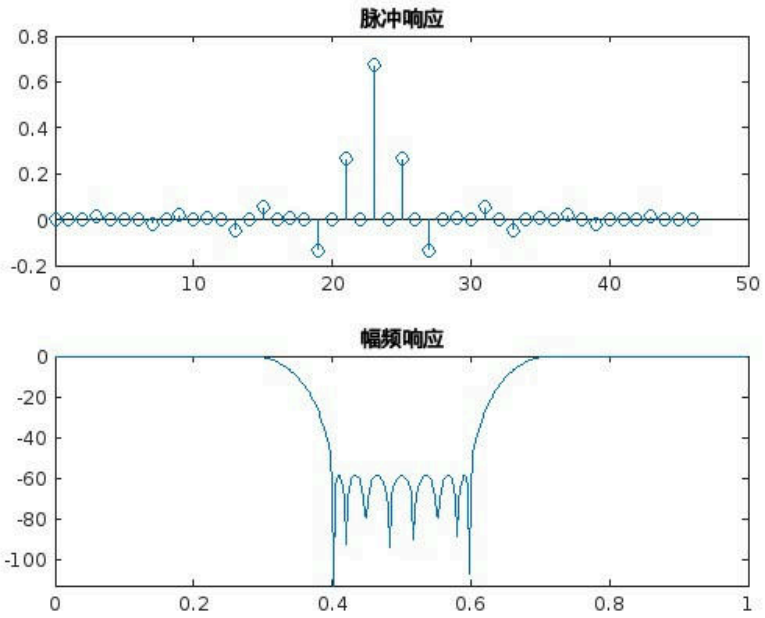


图 **21**　实验结果

- 实验分析与讨论（请手写）：请比较窗函数法和等波纹最佳逼近法的滤波器阶数和滤波效果。窗函数法设计出的滤波器阶数较低，但效果较差。等波纹最佳逼近法设计出的滤波器阶数较高，但效果较好，衰减基本在-60dB 以下。

实验五 综合实践项目：钢琴乐音识别技术研究

1 实验目的

通过钢琴乐音识别技术的研究，掌握满足复杂工程问题需求的离散时间系统的基本设计方法与分析技术，了解影响设计目标和技术方案的各种因素，并得出有效结论。

2 实验内容

钢琴乐音识别技术对钢琴乐音信号进行基频估计，然后根据基频大小来区分音高，从而实现对乐曲的识别。 课题针对钢琴音频信号进行乐音识别技术的研究。在对音频信号其进行分帧、分音程检测后，对各音程段信号进行离散傅里叶变换，分析频谱中所蕴含的音符信息，并与乐谱进行比对，并得出结论。 请查阅提供的文献资料，并结合自己查阅的资料，完成以下实验内容：

1. 基础要求（必做）：实现不同组别的钢琴音阶识别；
2. 进阶要求（选做）：实现钢琴乐曲《小星星》的整曲识别。
3. 拓展要求（选做）：对钢琴乐曲《小星星》的整曲节奏进行分析评价。

注：2 和 3 至少选做 1 个。

3 实验报告要求

1. 请写出钢琴乐音识别的技术路线； 音频采样->时域分析处理->频域分析->谱图表示->音符检测->模式匹配
2. 请阐述钢琴乐音识别的实现步骤；
    1. 读入音频文件
    2. 用帧峰检测法提取音频的包络信号：对音频信号进行分帧，记录每一帧内的峰峰值，并将每一帧的峰峰值记录在包络数组中
    3. 使用滤波算法，对包络信号进行滤波，本实验选择中值滤波
    4. 根据包络信号寻找音符的起始点，并去除较小的伪峰点
    5. 音程段提取与谱分析：找到每一段的起始位置和终止位置，提取，乘窗函数，添零，再做 FFT
    6. 音频信号谱图的可视化
    7. 画整个音频的短时傅里叶谱图
    8. 钢琴音频整曲识别：读入钢琴按键与频率的对应表，将音频幅度矩阵映射到按键矩阵中，画出键号对应的图，显示识别结果
3. 请对实验结果进行分析，并得出自己的结论；

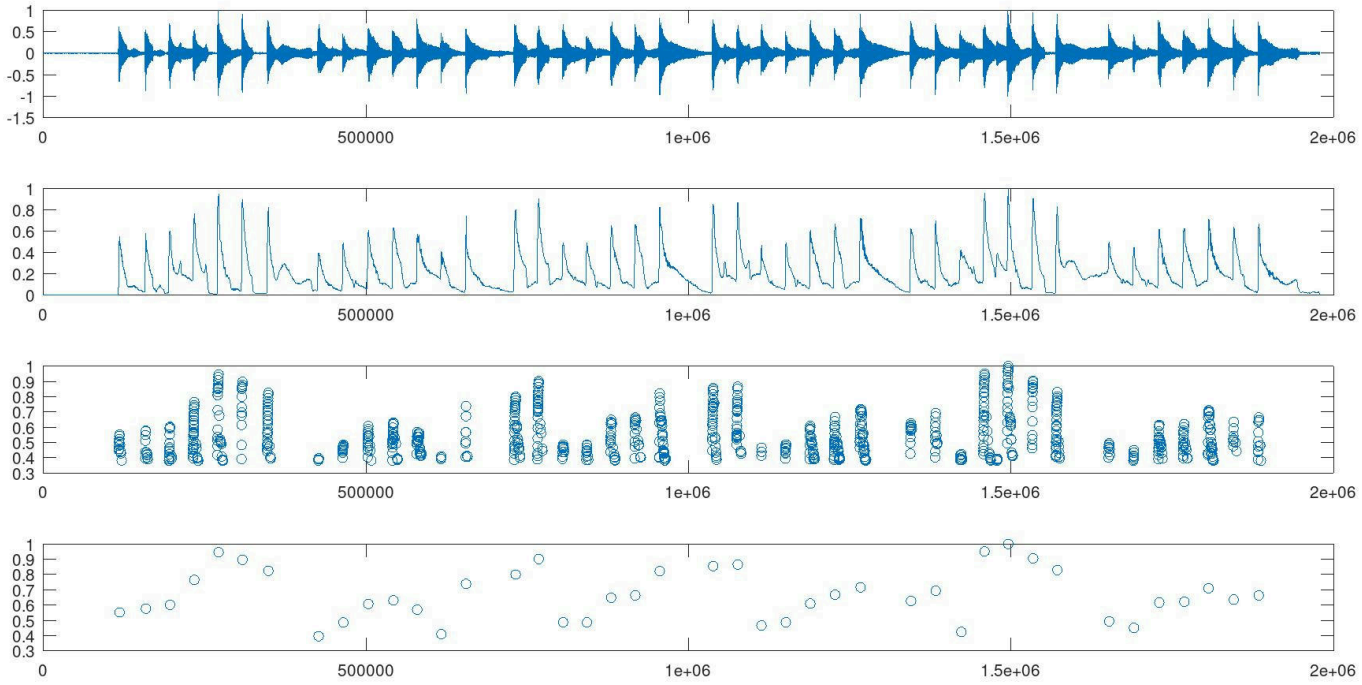


图 **22**　从读取音频到找到音频起始点。**plot1**：时域图；**plot2**：包络线；**plot3**：**findpeaks** 结果；**plot4**：去除伪峰点，只保留音符起始位置

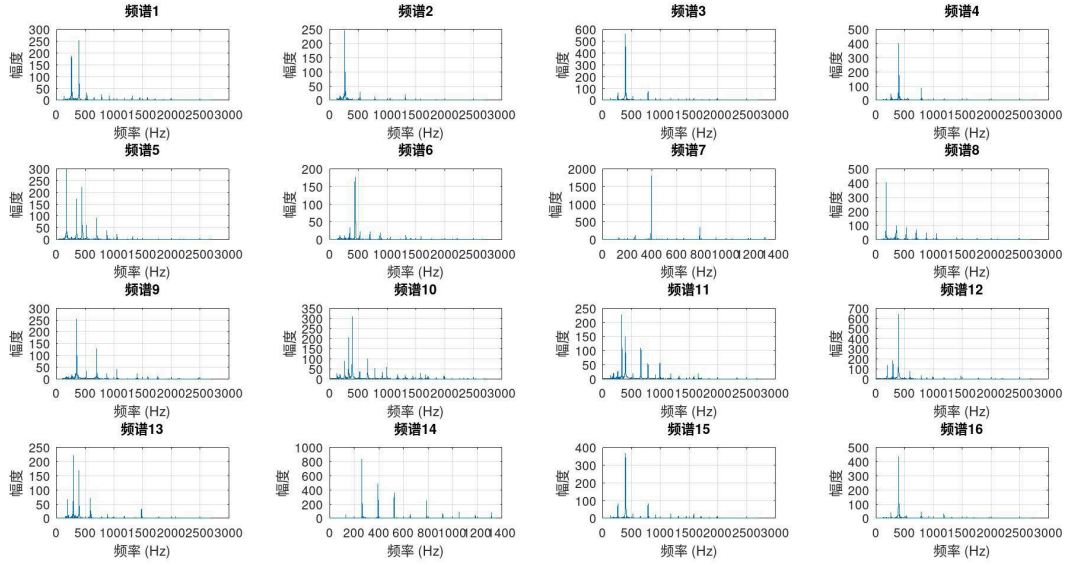


图 **23**　逐个音程画图（**1-16**）

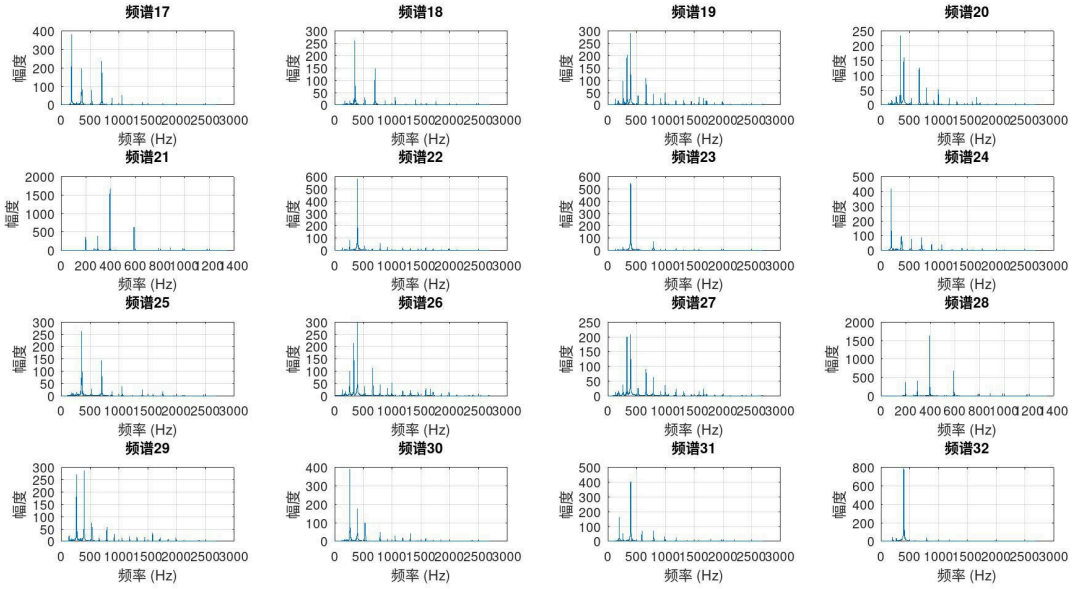


图 **24**　逐个音程画图（**17-32**）

使用音乐小星星画图，数量较多，后面还有一张，这里不放出。

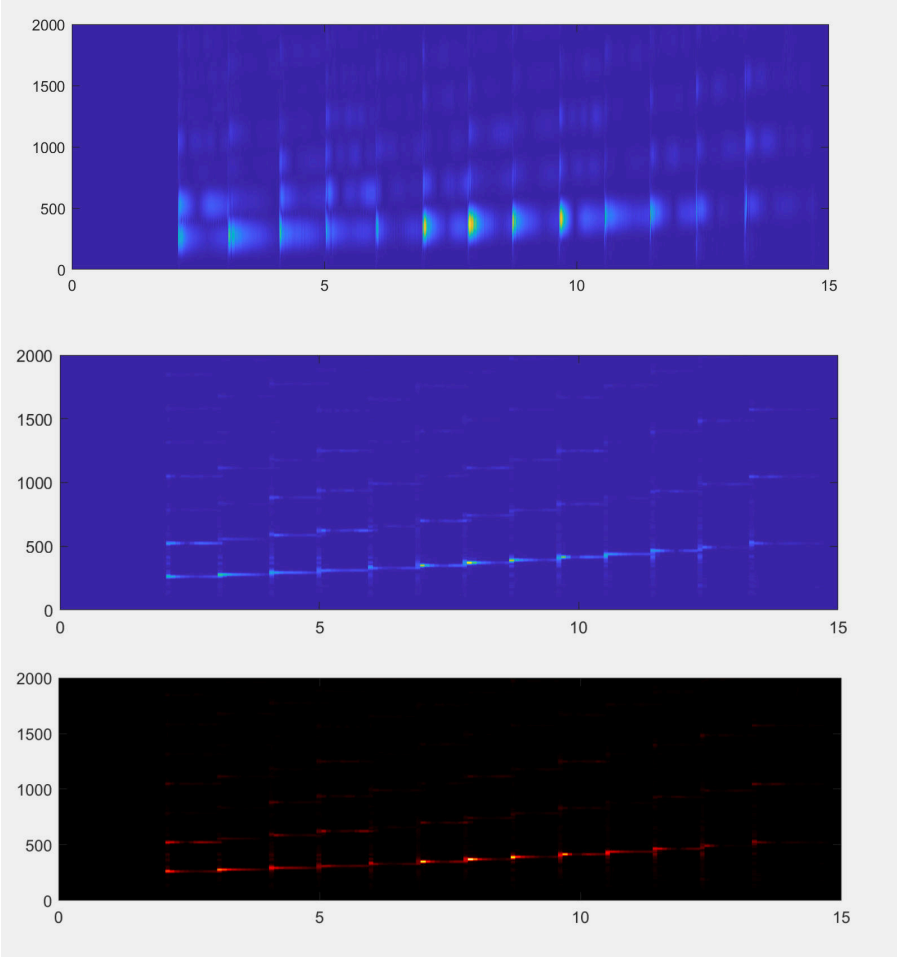


图 **25**　短时傅里叶谱图。上：**test_Num=400**，中、下：**test_Num=4000**

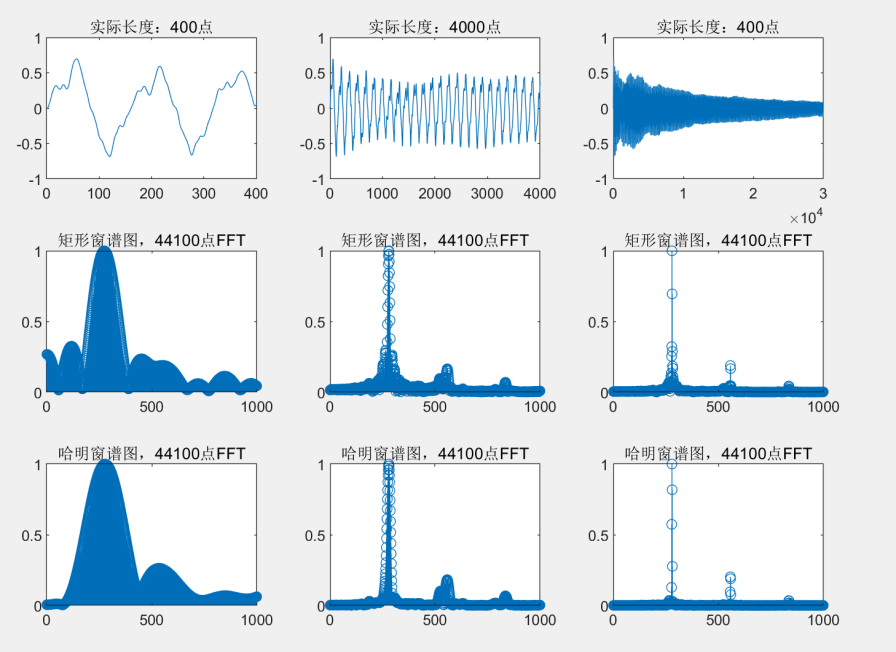


图 **26**　研究不同帧长下的频谱泄露现象，以帧长 **400** 为例。

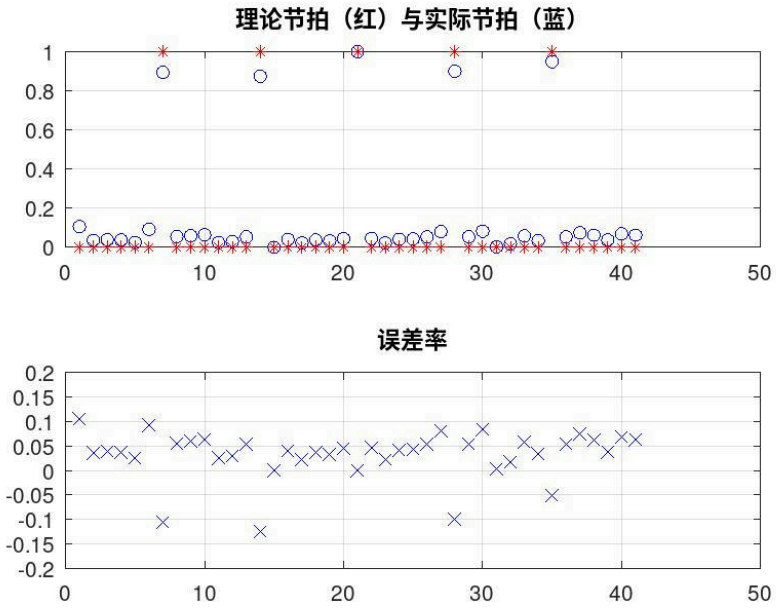


图 **27**　钢琴音频节奏评价

4. 请附上带注释的程序清单。

寻找音频起始点

5.1.m

```
close all
clear
hold on;

% 读取音频并处理（双声道转单声道）
[music, Fs] = audioread('../audio/star_69.wav');
music = music(:, 1) ./ max(music(:, 1));
subplot(411); plot(music);
time = length(music) / Fs;
```

```
% 计算包络
Frame_Num = 70;
envelope1 = [];

for i = 1:floor(length(music) / Frame_Num) - 1
    temp = music((i - 1) * Frame_Num +1:i * Frame_Num);
    envelope1 = [envelope1 max(temp) - min(temp)];
end

% 中值滤波
envelope1 = medfilt1(envelope1, 30);
envelope1 = envelope1 ./ max(envelope1);
envelope1_extend = repelem(envelope1, Frame_Num);
subplot(412);
plot(envelope1_extend);

% 寻找峰值
[peaky, peakx] = findpeaks(envelope1_extend);
_filter = peaky > 0.38;
peakx_filtered = peakx(_filter);
peaky_filtered = peaky(_filter);
subplot(413); plot(peakx_filtered, peaky_filtered, 'o');

% 过滤峰值。函数的作用是在检测到峰值后对其后的 step 长度切片，在每个切片中只保留值最大的一个峰值点
function [out_x, out_y] = x_fil(peak_x, peak_y, step)
    assert(length(peak_x) == length(peak_y));
    out_x = [];
    out_y = [];
    _start = peak_x(1);
    _maxy = peak_y(1);
    _maxx = _start;

    for i = 1:length(peak_x)

        if (peak_x(i) - _start) ≤ step

            if (peak_y(i) > _maxy)
                _maxy = peak_y(i);
                _maxx = peak_x(i);
            end

        else
            out_x = [out_x, _maxx];
            out_y = [out_y, _maxy];
            _start = peak_x(i);
            _maxy = peak_y(i);
            _maxx = _start;
        end

    end

    out_x = [out_x, _maxx];
    out_y = [out_y, _maxy];
```

```
end

[peakx_filtered_2, peaky_filtered_2] = x_fil(peakx_filtered, peaky_filtered,
30000);
subplot(414); plot(peakx_filtered_2, peaky_filtered_2, 'o');
```

逐个音程画图，外加没做完的音符识别

5.2.m

```
% 需要先执行过 5.1
close all
% 假设 music 是原音频的行向量，peakx_filtered_2 是音符的开始点向量
% 1. 取实际长度的 2/3
actual_length_factor = 2/3;
actual_lengths = round(diff([peakx_filtered_2, length(music)]) *
actual_length_factor);

% 2. 乘窗函数：在实际取到的音程段上乘以相同点数的窗函数
window = hann(max(actual_lengths));

% 从频域获取音符
function note_symbols = get_note_symbols(fft_frequency, fft_result)
    % 找到前两个幅度最大的频率索引
    [~, sorted_indices] = sort(abs(fft_result), 'descend');
    max_indices = sorted_indices(1:2);

    % 获取对应的频率
    max_frequencies = fft_frequency(max_indices);

    % 使用 A4（钢琴中央 C 的 440Hz）作为参考频率，计算音符符号
    reference_frequency = 440; % 可以根据需要调整
    note_numbers = round(12 * log2(max_frequencies / reference_frequency) +
69);

    % 使用数字谱进行映射（A0 是 1，C8 是 88）
    note_symbols = num2str(note_numbers - 20);
end

for i = 1:length(peakx_filtered_2)
    segment = music(peakx_filtered_2(i):(peakx_filtered_2(i) +
actual_lengths(i) - 1));
    processed_music = segment .* window'(1:length(segment));

    fft_points = 2^nextpow2(length(segment)); % 选择最接近窗口大小的 2 的幂次方作为
FFT 点数
    fft_result = fft(processed_music, fft_points);
    fft_result = abs(fft_result);

    % 计算频率轴
    fft_frequency = (1:fft_points) * Fs / fft_points;
    get_fre = 2000;
    fft_frequency = fft_frequency(1:get_fre);
    fft_result = fft_result(1:get_fre);

    if mod(i - 1, 16) == 0
        figure;
    end
```

```
    subplot(4, 4, mod(i - 1, 16) + 1); plot(fft_frequency, fft_result);
title(strcat('频谱', int2str(i))); xlabel('频率 (Hz)'); ylabel('幅度'); grid on;
    get_note_symbols(fft_frequency, fft_result)

end
```

短时傅里叶谱图

5.3.m

```
% 干净执行
close all
clear
% 画短时傅里叶谱图
[music, Fs] = audioread('../audio/M1_i1.wav');
music = music(:, 1) ./ max(music(:, 1));
F0 = 1; % 观察谱间隔暂定 1 Hz
test_Num = 400; % 画谱图的临时帧长
frame_N = floor(length(music) / test_Num); % 帧数
frame_t = (0:frame_N - 1) * test_Num / Fs; % 谱图横坐标单位调为时间(s)
fft_N = Fs / F0; % 每一帧 FFT 的点数，暂设为 44100 点
frame_f = (0:floor(fft_N - 1)) * F0; % 谱图纵坐标单位为 Hz
frame_A = zeros(fft_N, frame_N); % frame_A 记录某一帧的某一频率处的值，后面用颜色显示

% 计算短时傅里叶谱图
for i = 1:frame_N
    % 每一帧 400 点，均乘 400 点哈明窗后，做 44100 点 FFT
    start_idx = (i - 1) * test_Num + 1;
    end_idx = i * test_Num;
    fft_result = fft(music(start_idx:end_idx) .* hamming(test_Num), fft_N);
    frame_A(:, i) = abs(fft_result);
end

% 绘制短时傅里叶谱图
figure(5); subplot(211);
imagesc(frame_t, frame_f, frame_A);
axis xy;
M_T = ceil(length(music) / Fs);
axis([0 M_T 0 2000]);
```

不同帧长下的频谱泄露

5.4.m

```
% 前置：5.1, 5.3（非干净执行，需要修改 5.3）
% 关闭所有图窗，开始索引设为第一个峰值对应的帧数
close all;
starti = peakx_filtered_2(1) * Frame_Num;

% 设置测试信号的长度
test_N = 400;

% 提取测试信号
i = starti:starti + test_N;
test_music = music(i);

% 绘制测试信号时域波形
subplot(331);
```

```
plot(test_music);
title('实际长度：400 点');

% 计算测试信号的 FFT 并绘制矩形窗谱图（44100 点 FFT）
test_music_FFT = fft(test_music, 44100);
subplot(334);
stem(abs(test_music_FFT) / max(abs(test_music_FFT)));
title('矩形窗谱图，44100 点 FFT');
axis([0 1000 0 1]);

% 计算测试信号的 FFT 并绘制哈明窗谱图（44100 点 FFT）
test_music_FFT = fft(test_music .* hamming(length(test_music)), 44100);
subplot(337);
stem(abs(test_music_FFT) / max(abs(test_music_FFT)));
title('哈明窗谱图，44100 点 FFT');
axis([0 1000 0 1]);
```

钢琴音频节奏评价

5.5.m

```
% 前置：5.1
close all
Frame_Num = 100;
% 定义标准节拍和初始化实际节拍和误差
hold on;
N = 42;
ideal_Beat_T = [1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1
1 1 2 1 1 1 1 1 1 2];
Real_Beat_T = [];

% 计算实际节拍和误差
for i = 1:N - 1
    Real_Beat_T = [Real_Beat_T (peakx_filtered_2(i + 1) - peakx_filtered_2(i))
* Frame_Num / Fs * 69/60];
end

%归一化
ideal_Beat_T = normalize(ideal_Beat_T, "range");
Real_Beat_T = normalize(Real_Beat_T, "range");
Err_Beat_T = Real_Beat_T - ideal_Beat_T(1:end - 1);

% 绘制图表
subplot(2, 1, 1);
plot(1:N - 1, Real_Beat_T, 'bo');
hold on;
plot(1:N - 1, ideal_Beat_T(1:N - 1), 'r*');
grid on;
title('理论节拍（红）与实际节拍（蓝）');

subplot(212);
plot(1:N - 1, Err_Beat_T, 'bx');
grid on;
title('误差率');
```