MATLAB 高级编程与工程应用

语音合成大作业

BobAnkh

1. 语音预测模型

(1) 滤波器

对方程作 Z 变换，可以得到传递函数：

$$\mathrm{H(z)} = \frac{\mathrm{S(z)}}{\mathrm{E(z)}} = 1 - \mathrm{a_1 z^{-1}} - \mathrm{a_2 z^{-2}}$$

则可由下述公式求共振峰频率：

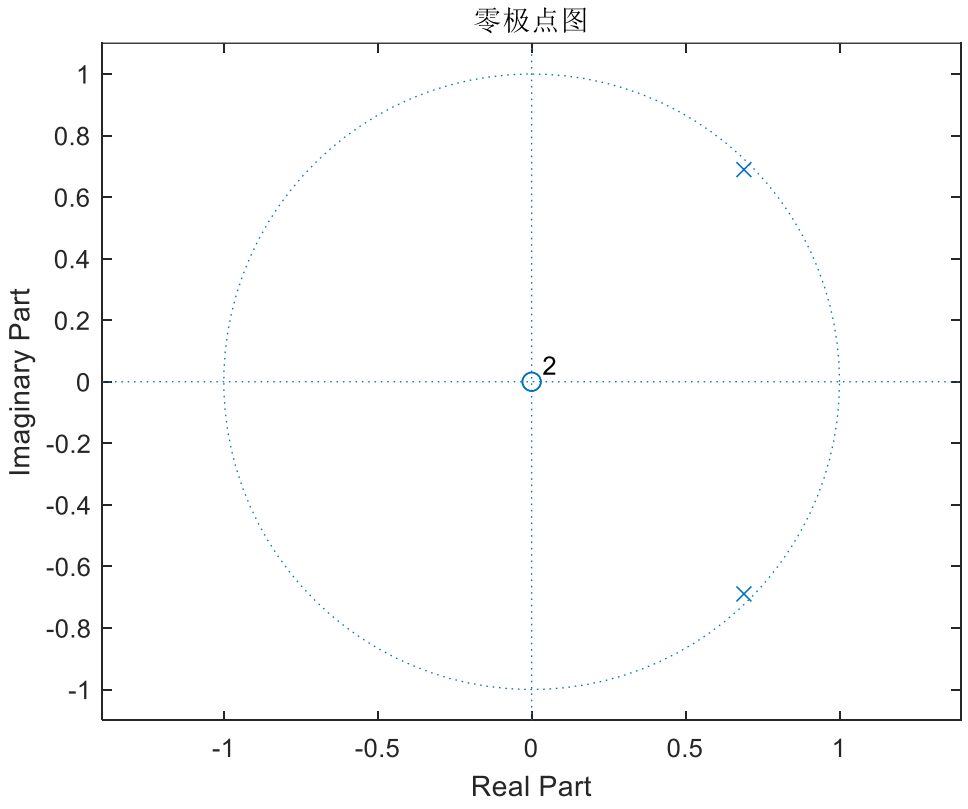
$$\mathrm{f} = \frac{\omega}{2\pi} = \frac{\Omega}{2\pi T} = \frac{abs\left(angle(p_i)\right)}{2\pi T}$$

其中，p_i是传递函数的极点，即Ω为极点幅角绝对值（振荡频率由其决定），T为采样周期，根据本实验说明中相关信息，采样是以每 10ms 采 80 个点进行的，故$T = 0.125ms$

使用 roots 函数求解出极点，根据上述计算式使用 matlab 计算得到共振峰频率：

$$\mathrm{f} = 999.9447\ \mathrm{Hz}$$

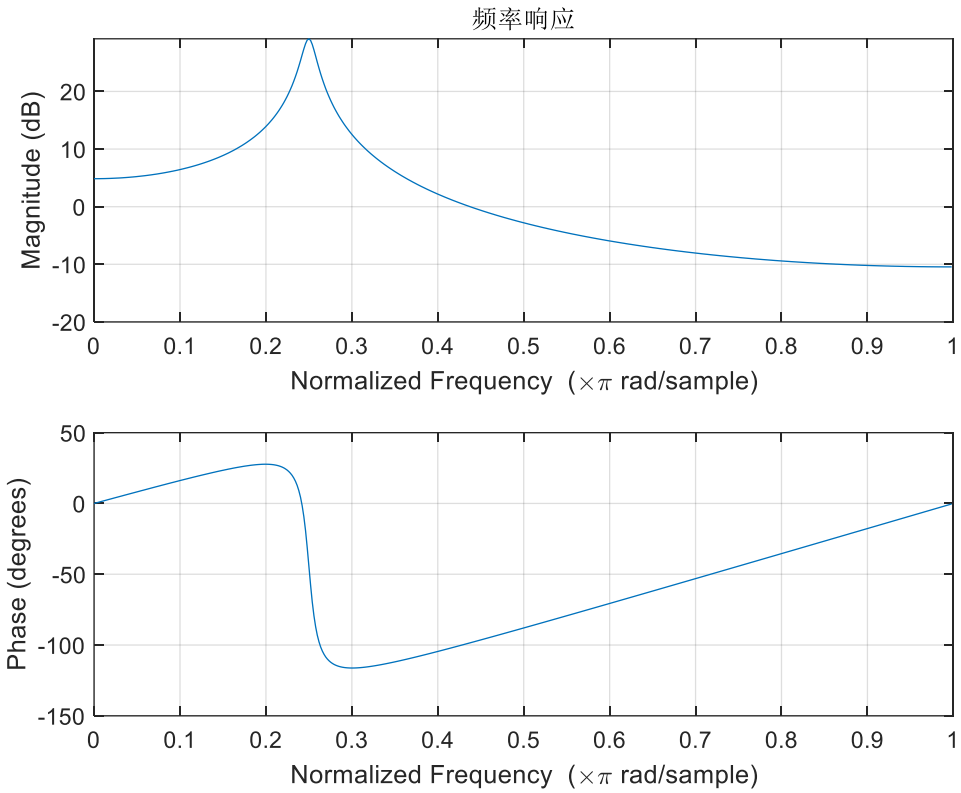
用 zplane 绘制零极点图如下：



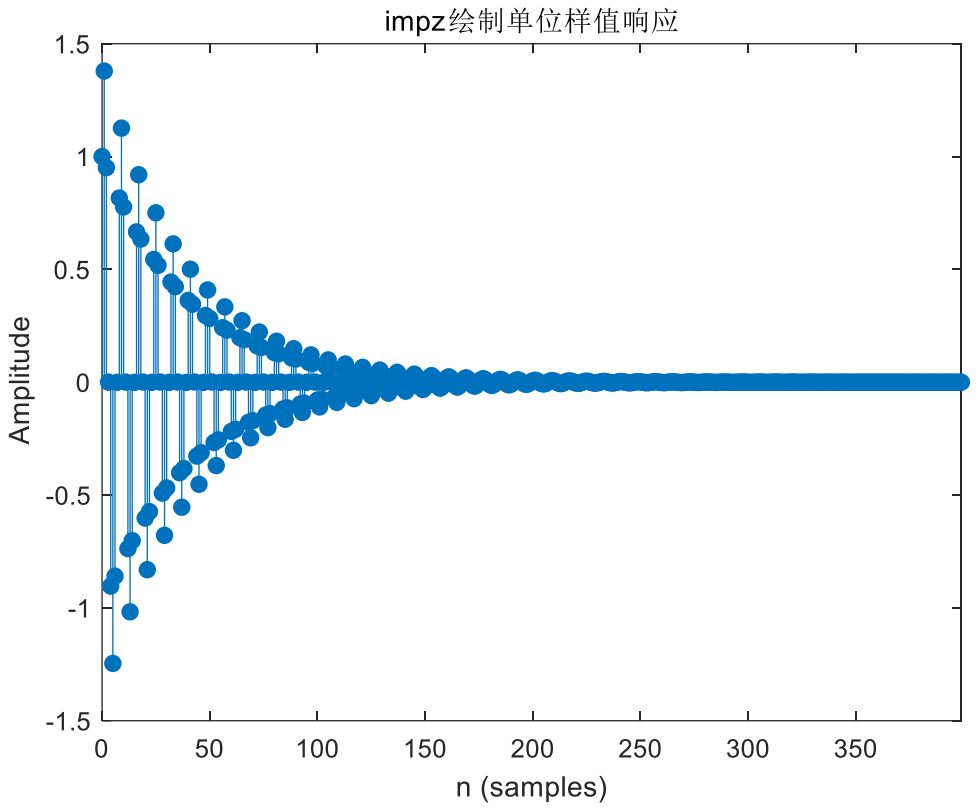
由零极点图可以看到原点处有一个二阶零点，而在靠近单位圆处，$\pm\frac{\pi}{4}$相位的位置上，

各有一个极点，而在这两个角度上，也正是共振峰处，并通过计算也可以印证此前共振峰频率计算的正确性。

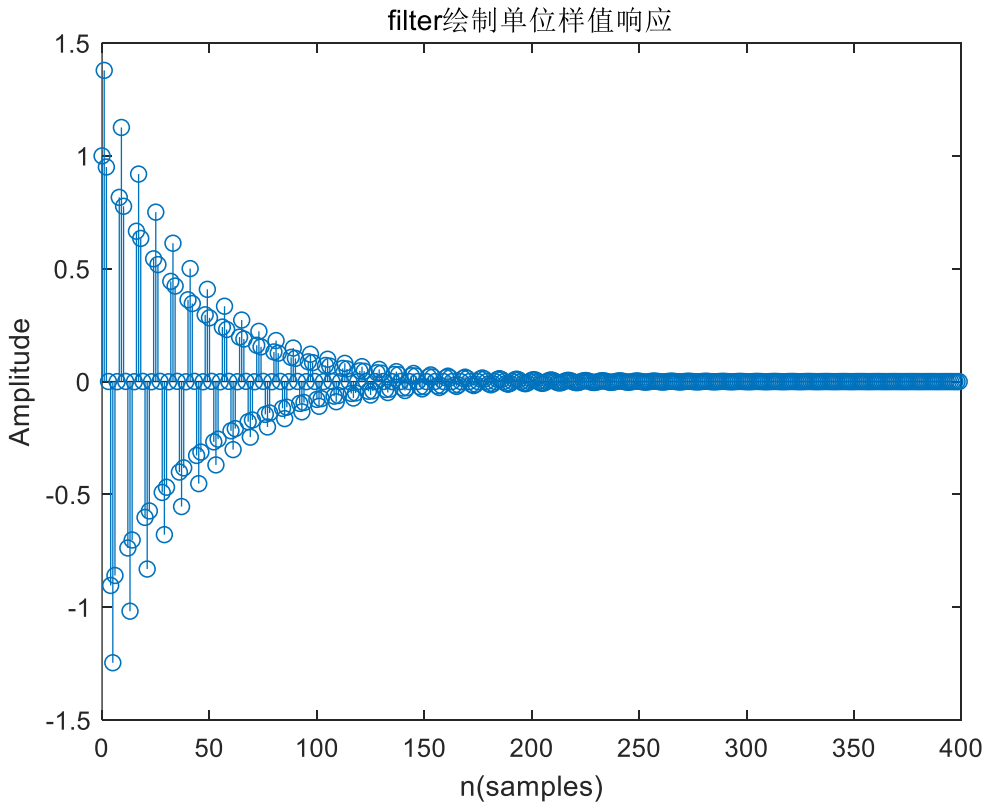
用 freqz 绘制频率响应如下：



用 impz 绘制单位样值响应如下：



用 filter 绘制单位样值响应如下：



使 filter 和 impz 均绘制 400 个点，比较发现用 filter 和用 impz 绘制得到的单位样值响应是一样的。

本题完整代码见 hw2_1.m。

(2)通过阅读理解 speechproc 的基本流程

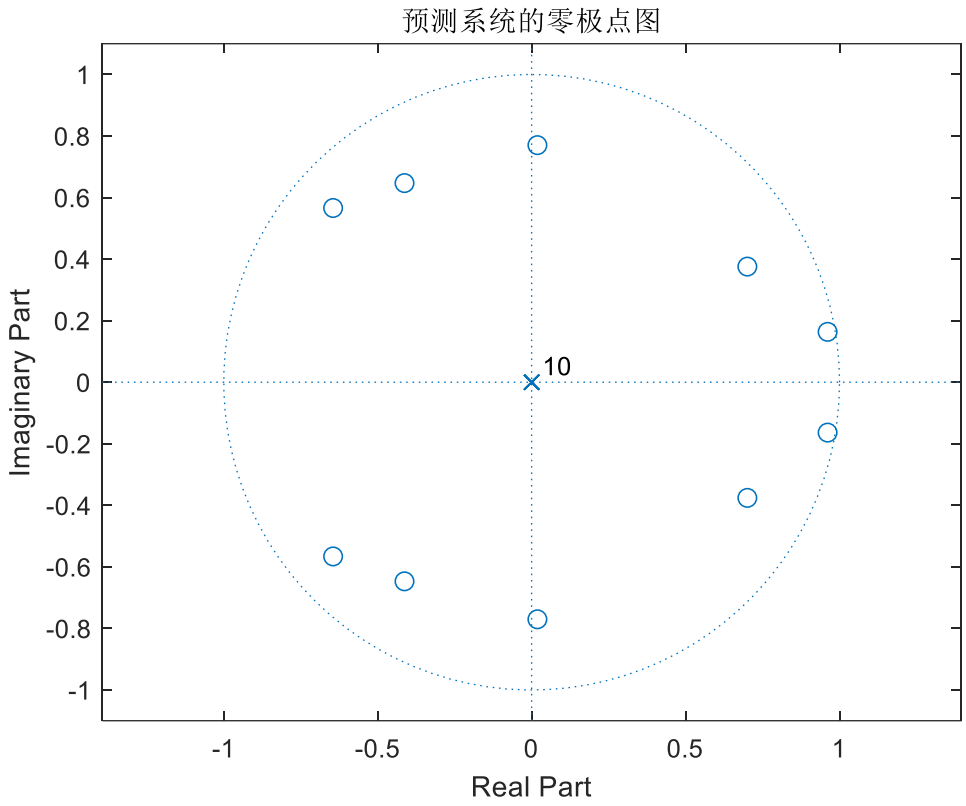
a) 定义相关参数：帧长、窗长、预测系数个数
b) 载入语音并计算帧数，定义各滤波器的状态、输入和输出
c) 采用循环依次处理每帧语音
   i. 加窗之后计算预测系数
   ii. 观测 27 帧时，预测系统的零极点图
   iii. 利用 filter 函数计算出激励并存入 exc 中
   iv. 利用 filter 函数和上一步计算出的 exc 重建语音，存入 s_rec 中
   v. 计算基音周期 PT 和合成激励的能量 G
   vi. 生成合成激励 exc_syn，并利用其和 filter 函数生成合成语音，存入 s_syn 中
   vii. 将合成激励的长度增加一倍得到 exc_syn_v，再次利用 filter 函数生成变速不变调的语音，存入 s_syn_v 中
   viii. 将基音周期减小一半，共振峰频率增加 150Hz，得到合成激励 exc_syn_t，再次利用 filter 函数生成变速不变调的语音，存入 s_syn_t 中
d) 试听语音
e) 保存所有文件

(3)在 27 帧时观察零极点图

此时输入是s(n)，输出是e(n)，所以使用zplane(b,a)函数绘图时，根据语音模型的公式和相关 matlab 代码可知，A即是此函数参数中的b，而此函数参数中的a为 1，由此使用如下代码绘图：

```
figure;
zplane(A, 1);
title("预测系统的零极点图");
```

绘制的零极点图如下：



可以看到原点处为 1 个十阶极点，而圆内还有着 5 对（10 个）关于实轴对称的零点。预测模型作为声道模型的逆系统，具备这样的零极点分布是符合预期的。

(4)用 filter 函数计算激励信号e(n)

filter 函数接受描述系统的参数 A，1，语音信号 s_f 和状态 zi_pre，输出激励 exc 并更新状态 zi_pre，相关代码如下：

```
    [exc((n-1)*FL+1:n*FL), zi_pre] = filter(A, 1, s_f, zi_pre);
```

(5)用 filter 函数计算重建信号s_rec(n)

与上一小题类似，输出重建语音信号 s_rec 并更新状态 zi_rec，相关代码如下：

```
[s_rec((n-1)*FL+1:n*FL), zi_rec] = filter(1, A, exc((n-
1)*FL+1:n*FL), zi_rec);
```

(6)试听信号并对比绘图

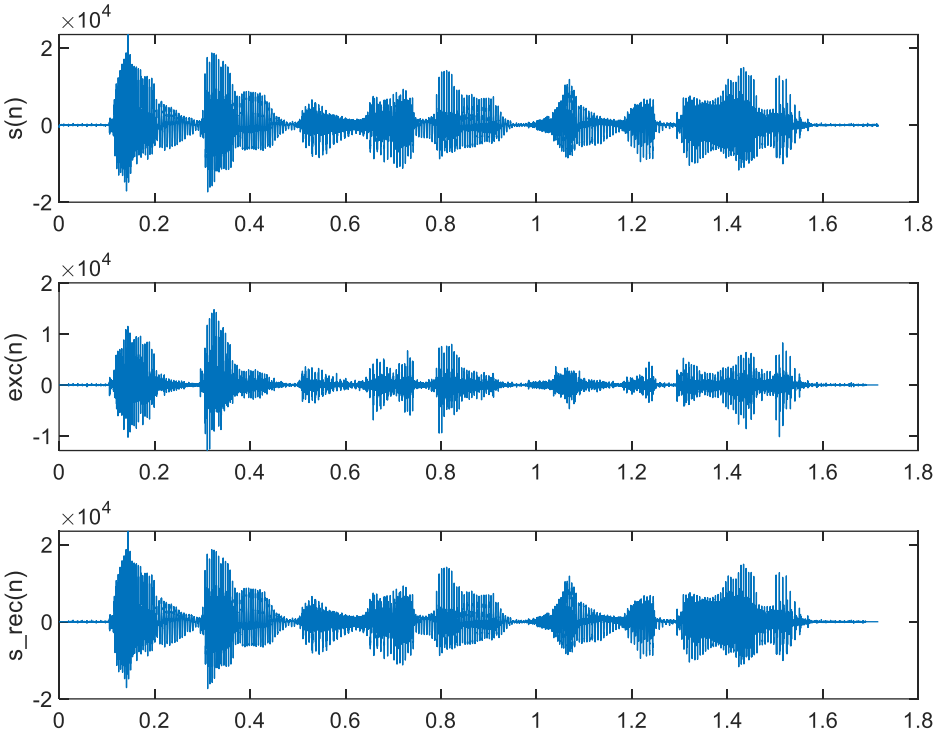
使用 sound 函数来听这三个信号，查阅 matlab 相关文档可知，需要将音频信号的幅值压缩到[-1, 1]的区间内来取得更好的播放效果。而此音频本身是 16 位深度的，所以只需要将信号幅度除以 32768 即可得到符合要求的信号。经过尝试发现，不除该值会使得音频播放出来充斥着大量的背景杂声，而除去改值后音频播放则比较正常。采用 8000Hz 采样频率进行

播放。相关代码如下：

```
sound([s;exc;s_rec] / 32768, 8000);
```

主观上听起来，原始语音信号和重建信号听起来基本相同，察觉不出差异，而且是清晰自然的人音；而激励信号则听起来音量轻一些，而且更像是老式收音机的声音，音色有明显颗粒感，同时杂音也多一些。

利用 plot 函数绘制 3 个信号的时域波形图：

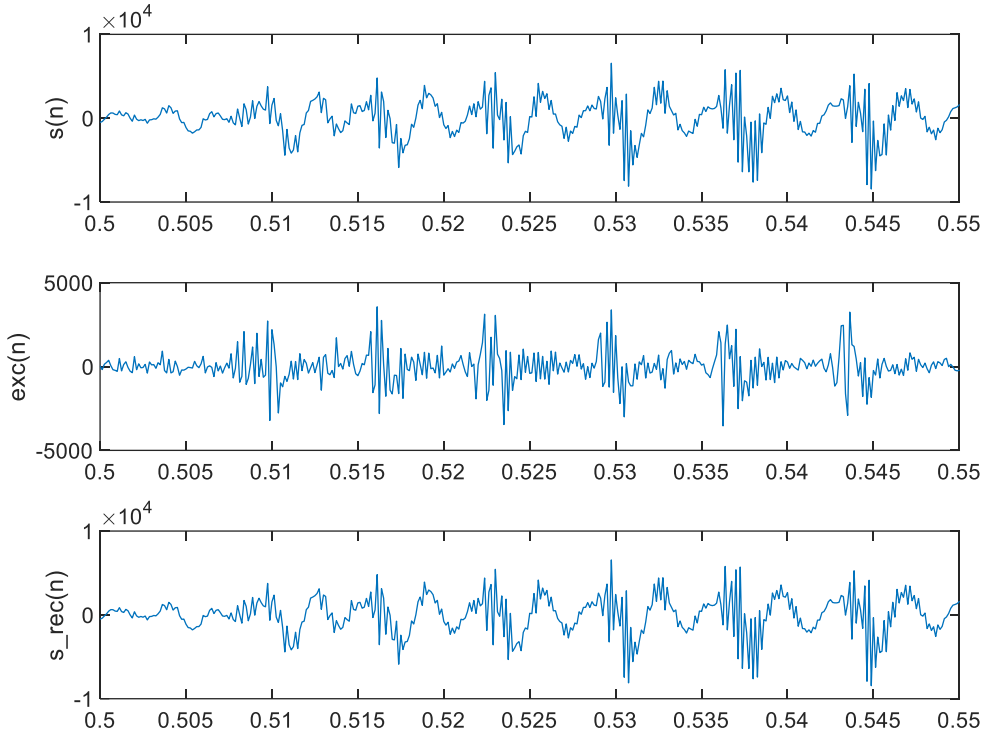


可以使用如下代码来查看 0.5s~0.55s 的细节波形：

```
set(gca, 'XLim', [0.5, 0.55]);
```

（注：在提交的代码中将上述查看细节波形的代码注释了，如有需要取消注释即可）

3 个信号的细节波形如下：



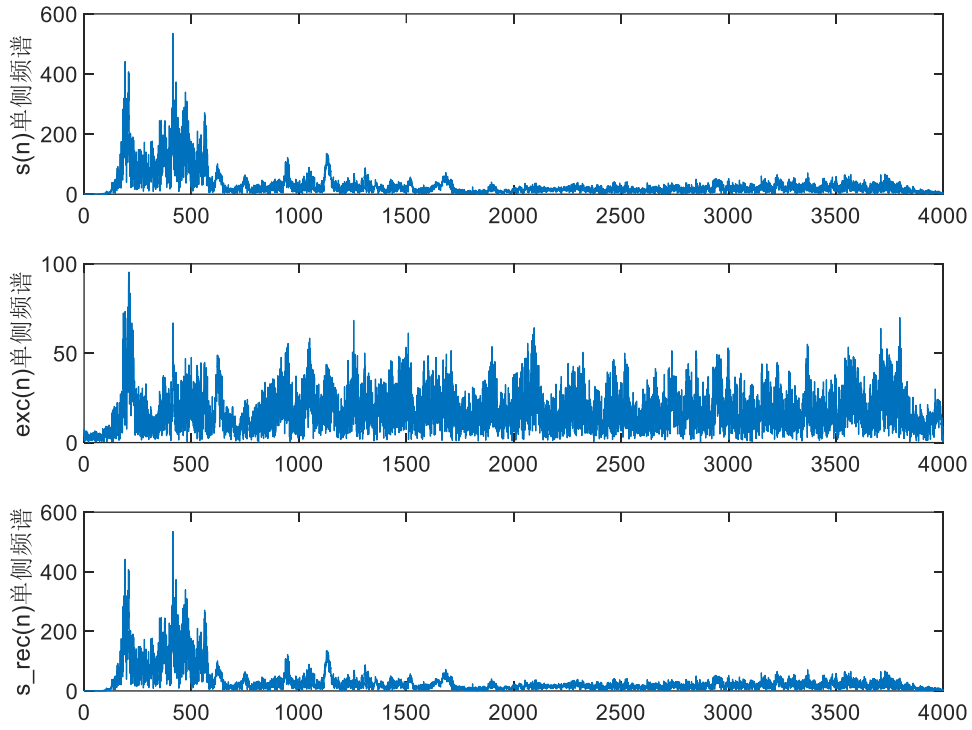
由上述多张时域波形图像可以看到，原始语音信号s(n)和重建信号ŝ(n)的波形基本上完全一致，而激励信号e(n)则相对而言可以看出明显的不同，比较同一时间处，可以看到激励信号e(n)普遍幅度要小一些，这也是其播放时音量要小一些的原因，而且其变化也更频繁一些，推测其应该有更多的高频分量。

观察其频谱异同。根据 matlab 官方文档中示例的方法，采用 fft 变换到频域并得到单侧幅值频谱后用 plot 绘制出，相关代码如下：

```
figure;
fft_freq = 8000 * (0 : (L / 2))' / L;
subplot(3, 1, 1);
S = abs(fft(s) / L);
S = S(1 : L / 2 + 1);
S(2 : end - 1) = 2 * S(2 : end - 1);
plot(fft_freq, S);
ylabel('s(n)单侧频谱');
subplot(3, 1, 2);
EXC = abs(fft(exc) / L);
EXC = EXC(1 : L / 2 + 1);
```

```
EXC(2 : end - 1) = 2 * EXC(2 : end - 1);
plot(fft_freq, EXC);
ylabel('exc(n)单侧频谱');
subplot(3, 1, 3);
S_REC = abs(fft(s_rec) / L);
S_REC = S_REC(1 : L / 2 + 1);
S_REC(2 : end - 1) = 2 * S_REC(2 : end - 1);
plot(fft_freq, S_REC);
ylabel('s\_rec(n)单侧频谱');
```

绘制得到的单侧频谱图如下：



由上图可以看到，s(n)和s_rec(n)的频谱基本一致，而且主要集中在低频分量；而e(n)则相对高频分量较多，且普遍幅值要小一些，这也在一定程度上解释了听感上的差异和音量小的原因。

2. 语音合成模型

(7)生成单位样值串

根据单位样值串表达式和相关信息进行计算：

$$N = \text{round}(\frac{fs}{f}) = \frac{8000}{200} = 40$$

$$NS = \text{round}(\text{duration} * f) = 1 * 200 = 200$$

其中fs是采样频率，f是单位样值串频率，duration是信号持续时间。

为了方便使用，将其封装为一个函数 UnitSample：

```
function s = UnitSample(duration, fs, f)
%UnitSample 生成单位样值串
%   duration: 信号持续时间
%   fs: 采样频率
%   f: 单位样值串频率

s = zeros(round(fs * duration), 1);
NS = round(duration * f);
N = round(fs / f);
ni = 0:NS - 1;
s(ni * N + 1) = 1;
end
```

为了使函数更普适，所以在不少地方使用了 round 函数。

分别生成 200Hz 和 300Hz 的信号并试听，可以明显听出 300Hz 的信号的音调要高一些

生成并试听信号的代码见 hw2_7.m

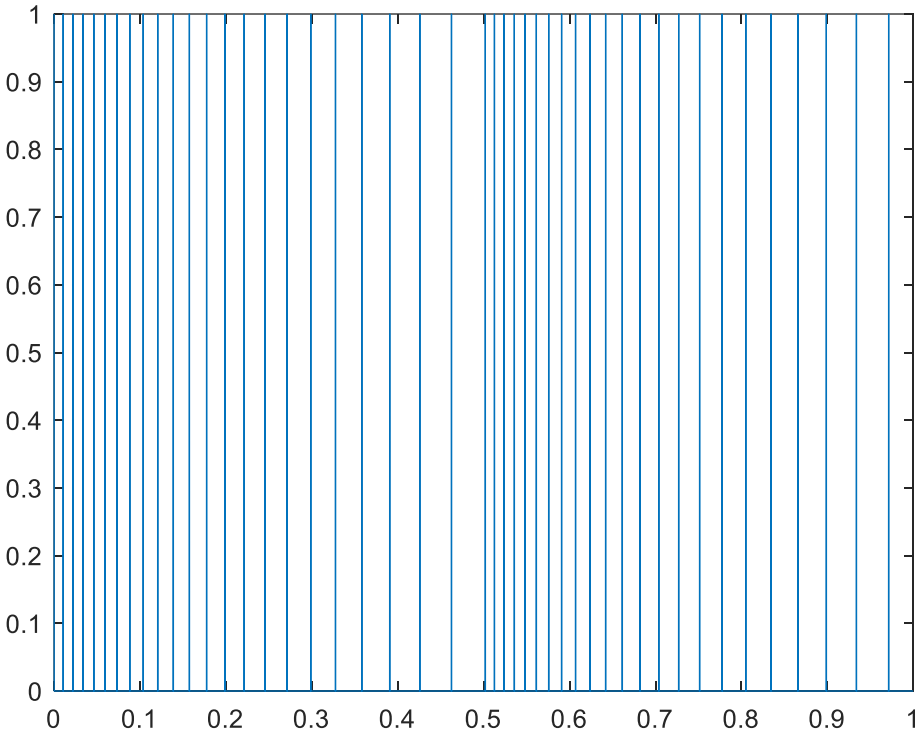
(8)生成随时间变化基音周期的单位样值串

本题的实现思路是：生成一个脉冲，根据当前所处的段序号计算到下一个脉冲点的 PT，然后据此移动到下一个脉冲点处，循环上述过程来生成该单位样值串。为了方便使用，同样将其封装为 1 个函数。相关代码如下：

```
function s = VariedUnitSample(duration, fs)
%VariedUnitSample 生成单位样值串
%   duration: 信号持续时间
%   fs: 采样频率
```

```
s_len = round(fs * duration);
s = zeros(s_len, 1);
seg = round(0.01 * fs);

% 通过循环来生成信号
pos = 1;
while pos <= s_len
    s(pos) = 1;
    m = ceil(pos / seg);
    PT = 80 + 5 * mod(m, 50);
    pos = pos + PT;
end
end
```

生成了 1s 的信号，听起来感觉有点像是通过某些空间的气流音，颗粒感比较足，带有一定电音的感觉，也有一定的节奏感。绘制出波形如下：



可以看出，确实是一个基音周期不断变化的冲激串

生成并试听信号的代码见 hw2_8.m

(9)用激励e(n)通过(1)中滤波器输出s(n)

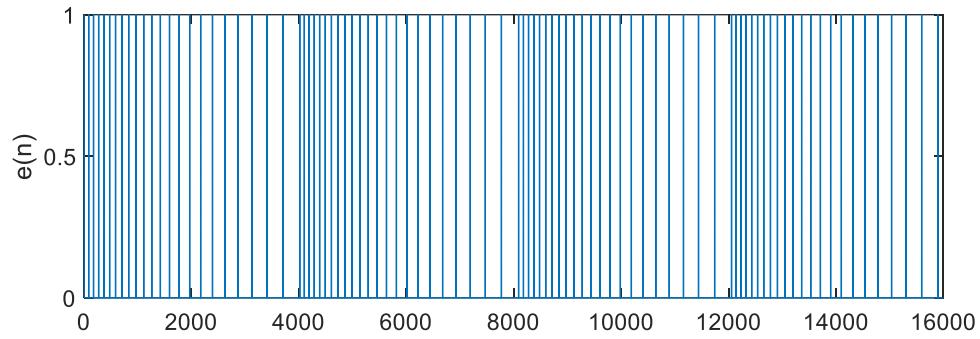
用如下代码生成信号并试听音频：

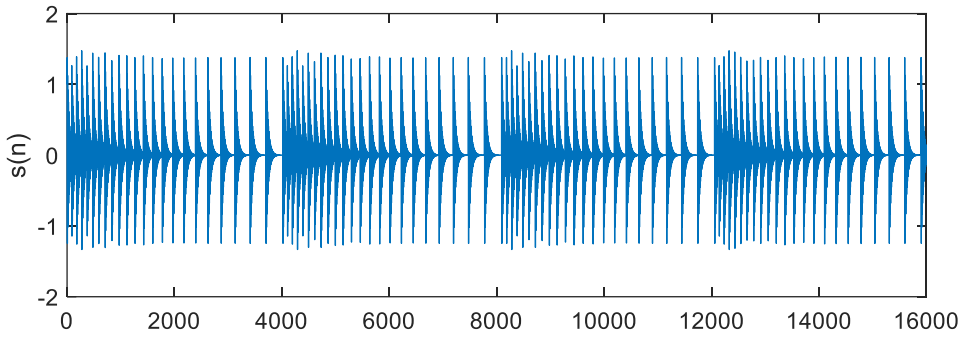
```
fs = 8000;
duration = 2;
% 生成e(n)
en = VariedUnitSample(duration, fs);

a1 = 1.3789;
a2 = -0.9506;
b = 1;
a = [1, -a1, -a2];
% 生成s(n)
sn = filter(b, a, en);
% 试听两信号
sound([en; sn/max(abs(sn))], fs);
```

听起来感觉滤波之后更饱满了，有一种敲木鱼的声音的音质，但是仍然有颗粒感，并且有一种在发出“啊”音的感觉。完整代码见 hw2_9.m

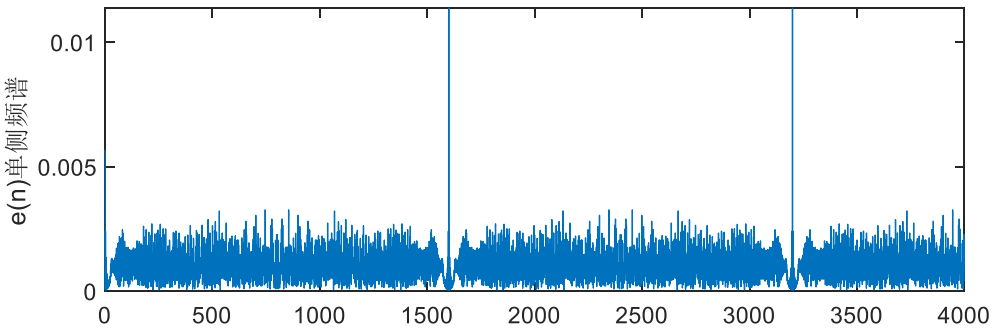
绘制出时域波形和单侧频域波形如下：

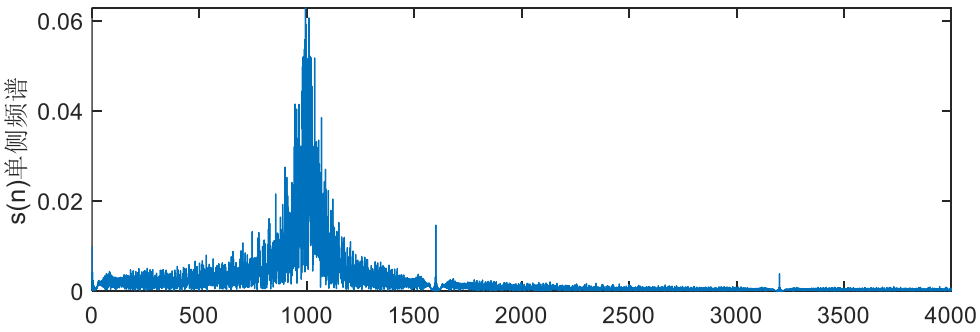
时域波形：





频域波形：





时域上看，s(n)对比e(n)感觉像是添加了包络，使得听感中的颗粒感减少，更加饱满，也可以从频域中看到通过滤波器之后的 s(n)能量主要集中在 1000Hz 左右，低频增强了，考

虑到该滤波器的共振峰频率也在 1000Hz 左右，这样的结果是符合滤波器幅频特性的。

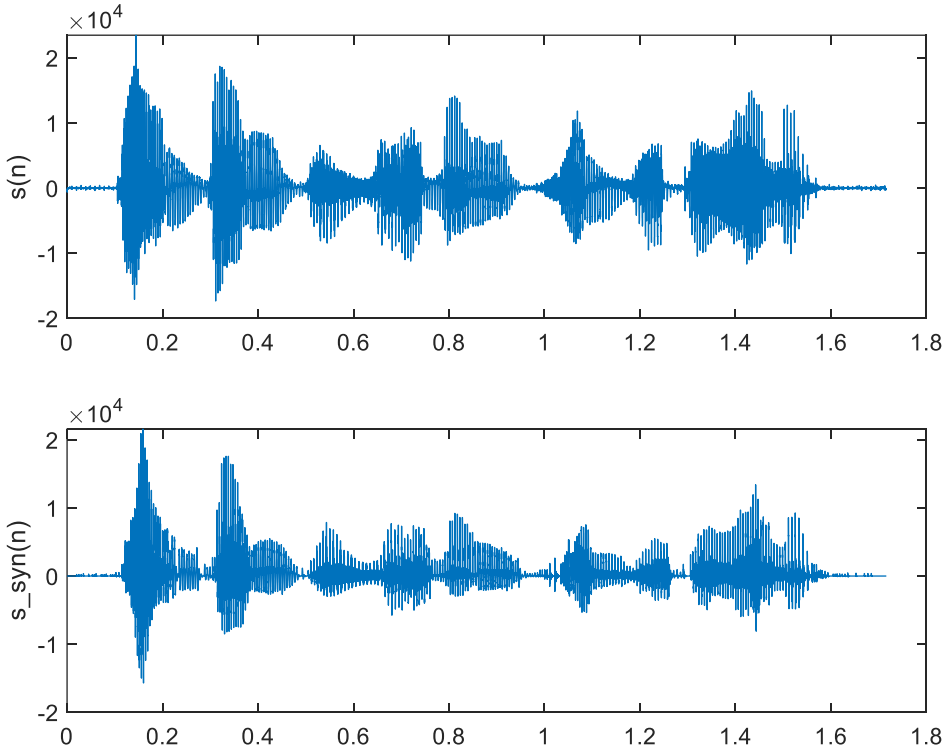
(10)　合成语音

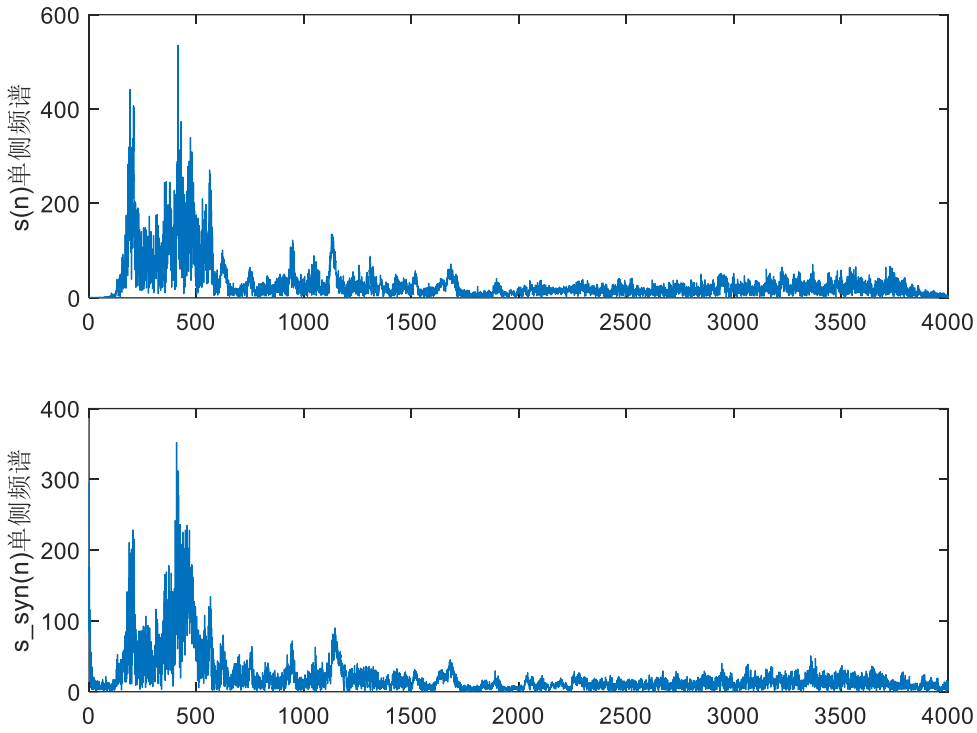
参考了(8)中的做法实现了(10)，在代码开头定义了保存滤波器状态的变量和激励信号的起始位置。循环生成激励并得到合成语音信号的相关代码如下：

```
while pos <= n * FL
            exc_syn(pos) = G;
            pos = pos + PT;
        end
        [s_syn((n-1)*FL+1:n*FL), zi_syn] = filter(1, A, exc_syn((n-
1)*FL+1:n*FL), zi_syn);
```

试听相关语音，感觉合成语音和原始语音基本差不多，听感上合成语音稍微带有一些颗粒感。

绘制出时域波形和单侧频谱波形如下：





可以看到，频域分布基本上是类似的，这也是听感上没有太大差异的原因，但是看时域波形仍然有些差异，我认为这也是相较于原始语音仍存在颗粒感的原因。

3. 变速不变调

(11) 合成变速不变调的语音

与第(10)题类似方式，生成变速不变调的语音，相关代码如下：

```
while pos_v <= n * FL_v
            exc_syn_v(pos_v) = G;
            pos_v = pos_v + PT;
        end
        [s_syn_v((n-
1)*FL_v+1:n*FL_v), zi_syn_v] = filter(1, A, exc_syn_v((n-
1)*FL_v+1:n*FL_v), zi_syn_v);
```

试听起来感觉声调基本没有变化，而速度慢了一倍，同时颗粒感也更加明显了。

同样生成了该信号的时域波形和单侧频谱，与变调不变速的语音信号的相关图像放在一起，图见后文。

4. 变调不变速

(12) 提高共振峰频率

由第(1)题中公式可知，只需要改变极点的幅角即可实现共振峰频率的提高。所以从系统参数中获得极点后，根据其原幅角正负乘上$e^{j\theta}$或$-e^{j\theta}$实现改变，相关代码如下：

```
a1 = 1.3789;
a2 = -0.9506;
b = 1;
a = [1, -a1, -a2];

[z, p, k] = tf2zpk(b, a);
p = p .* exp(1i * sign(imag(p)) * 150 * 2 * pi / 8000);
[b, a] = zp2tf(z, p, k);
```

由此输出数组 a 的值，可以得到本题所要求的值：

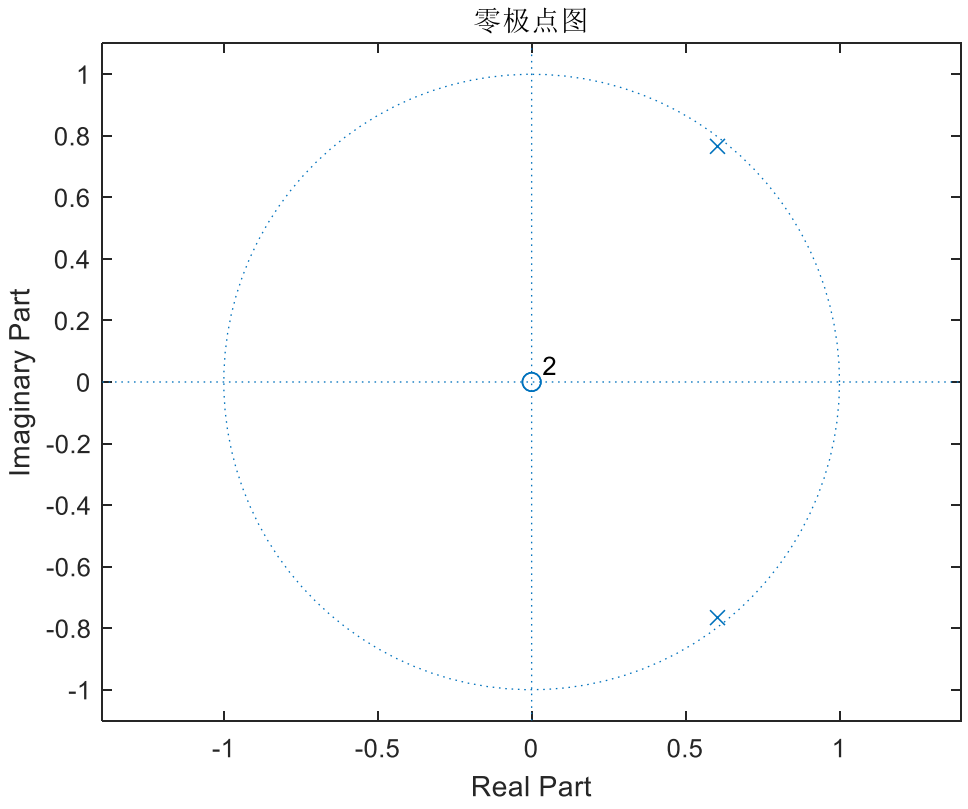
$$a_1 = -1.207284, a_2 = 0.9506000$$

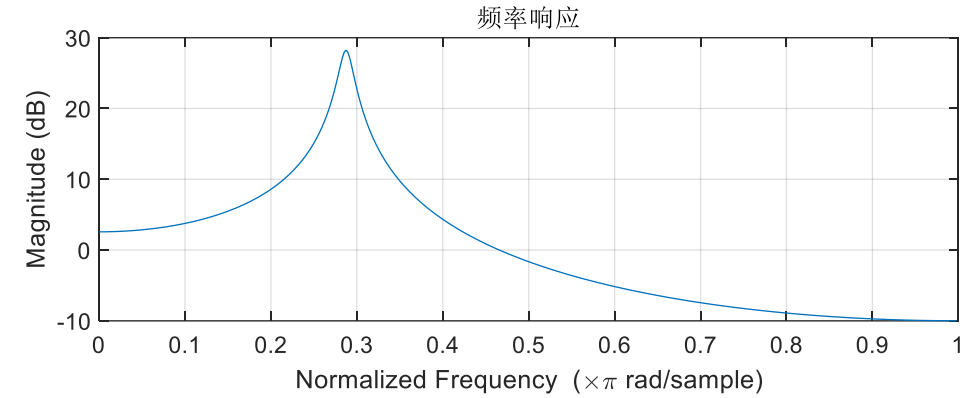
同时检验并输出此时共振峰频率，得到：

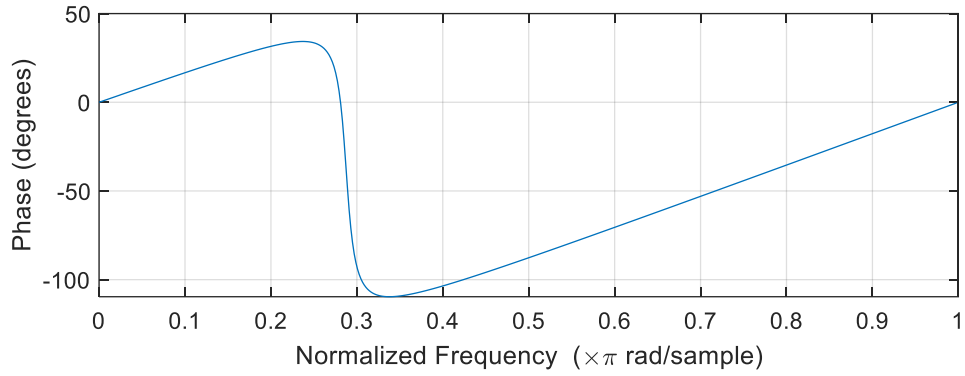
$$f = 1149.9447\ \text{Hz}$$

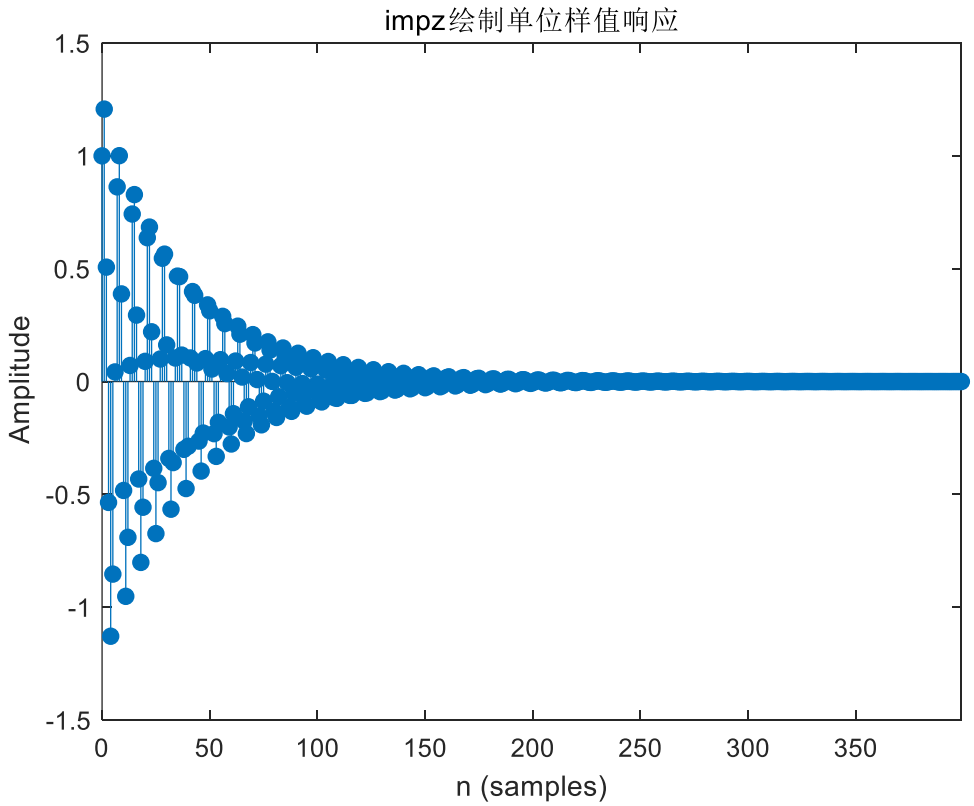
与第(1)题中得到的共振峰频率正好相差 150Hz，符合要求

同样生成其零极点图、频率响应和单位样值响应：









从频率响应图像的变化中也可以较为明显直观地看出共振峰频率的变化。

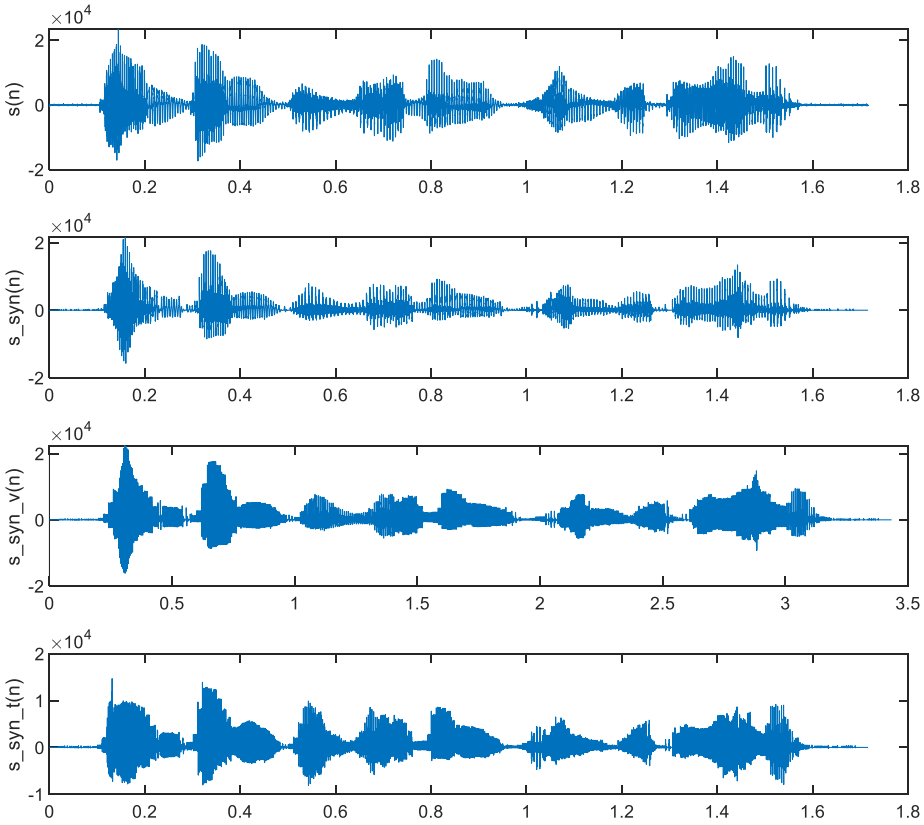
(13)　合成变调不变速的语音

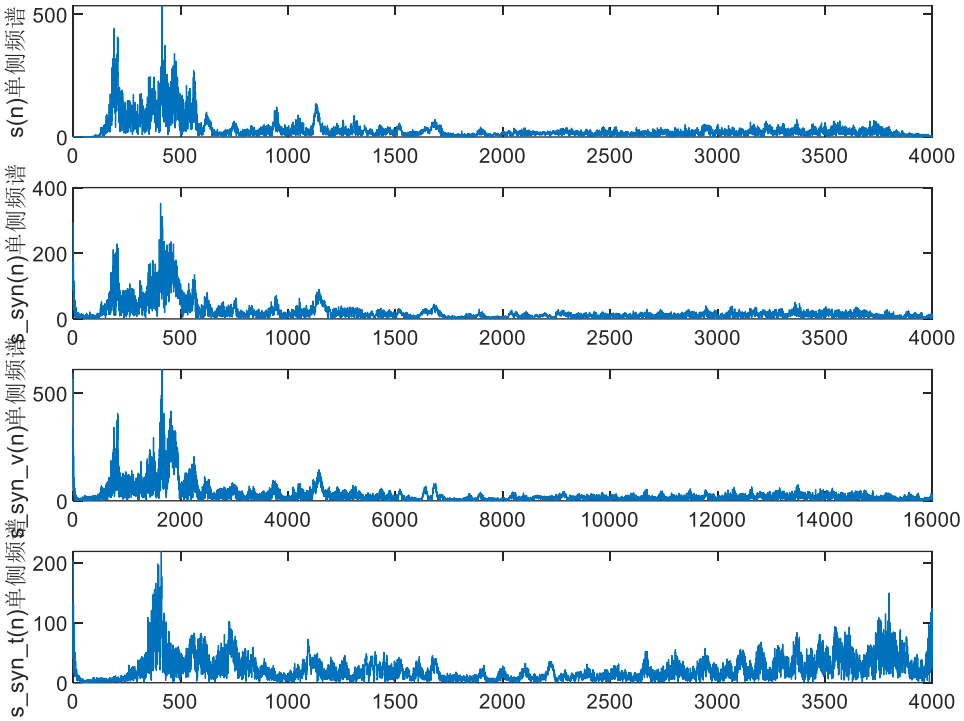
与第(10)题和第(12)采用类似方式，生成变调不变速的语音，相关代码如下：

```
while pos_t <= n * FL
            exc_syn_t(pos_t) = G;
            pos_t = pos_t + round(PT / 2);
        end
        [z, p, k] = tf2zpk(1, A);
        p = p .* exp(1i * sign(imag(p)) * 150 * 2 * pi / 8000);
        [~, A1] = zp2tf(z, p, k);
        [s_syn_t((n-
1)*FL+1:n*FL), zi_syn_t] = filter(1, A1, exc_syn_t((n-
1)*FL+1:n*FL), zi_syn_t);
```

试听该语音，感觉有明显的音调上升，变得更加尖了，但是总体时长和速度与原始语音并没有变化。

同样绘制该信号的时域波形和单侧频谱，与之前的几个信号放在一起呈现，4 个信号从上到下分别是原始语音信号、合成语音信号、变速不变调语音信号、变调不变速语音信号：





观察时域波形，可以看到变速不变调信号的波形虽然持续时间是其他信号的两倍，但是和语音合成信号的波形是基本相同的，这也与其音调没有变化是对应的；而变调不变速信号的波形则显得更为致密，并且波形发生了改变，这与其听感音调变高也是对应的。

观察单侧频谱，可以看到前 3 个信号的频谱基本上还是很相似的，而唯独变调不变速信号的多出了不少高频分量，这与其听起来音调变高也是对应的。

5. 总结

本次实验大部分是建立在 speechproc.m 已有基础上进行的，所以整体进行的比较顺利。本次实验中对于不少 matlab 的函数需要反复使用，比如 filter 函数等，所以对于各个常用函数的各个参数和使用上的一些注意事项也有了比较深入清晰的了解。而且，对于使用 matlab 绘图的操作通过反复练习有了更好的了解和掌握，这也定会有利于将来的学习科研工作。同时，我以前从未接触过语音合成和分析等方面，这一次挑选完成语音合成大作业也是希望借此机会更多地了解一些，而这一个大作业也确实让我对于语音合成的众多方面，包括语音合成、语音预测和语音模型等方面，有了更加清晰的认知，确实有很多收获。

6. 文件附录清单

随本报告提交的附件名称	解释说明
hw2_1.m	第(1)问主调程序
hw2_7.m	第(7)问主调程序
hw2_8.m	第(8)问主调程序
hw2_9.m	第(9)问主调程序
hw2_12.m	第(12)问主调程序
speechproc.m	第(3) (4) (5) (6) (10) (11) (13)问主调程序，第(2)问也与其相关
UnitSample.m	第(7)问需要的函数
VariedUnitSample.m	第(8)问需要的函数
voice.pcm	本大作业需要用到的语音文件
文件夹 save_pcm	本文件夹中包含所有由 speechproc.m 所保存下来的信号的 pcm 文件（也可直接由 speechproc.m 根据 voice.pcm 生成）