

语音合成大作业

漆耘含

2016011058

一、 练习题

1. 给定差分方程， $e(n)$ 是输入信号， $s(n)$ 是输出信号，则滤波器的传递函数是什么？如果 $a_1=1.3789$ ， $a_2=-0.9506$ ，上述合成模型的共振峰频率是多少？用 `zplane`，`freqz`，`impz` 分别汇出零极点图，频率响应和单位样值相应。用 `filter` 汇出单位样值相应，比较和 `impz` 的是否相同。

答：根据差分方程：

$$e(n) = s(n) - a_1s(n-1) - a_2s(n-2)$$

可以得到其传递函数为：

$$H(Z) = \frac{1}{1 - a_1Z^{-1} - a_2Z^{-2}}$$

若 $a_1=1.3789$ ， $a_2=-0.9506$ ，则可利用 `zplane`，`freqz`，`impz` 函数分别绘制出零极点图、频率响应和单位样值响应图，如图 1、2、3 所示：

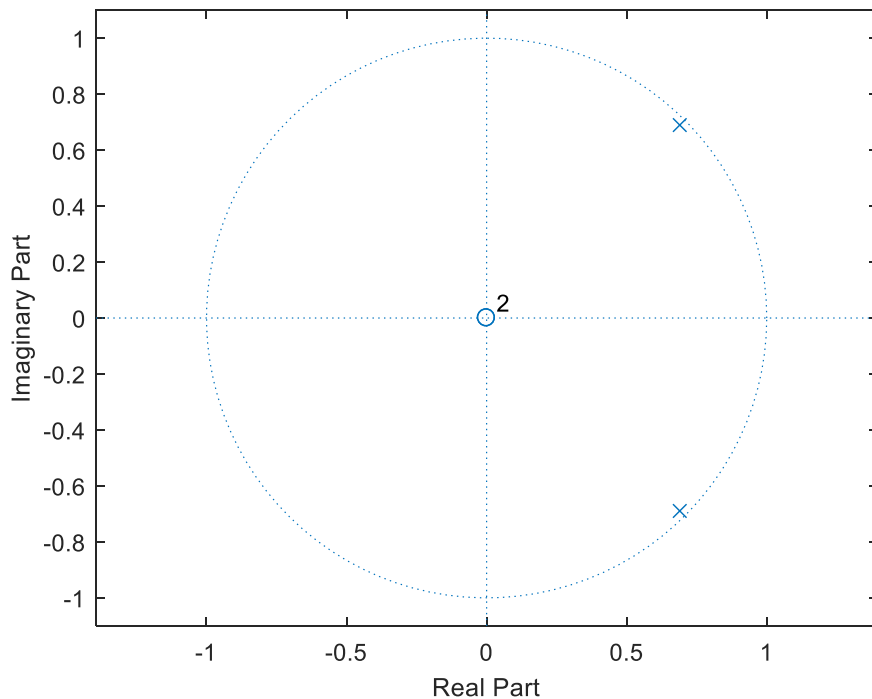


图 1 `zplane` 绘制出的零极点图

在给定差分方程和边界条件之后，可以用 `filter` 绘制出单位样值响应，如图 4 所示：

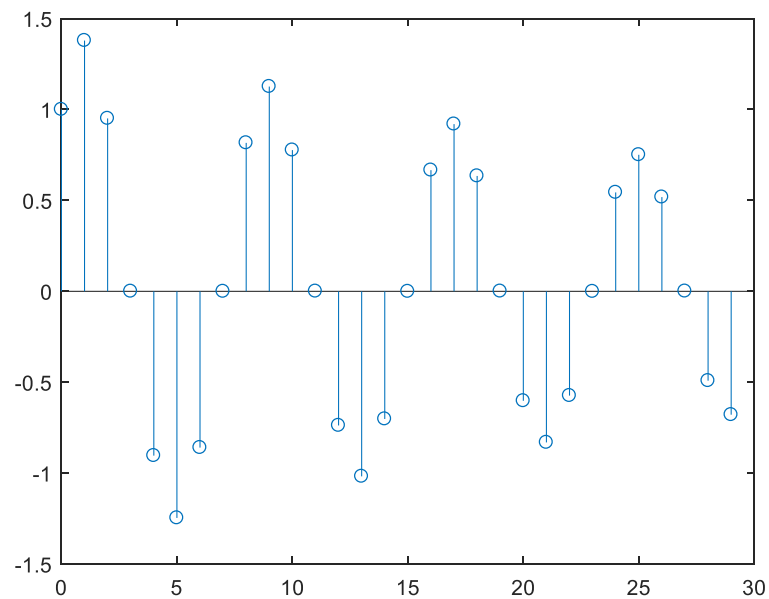


图 4 用 `filter` 绘制出的单位样值响应

从图形上来分析，`impz` 和 `filter` 绘制出的单位样值响应是相同的。

```
代码：
clear;
clc;
syms z;
a1 = 1.3789;
a2 = -0.9506;
H = 1/(1-a1*z^-1-a2*z^-2);
a = [1 -a1 -a2];
b = 1;
```

```

N = 30;
n = 0:1:N-1;
e = (n==0);
figure
zplane(b, a);
%impz(b, a, n);
%hf = filter(b, a, e);
%stem(n, hf);
%freqz(b, a);

```

2. 阅读speechproc.m 程序，理解基本流程。程序中已经完成了语音分帧、加窗、线性预测、和基音周期提取等功能。注意：不要求掌握线性预测和基音周期提取的算法原理。
3. 运行改程序到27帧停住，用（1）中的方法观察零极点图。

答：直接用zplane绘制出零极点图，如图5：

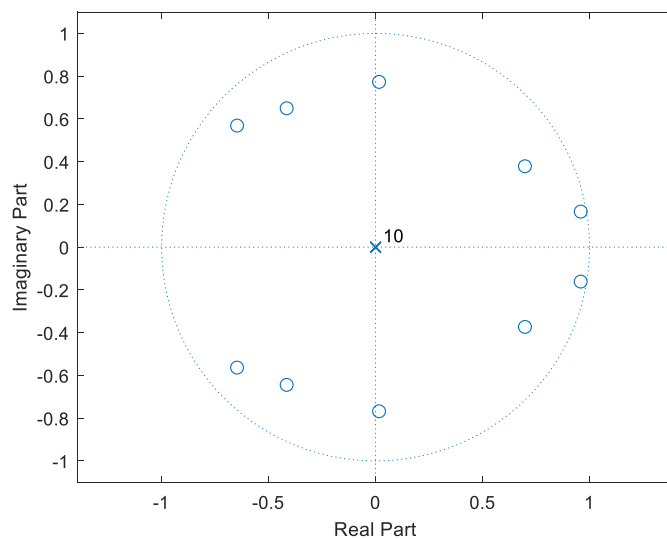


图5 在第27帧时的零极点图

代码:

```
if n == 27

    % (3) 在此位置写程序, 观察预测系统的零极点图
    figure, zplane(A);

end
```

4. 在循环中添加程序: 对每帧语音信号 $s(n)$ 和预测模型系数 $\{a_i\}$, 用 `filter` 计算激励信号 $e(n)$ 。注意: 在系数变化的情况下连续滤波, 须维持滤波器的状态不变, 要利用 `filter` 的 `zi` 和 `zf` 参数。

答:

现在是已知输出求激励, 即相当于把已知输出当做激励, 求逆系统的输出, 则传递函数和原来相比需要上下颠倒一下, 即 `filter` 的 `a` 和 `b` 参数需要反一下。

又因为系数是在不断变化, 因此需要维持滤波器的状态不变, 方法是把上一次滤波的 `zf` 状态当做下一次滤波的 `zi` 状态即可。

代码:

```
s_f = s((n-1)*FL+1:n*FL);    % 本帧语音, 下面就要对它做处理

% (4) 在此位置写程序, 用filter函数s_f计算激励, 注意保持滤波器状态
[exc_1, zf]=filter(A, 1, s_f, zi_pre);
```

```
zi_pre = zf;  
  
% exc((n-1)*FL+1:n*FL) = ... 将你计算得到的激励写在这里  
exc((n-1)*FL+1:n*FL) = exc_1;
```

5. 在循环中添加程序：用你计算得到的激励信号 $e(n)$ 和预测模型系数 $\{a_i\}$ ，用`filter`计算重建语音 $s(n)$ 。同样要注意滤波器的状态不变。

答：

根据第4问中得到的 $e(n)$ ，把它当做激励，重新输入系统中，用`filter`即可实现，保持滤波器状态和第4问的方法相同。

重建的 $s(n)$ 信号，如图6所示：

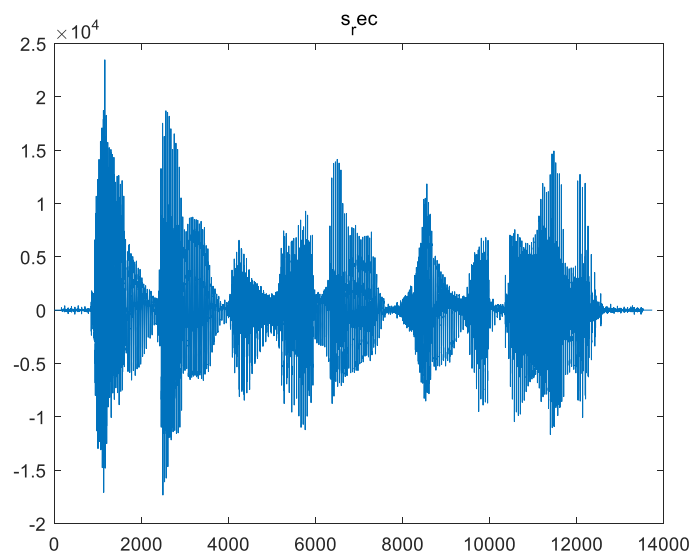


图6 重建的 $s(n)$

代码:

% (5) 在此位置写程序, 用filter函数和exc重建语音, 注意保持滤波器状态

```
[s_r,zf_rec] = filter(1,A,exc_l,zi_rec);
```

```
zi_rec = zf_rec;
```

% s_rec((n-1)*FL+1:n*FL) = ... 将你计算得到的重建语音写在这里

```
s_rec((n-1)*FL+1:n*FL) = s_r;
```

6. 在循环结束后添加程序: 用sound试听 (4) 中的 $e(n)$ 信号, 比较和 $s(n)$ 以及重建的 $s(n)$ 的信号有何区别。对比画出三个信号, 选择一小段, 看看有何区别。

答:

绘制出三者的图形, 如图7所示:

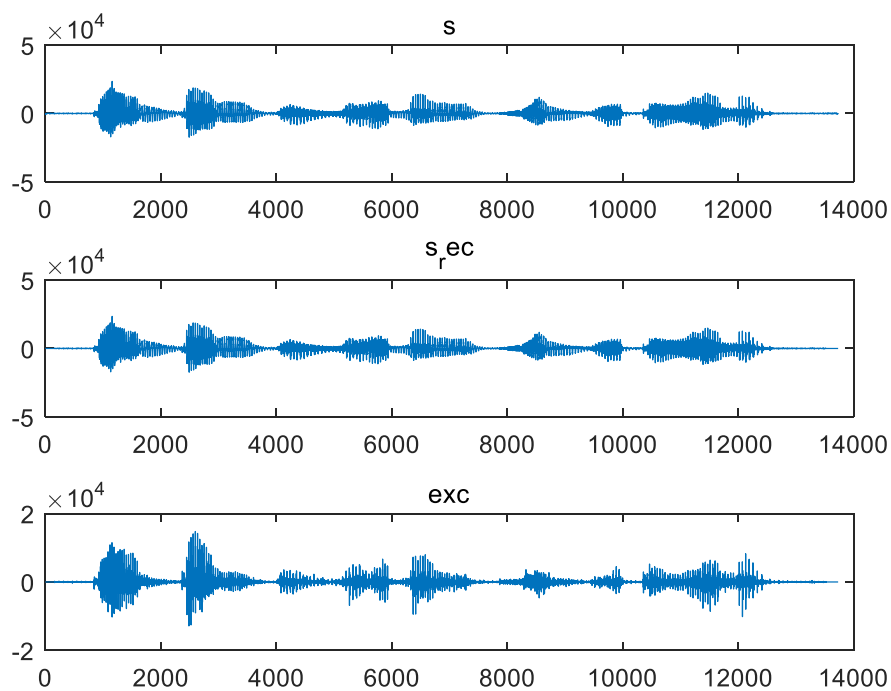


图7 $s(n)$ 、重建 $s(n)$ 、 $e(n)$

从图中可以看出， $e(n)$ 的噪声比较大， $s(n)$ 和重建的 $s(n)$ 噪声比较小。实际用sound听也是如此。

代码：

% (6) 在此位置写程序，听一听 s ， exc 和 s_rec 有何区别，解释这种区别

```
sound (s/max(abs(s)),8000);
%sound (exc/max(abs(exc)),8000);
%sound (s_rec(max(abs(s_rec))),8000);
```

```
figure
subplot(3,1,1),plot(s),title('s');
subplot(3,1,2),plot(s_rec),title('s_rec');
```



```
subplot(3,1,3),plot(exc),title('exc');  
  
sound (s/max(abs(s)),8000);  
sound (exc/max(abs(exc)),8000);  
sound (s_rec/max(abs(s_rec)),8000);
```

7. 生成一个8kHz抽样的持续1秒钟的数字信号，该信号是一个频率为200Hz的单位样值串，即：

$$\mathbf{x}(\mathbf{n}) = \sum_{i=0}^{NS-1} \delta(\mathbf{n} - iN)$$

考虑该信号的N和NS分别为何值？用sound试听这个声音信号。再生成一个300Hz的单位样值串并试听，有何区别？实际上，这个信号是后面要用到的以基音为周期的人工激励信号e(n)。

答：

N应该等于8000/freq，NS等于8000/N=freq

生成的200Hz和300Hz的单位样值串如图8所示：

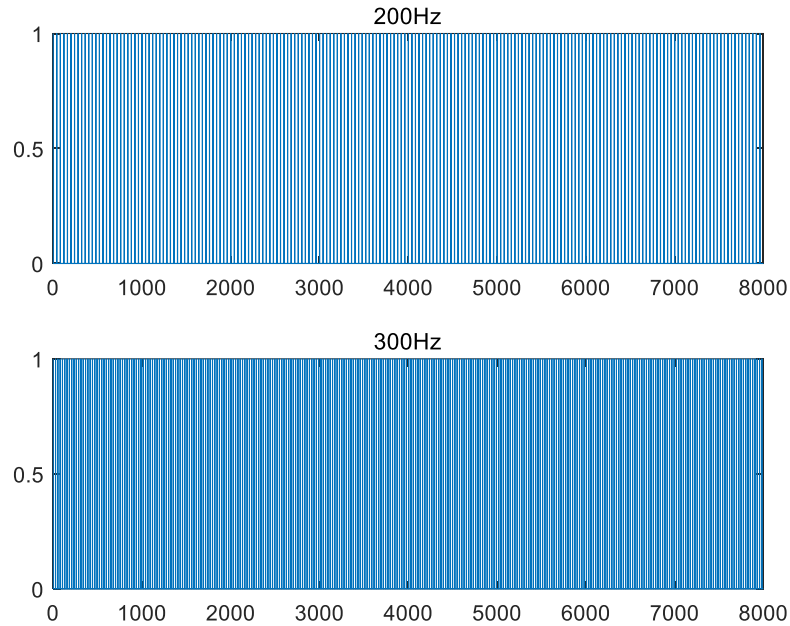


图8 200Hz和300Hz的信号

300Hz的声音比200Hz的声音要高一些。

代码：

% (7) 生成一个8kHz 抽样的持续1 秒钟的数字信号，该信号是一个频率为200Hz 的单

%位样值\串”，用sound 试听这个声音信号。再生成一个300Hz 的单

%位样值\串”并试听，有何区别？事实上，这个信号将是后面要用到的以基音为周期的人工

%激励信号e(n)

```
Fs = 8000;
```

```
Ns1 = 200;
```

```
Ns2 = 300;
```

```
t = 0:1:Fs-1;
```

```
N1 = Fs/Ns1;
```

```
N2 = floor(Fs/Ns2); %取整
```

```

x1=(mod(t,N1) == 0);%每隔N1就取值为1，否则为0
x2=(mod(t,N2) == 0);

figure
subplot(2,1,1),plot(x1),title('200Hz');
subplot(2,1,2),plot(x2),title('300Hz');
%sound(double(x1),8000);
%sound(double(x2),8000);

```

8. 真实语音信号的基音周期总是随着时间变化的。假设每段内基音周期固定不变，但段和段之间不同，具体为：

$$PT = 80 + 5\text{mod}(m, 50)$$

其中PT表示基音周期， m 表示段序号。生成1秒钟的上述信号并试听。

（提示：用循环逐段实现，控制相邻两个脉冲的间隔为其中某个脉冲所在段的PT值）

答：

用一个循环即可实现， $1s=1000ms$ ，段号 $m=n/80$ ，生成的信号如图9所示：

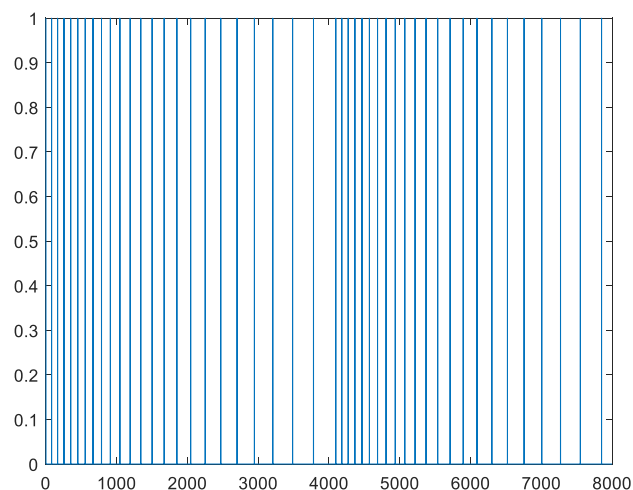


图9 生成的信号

代码:

% (8) 真实语音信号的基音周期总是随着时间变化的。我们首先将信号分成若干个10

%毫秒长的段,假设每个段内基音周期固定不变,但段和段之间则不同,具体为 $PT = 80 + 5\text{mod}(m; 50)$

%其中PT 表示基音周期, m 表示段序号。生成1 秒钟的上述信号并试听。

```
e = zeros(1,Fs);
```

```
n=1;
```

```
while n <=8000    %用循环来实现
```

```
    e(n) =1;
```

```
    m = floor(n/80);%计算段号
```

```
    PT = 80+5*mod(m, 50);%根据段号计算当前基音周期
```

```
    n = n+PT;
```

```
end
```

```
%sound(e);
```

```
%figure,plot(e);
```

9. 用filter将(8)中的激励信号 $e(n)$ 输入到(1)的系统中计算输出 $s(n)$,试听和 $e(n)$ 的区别。

答:

将（8）中的信号当做激励，用（1）中的参数，可得到 $s(n)$ ，如图10所示:

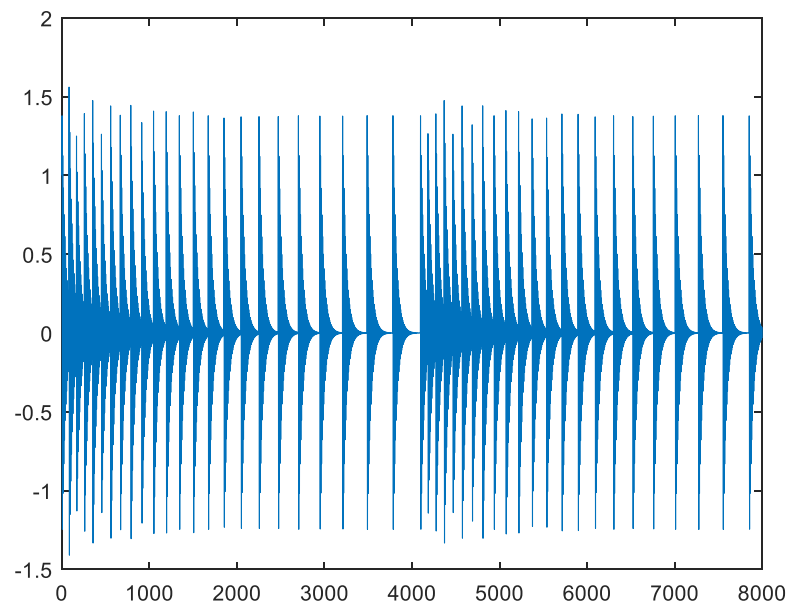


图10 经过（1）系统之后的输出

代码:

% (9) 用filter 将 (8) 中的激励信号 $e(n)$ 输入到 (1) 的系统中计算输出 $s(n)$, 试听和 $e(n)$ 有何区别

```
b = [1];  
a = [1, -1.3789, 0.9506];  
s = filter(b, a, e);  
%sound(s/max(abs(s)));  
%figure, plot(s)
```

10. 利用每一帧计算得到的基音周期和（8）中的方法，生成合成激励信号 $G_x(n)$ ，用filter函数将 $G_x(n)$ 送入合成滤波器得到合成语音。试听和原始语音的差别

答：

和（8）、（9）中的方法一样，并且已知PT，则用一个循环来生成 $G_x(n)$ ，再通过当前的系统得到新生成的 $s(n)$ ，如图11所示：

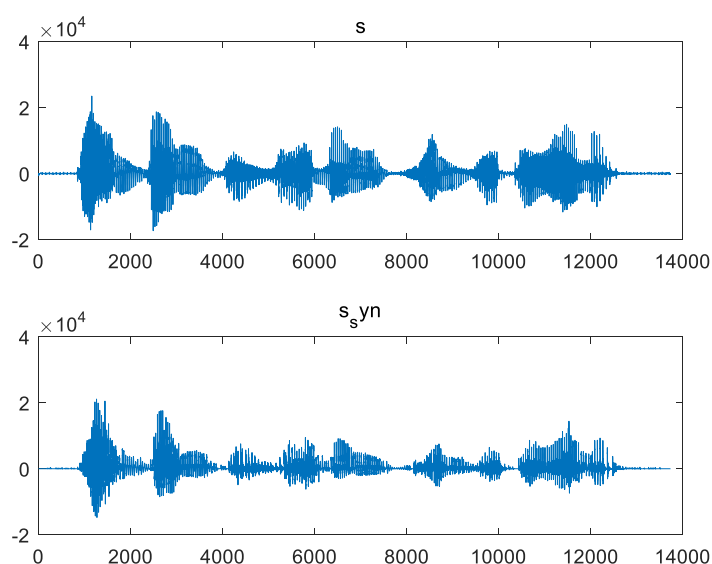


图11 原始信号 $s(n)$ 和新合成的信号 $s_{syn}(n)$

用sound来听，发现原始语音更加清晰，但新合成的语音也很清晰，只是有些许的噪音。

代码：

```
% (10) 在此位置写程序，生成合成激励，并用激励和filter函数产生合成语音
```

```
for i = (n-1)*FL+1:n*FL
```

```

exc_syn(i) = (mod(i,PT)==0)*G; %生成激励信号
end
[s_syn1,zf_s] = filter(1,A,exc_syn((n-1)*FL+1:n*FL),zi_s);
zi_s = zf_s; %保持滤波器状态不变
s_syn((n-1)*FL+1:n*FL) = s_syn1;

sound(s_syn/max(abs(s_syn)));

```

11. 变速不变调：将合成激励的长度增加一倍，再用同样的方法合成出语音。

答：方法是在合成激励信号的时候，合成出两倍长度的激励信号即可。合成信号如图12所示：

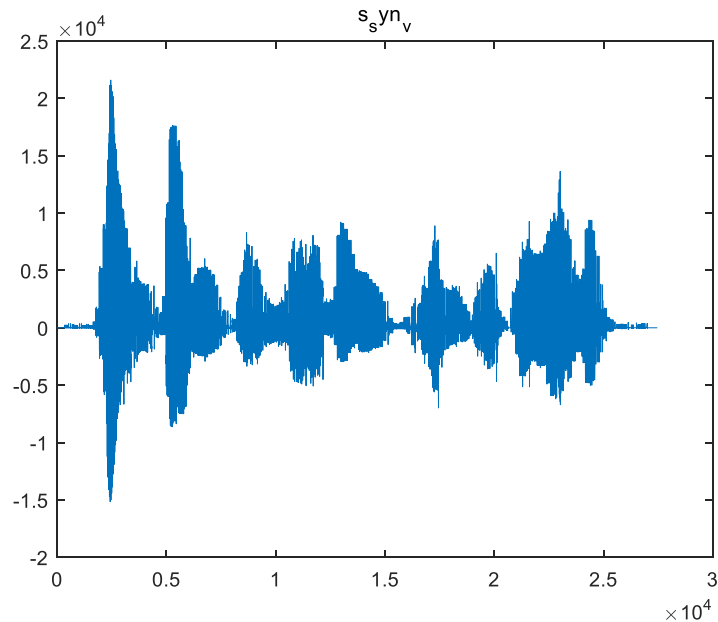


图12 合成的变速不变调信号

代码：

% (11) 不改变基音周期和预测系数，将合成激励的长度增加一倍，再作为filter

% 的输入得到新的合成语音，听一听是不是速度变慢了，但音调没有变。

```
FL_v = FL*2;
```

%因为时间变为两倍，因此生成的激励也应该为两倍长度

```
for i = (n-1)*FL_v+1:n*FL_v
```

```
    exc_syn_v(i) = (mod(i,PT)==0)*G;
```

```
end
```

```
[s_syn2,zf_s_v]=filter(1,A,exc_syn_v((n1)*FL_v+1:n*FL_v),zi_s_v);
```

```
zi_s_v = zf_s_v;
```

```
s_syn_v((n-1)*FL_v+1:n*FL_v) = s_syn2;
```

```
sound(s_syn_v/max(s_syn_v),8000);
```

12. 变调不变速：重新考察（1）的系统，将其共振峰频率提高150Hz之后， a_1 和 a_2 分别为多少？

答：先通过tf2zp将差分方程求出零极点，然后对极点进行操作，之后再 用zp2tf函数从零极点变换回差分方程。

得到的结果为： $a_1=1.2073$ ， $a_2=-0.9506$

变换前后零极点图如图13所示：

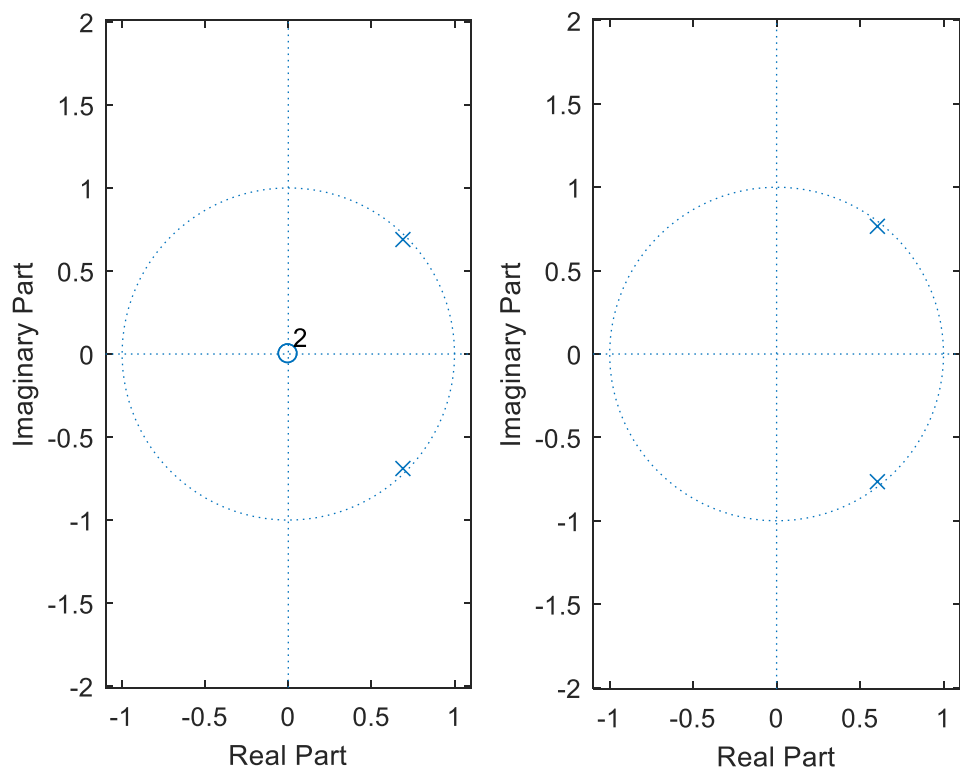


图13 变换前后的零极点图

代码:

% (12) 提高共振峰频率

```
[z, p, k]=tf2zp(b, a);
```

```
p_i(1)=p(1)*exp(2*pi*1j*150/8000); %对上半平面的极点进行处理
```

```
p_i(2)=p(2)*exp(-2*pi*1j*150/8000); %对下半平面的极点进行处理
```

```
[e, s]=zp2tf(z, p_i, k); %从零极点转换到状态方程
```

```
disp (s);
```

```
figure
```

```
subplot(1,2,1), zplane(b, a);
```

```
subplot(1,2,2), zplane(e, s);
```

13. 仿照（10），将基音周期减小一半，将所有的共振峰频率增加150Hz，重新合成语音。

答：基音周期减小一半，即 $PT=PT/2$ ，共振峰频率提高150Hz可以参照（12）中的方法进行变换，最后通过filter合成变调不变速的语音，如图14所示：

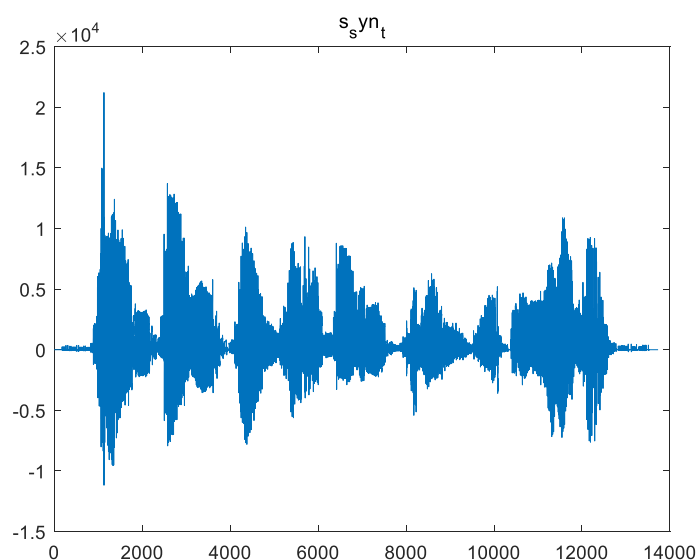


图14 变调不变速的合成信号

代码：

%（13）将基音周期减小一半，将共振峰频率增加150Hz，重新合成语音，听听是啥感受～

```
PT_t = floor(PT/2);
```

```
for i = (n-1)*FL+1:n*FL
```

```
    exc_syn_t(i) = (mod(i,PT_t)==0)*G;
```

```
end
```

```

[z, p, k]=tf2zp(1, A);
a = angle(p);

% 因为极点不止两个，因此当角度大于0的时候，乘上
exp(2*pi*1j*150/8000)，而当角度小于0的时候，乘上 exp(-
2*pi*1j*150/8000)

for i = 1:length(p)
    if(a(i)>0)
        p_t(i) = p(i)*exp(2*pi*1j*150/8000);
    else
        p_t(i) = p(i)*exp(-2*pi*1j*150/8000);
    end
end
[e_cvt, s_cvt] = zp2tf(z, p_t, k); %从零极点转换
[s_syn3, zf_s_t]=filter(e_cvt, s_cvt, exc_syn_t((n1)*FL+1:n*FL)
, zi_s_t); %通过filter函数计算响应
zi_s_t = zf_s_t;
s_syn_t((n-1)*FL+1:n*FL) = s_syn3;

sound(s_syn_t/max(abs(s_syn_t)), 8000);

```

合成信号、变速不变调信号、变调不变速信号如图15所示：

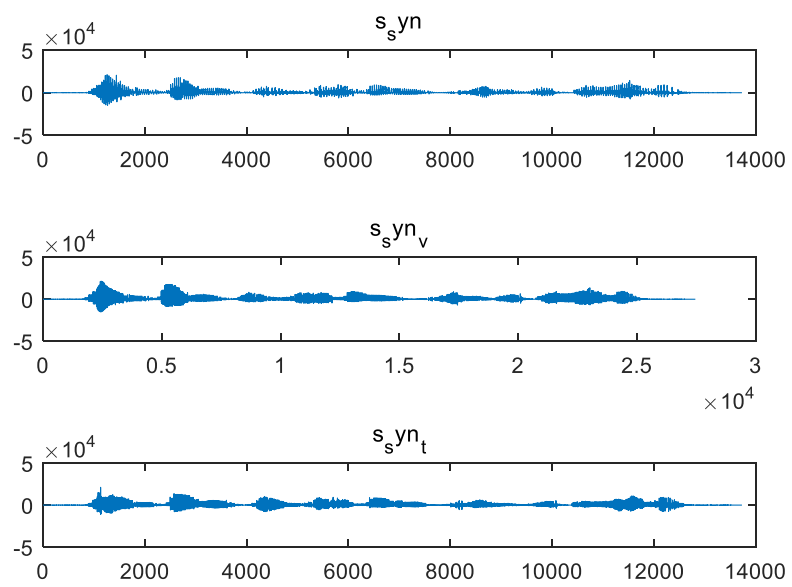


图15 合成信号、变速不变调信号、变调不变速信号

二、 总结

通过本次matlab大作业，让我初步了解到人发生的机理，同时还通过信号与系统的方法，来对语音信号进行处理，实现语音重建、语音变速不变调、语音变调不变速的功能，充满了趣味性。

本次matlab大作业让我对matlab编程有了更深刻的认识，对其操作更加熟悉，掌握了很多信号系统相关的matlab编程。在编程的时候，碰到了很多困难，特别是在理解基音周期那一块的时候，不知道该怎么处理，后来向同学请教，看书，慢慢弄明白了。在完成本次大作业之后，深感自己能力的不足，只是完成了大作业的基本要求，并没有在此基础上进行创新和拓展，希望在下一个大作业中能够有所突破！