语音合成大作业

班级：wu52

姓名：yfreedomliTHU

学号：

日期：2017.7.19

1.2.1 语音预测模型

（1）根据给定的输入信号 e(n)以及输出信号 s(n)，对等式两边进行 Z 变换，通过化简，可以得到传递函数的形式为：

$$H(z)=\frac{1}{1-a_1z^{-1}-a_2z^{-2}}$$

当$a_1=1.3789$　$a_2=-0.9506$ 时，根据$f=\frac{\omega}{2\pi}=\frac{\Omega}{2\pi T}$，所以对应可以计算得到共振峰频率为 999.9447Hz.

代码如下：

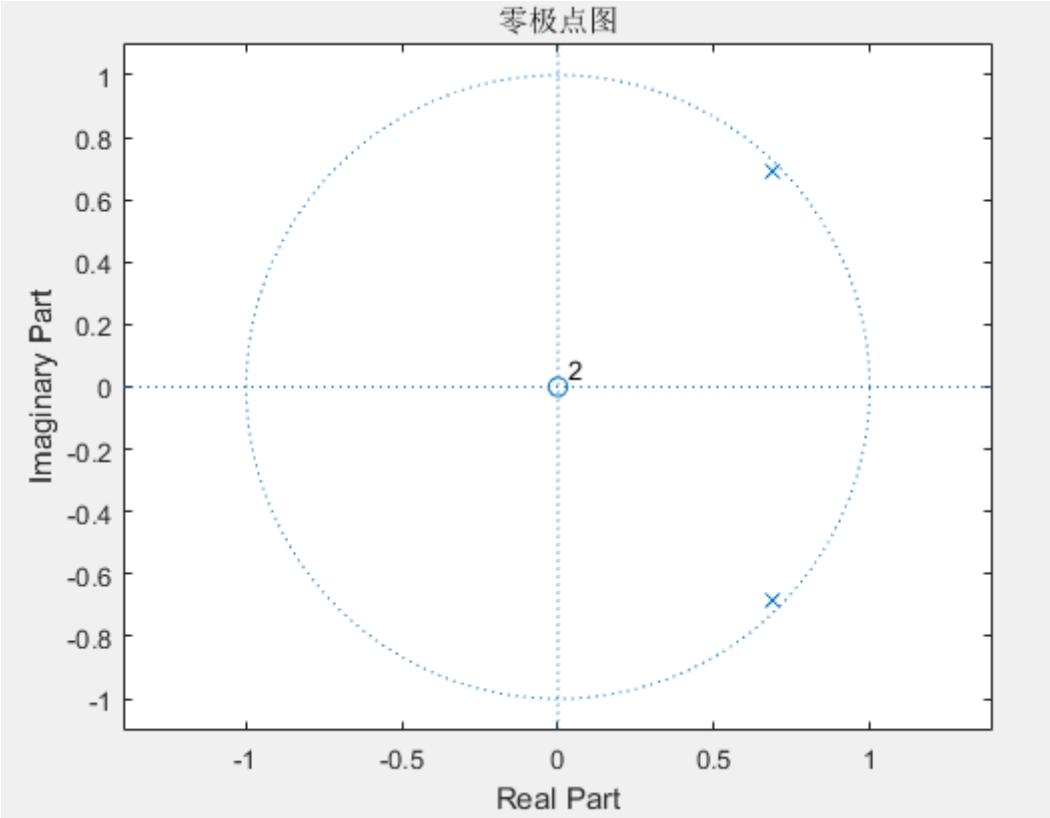
```
close all, clear all, clc;
a = [1,-1.3789,0.9506];
b = [1];
n=[0:200]';
[z,p,k] = tf2zp(b,a);                  % 得到零极点
peak_freq = angle(p(1))/(2*pi)*8000; % 共振峰频率
%plot
figure;
zplane(b,a),title('零极点图');          % 画出零极点图
figure;
freqz(b,a);                            % 画系统函数的频率响应
figure;
hi=impz(b,a,n);                        % 用impz求系统单位样值响应
subplot(1,2,1);
stem(n,hi,'b-');
subplot(1,2,2);                        % 用filter求系统单位样值相应
x = (n==0);                            % 以单位样值序列为激励信号
hf=filter(b,a,x);
stem(n,hf,'k-');
```

共振峰频率计算结果如下：

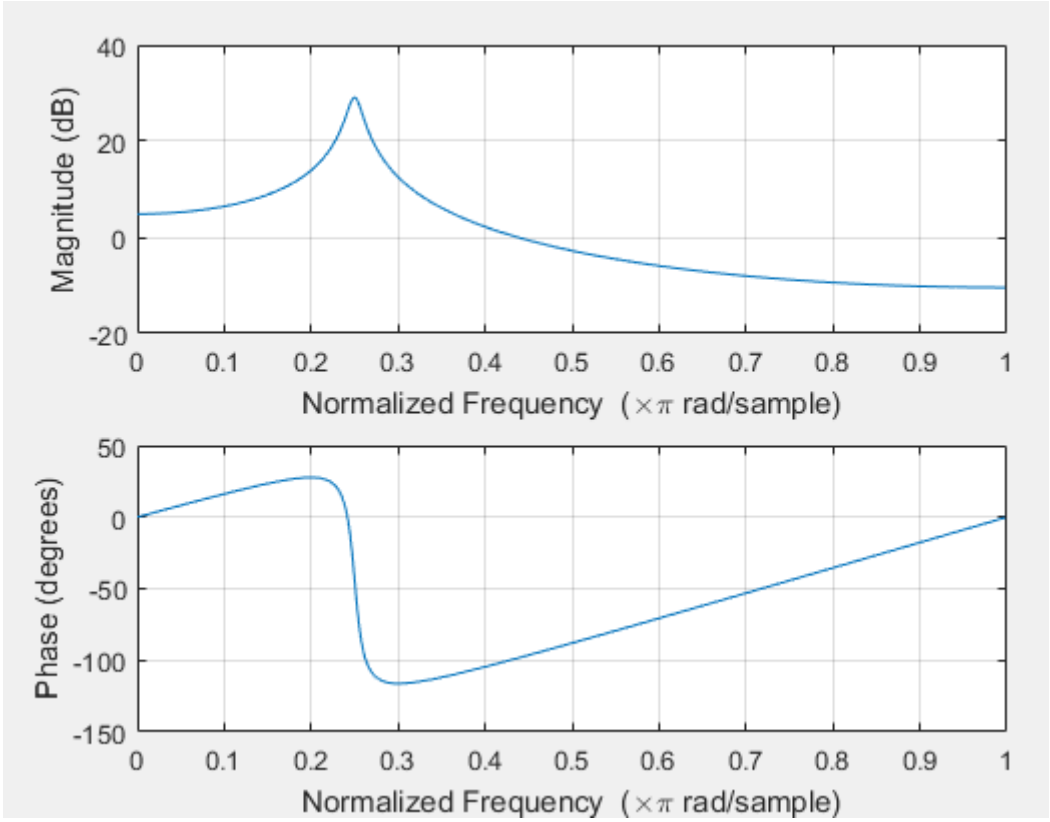
```
peak_freq =

  999.9447
```

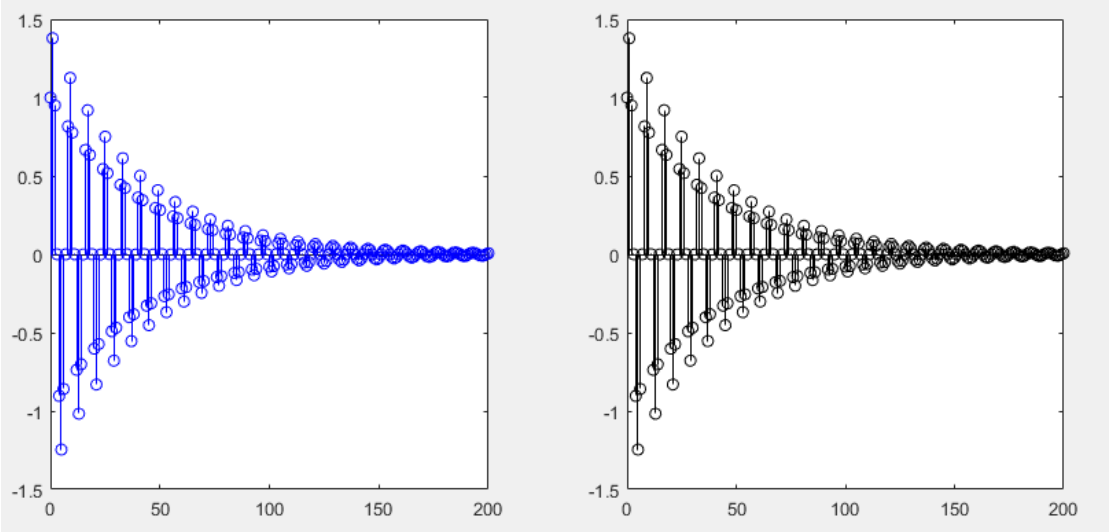
零极点分布图：


零极点图
Imaginary Part
Real Part
2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

频率响应：


Magnitude (dB)
40
20
0
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (×π rad/sample)
Phase (degrees)
50
0
-50
-100
-150
Normalized Frequency (×π rad/sample)

单位样值响应：（左为 impz，右为 filter）



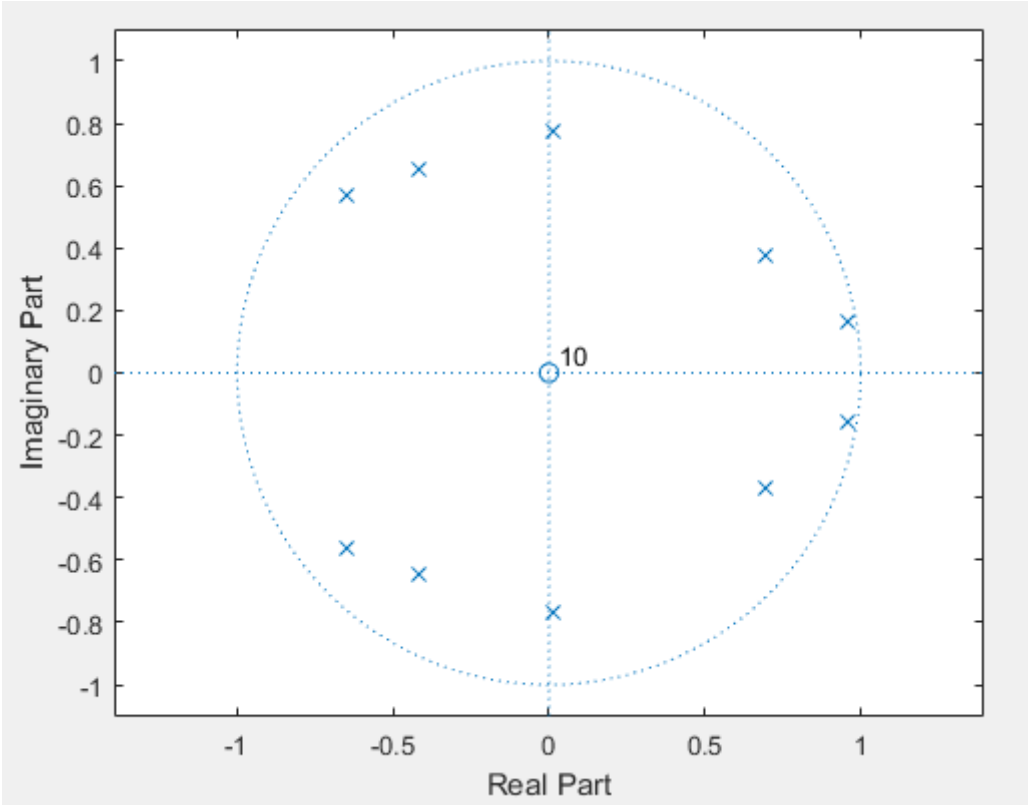
从结果可以看出，impz 和 filter 得到的图像的确完全相同。

（2）对程序的阅读后已基本理解相应的基本流程。

（3）在 speechproc.m 添加如下代码：

```
if n == 27
% (3) 在此位置写程序，观察预测系统的零极点图
figure;
zplane(1,A);
end
```

绘制的零极点分布如下图所示：



（4）该问要求用 filter 函数计算激励，同时需要保持滤波器状态

```
% (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
[result,zi_pre]=filter(A,1,s_f,zi_pre);%保持滤波器状态需要用上一帧的末状态更新下一帧的初始状态
exc((n-1)*FL+1:n*FL) =result;  %计算得到的激励
```

说明：由于需要保持滤波器的状态不变，则需要满足上一帧的末状态即为下一帧的初状态。利用 filter 函数计算激励，即利用输出来求激励，所以该逆系统传递函数应该为原传递函数的倒数，所以 filter 里面的参数为 A,1。

（5）用 filter 函数和 exc 重建语音

重建语音过程比较简单明了，直接利用上一问得到的激励信号 exc 通过传递函数即可，注意这里不再需要对传递函数去导数，filter 里面的参数应该为 1，A。同样地，为了保持滤波器状态不变，也需要用上一帧的末状态更新下一帧的初状态。

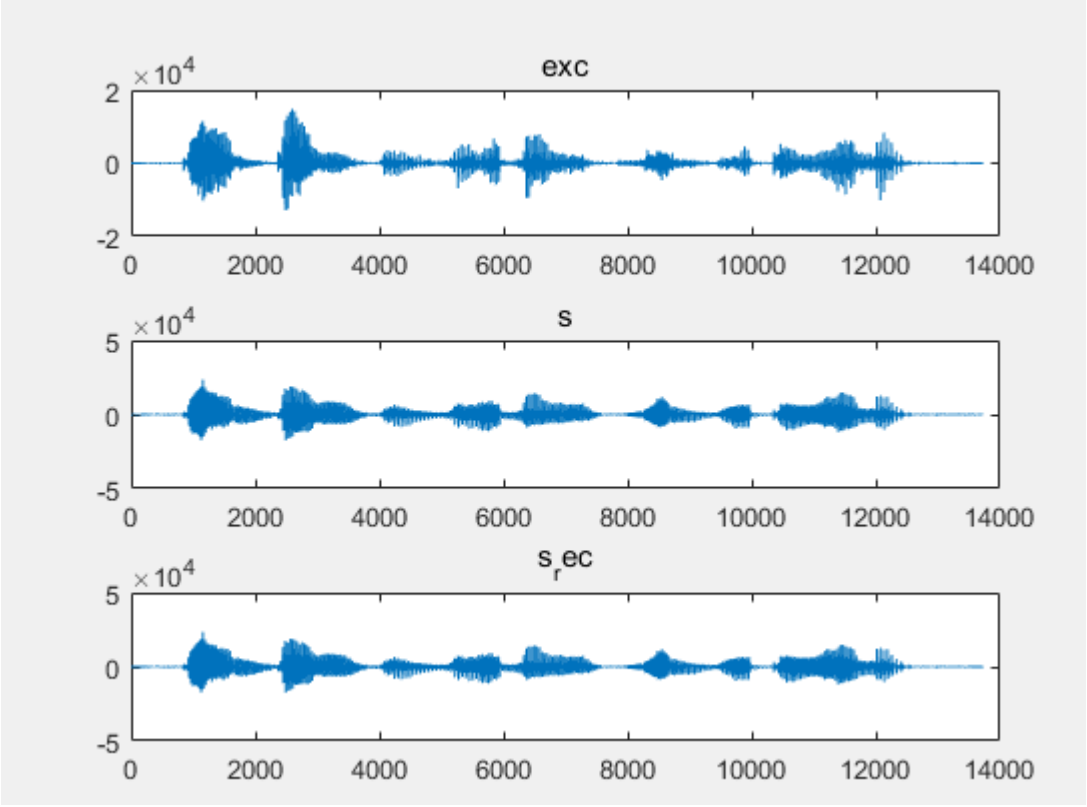
相应代码如下：

```
% (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_rec_result,zi_rec]=filter(1,A, exc((n-1)*FL+1:n*FL),zi_rec); %保持滤波器状态
s_rec((n-1)*FL+1:n*FL) =s_rec_result; % 计算得到的重建语音写在这里
```

（6）在循环结束后添加程序：用 sound 试听（6）中的 e(n) 信号，比较和 s(n) 以及 ˆs(n)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。

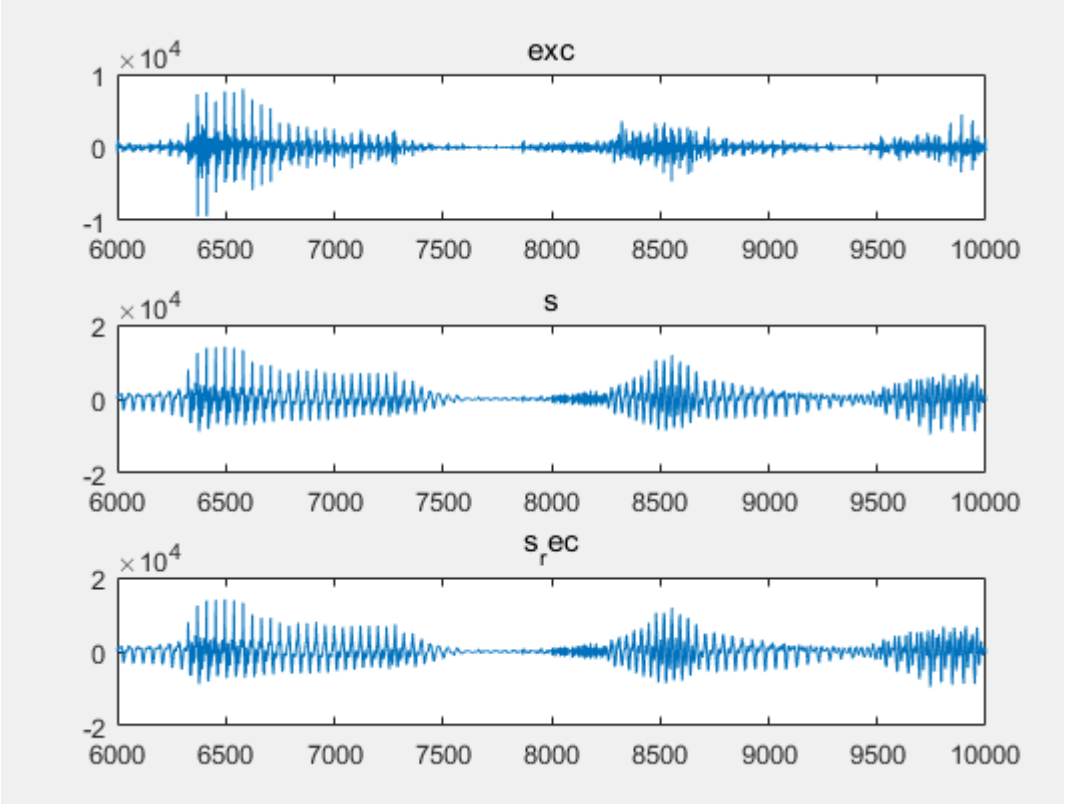
试听完这三个信号后，发现激励信号 e(n)的噪音比较大，分辨度不高，相比之下，s(n)以及ˆs(n)信号更清楚，并且发现 s(n) 以及ˆs(n)信号清晰度基本相同，人耳很难加以辨别。

直接选取三段语音作图如下：



可以看出 s 与 s_rec 的图像十分接近，但为了进一步看清楚信号变化，取各信号的

[6000:10000]部分重新作图如下：



从该图可以确定 s 与 s_rec 的图形确实基本相同，这也验证了我们之前的结论。

代码如下：

```
% (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
% 后面听语音的题目也都可以在这里写，不再做特别注明

sound(exc);
sound(s);
sound(s_rec);
figure;
subplot(3,1,1);
plot(6000:10000,exc(6000:10000));
title('exc');
subplot(3,1,2);
plot(6000:10000,s(6000:10000));
title('s');
subplot(3,1,3);
plot(6000:10000,s_rec(6000:10000));
title('s_rec');
```

（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

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

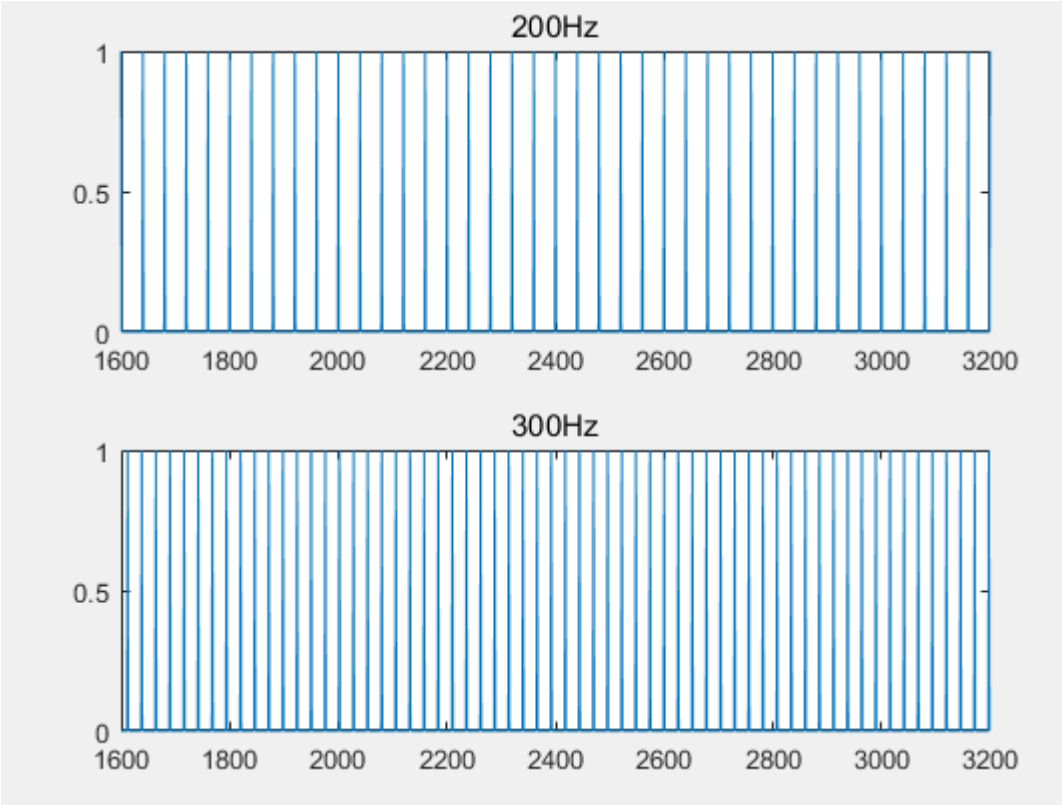
考虑该信号的 N 和 NS 分别为何值？用 sound 试听这个声音信号。再生成一个 300Hz 的单位样值”串"并试听，有何区别？事实上，这个信号将是后面要用到的以基音为周期的人工激励信号 e(n) 。

根据表达式，易得到：

200Hz 时，NS=200，N=floor(8000/200)=40；

300Hz 时，NS=300，N=floor(8000/300)=26；

区别：通过人耳对生成的声音信号进行试听，发现 300Hz 的音调明显高于 200Hz，这与理论相符，为了进一步验证，选取相同长度的部分语段作出图像如下：



可以看出，在相同区间内，300Hz 的信号分布确实比 200Hz 更密集，前面的结论正确。
代码保存于 solvecode02.m 文件中，具体代码如下：

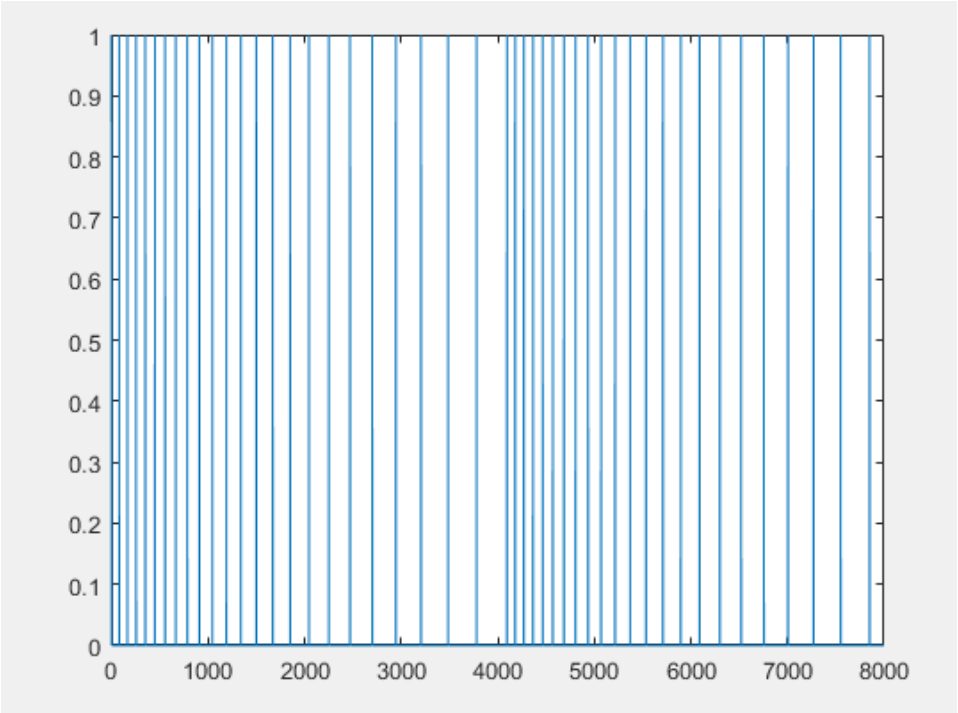
```
close all,clear all,clc;
%生成8KHz抽样的持续1秒钟的数字信号，是频率为200Hz的单位样值“串”
fs=8000;
NS1=200;
NS2=300;
N_200=floor(fs/200);
N_300=floor(fs/300);
T=1:fs;
L=length(T);
result1=(mod(T,N_200)==0);
result2=(mod(T,N_300)==0);
figure;
subplot(2,1,1);
plot(0.2*L:0.4*L,result1(0.2*L:0.4*L));
title('200Hz');
subplot(2,1,2);
plot(0.2*L:0.4*L,result2(0.2*L:0.4*L));
title('300Hz');
sound(double(result1),fs);
sound(double(result2),fs); %需要转为dpule才不会报错
```

（8）按前面默认的频率 8000Hz，则 1s 采样 8000 个点，由于 10ms 为一个片段，所以当前段数 m=floor(i/80),从而可以计算出基音周期 PT，把 PT 加到 I 上即可，直到 i=8000，从而通过循环得到所需语音信号。代码存在 solvecode03.m 中。
生成信号的代码如下：

```
close all,clear all,clc;
fs=8000;
i=1;
x=zeros(8000,1);
while(i<fs)      %循环实现
    x(i)=1;
    %由于每段为10ms，所以m=i/80取整数,根据提示，基音周期
    PT=80+5*mod(floor(i/80),50);
    i=i+PT;
end
sound(x,fs);
plot(x);
```

试听后发现可以明显分辨出语音有分段现象，可以听到两段声音，语音频率也在变化，大致是频率从较大频率减小，然后第二段又从较大频率减小，具体变化如下图所示。

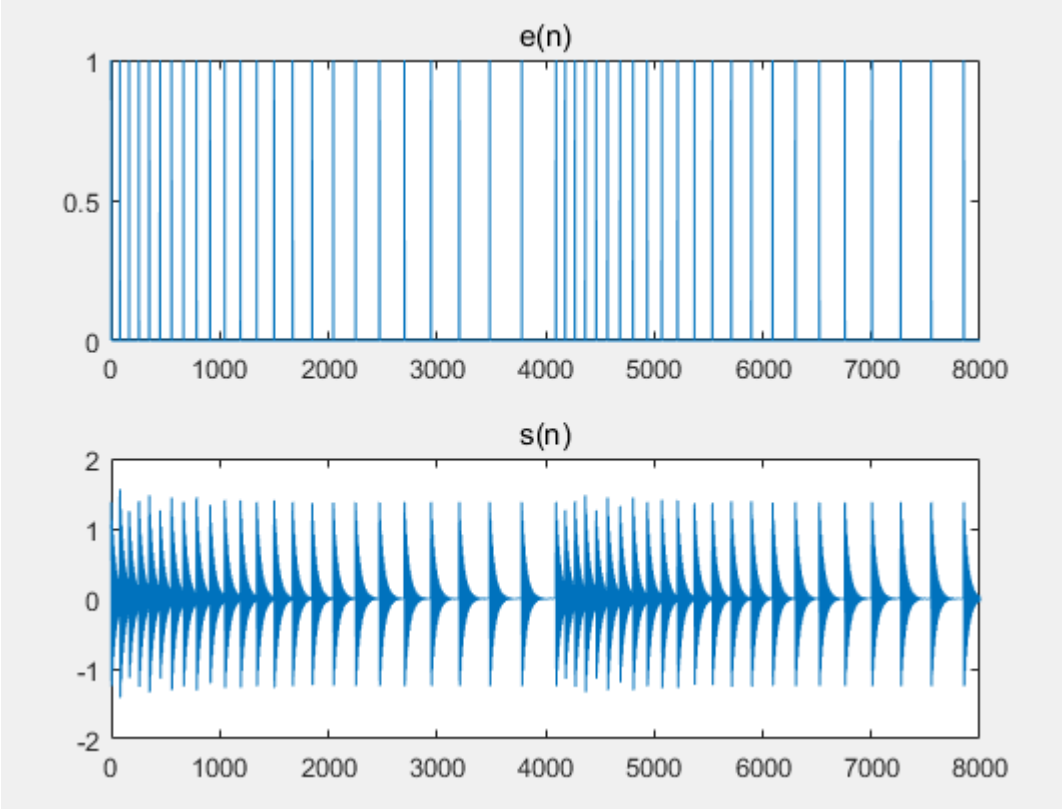
作出的图像如下



（9）.直接在上题的代码中添加即可，代码同样保存在 solvecode03.m 中

```
close all,clear all,clc;
fs=8000;
i=1;
x=zeros(8000,1);
while(i<=fs)      %循环实现
    x(i)=1;
    %由于每段为10ms，所以m=i/80取整数,根据提示，基音周期
    PT=80+5*mod(floor(i/80),50);
    i=i+PT;
end
%sound(x,fs);
% plot(x);
a = [1,-1.3789,0.9506];
b = [1];
s=filter(b,a,x);
sound(s,fs);
subplot(2,1,1);
plot(x);
title('e(n)');
subplot(2,1,2);
plot(s);
title('s(n)');
```

和 e(n)相比，可以听出 s(n)的音调似乎变得更高了。用 plot 作出两者的图像如下：



可以看出通过（1）中的系统后得到的 s(n)与 e(n)相比波形发生了改变，包络改变导致了声音的改变。

（10）.重改 speechproc.m 程序。利用每一帧已经计算得到的基音周期和（8）的方法，生成合成激励信号 Gx(n)（G 是增益），用 filter 函数将 Gx(n)送入合成滤波器得到合成语音 ~s(n)。试听和原始语音有何差别。

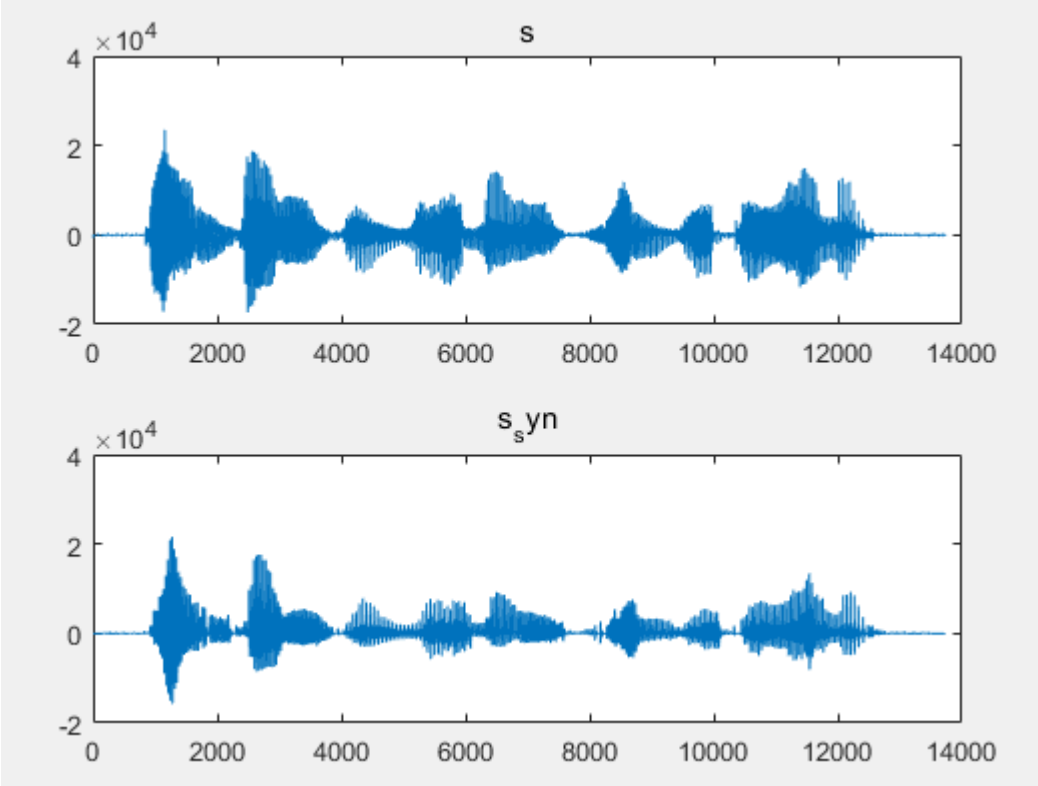
代码如下：

```
if (n<=3)
    num=(n-1)*FL+1;              %初始起点num为n=3时的起点,需注意
end
while (num<=n*FL)                  %按照（8）的规则生成
    exc_syn(num) = G;              % 生成激励
    num = num + PT;
end
[s_syn_result,zi_syn]=filter(1,A,exc_syn((n-1)*FL+1:n*FL),zi_syn);
% exc_syn((n-1)*FL+1:n*FL) = ... 将你计算得到的合成激励写在这里
s_syn((n-1)*FL+1:n*FL) =s_syn_result;    %计算得到的合成语音写在这里
```

本题有第（8）问作为背景其实比较简单，但是需要注意的是 num 的初值问题，在开始的时候，我在进入 while 循环之前，时钟吧 num 的处置设为当前片段的起点，即(n-1)*FL+1,但是得到的语音效果和未处理原声差不多，而且感觉声调似乎也改变了，感觉不是很对。后来发现每段的初值并不一定一定在（n-1）*FL+1 处有激励，这应该取决于上一段末尾的 num+PT 是否恰好等于下一段的端点（即(n-1）*FL+1),而往往二者是不相等的，所以原来的初始化是错误的。由于只需要在 n=3 时加初始化，其余 num 直接接上一段语音末尾的 num+PT 即可，所以在前面初始化时加了 if 条件判断，问题得以解决。用 sound 函数听后发现效果比未处理前好了一些，噪声得到了减少，与 s_rec 比较接近，二者降噪后效果都

有所提升，但具体哪个更好，感觉差别不是很大。

进一步对比 s_syn 与原信号 s 的波形：



发现降噪的同时似乎对原信号也有一定的破坏，表现为部分下半部分波形消失。

（11）. 仿照（10）重改 speechproc.m 程序，只不过将（10）中合成激励的长度增加一倍，即原来 10 毫秒的一帧变成了 20 毫秒一帧，再用同样的方法合成出语音来，如果你用原始抽样速度进行播放，就会听到慢了一倍的语音，但是音调基本没有变化。

令 FL_v=2*FL;保持采样率不变，仅仅将激励长度增加一倍，可以听到音调不变，语速变慢，确实如题目描述所言。

代码如下：

```
% (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=2*FL;
if (n<=3)
    num=(n-1)*FL+1;              %初始起点num为n=3时的起点,需注意
end
while (num<=n*FL_v)                  %按照（8）的规则生成
    exc_syn_v(num) = G;              % 生成激励
    num = num + PT;
end
[s_syn_v_result,zi_syn_v]=filter(1,A,exc_syn_v((n-1)*FL_v+1:n*FL_v),zi_syn_v);
% exc_syn_v((n-1)*FL_v+1:n*FL_v) = ... 将你计算得到的加长合成激励写在这里
s_syn_v((n-1)*FL_v+1:n*FL_v) = s_syn_v_result;     % 将你计算得到的加长合成语音写在这里
```

（12）重新考察（1）中的系统，将其共振峰频率提高 150Hz 后的 a1 和 a2 分别是多少？

由于频率增加 150Hz，所以一种比较简便的方法就是直接用增加的频率换算出新的 p，然后利用 zp2tf 函数即可得到 a 的值。

代码如下：

```
close all, clear all, clc;
a = [1,-1.3789,0.9506];
b = [1];
n=[0:200]';
[z,p,k] = tf2zp(b,a);                    % 得到零极点
peak_freq = angle(p(1))/(2*pi)*8000; % 共振峰频率
%plot
figure;
zplane(b,a),title('零极点图');          % 画出零极点图
figure;
freqz(b,a);                              % 画系统函数的频率响应
figure;
hi=impz(b,a,n);                          % 用impz求系统单位样值响应
subplot(1,2,1);
stem(n,hi,'b-');
subplot(1,2,2);                          % 用filter求系统单位样值相应
x = (n==0);                              % 以单位样值序列为激励信号
hf=filter(b,a,x);
stem(n,hf,'k-');
%12问，共振峰频率提高150Hz 后的a1 和a2 分别是多少？
p11=p(1)*exp(i*150*2*pi/8000);      %直接计算新的p
p12=p(2)*exp(-i*150*2*pi/8000);
p1=[p12,p11]';                          %拼接得到二维的p1
[b1,a1]=zp2tf(z,p1,k);                %由新的p利用zp2tf函数即可得到a与b的值
```

运行结果：

```
a1 =

    1.0000   -1.2073    0.9506
```

所以 a1=1.2073,a2=-0.9506

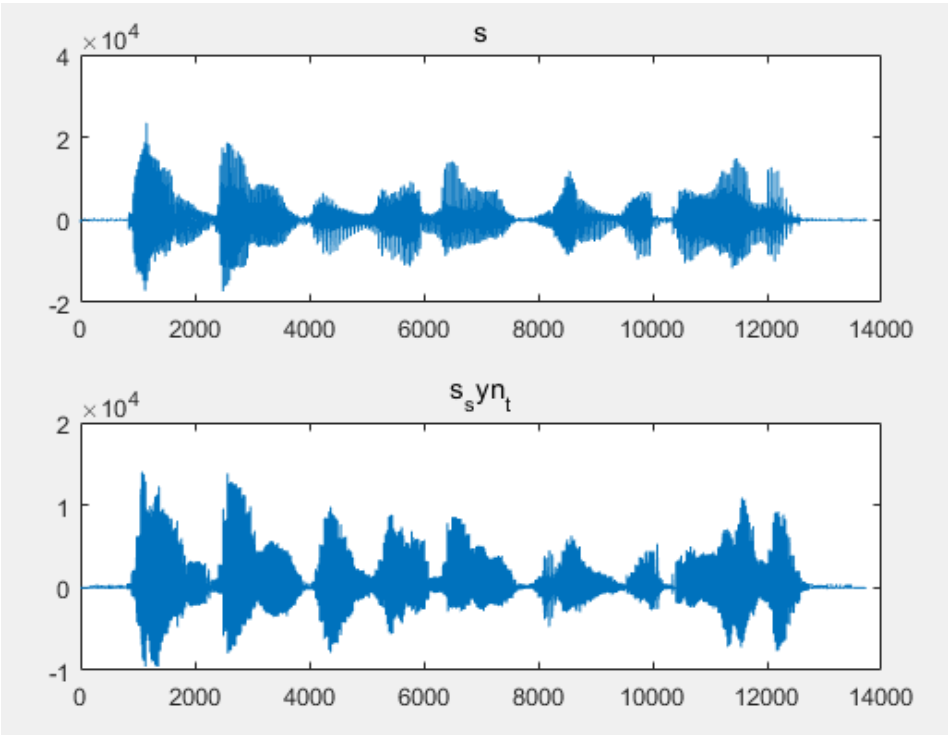
（13）仿照（10）重改 speechproc.m 程序，但要将基音周期减小一半，将所有的共振峰频率都增加 150Hz ，重新合成语音，听听是何感受。

在前面的基础上修改比较简单，需要注意的是，刚开始的时候用了 residuez 函数来得到几点，在第一问也对了，但是在本题中使用 residuez 函数不对，会报错，显示 s_syn_t 不是示数或浮点数，进一步查明资料后，这里是传递函数与零极点的转换，确实应该使用 tf2zp 和 zp2tf 函数，修改之后结果就正确了。

```
% (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~
%共振峰频率增加150Hz
 [z,p,k] = tf2zp(1,A);              % 得到零极点
 for k1=1:length(p);
     if(angle(p(k1))>0)
        p1(k1)=p(k1)*exp(i*150*2*pi/8000);      %直接计算新的p
     else
        p1(k1)=p(k1)*exp(-i*150*2*pi/8000);
     end
 end
 [B1,A1]=zp2tf(z,p1,k);          %由新的p利用residuez函数即可得到a与b的值
 if (n<=3)
     num2=(n-1)*FL+1;              %初始起点num为n=3时的起点,需注意
 end
 while (num2<=n*FL)                    %按照（8）的规则生成
     exc_syn_t(num2) = G;                % 生成激励
     num2 = num2 + floor(PT/2);
 end
  [s_syn_t_result,zi_syn_t]=filter(B1,A1,exc_syn_t((n-1)*FL+1:n*FL),zi_syn_t);
  % exc_syn_t((n-1)*FL+1:n*FL) = ... 将你计算得到的变调合成激励写在这里
  s_syn_t((n-1)*FL+1:n*FL) =s_syn_t_result;  % ...    将你计算得到的变调合成语音写在这里
```

试听结果确实如题目描述的那样，音调确实增加，语速未变，听起来像是女声，而不再是男声了。

比较原声 s 与 s_syn_t 的图像如下：



可以看出，变调后，波形更密集，但大体包络和长度没变，表现为变调不变速。

原创说明：

本次实验基本由本人独立完成，但也有和同学讨论，如第（10）问中刚开始发现初始化取值不对，得到的声音效果比原来还差，和同学就这个问题进行了讨论。

实验感想：

总体来说，本次实验比较简单，更多的是对 matlab 课上的一些函数的巩固，涉及真正的语音合成部分较少，比信号与系统大作业简单多了，虽然都是语音方面的大作业。并且通过本次大作业对给出的函数的阅读以及理解，对语音信号处理（比如预处理进行分帧，加窗等）的方法有了初步了解。

附件：

code 文件夹里面存放的是单独小问的解决代码

对原 matlab 代码文件的补充以及生成的语音文件在 code1 文件夹中。