

信号与系统— MATLAB 综合实验之语音合成

班级：无 06

姓名：闫珺博

学号：2020010796

日期：2022.07.31

1.2.1_(1)

给定差分方程 $e(n) = s(n) - a_1s(n-1) - a_2s(n-2)$, $e(n)$ 是输入信号, $s(n)$ 是输出信号, 求传递函数、绘出零极点图, 频率响应, 用 `impz` 和 `filter` 分别画出单位样值响应并比较。

两侧同做 z 变换得传递函数 $H(z) = \frac{S(z)}{E(z)} = \frac{1}{z^2 - a_1z - a_2}$

```

1 FS=8000;
2 a1=1.3789;
3 a2=-0.9506;
4 num=[0 0 1];
5 den=[1 -a1 -a2];
6 omega=roots(den);
7 f=(abs(angle(omega(1))))*8000/(2*pi);%共振峰频率
8 tf(num,den,1)%传递函数
9 figure(1)
10 zplane(num,den);%画z平面零极点
11 figure(2)
12 freqz(num,den);%频率响应
13 figure(3)
14 yim=impz(num,den);%用impz求的单位样值响应
15 n=0:391;
16 stem(n,yim);
17 x=((n-0)==0);%单位样值信号
18 y=(filter(num,den,x))';%输入单位样值信号后的响应
19 figure(4)
20 stem(n,y);%画用filter求的单位样值响应
21 figure(5)
22 stem(n,y-yim);%两种单位样值响应的误差
23 %(9)
24 e_n=hw122_8();
25 s_n=filter(num,den,e_n);
26 sound(s_n,FS);

```

核心代码

ans =

1

 $z^2 - 1.379 z + 0.9506$

Sample time: 1 seconds

Discrete-time transfer function.

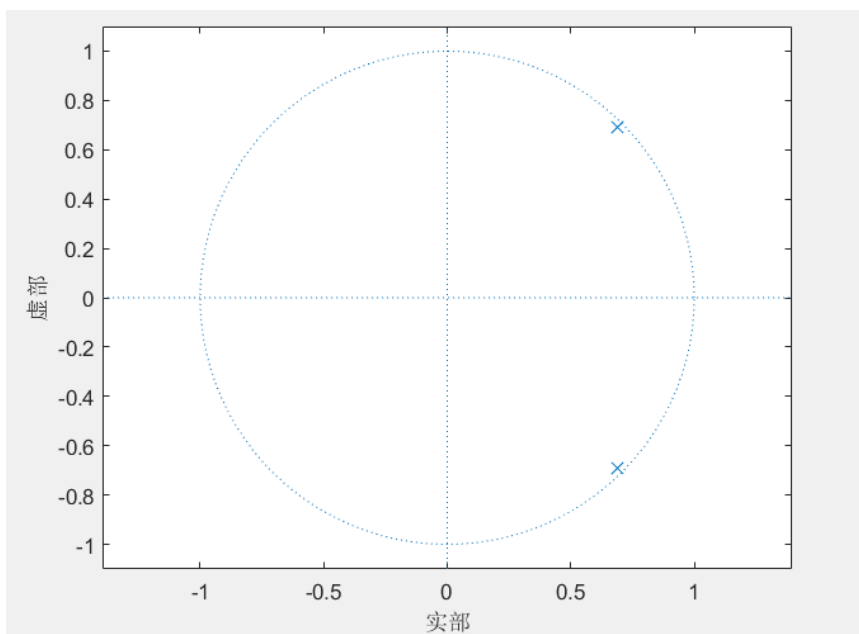
$a_1 = 1.3789, a_2 = -0.9506$ 时的传递函数

共振峰频率公式 $f = \frac{\omega}{2\pi} = \frac{\Omega}{2\pi T}$ ，其中 $T = \frac{1}{8000}$ ，roots 函数求根后使用 angle 函数得到其中一个根的辐角，由于 den 系数都是实数，两个根的辐角绝对值相同，用 abs 函数取正值即可得到 Ω ，代入得到 f，其他图像简单代入函数就可得到。

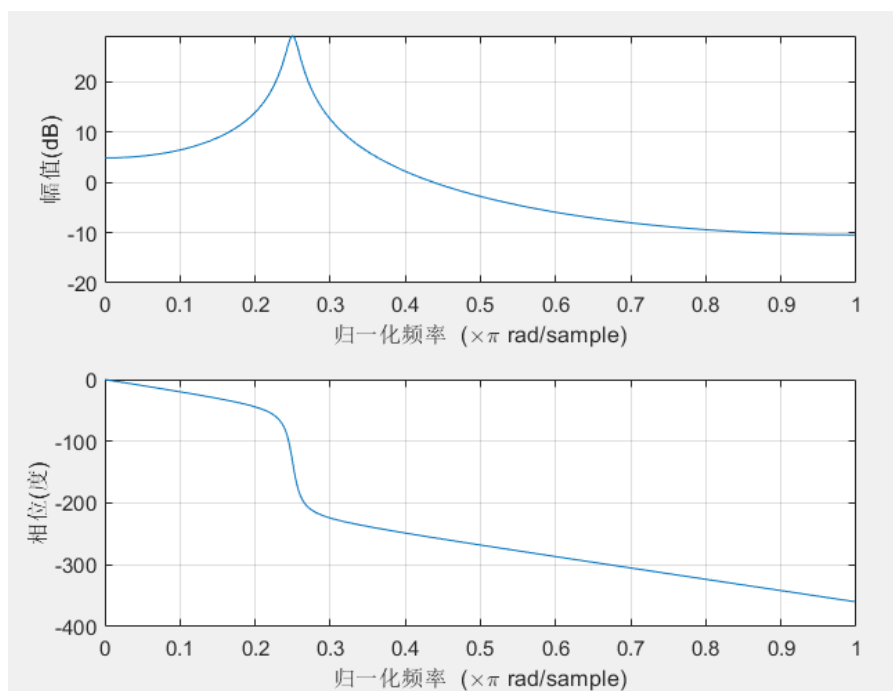
f =

999.9447

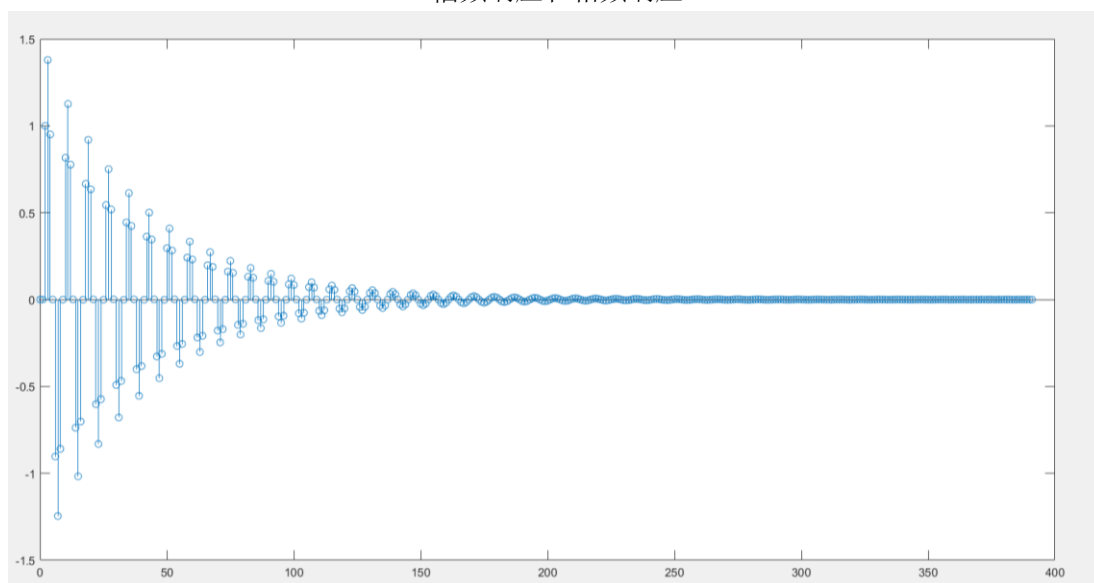
共振峰频率 999.9447Hz



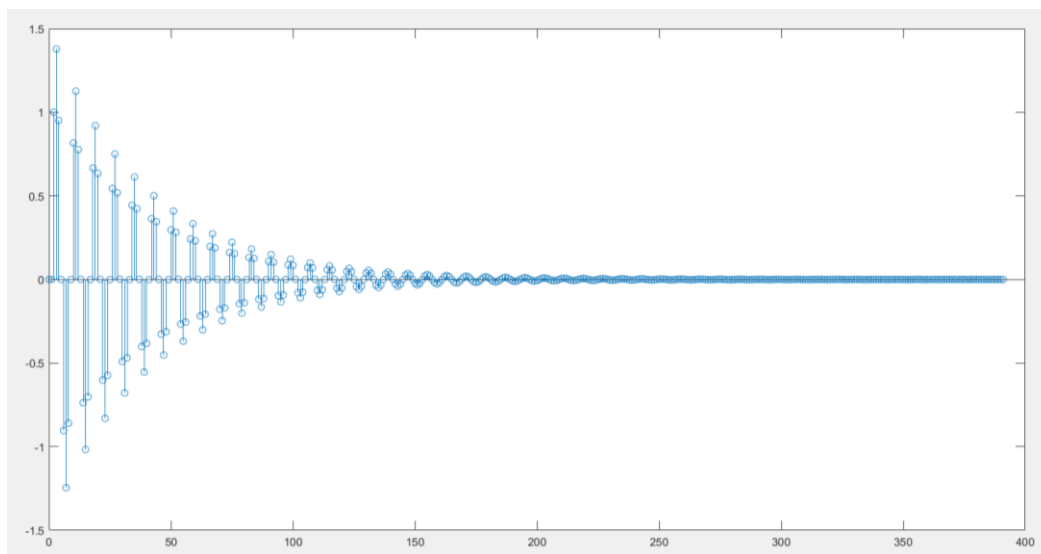
zplane 绘出的零极点图



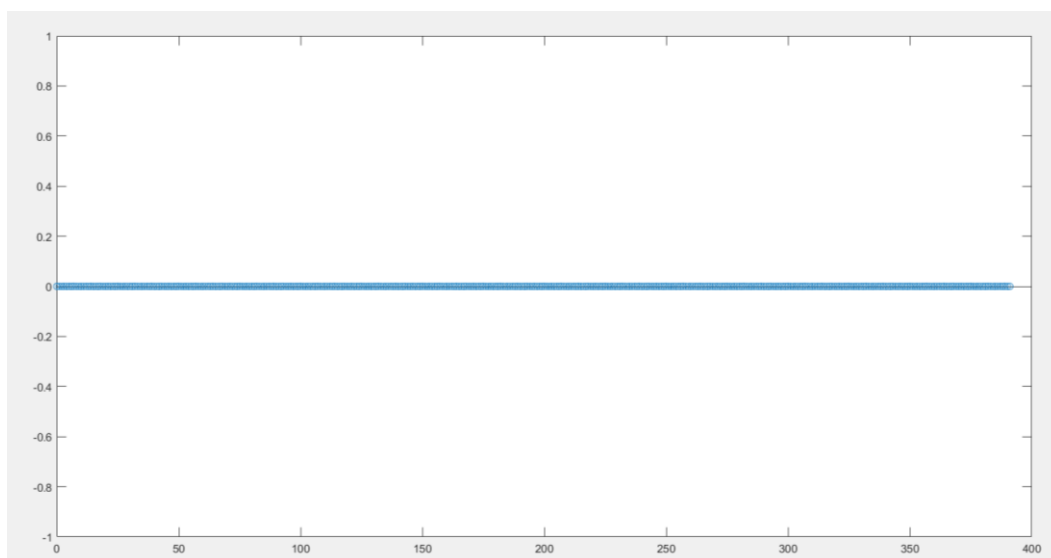
幅频响应和相频响应



用 impz 画出的单位样值响应



用 `filter` 画出的单位样值响应



用 `impz` 和 `filter` 得到单位样值响应相减的结果

1.2.1_(2)

阅读 `speechproc.m` 程序，理解基本流程。

```
function speechproc()

% 定义常数
FL = 80;           % 帧长
FL_v=160;
WL = 240;          % 窗长
P = 10;            % 预测系数个数
s = readspeech('voice.pcm',100000); % 载入语音s
L = length(s);     % 读入语音长度
FN = floor(L/FL)-2; % 计算帧数
% 预测和重建滤波器
exc = zeros(L,1);  % 激励信号（预测误差）
zi_pre = zeros(P,1); % 预测滤波器的状态
s_rec = zeros(L,1); % 重建语音
zi_rec = zeros(P,1);
% 合成滤波器
exc_syn = zeros(L,1); % 合成的激励信号（脉冲串）
s_syn = zeros(L,1);   % 合成语音
% 变调不变速滤波器
exc_syn_t = zeros(L,1); % 合成的激励信号（脉冲串）
s_syn_t = zeros(L,1);   % 合成语音
% 变速不变调滤波器（假设速度减慢一倍）
exc_syn_v = zeros(2*L,1); % 合成的激励信号（脉冲串）
s_syn_v = zeros(2*L,1);   % 合成语音

hw = hamming(WL); % 汉明窗
```

程序开始定义了帧长，窗长等常数，其中 FL_v 是我定义的变速不变调帧长；还有预测重建滤波器、合成滤波器、变调不变速滤波器以及变速不变调的激励 $e(n)$ 和语音 $s(n)$ ；以及加权用的汉明窗。之后依次处理每帧语音，计算得到预测系数，依次计算得到四种滤波器输出的语音信号；循环之外定义了三个函数，用来读写语音文件和计算基音周期。

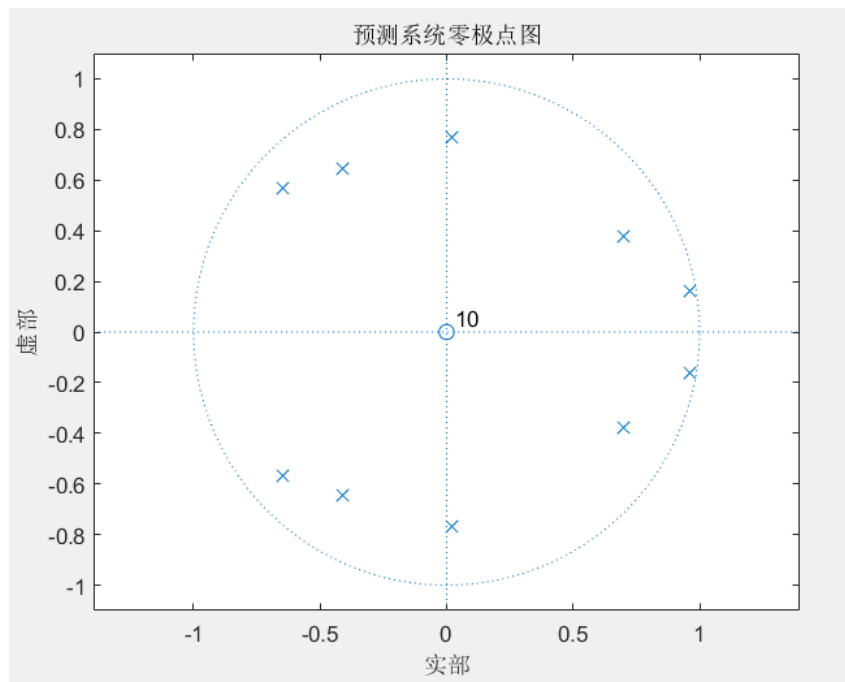
1.2.1_(3)

运行该程序到 27 帧时停住，用（1）中的方法观察零极点图。

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

figure(1)
zplane(1,A);
title('预测系统零极点图');
end
```

直接使用 `zplane` 函数和预测系数观察



1.2.1_(4)

对每帧语音信号 $s(n)$ 和预测系数 $\{a_i\}$ ，用 `filter` 计算激励信号 $e(n)$

% (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_n,zf_pre]=filter(A,1,s_f,zi_pre);
zi_pre=zf_pre;
exc((n-1)*FL+1:n*FL) = exc_n;%将你计算得到的激励写在这里
```

`zi_pre` 和 `zf_pre` 参数用来维持滤波器的状态不变

1.2.1_(5)

用计算得到的激励信号 $e(n)$ 和预测系数 $\{a_i\}$ ，用 `filter` 计算重建语音 $\hat{s}(n)$ 。

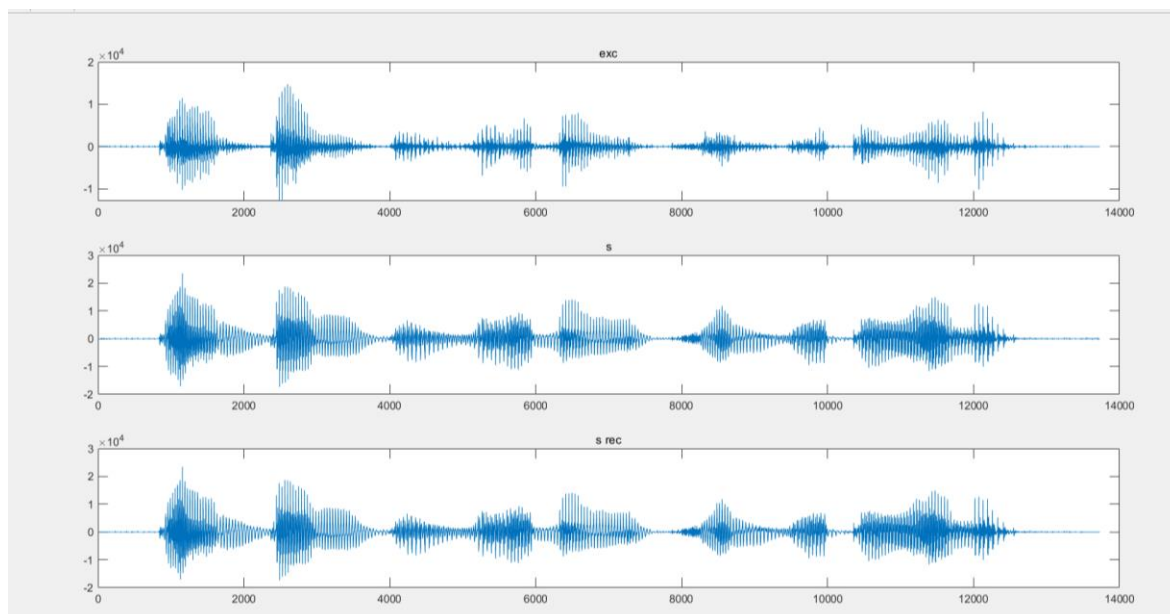
% (5) 在此位置写程序，用`filter`函数和`exc`重建语音，注意保持滤波器状态

```
[s_rec_n,zf_rec]=filter(1,A,exc((n-1)*FL+1:n*FL),zi_rec);
zi_rec=zf_rec;
s_rec((n-1)*FL+1:n*FL) = s_rec_n; %将你计算得到的重建语音写在这里
```

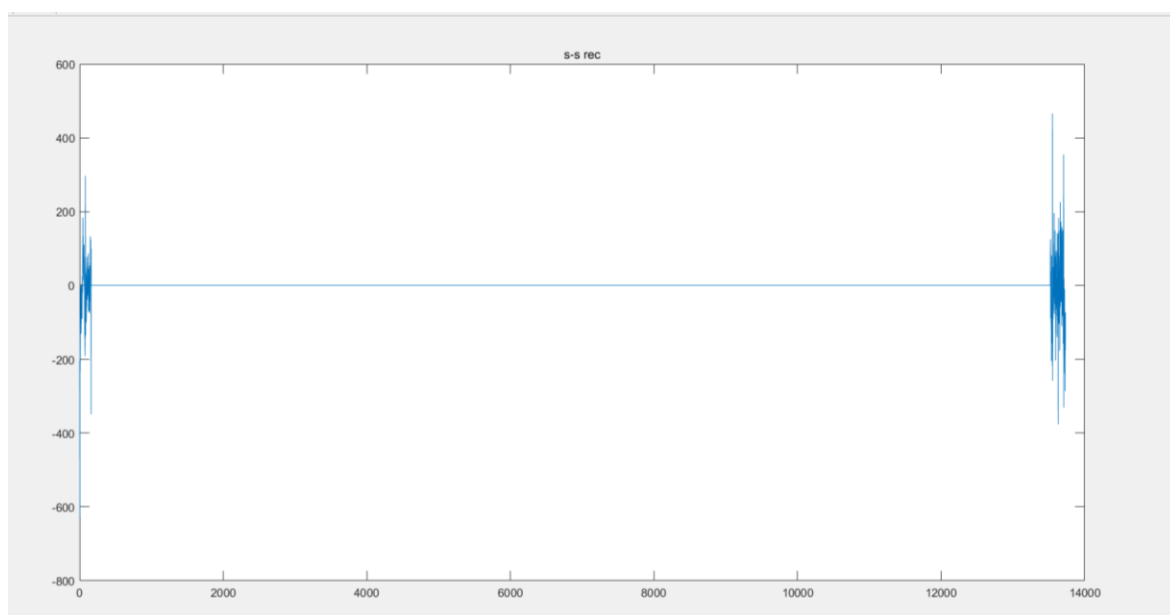
`zi_rec` 和 `zf_rec` 参数同样用来维持滤波器的状态不变，预测和重建滤波器由于是输入输出互换，因此 `b`，`a` 参数互换即可。

1.2.1_(6)

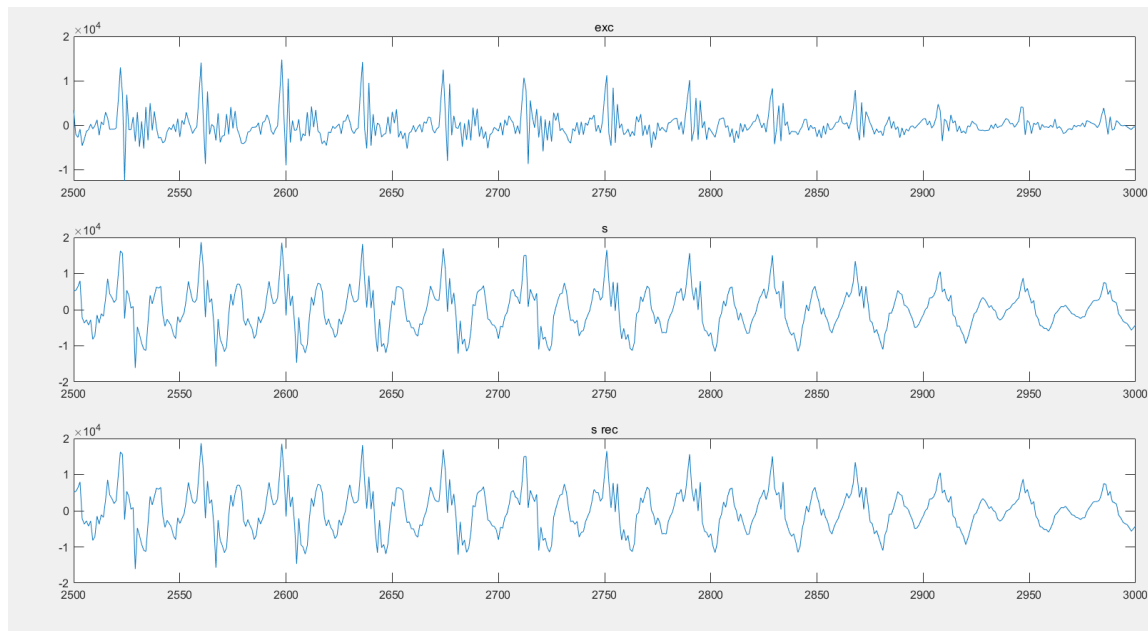
试听 $e(n)$ ，能分辨说话内容的同时能听到明显的噪声， $s(n)$ 和 $\hat{s}(n)$ 基本听不出差别，比较清晰噪声比 $e(n)$ 小



三个信号的整体波形，可以看出三个波形总体上趋势相同，但是 $e(n)$ 变化的更剧烈一些，从最大振幅和增减速度上看都比语音信号和重建语音更不光滑。



语音信号和重建信号作差得到的信号，由于重建从第三帧开始 n 很小的时候有明显波动，中间段基本是 0，差值很小。



选取 2500-3000 部分波形，可以看到语音信号和重建信号波形基本相同，激励信号相比之下高频分量更多，波形更不光滑。

1.2.2_(7)

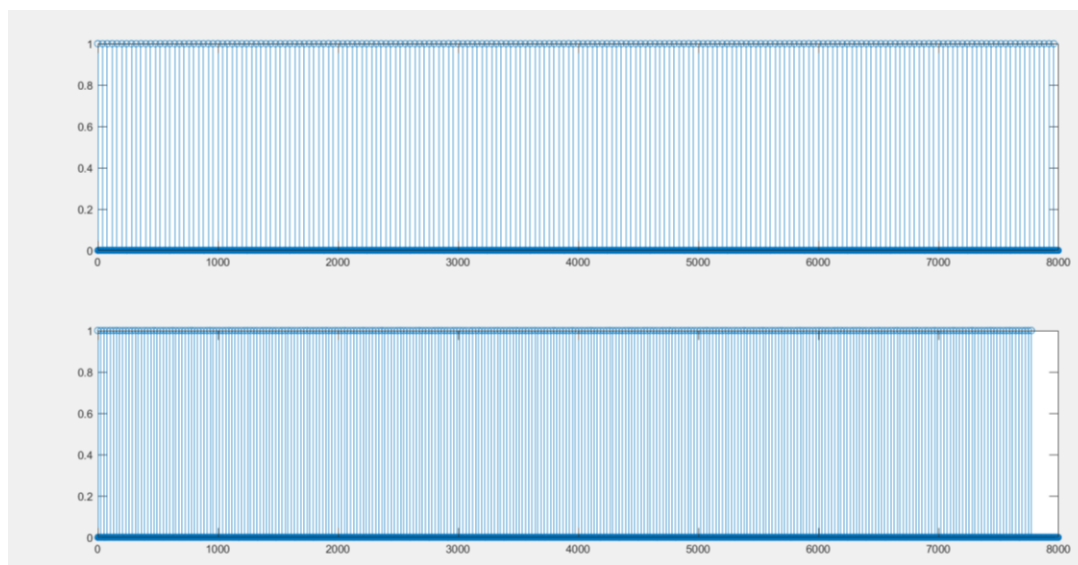
生成一个 8kHz 抽样的持续 1 秒钟的数字信号，该信号是一个频率为 200Hz 的单位样值“串”，即 $x(n) = \sum_{i=0}^{NS-1} \delta(n - iN)$ ，该信号的 N 和 NS 分别为何值？用 sound 试听这个声音信号。再生成一个 300Hz 的单位样值“串”并试听，有何区别？

由于 $8000=40 \times 200$ ，这个单位样值串的 200 个非零点(持续 1s 且频率 200Hz，所以是 200 个点)都能被抽到，因此 $NS=200$ ；抽样之后的信号其实没有变化，频率还是 200Hz，因此 $N = \frac{8000}{200} = 40$ 。

```
FS=8000;
n=(0:8000);
xn_1=fix(n/40)<200&fix(n/40)>=0&fix(n/40)==n/40;%得到200个脉冲
%sum(xn_1);%得到xn_1脉冲个数
%xn_2=fix(3*n/80)<300&fix(3*n/80)>=0&fix(3*n/80)==3*n/80;%这么写只能得到100个脉冲，太少了效果不好
xn_2=fix(n/26)<300&fix(n/26)>=0&fix(n/26)==n/26;%80/3向下取整得到26，有300个脉冲，效果较好
%sum(xn_2);%得到xn_2脉冲个数
sound(double(xn_1),FS);
sound(double(xn_2),FS);
figure(1)
subplot(2,1,1);
stem(n,xn_1);
subplot(2,1,2);
stem(n,xn_2);
```

生成 200Hz 时的判断条件是 $n = Ni$ ，即 $i = \frac{n}{40}$ ，因此仅当 $\frac{n}{40}$ 是在 0-199 之间的整数时序

列取 1，否则取 0；300Hz 时由于 $\frac{8000}{300}$ 不是整数，向下取整到 26，否则得到的脉冲只有 100 个，频率比 200Hz 的还要低。



上面的图是 200Hz，下面的图是 300Hz。

听起来 200Hz 的信号比 300Hz 的低沉一些，都是“嘀”的声音。

1.2.2_(8)

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

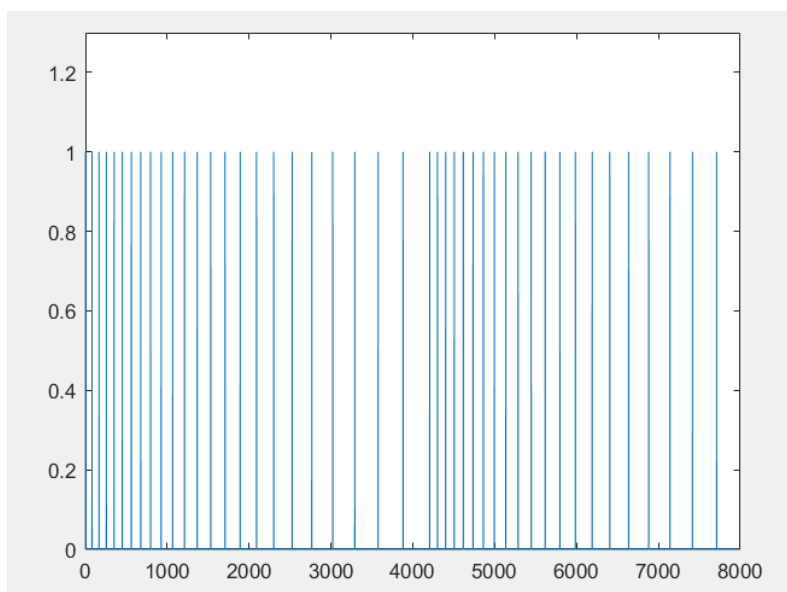
```

1 function e_n=hw122_8()
2     FS=8000;
3     e_n=zeros(8000,1);
4     n_loop=1;
5     while n_loop<=8000
6         m=round(n_loop/80);
7         PT=80 + 5*mod(m, 50);
8         e_n(n_loop,1)=1;
9         n_loop=n_loop+PT;
10
11     end
12     sound(e_n,FS)
13 end

```

激励信号 $e(n)$ 是 8000×1 的向量，直接用段序号 m 循环在段与段之间比较难处理，使用 n_loop 循环，由于 10ms 一段，1s 的语音共 8000 个样点，因此段序号 m 是 $\frac{n_loop}{80}$ 最近取整。 n_loop 每次循环加上 PT 保证两个脉冲之间间距是某一个脉冲所在段的 PT 。

试听激励信号的声音很像是拉链在尼龙材料上的摩擦声，能比较明显听出有两次摩擦声。



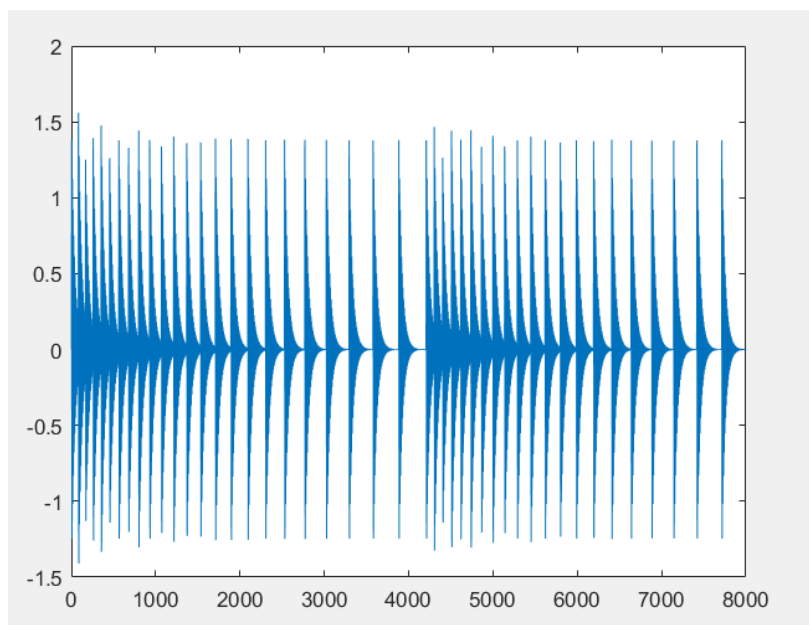
观察序列也可看出，中间较稀疏，两侧较密，反映到听到的声音就是中间有一处停顿
1.2.2_(9)

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。

在 `hw122_8` 中写子函数把（8）中的 $e(n)$ 传入（1）的系统。

```
23      %(9)  
24      e_n=hw122_8();  
25      s_n=filter(num,den,e_n);  
26      sound(s_n,FS);
```

试听 $s(n)$ 有些像打嗝的声音，相比 $e(n)$ 更像人能发出的声音，和 $e(n)$ 同样，声音明显分为两次。



和 $e(n)$ 类似，中间较稀疏，整体上波形变化比 $e(n)$ 缓和

1.2.2_(10)

重改 `speechproc.m` 程序。利用计算得到的基音周期和（8）中循环产生激励信号的方

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 $\hat{s}(n)$ 。试听和原始语音有何差别。

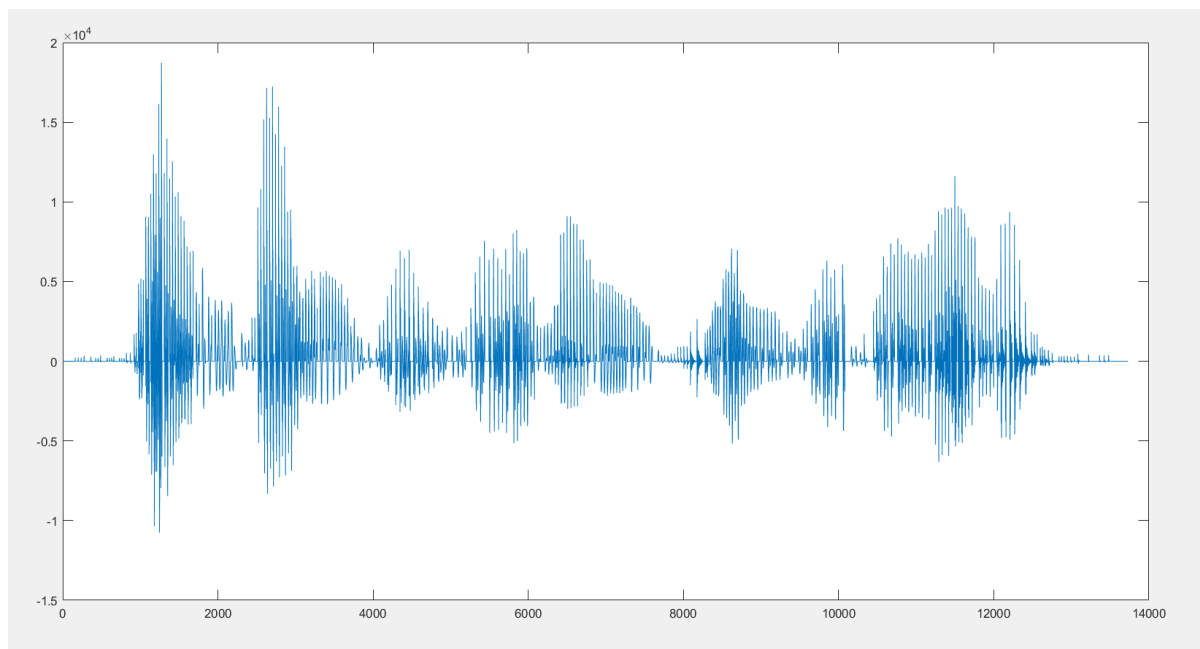
```
% (10) 在此位置写程序，生成合成激励，并用激励和filter函数产生合成语音
if(n==3)
    n_loop=(n-1)*FL+1;%初始化
end
x_n=zeros(FL,1);
while n_loop<= n*FL
    x_n(n_loop-((n-1)*FL),1)=1;
    n_loop=n_loop+PT;
end

exc_syn((n-1)*FL+1:n*FL) =G*x_n;%将你计算得到的合成激励写在这里
s_syn((n-1)*FL+1:n*FL) = filter(1,A,exc_syn((n-1)*FL+1:n*FL));%将你计算得到的合成语音写在这里
```

其中 x_n 是每一帧要计算的 $FL*1$ 的激励信号片段， n_loop 相当于记忆了上一次脉冲的准确位置，如果脉冲超过了当前 PT 对应帧的位置，则此帧的激励信号片段全是 0。

尝试像前面练习题一样用 `zi`、`zf` 保持滤波器状态，听起来更不清晰了，因此没有加上。

试听 $\hat{s}(n)$ 相较原语音噪音更重，感觉偏离正常人说话的声音，但能分辨出说的内容。



得到波形

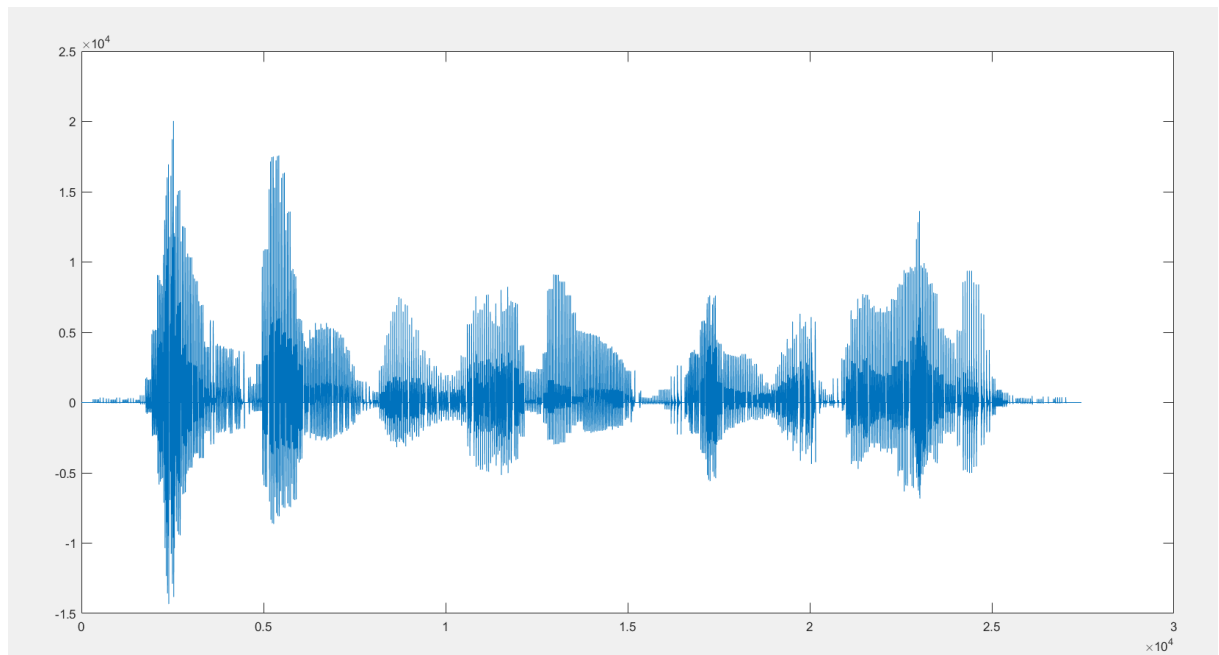
1.2.3_(11)

将 (10) 中合成激励的长度增加一倍,用同样的方法合成出语音，并用原始抽样速度播放。

```
% (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。
if(n==3)
    n_loop_v=(n-1)*FL_v+1;%初始化
end
x_n_v=zeros(FL_v,1);
while n_loop_v<= n*FL_v
    x_n_v(n_loop_v-((n-1)*FL_v),1)=1;
    n_loop_v=n_loop_v+PT;
end

exc_syn_v((n-1)*FL_v+1:n*FL_v) =G*x_n_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) =filter(1,A,exc_syn_v((n-1)*FL_v+1:n*FL_v));%将你计算得到的加长合成语音写在这里
```

除了 FL_v 替换了 (10) 中的 FL 外程序没有区别
试听基本是 (10) 中合成语音的不变调慢速版，噪音也慢速播放但还能分辨出讲话内容。



得到波形

1.2.4_(12)

将 (1) 中的系统共振峰频率提高 150Hz 后的 a_1 和 a_2 分别是多少？

```
1      a1=1.3789;
2      a2=-0.9506;
3      num=[0 0 1];
4      den=[1 -a1 -a2];
5      omega1=roots(den);
6      omega2(1)=omega1(1)*exp(150*pi/4000*1i);
7      omega2(2)=conj(omega2(1));
8      poly(omega2)
```

用 roots 求出两个共轭极点，存放在 omega 中，虚部是正数的排序在前，同 (1) 中共振峰频率计算公式可以得出 $\Omega = 2\pi T f$ ，因此在虚平面的辐角要转动 $2\pi T f = 150 \frac{\pi}{4000}$ ，乘上 e 的虚指数幂改变辐角，得到一个新极点，取共轭得到另一个极点，用 poly 得到最高次项归一化的以两个极点为根的多项式，得到 $a_1 = 1.2073$ 和 $a_2 = -0.9506$

```
>> hw124_12

ans =

    1.0000   -1.2073    0.9506
```

1.2.4_(13)

将 (10) 中基音周期改为一半,将共振峰频率增加 150Hz, 再合成出语音, 并试听。

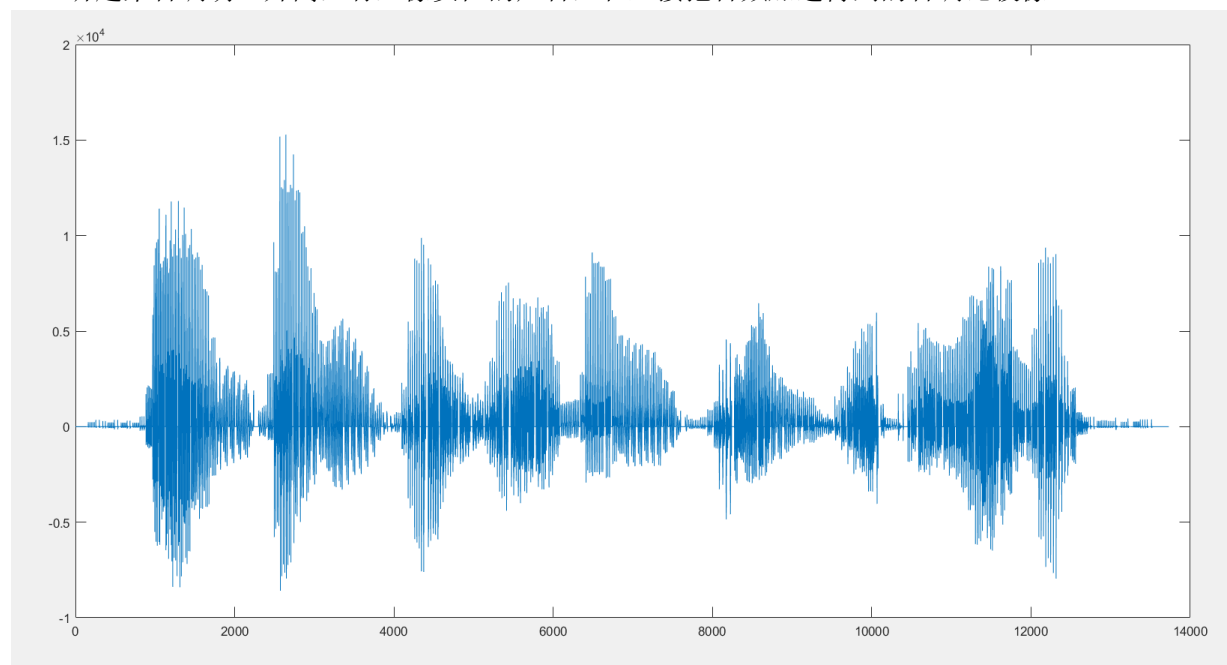
```
% (13) 将基音周期减小一半, 将共振峰频率增加150Hz, 重新合成语音, 听听是啥感受~
if(n==3)
    n_loop_t=(n-1)*FL+1;%初始化
end
x_n_t=zeros(FL,1);
while n_loop_t<= n*FL
    x_n_t(n_loop_t-((n-1)*FL),1)=1;
    n_loop_t=n_loop_t+round(PT/2);
end
[z_t,p_t,k_t]=tf2zp(1,A);
for i=1:length(p_t)
    if(imag(p_t(i))<0)
        p_t(i)=p_t(i)*exp(-150*pi/4000*1i);
    else
        p_t(i)=p_t(i)*exp(150*pi/4000*1i);
    end
end
end

[~,A_t]=zp2tf(z_t,p_t,k_t);
exc_syn_t((n-1)*FL+1:n*FL) =G*x_n_t;%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) =filter(1,A_t,exc_syn_t((n-1)*FL+1:n*FL)); %将你计算得到的变调合成语音写在这里
```

计算激励部分除了基音周期改为 $\text{round}(PT/2)$ 没有改变, 也即只是把 PT 改为离 $PT/2$ 最近的整数。

由初始系统的预测系数 A 和分子 1 通过 tf2zp 得到零极点和增益 k_t , 改变共振峰频率只需要改变极点, 类似 1.2.4_(12) 中的方法根据虚部正负乘上 e 的虚指数幂实现共振峰频率偏移。得到新极点后和原零点原增益通过 zp2tf 函数得到新的预测系数 A_t , 带入 filter 实现滤波。

听起来音调明显升高, 有些像女性的声音, 和直接把音频加速得到的音调比较像。



得到波形