信号与系统－ MATLAB 综合实验之语音合成[1]

谷源涛　　应启珩　　郑君里

二〇〇九年九月二日

[1]摘录于草稿，可能和纸质出版物不完全相同。

目录

第一章　语音合成

本章中将基于数字滤波器和 z 变换等基础知识，应用第一篇讲授的 MATLAB 编程技术，在语音分析合成领域做一些练习。通过本章的练习，可以增进对 z 变换和滤波器的理解，熟练运用 MATLAB 基本指令。本章包括两部分，第一部分介绍语音生成和分析的基本知识，第二部分给出详细的练习内容和编程步骤。相信读者对此会产生强烈兴趣。

第一节　背景知识

1.1.1　发声机理

从物理原理来看，语音信号是由肺挤压出的空气激励发声器官振动产生的。发声器官包括喉、声道和嘴。喉位于气管的上端，实际上是由气管末端的一圈软骨构成的一个框架。喉中有两片肌肉，它们和周围的韧带称为声带。声带张开时空气可以自由地流过喉和气管，如正常呼吸时；声带闭合，将喉封住，所以吃东西时食物不会落入气管。两片声带之间的空隙称为声门。说话时声带相互靠拢但不完全封闭，这样声门变成一条窄缝，当气流通过时其间压力减小，从而声带完全合拢使气流不能通过；在气流被阻断时压力恢复正常，因而声带间形成空隙，气流再次通过。这一过程周而复始，就形成了一串周期性的脉冲气流送入声道。如图 1.1 所示。这个脉冲串的周期称为“基音周期”，其倒数是“基音频率”。男性说话的基音频率在 60－200 Hz 范围内，女性和小孩在 200－450 Hz 之间。以上这种方式发出的音就是浊音。

气流从喉向上经过口腔或者鼻腔后向外辐射，经过的传输通道称为声道。气流流过声

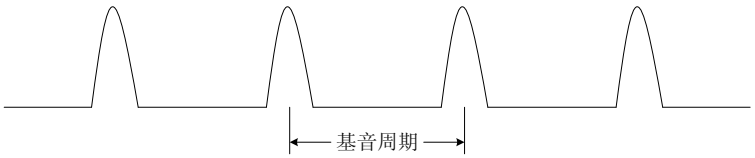

图 1.1: 典型的声门脉冲串波形

道犹如通过一个具有某种谐振特性的腔体。如图 1.2 所示。输出气流的频率特性既取决于声门脉冲串的特性，又取决于声道特性。声道包括口腔和鼻腔两部分，对成年男性而言，口腔段约 17cm，鼻腔段约 13cm，气流在软腭的控制下分别流向这两个通道。所以声道的截面积是变化的，而声道的频率特性主要取决于声道截面的最小值（收紧点）出现的位置，除了软腭控制一些外，收紧点主要由舌头的位置来决定。

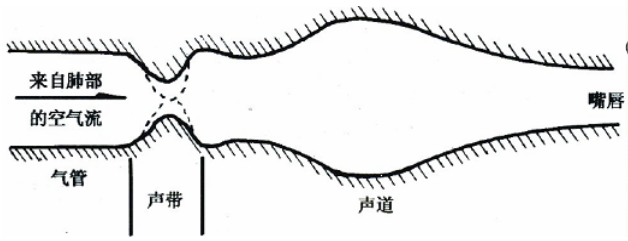

图 1.2: 声管构造示意图

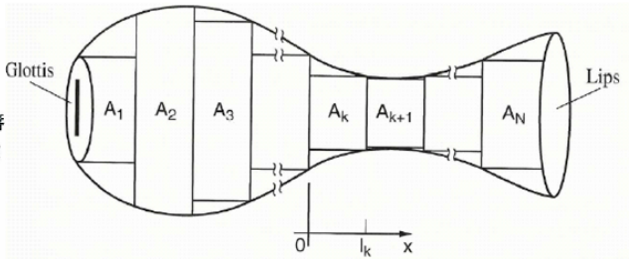

图 1.3: 级联无损声管模型

语音的另一种产生方式是声门完全闭合，此时声道不是受声门周期脉冲气流的激励，而是利用口腔内存有的空气释放出来而发声。该气流在口腔中形成湍流，因而表现为随机噪声。这种方式发出的音就是清音。（男生如果把手放在脖子前面喉结上部的倒三角位置，发浊音“啊”的音时可以感觉到声管的震动，发清音“是” 的音时就感觉不到。）

1.1.2　语音信号的时域特征

一段女声发音“MATLAB”的波形如图 1.4 所示，可以看出语音能量的起伏从而大致分辨出话语中的每个音节在此波形中的位置。我们把时间轴拉宽后在图 1.5 中观察两个细节部分，可以看出语音的浊音段能量较大（右上图），有明显的周期特征，而清音段能量很小（右下图），类似于噪声随机变化。

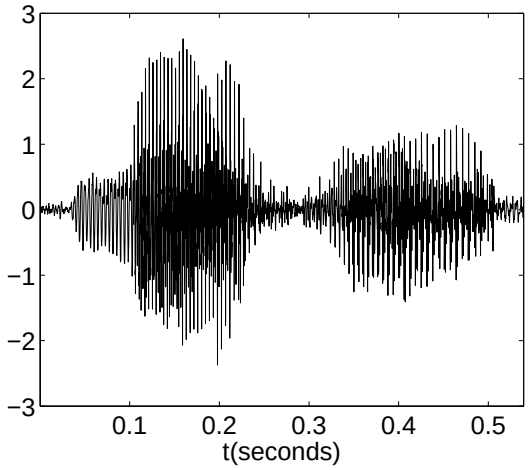

图 1.4: 女声发音“MATLAB”

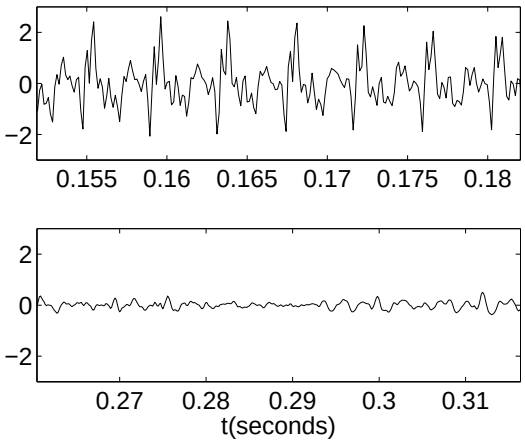

图 1.5: 女声发音“MATLAB”细节

1.1.3　语音模型

语音生成模型

通过对声管的研究，发现它可以用若干段截面积不等的均匀管道级联起来描述，如图 1.3，一般称作级联无损声管模型。采用流体力学的方法可以证明每一截均匀管道能够用一个单极点模型来近似，这样 N 段管道组成的声管就可以用一个 N 阶全极点滤波器表述，即

$$V(z) = \frac{G}{\prod_{k=1}^{N}(1 - p_k z^{-1})} = \frac{G}{1 - \sum_{k=1}^{N} a_k z^{-k}} \tag{1.1}$$

对于典型的男声，$N = 10$，所有的极点 p_i 要分别构成共轭对以保证 $\{a_i\}$ 系数都是实数。再综合考虑清音信号，就可以得到语音信号产生的离散语音模型，如图 1.6 所示。

准确的清浊音判决远远超出了本书的范畴，因而我们将对上述模型进行充分简化。首先去掉随机信号激励部分，我们认为激励信号是一个脉冲序列，不考虑有无周期。其次去掉声门脉冲模型和口唇的辐射模型，从而得到图 1.7 所示最简单的语音模型，现在我们用 z 变换的知识就可以应对了。

假设激励信号用 $e(n)$ 表示，语音信号用 $s(n)$ 表示，根据全极点模型表达式，有

$$s(n) = \sum_{k=1}^{N} a_k s(n-k) + Ge(n) \tag{1.2}$$

从而我们可以用声管模型对激励信号进行滤波得到语音信号。

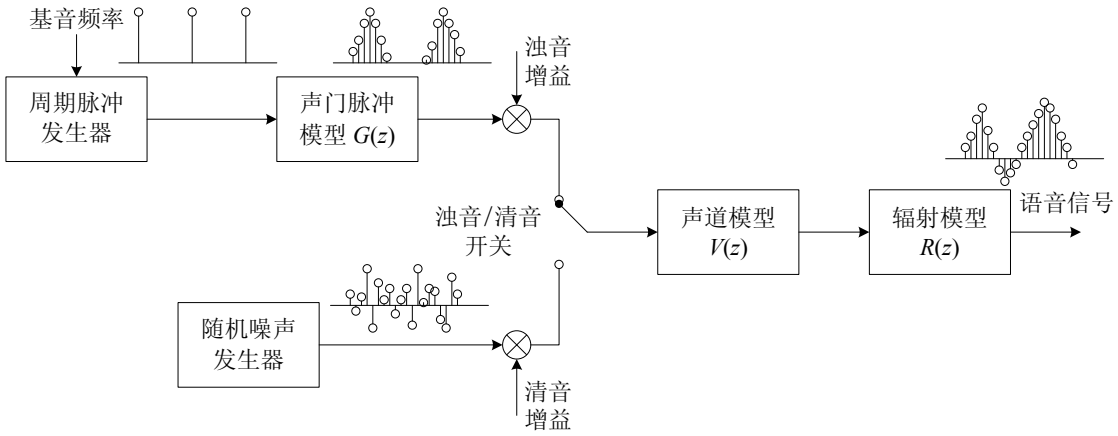

图 1.6: 产生语音信号的离散时域模型

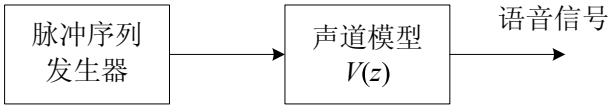

图 1.7: 简化的语音生成模型

语音预测模型

我们可以采集到语音信号 $s(n)$，也已经知道了它的生成模型如图 1.7 所示，但不知道激励 $e(n)$ 和模型 $V(z)$ 中的 $\{a_i\}$ 系数。根据原著第七章 7.7 节，我们知道这是一个解卷积问题，而且它是更复杂的盲解卷，因为激励和滤波器系数两者都不知道。如果进一步做些合理的假设，这个问题还是可以解决的，比如约束 $e(n)$ 是一个周期脉冲序列和一个高斯白噪声序列之和，我们就可以用一些信号处理方法，如自相关法和自协方差法求出系数 $\{a_i\}$ 来，并且有 Durbin 递推算法和 Schur 递推算法等快速方法。

假设我们已经知道了系数 $\{a_i\}$，那么将图 1.7 的输入和输出对换，就构成了语音的预测模型，即语音信号 $s(n)$ 送入预测滤波器，得到预测残差 $e(n)$，

$$e(n) = s(n) - \sum_{k=1}^{N} a_k s(n-k)$$

这种预测模型在通信中用来增加每个信道上传输语音信号的通道数。假设信号的发端和收端都知道预测系数 $\{a_i\}$，那么发端只需要把残差 $e(n)$ 传到收端即可，因为收端可以用 $e(n)$ 作为上述差分方程的激励得到重建语音。在发端，语音 $s(n)$ 是滤波器的输入，而误差 $e(n)$ 是输出。事实上，$\{a_i\}$ 系数当然也需要从发端传到收端，但因为语音具有短时平稳性，

即在短时间内（比如 10 毫秒），$\{a_i\}$ 系数可以认为不发生变化，所以也不必太频繁的传输，因而采用预测技术后，总是可以大幅度地降低语音的带宽。这种通过线性预测方法压缩语音数据量的技术叫做线性预测编码（ Linear Prediction Coding, LPC ）技术。

语音重建模型

如果已知激励信号 $x(n)$（先不考虑是如何得到的）和滤波器系数 $\{a_i\}$，我们就可以利用语音生成模型重建语音了，

$$\hat{s}(n) = x(n) + \sum_{k=1}^{N} a_k \hat{s}(n-k)$$

但我们把上述模型称作语音重建模型，为了同生成模型区分开，$\hat{s}(n)$ 称为重建语音。如果 $x(n)$ 正好等于 $e(n)$，那么重建语音就和原始语音 $s(n)$ 完全相同。

语音的非平稳性（虽然短时平稳）导致预测系数 $\{a_i\}$ 是时变的，一般每 10 − 20 毫秒就会发生一些变化以产生不同的音节。在这种情况下，滤波过程也要分段进行，即每次用不同的滤波器系数，但相邻两次滤波必须要保持滤波器的状态不发生变化。

谐振和共振峰频率

语音生成模型的每一对共轭极点都对应一个衰减的正弦信号的特征响应。例如一对共轭极点 $\mid p_i \mid \mathrm{e}^{\pm j\Omega}$ 在时域冲激响应中的贡献是 $A \mid p_i \mid^n \cos(\Omega n + \varphi)$ 。其中极点幅度决定衰减速度，幅角决定振荡频率。

对语音合成，用数字的正弦信号表示抽样后的连续正弦信号。在这种情况下，模拟频率和数字频率的关系是 $\Omega = \omega T$,其中 T 表示抽样间隔，ω 表示模拟频率（弧度），对应的 $f = \omega/2\pi$ 称作共振峰频率，它定义了声管的谐振频率。典型的男声（ $N = 10$ ）可以用 5 个共振峰频率来描述。当模型参数变化时，共振峰频率也随着变化，从而产生不同的声调。

1.1.4 分析和合成语音

我们的分析和合成系统如图 1.8 所示。

首先要分析一段（一般是 10 毫秒）语音得到它的最佳 $\{a_i\}$ 系数。给定这些系数后，我们就可以用适当的输入来合成语音。对于浊音信号，一种可取的激励模型就是以特定频率重复的单位样值序列，这个频率就是基音频率。对清音，最好选择随机噪声或白噪声作为

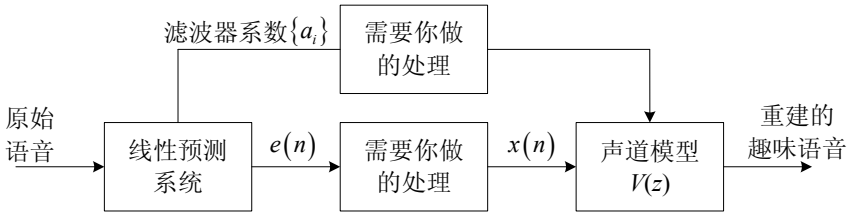

图 1.8: 分析和合成语音的系统框图

输入。但在不做清浊音判决的情况下，全部采用周期激励的合成质量也是可以接受的（我们就是这么做的）。

分析过程是：先将语音采样信号以 10 毫秒进行分段，然后对得到的每段数据进行统计分析并计算相邻样点的相关性，最终求得最佳预测系数。合成过程就是利用这些预测系数，以及周期的单位样值序列作为输入，依次得到每段合成语音。

变速不变调

变速不变调技术广泛应用于消费类电子产品，如英语复读机等。所谓变速不变调，是指声音播放时，速度的改变不会导致音调的变化。一般来说，用随身听听音乐，快进播放和慢速播放，其音调是不一致的，如快进播放，频率会变高，男声听起来会感觉是女声。（原因是什么？想一想 $\Omega = \omega T$，如果 Ω 不变 T 却减小了，则 ω 会怎样？）为了实现变速不变调，首先需要将表示“调”的内容从语音中分离出来，由前述语音预测模型，表示“调”的有两个部分，一是共振峰频率，即预测模型的参数；二是基音周期，即激励信号的参数。下面的工作就是在不改变这两种参数的前提下改变数据长度的问题了。即将对应于 10 毫秒的 80 个样点的激励变成对应于 20 毫秒的 160 个样点（注意保持单位样值的周期不变），在这 20 毫秒内保持预测模型系数不变，不就可以合成出 20 毫秒的合成语音了么？新语音的声调和原有语音是完全相同的，只不过时间变长了而已。

变调不变速

前面说过，最简单的男声变女声只要让随身听快进播放就可以了。但快进播放改变的不仅是声调，语速也会发生很大变化，快得让人听不清楚。为了解决这个问题，就需要用变调不变速的技术。同上分析过程一样，还是需要在共振峰频率和基音周期上做改变。女声和男声的最大区别是频率高，一方面表现在基音频率高，另一方面共振峰对应的谐振频率也更高一些，所以我们可以考虑将激励信号的频率增加（注意不改变信号长度），同时将共

振峰频率也相应增大一些（即极点的幅角绝对值增大，或者说上半平面的极点逆时针旋转，下半平面的顺时针旋转，但注意两者都要旋转同样角度而且不要转过负实轴）。这样得到的合成语音会更“女声”一些。

后两项技术是典型的语音信号数字处理技术。它们的基础是 z 变换和线性预测模型，用传统的模拟信号处理方法不可能实现，这正体现了数字信号处理的优点。最后还需指出，这两项技术并不矛盾，事实上，它们可以完美地结合在一起，你能做出一种速度和音调都发生变化的合成语音算法来么？

第二节　练习题

1.2.1　语音预测模型

（1）给定

$$e(n) = s(n) - a_1 s(n-1) - a_2 s(n-2)$$

假设 $e(n)$ 是输入信号，$s(n)$ 是输出信号，上述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 分别绘出零极点图，频率响应和单位样值响应。用 filter 绘出单位样值响应，比较和 impz 的是否相同。

（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。

```
function speechproc()
    % 定义常数
    FL = 80;                                    % 帧长
    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
% 依次处理每帧语音
for n = 3:FN
    % 计算预测系数(不需要掌握)
    s_w = s(n*FL-WL+1:n*FL).*hw;       %汉明窗加权后的语音
    % A是预测系数,E会被用来计算合成激励的能量
    [A E] = lpc(s_w, P);               %用线性预测法计算P个预测系数
    if n == 27
    % (3) 在此位置写程序,观察预测系统的零极点图
    end
    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函数和s_f计算激励,注意保持滤波器状态
    % exc((n-1)*FL+1:n*FL) = ...       将你计算得到的激励写在这里
```

```
    % (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-1)*FL+1:n*FL) = ...       将你计算得到的重建语音写在这里
    % 注意下面只有在得到exc后才会计算正确
    s_Pitch = exc(n*FL-222:n*FL);
    PT = findpitch(s_Pitch);          % 计算基音周期PT(不要求掌握)
    G = sqrt(E*PT);                   % 计算合成激励的能量G(不要求掌握)
    % (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
    % 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) = ...       将你计算得到的合成语音写在这里
    % (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，而且音调没有变。
    % 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) =  将你计算得到的加长合成语音写在这里
    % (13) 将基音周期减小一半，共振峰频率增加150Hz，重新合成听听感受
    % 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) =    将你计算得到的变调合成语音写在这里
end
% (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
% 保存所有文件
writespeech('exc.pcm',exc);
writespeech('rec.pcm',s_rec);
writespeech('exc_syn.pcm',exc_syn);
writespeech('syn.pcm',s_syn);
writespeech('exc_syn_t.pcm',exc_syn_t);
writespeech('syn_t.pcm',s_syn_t);
writespeech('exc_syn_v.pcm',exc_syn_v);
writespeech('syn_v.pcm',s_syn_v);
```

```
return
% 从PCM文件读入语音
function s = readspeech(filename, L)
    fid = fopen(filename, 'r');
    s = fread(fid, L, 'int16');
    fclose(fid);
return
% 把语音写入PCM文件
function writespeech(filename,s)
    fid = fopen(filename,'w');
    fwrite(fid, s, 'int16');
    fclose(fid);
return
% 计算一段语音的基音周期，不要求掌握
function PT = findpitch(s)
    [B, A] = butter(5, 700/4000);
    s = filter(B,A,s);
    R = zeros(143,1);
    for k=1:143
        R(k) = s(144:223)'*s(144-k:223-k);
    end
    [R1,T1] = max(R(80:143));
    T1 = T1 + 79;
    R1 = R1/(norm(s(144-T1:223-T1))+1);
    [R2,T2] = max(R(40:79));
    T2 = T2 + 39;
    R2 = R2/(norm(s(144-T2:223-T2))+1);
    [R3,T3] = max(R(20:39));
```

```
    T3 = T3 + 19;
    R3 = R3/(norm(s(144-T3:223-T3))+1);
    Top = T1;
    Rop = R1;
    if R2 >= 0.85*Rop
        Rop = R2;
        Top = T2;
    end if R3 > 0.85*Rop
        Rop = R3;
        Top = T3;
    end
    PT = Top;
return
```

（3）运行该程序到 27 帧时停住，用（1）中的方法观察零极点图。

（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，需维持滤波器的状态不变，要利用 filter 的 zi 和 zf 参数。

（5）完善 speechproc.m 程序，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 $\hat{s}(n)$。同样要注意维持滤波器的状态不变。

（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 $\hat{s}(n)$ 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。

1.2.2 语音合成模型

（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 $e(n)$ 。

（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

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

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。）

（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。

（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 $\tilde{s}(n)$ 。试听和原始语音有何差别。

1.2.3 变速不变调

（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。

1.2.4 变调不变速

（12）重新考察（1）中的系统，将其共振峰频率提高 150Hz 后的 a_1 和 a_2 分别是多少？

（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。

✍ MATLAB 知识点 (1) —低级文件访问

为充分满足用户的各种要求，MATLAB 也提供包括 fopen ， fread ， fwrite ， fseek 和 fclose 等在内的一套直接访问二进制文件的函数，这些函数名与 ANSIC 编程语言中的标准函数非常相似，请读者查阅帮助学习。
