UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering

ECE 498MH Signal and Image Analysis

Homework 9

Fall 2013

Assigned: Friday, November 15, 2013 Due: Friday, November 22, 2013

Reading:

Problem 9.1

Suppose that the glottis closes at time $n = 0$. After the glottis closes, the vocal tract "rings"—ringing means that it responds with decaying sine waves. For example, the sine wave corresponding to the first formant is

$$s[n] = \begin{cases} e^{-\pi B_1 n/F_s} \sin\left(2\pi F_1 n/F_s\right) & n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

That $s[n]$ is just a little too complicated to solve using pencil and paper, so instead, let's work with a signal that decays without oscillating:

$$s[n] = \begin{cases} e^{-\pi B_1 n/F_s} & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

(a) Remember that the autocorrelation is defined to be $R[m] = E\left[s[m]s[m-\tau]\right]$. For this problem, we will approximate the expected value with a long-term average, thus let's define

$$R[\tau] = \sum_{m=-\infty}^{\infty} s[m]s[m-\tau] \tag{2}$$

Using the definitions in Eq. 1 and 2, find $R[\tau]$. Hint: use the formula $\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}$.

(b) Use your answer from part (a) to find the particular values $R[0]$ and $R[1]$.

(c) Remember that the general formula for the LPC coefficients is $\vec{r} = R\vec{a}$, where $\vec{a}$ is the vector of LPC coefficients, R is the matrix of autocorrelation coefficients, and $\vec{r}$ is a vector of autocorrelation coefficients. Let's simplify to the case of a first-order LPC filter, $s[n] = e[n] + a_1 s[n-1]$. In that case, the formula for the LPC coefficients simplifies to

$$R[1] = R[0]a_1 \tag{3}$$

so $a_1 = R[1]/R[0]$. Find a_1 for the signal in parts (a) and (b).

(d) The LPC synthesis filter is then

$$s[n] = a_1 s[n-1] + \delta[n] \tag{4}$$

Check to see that Eq. 1 satisfies Eq. 4, using the coefficient you found in part (c).

Matlab Exercises

Problem 9.2

In this problem, let's try to build an auto-tuner.

(a) Use `wavrecord` or `audiorecorder` to record your own voice at $F_s = 8000$ samples/second. Probably you just want to record yourself saying a vowel. Find a 100ms segment near the peak of the vowel, cut it out, and call it `x`. Create a time axis `t=[0:799]/8000;`, and `plot(t,x);` to show the waveform. Print and hand in.

(b) Use the `xcorr` function to find the autocorrelation coefficients, $Rx[0]$ through $Rx[12]$. Place these in a vector `Rx` such that $Rx(m) = R[m-1]$. Create a 12x1 vector `rvec`, and a 12x12 `Rmat` as follows:

```
rvec=zeros(12,1);
Rmat=zeros(12,12);
for m=1:12,
rvec(m) = Rx(m+1);
for n=1:12,
  Rmat(m,n) = Rx(abs(m-n)+1);
end
end
```

You should now be able to find the LPC coefficients by typing `avec=inv(Rmat)*rvec;`.

(c) Plot the absolute value of the fft of `x` in the top subwindow:

```
  omega=[0:799]*2*pi/800;subplot(2,1,1);plot(omega,abs(fft(x)));
```

In the bottom subwindow, plot the function $|H(e^{j\omega})| = 1/|A(e^{j\omega})|$, thus

```
A = zeros(size(z));
H = zeros(size(z));
for k=1:size(HMAG),
  A(k)=1;
  for i=1:12,
   A(k) = A(k) - avec(i)*exp(-j*omega(k)*i);
  end
  H(k) = 1/A(k);
end
subplot(2,1,2);
plot(omega,abs(H));
```

You should see that they have similar shape.

(d) The analysis filter is $A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i}$. Filtering by $A(z)$ is the same as convolving with the filter `afilt=[1,-avec'];`. Try it: `e=conv(x,afilt);`. Plot it: `plot(e);` You should see something that has impulses once per pitch period, and little other information.

(e) Chop out one complete pitch period, call it `oneperiod`. Cut from one zero-valued sample to another one about T_0 seconds later—thus you should probably chop about halfway between two glottal closure instants, not right at the GCIs.

Create an autotuned excitation signal, `autoex`, by repeating your one period 20 times in a row. Space the periods out at exactly 440.44 Hz, thus at $T_0 = 1/440.44$, thus at $N_0 = 8000/440.44 = 18$ samples:

```
T0=length(oneperiod);
autoex=zeros(1,18*19+T0);
for k=1:20,
  autoex(18*(k-1)+[1:T0])=autoex(18*(k-1)+[1:T0])+oneperiod;
end
```

Now use the `filter` function, together with `afilt`, to resynthesize a vowel in your own voice, but at exactly 440.44Hz.

Create a plot with four subplots. In the top one, show the original 100ms vowel. In the second one, show the excitation signal $e[n]$. In the third one, show the synthesized autotune excitation signal. In the fourth one, show the synthesized autotune vowel after LPC resynthesis. Hand in this plot.

Play the autotune vowel. It should sound like you, as you would sound if you were using autotune to create a pop song.