CS630: Speech Technology
LAB-4: Basics of DSP

OBJECTIVES:

To study quantization and aliasing effects.
To synthesize vowels /a/, /i/ and /e/

Tasks

(a) Record a short utterance of speech for 2 to 3 sec (8 kHz, 16 bits).

(b) Listen to speech at different bit rates (16 bits/sample, 8 bits/sample and 1 bit/sample).

(c) Listen to speech at different sampling rates (8 kHz, 4 kHz and 2 kHz).

(d) Synthesize vowels $/i/$, $/e/$ and $/a/$ for the actual pitch period, half the pitch period and twice the pitch period.

(e) Write a brief note on the observations.

1 PROCEDURE:

1. **Recording Speech Signal**

   - Record a short sentence of speech using 'brec' command brec -s 8000 -b 16 -t 2 -w samplefile.wav

     brec -s 8000 -b 16 -t 2 -w samplefile.wav

     where "brec" is the linux command

     "s" is the Sampling frequency in Hz

     "b" is the number of bits/sample used for quantization

     "t" is the time interval used to record the speech signal

     "w" format (wav) in which the speech signal to be represented

2. **Study of Quantization Effect**

   - The short segment of speech recorded is at a sampling frequency of $8kHz$ and resolution of 16 bits/sample.
   - Load this segment of speech into MATLAB workspace using
     $- >> p =$ wavread(filename1.wav);
     where wavread is MATLAB function which reads contents of a file in wave format into a array.
   - Change resolution from 16 bits/sample to 8 bits/sample
     $- >>$ wavwrite(p,8000,8,filename2.wav);
     $- >> p1 =$ wavread(filename2.wav);
   - Change resolution from 16 bits/sample to 1bit/sample $- >> p2 = [p1 > 0]$;
   - Now p, p1, p2 respectively contains same speech at different resolutions 16 bits/sample, 8bits/sample and 1 bit/sample, but all at the same sampling frequency 8 kHz.
   - Play these sequences and note the observations:
     $- >>$ sound(p,8000); $- >>$ sound(p1,8000); $- >>$ sound(p2,8000);

3. **Study of Aliasing Effect**

- The short segment of speech recorded is at a sampling frequency of 8 kHz and 16 bits/sample resolution.
- Load this segment into MATLAB workspace using
  $->> q1 = wavread(filename1.wav);$
- Choose every alternate sample from q1
  $->> i = [1 : 2 : length(q1)];$
  $->> q2 = q1(i);$

- Choose every alternate sample from q2
  $->> j = [2 : 2 : length(q2)];$
  $->> q3 = q2(j);$

- Now q1, q2 and q3 respectively contains 8000, 4000, 2000 samples/sec. These are equivalent to sampling the speech respectively at 8 kHz, 4 kHz and 2 kHz (Low pass filtering is not done for studying the aliasing effect).
- Play these sequences and note the observations:
  $-$ >> sound(p,8000);
  $-$ >> sound(p1,8000);
  $-$ >> sound(p2,8000);

4. **Synthesizing vowels** $/a/, /i/and/e/$

- Given formant frequencies F_1, F_2 and F_3 of the vowel, pitch period T_0 and sampling frequency F_s at which a vowel is to be synthesized.
- The system used in the production of vowels is an all pole filter. Compute the parameters of the filter using formant frequencies and sampling frequency. The parameters are the coefficients of the numerator and denominator polynomials (b_k s anda_k s) of the system transfer function.
- The source used for excitation is a periodic unit sample sequence.Generate unit sample sequence of required length, depending on the length of the speech segment to be generated.

- Obtain response of the system with excitation generated using
  − >> y = filter(b, a, x) ;
  − >> sound(y);

Illustration of Synthesizing Vowel $/a/$:

- Given,
  -Formant frequencies F_1 = 560 Hz, F_2= 1180 Hz, and F_3= 2480 Hz
  -Pitch period, $T_0 = 7.5$ msec
  -Sampling frequency, $F_s = 10000Hz$

- The system $H(z)$, shown in Figure 1 is an all-pole filter. When input excitation $x(n)$ is unit sample sequence, then its response $y(n)$ will be damped sinusoids as shown in Figure 2.

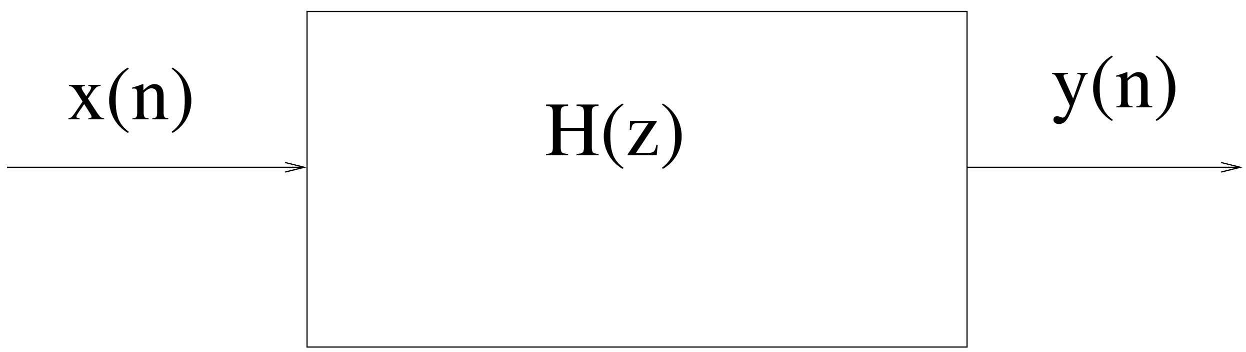

Figure 1: All-pole filter.

- The Transfer function $H(z)$ is given by,

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k.z^{-k}}{\sum_{k=0}^{N-1} 1 + a_k.z^{-k}} = \frac{1}{1 + \sum_{k=1}^{6} 1 + [a_k z^{-k}]} \qquad (1)$$

- For a given formant frequency F_i, let its bandwidth be $B_i = 0.1 * F_i$
- For each (F_i, B_i), system parameters are computed using

$$H_i(z) = \frac{1}{1 - 2.exp(-pi.B_i.T).cos(2.pi.F_i.T).z^{-1} + exp(-2.pi.B_i.T).z^{-2}} \quad (2)$$

where T is sampling rate = $\frac{1}{Fs}$ = 100*10^{-6} sec

- Substituting $F_1 = 560$ Hz, $B_1 = 56$Hz in Equation 1 and simplifying we get

$$H_1(z) = \frac{1}{1 - 1.8447 * z^{-1} + 0.9654 * z^{-2}} \quad (3)$$

- Substituting $F_2 = 1180$ Hz, $B_2 = 118$ Hz in equation 1 and simplifying we get

$$H_2(z) = \frac{1}{1 - 1.4213 * z^{-1} + 0.9285 * z^{-2}} \quad (4)$$

- substituting $F_3 = 2480$ Hz, $B_3 = 248$ Hz in Equation 1 and simplifying we get

$$H_3(z) = \frac{1}{1 - 0.0232 * z^{-1} + 0.8557 * z^{-2}} \quad (5)$$

- The system transfer function $H(z)$ is given by, $H(z) = H_1(z).H_2(z).H_3(z)$
- Substituting for $H_1(z), H_2(z), H_3(z)$ and simplifying we get

$$H(z) = \frac{1}{1 - a_1 z^{-1} + a_2 z^{-2} - a_3 z^{-3} + a_4 z^{-4} - a_5 z^{-5} + a_6 z^{-6}} \quad (6)$$

  where a = $[1, a_1, a_2, a_3, a_4, a_5, a_6]$ is shown below:

- From Equation 6 we have b= [1];
  a= $[1, -3.2892, 5.4471, 4.8318, -2.6604, 0.7669]$;

- We can also compute values of a_k s from $H_1(z), H_2(z) and H_3(z)$ by convolution
  $-$ >> t = conv([1,-1.8447,0.9654], [1, -1.4213, 0.9285]);
  $-$ >> a = conv(t,[1, -0.0232, 0.8557]);

- The values obtained by convolution are ,
  $a = [1, -3.2892, 5.472, -5.9844, 4.8321, -2.6606, 0.7670]$;
  This gives the system parameters.

- Generate the unit sample sequence to be used as the excitation at the actual pitch period (7.5 msec i.e., around 60samples at 8 kHz) as
  − >>x1 = zeros(1, 4000);
  − >>for i =1:60:length(x1)
  x1(i) = 1;
  end;

- Similarly, excitation sequences at half the pitch period and twice the pitch period are generated as follows :
  − >>x2 = zeros(1, 4000);
  − >>for i =1:30:length(x1)
  x1(i) = 1;
  end;
  − >>x3 = zeros(1, 4000);
  − >>for i =1:120:length(x1)
  x1(i) = 1;
  end;

- Now using the excitation sequence, the output sequence is generated using − >>y1 = filter(b,a,x1);
  Figure 3 shows the excitation sequence and the signal synthesized for the half pitch period case. − >>y2 = filter(b,a,x2);
  Figure 4 shows the excitation sequence and the signal synthesized for the pitch period case.
  − >>y3 = filter(b,a,x3);

  Figure 5 shows the excitation sequence and the signal synthesized for twice the pitch period case.

- Play the output sequence using sound command − >>sound(y1);
  − >>sound(y2);
  − >>sound(y3);
  Figure 6,7,8 shows the synthesized signal for the sound unit /a/ and the corresponding spectrograms for excitation with a period of half the pitch, pitch, twice the pitch respectively.

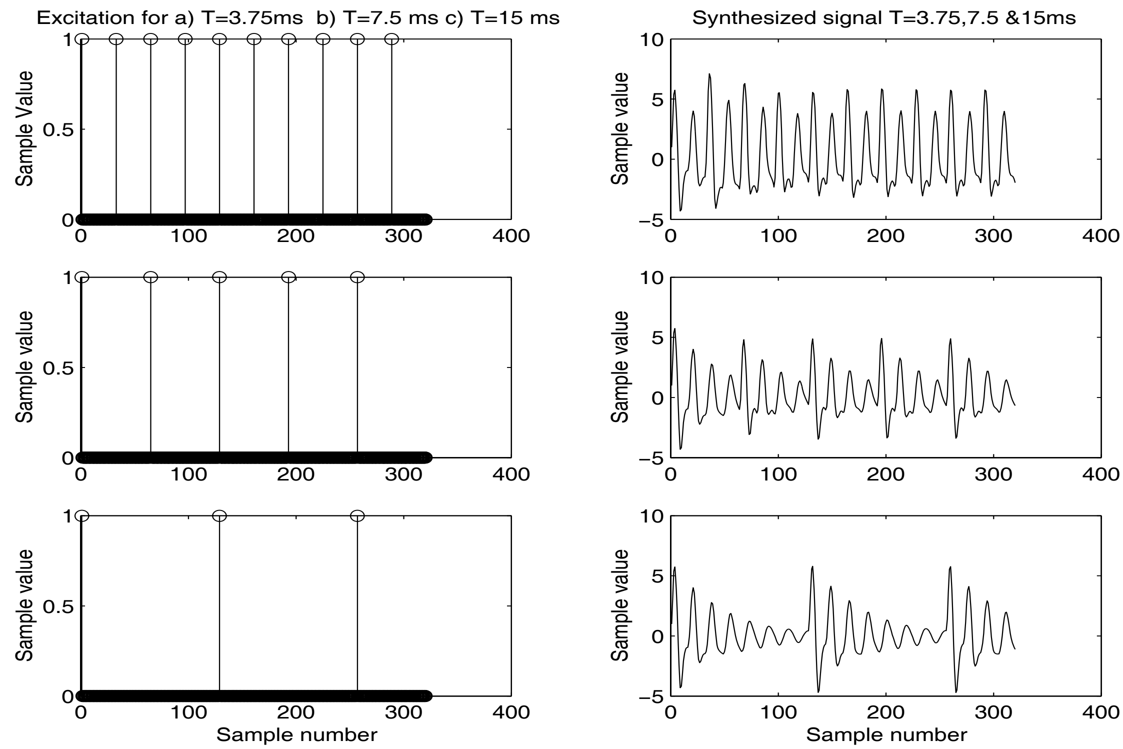

Figure 2: The excitation and the response of the all-pole filter for impulse sequence excitation with a period of 3.75ms, 7.5ms and 15ms respectively.

2 CONCLUSIONS

- The quality of speech is directly proportional to the number of quantization levels (that is, number of bits used for quantization).Though quantization results in loss of information the human perception mechanism can still get the information present in the speech signal.
- It is interesting to note that even with 1 bit/sample, most of the speech is intelligible. Hence we can conclude that information lies in the sequence and not in the set of numbers.
- The aliasing effect is perceived as distortions in the output signal. This is due to the effect of overlapping of frequency components.

- The synthesized vowel sounds as that of vowel except at the cost of naturalness. The formants are clearly defined and there is no variability in the signal as it is a synthetic siganl.

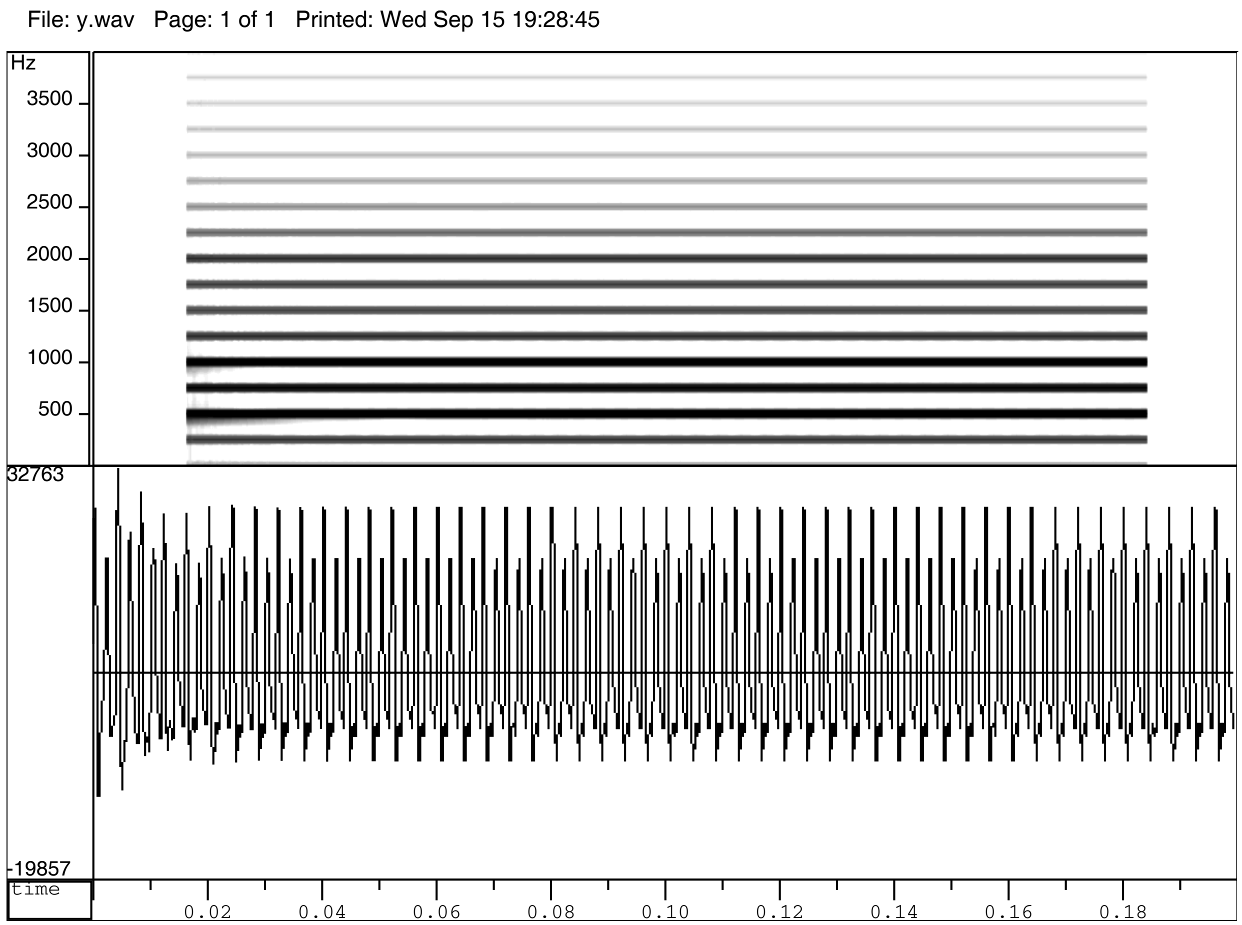

Figure 3: Waveform of the the synthesized vowel /a/ and its spectrogram for excitation at half the pitch period.

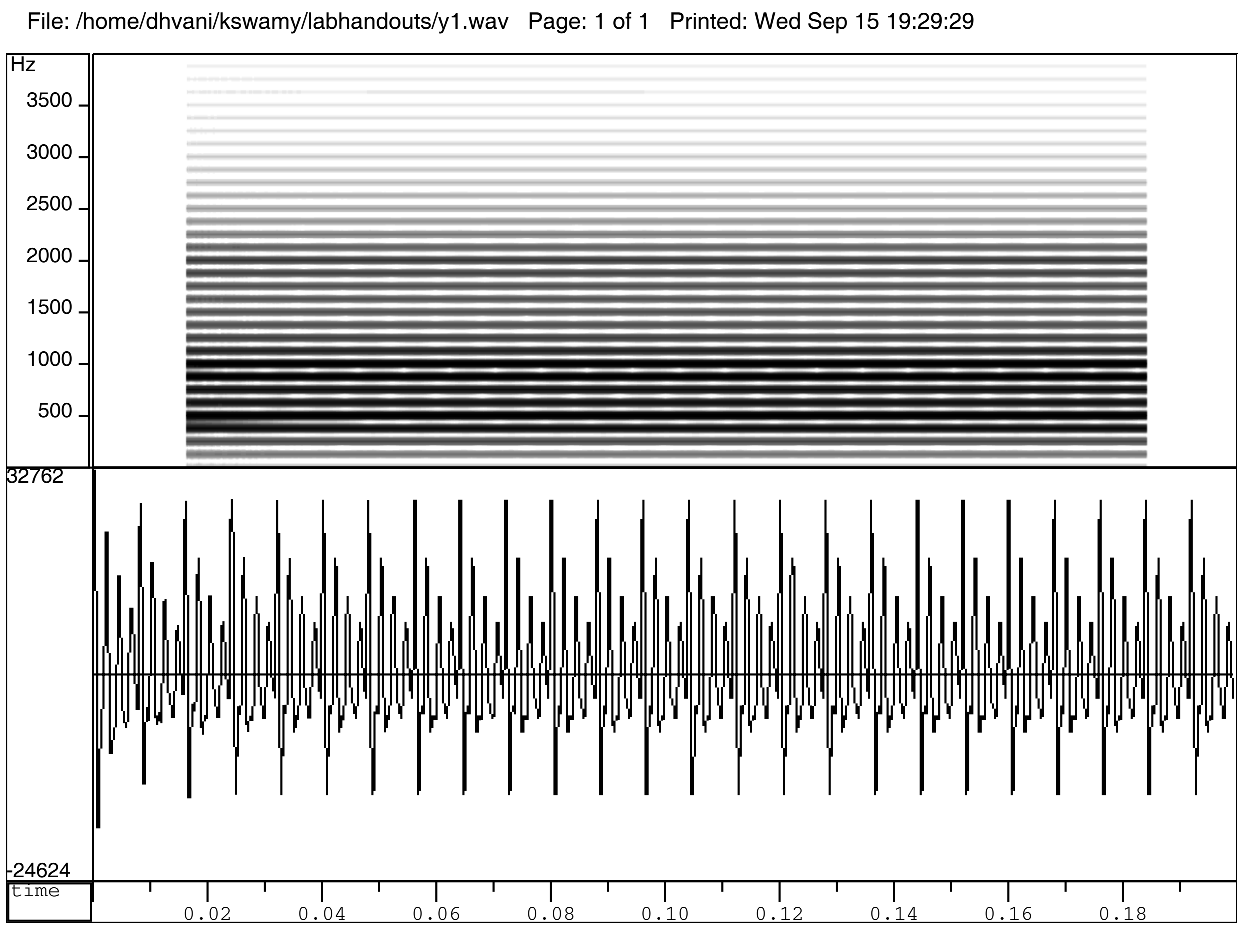

Figure 4: Waveform of the the synthesized vowel /a/ and its spectrogram for excitation at the pitch period.

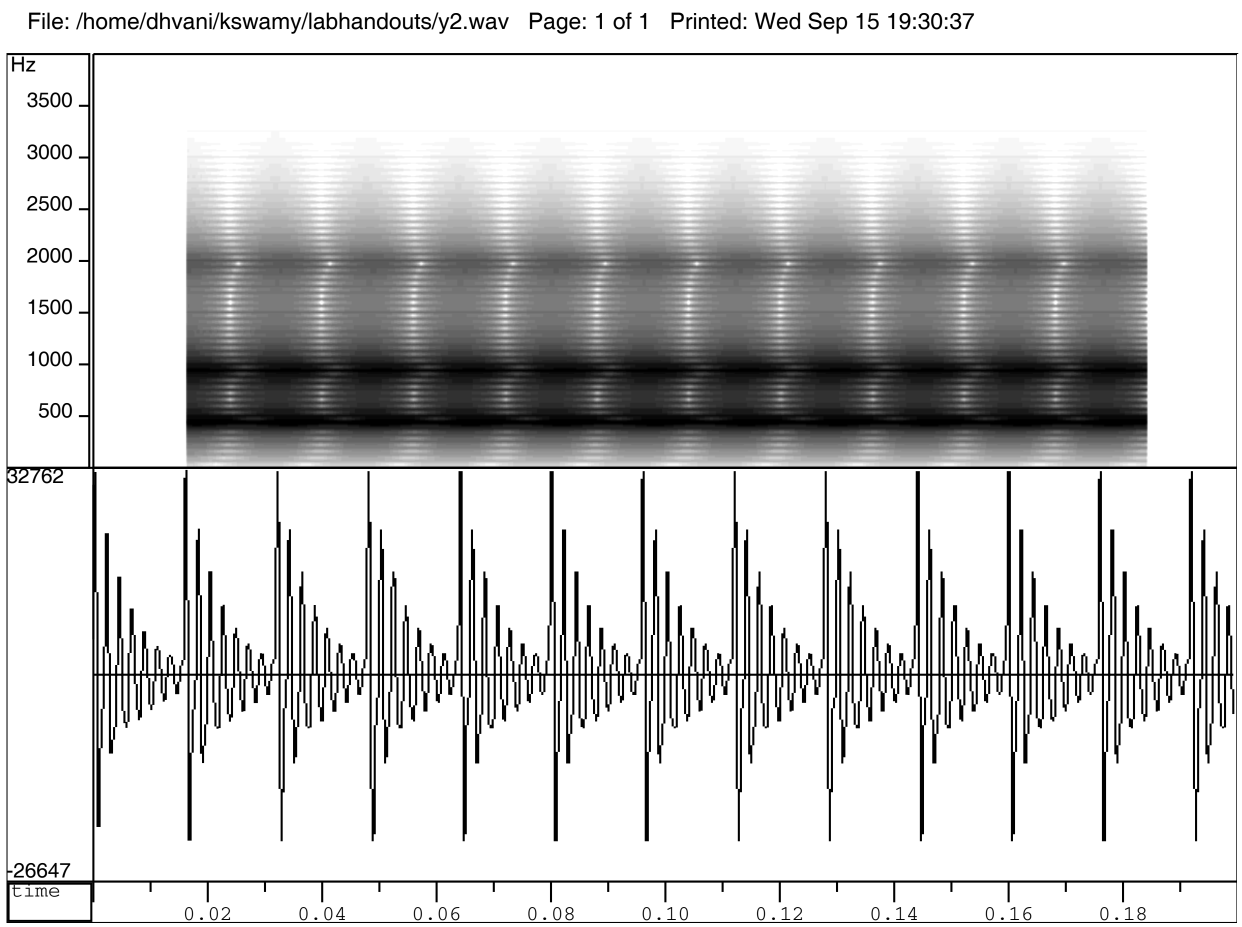

Figure 5: Waveform of the the synthesized vowel /a/ and its spectrogram for excitation at twice the pitch period.