Lab 2
Time domain methods for speech processing

1. Objectives

i. To become familiar with the time domain methods for speech processing.

ii. To calculate

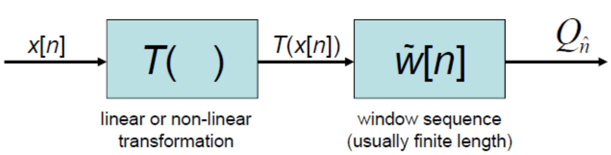
  i. Short-time energy
  ii. Average magnitude
  iii. Short-time average zero crossing rate
  iv. Short-time auto correlation function

iii. To check the effect of windowing in the calculation of different functions.

2. Generic Short-Time Processing

Generic short-time processing is represented as:

$$\boxed{Q_{\hat{n}} = \left. \left(\sum_{m=-\infty}^{\infty} T(x[m])\, \tilde{w}[n-m] \right) \right|_{n=\hat{n}}}$$



3. Short-Time Speech Measurements for different windows

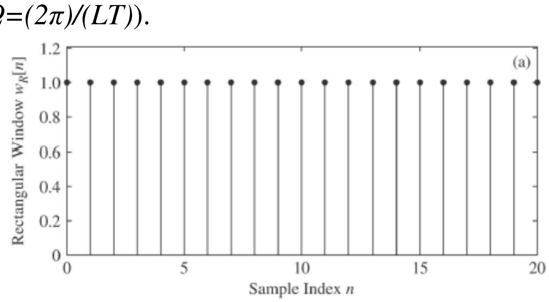
The choice of the window in short-time speech processing determines the nature of the measurement representation. A long window would result in very little changes of the measurement in time whereas the measurement with a short window would not be sufficiently smooth. Two representative windows are demonstrated, Rectangular and Hamming. The latter has almost twice the bandwidth of the former, for the same length. Furthermore, the attenuation for the Hamming window outside the passband is much greater.

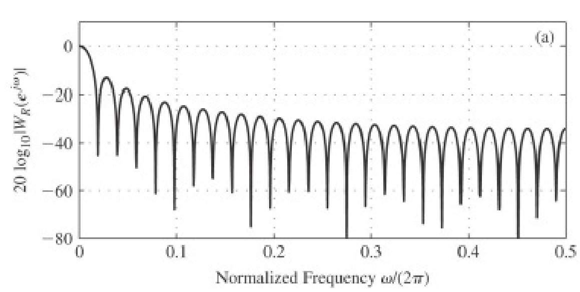
Rectangular window:

$h[n]=1$, $0 \le n \le L-1$ and 0 otherwise

$$H(e^{j\Omega T}) = \frac{\sin(\Omega L T/2)}{\sin(\Omega T/2)} e^{-j\Omega T(L-1)/2}$$

first zero location i.e. nominal cutoff frequency of the equivalent "lowpass" filter at $f=Fs/L=1/(LT)$ (or $\Omega=(2\pi)/(LT)$).

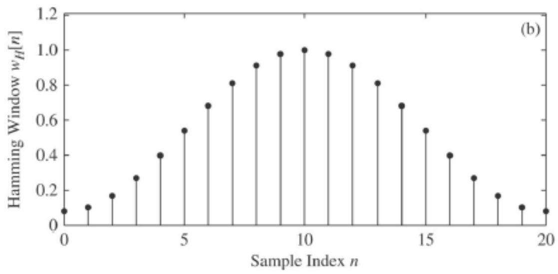


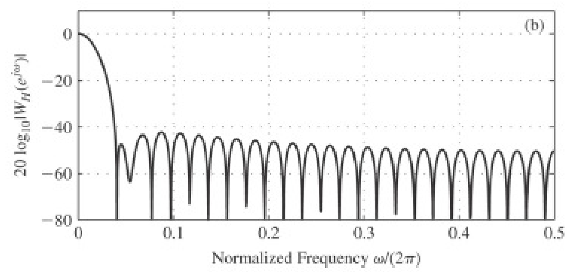


Hamming window (raised cosine window):

h[n]=0.54-0.46 cos(2πn/(L-1)), 0≤n≤L-1 and 0 otherwise

$$\tilde{w}_H[n] = 0.54\tilde{w}_R[n] - 0.46 * \cos(2\pi n / (L-1))\tilde{w}_R[n]$$





Sample code:

```
% Sampling frequency in Hz (for the data used in this exercise)
Fs = 16000;
```

Check the input audio file and properly choose sampling frequency.

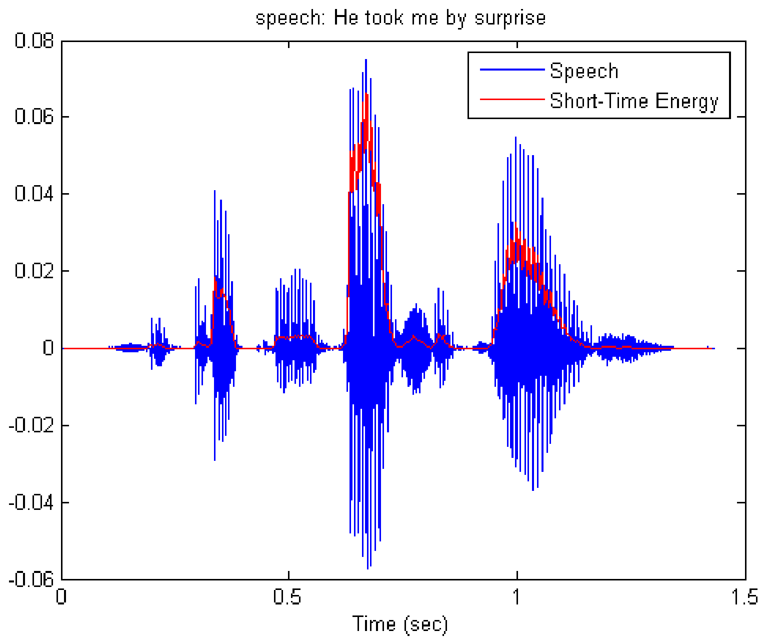
```
% Rectangular and Hamming window, chosen from the variety of windows
% offered in MATLAB.
wRect = rectwin(51);
wHamm = hamming(51);

% Calculate the magnitude of the Fast Fourier Transform of the windows
fftLength = 1024;
magFWHamm = abs(fft(wHamm, fftLength));
magFWRect = abs(fft(wRect, fftLength));

subplot(2,1,1);
plot(linspace(0,0.5,ceil(fftLength/2)),
20*log10(magFWRect(1:ceil(fftLength/2))));
ylabel('dB');
legend('Rectangular Window');
subplot(2,1,2);
plot(linspace(0,0.5,ceil(fftLength/2)),
20*log10(magFWHamm(1:ceil(fftLength/2))));
ylabel('dB');
xlabel('Normalized Frequency');
legend('Hamming Window');
```

4. Short-Time energy calculation

Short-Time energy is a simple short-time speech measurement. It is defined as:

$$E_{\hat{n}} = \sum_{m=-\infty}^{\infty} \left[x[m]\tilde{w}[\hat{n}-m] \right]^2$$

This measurement can in a way distinguish between voiced and unvoiced speech segments, since unvoiced speech has significantly smaller short- time energy. For the length of the window a practical choice is 10-20 msec that is 160-320 samples for sampling frequency 16 kHz. This way the window will include a suitable number of pitch periods so that the result will be neither too smooth, nor too detailed.



In this lab, a sample sentence is chosen and results are shown using that sample sentence. The students must generate a sample sentence from Praat or similar audio software as in Lab 1 with distinct voiced and unvoiced sound and use it for rest of the calculation steps.

Sample code:

```
% A sentence stored in SPHERE format is read. The sentence is :
% 'He took me by surprise'. A function of voicebox is used.
speechSignal = readsph(fullfile(wavpath,'SI1899a.WAV'));

% A hamming window is chosen
winLen = 301;
winOverlap = 300;
wHamm = hamming(winLen);

% Framing and windowing the signal without for loops.
sigFramed = buffer(speechSignal, winLen, winOverlap, 'nodelay');
sigWindowed = diag(sparse(wHamm)) * sigFramed;

% Short-Time Energy calculation
energyST = sum(sigWindowed.^2,1);

% Time in seconds, for the graphs
t = [0:length(speechSignal)-1]/Fs;

subplot(1,1,1);
plot(t, speechSignal);
title('speech: He took me by surprise');
xlims = get(gca,'Xlim');
```

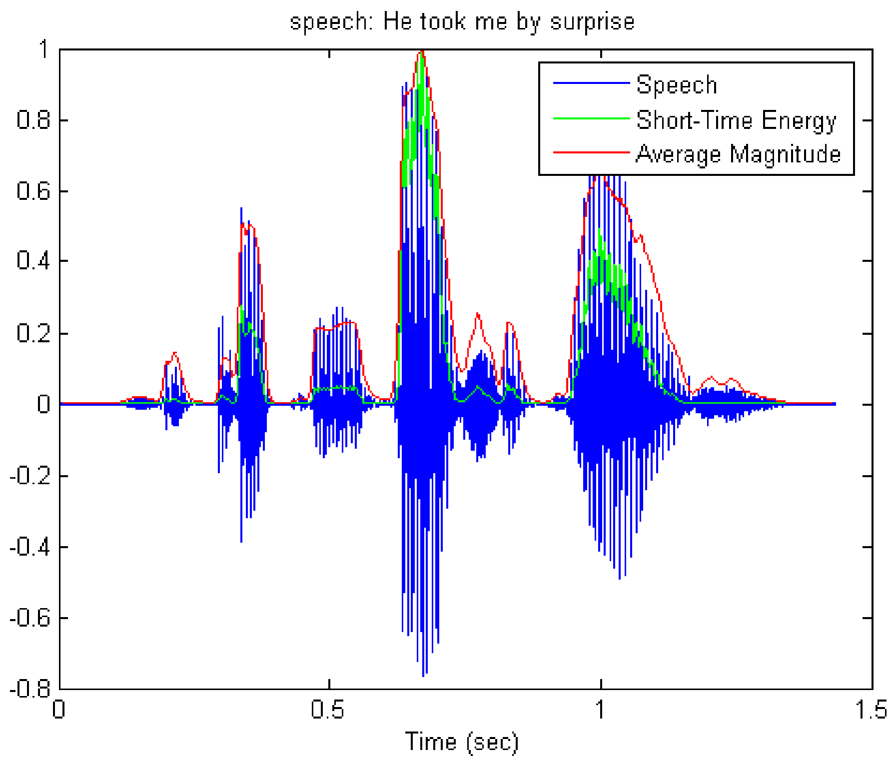
```
hold on;

% Short-Time energy is delayed due to lowpass filtering. This delay is
% compensated for the graph.
delay = (winLen - 1)/2;
plot(t(delay+1:end - delay), energyST, 'r');
xlim(xlims);
xlabel('Time (sec)');
legend({'Speech','Short-Time Energy'});
hold off;
```

5. Short-Time average magnitude calculation

Average magnitude function defined as

$$M_{\hat{n}} = \sum_{m=-\infty}^{\infty} |x[m]|\, \tilde{w}[\hat{n}-m]$$

does not emphasize large signal levels so much as short-time energy since its calculation does not include squaring. The signals in the graphs are normalized.

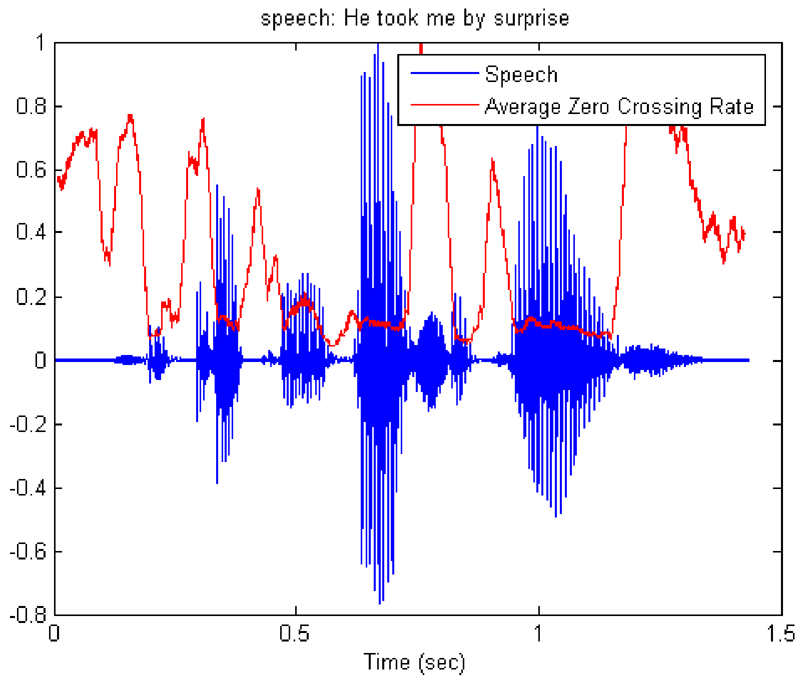


6. Short-Time average zero crossing rate

Average Zero-Crossing Rate is defined as

$$Z_n = \sum_{m=-\inf}^{m=\inf} |sgn[x(m)] - sgn[x(m-1)]| w(n-m)$$

w(n) = 1/2N, 0 <= n <= N-1. This measure could allow the discrimination between voiced and unvoiced regions of speech, or between speech and silence. Unvoiced speech has in general, higher zero-crossing rate. The signals in the graphs are normalized.



Hint:

- Use **sign** function in MATLAB to check sign and consider no sign change at the beginning of sample.

7. Short-Time autocorrelation calculation

Short-time autocorrelation is defined as

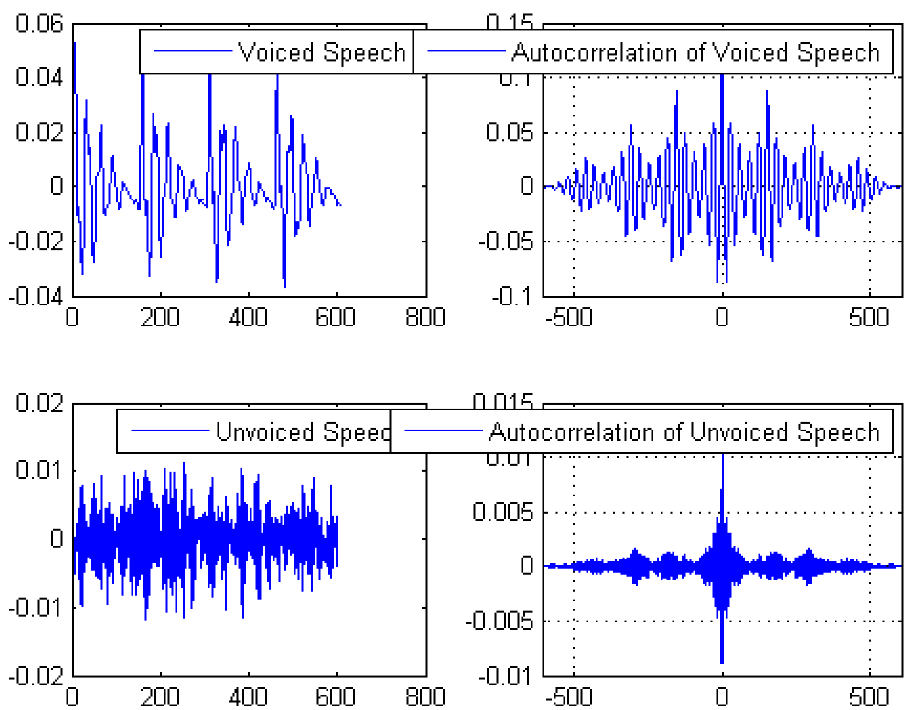
$$R_n(k) = \sum_{m=0}^{m=N-1-k} [x(n+m)w'(m)][x(n+m+k)w'(k+m)]$$

and is actually the autocorrelation of a windowed speech segment. The property of the short-time autocorrelation to reveal periodicity in a signal is demonstrated. Notice how the autocorrelation of the voiced speech segment retains the periodicity. On the other hand, the autocorrelation of the unvoiced speech segment looks more like noise. In general, autocorrelation is considered as a robust indicator of periodicity.

Sample code:

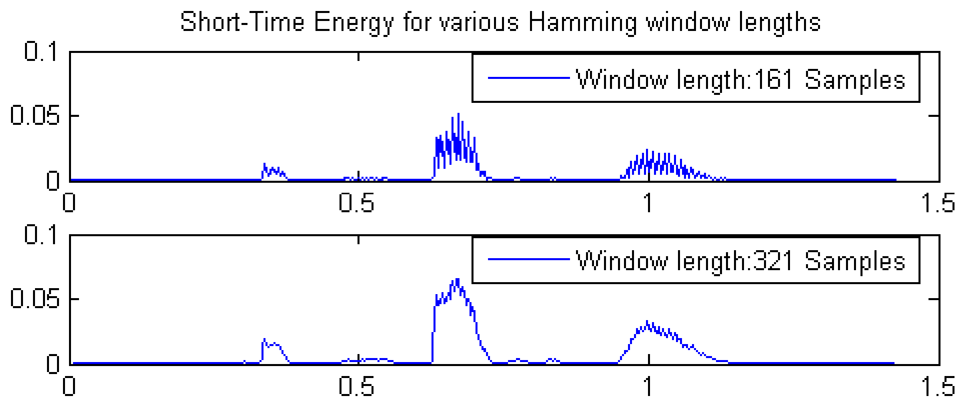
```
% Voiced speech segment.
ss1 = speechSignal(16095:16700);
```

```
% An unvoiced speech segment.
ss2 = speechSignal(12200:12800);

% Calculation of the short time autocorrelation
[ac1, lags1] = xcorr(ss1);
[ac2, lags2] = xcorr(ss2);

subplot(2,2,1);
plot(ss1);
legend('Voiced Speech')
subplot(2,2,2);
plot(lags1, ac1);
xlim([lags1(1) lags1(end)]);
legend('Autocorrelation of Voiced Speech')
grid on;
subplot(2,2,3);
plot(ss2);
legend('Unvoiced Speech')
subplot(2,2,4);
plot(lags2, ac2);
xlim([lags1(1) lags1(end)]);
legend('Autocorrelation of Unvoiced Speech')
grid on;
```



8. Short-Time energy calculation with Hamming window of variable length

Effects of the choice of window length are demonstrated. As the length increases, short-time energy becomes smoother, as expected. It should be noticed that the measurement is not taken for every sample. Due to the lowpass character of the window, short-time energy is actually band limited to the bandwidth of the window, which is much smaller than 16 KHz. Actually for the lengths we are interested in; it is less than 160 Hz. So, we could calculate the energy every 50 samples that is windows shifting of 50 samples can be employed.



9. **Perform the following calculation and plot the respective results. Discuss on the each results and highlight the major observation.**
   i. Short-Time energy calculation with Hamming window of variable length (161,321,401,501)
   ii. Short-Time energy calculation with Rectangular window of all these lengths and Compare with the Hamming window results.
   iii. Calculate and plot average magnitude, average zero crossing rate and auto correlation for the varying window.

NOTE: The lab report can be prepared submitted as softcopy and can be uploaded to Speech processing dropbox folder or sent via mail.

INLAB REPORT:

Hand in the following:

- Specify signal bandwidth and sampling frequency and window size for that sampling frequency.
- The plot of all the four functions for a speech signal by clearly showing the location of voiced and unvoiced sounds with both hamming and rectangular window.
- Show all the plots with different window size and describe clearly the effect of windowing.
- Estimate the pitch period at voiced locations.

It is highly unlikely that two or more students have same speech signal input and results. There will penalty if such case is discovered.