EE-414 Speech Processing Lab

Lab-6 - 180020002

Aim:

● To understand the need for short term processing of speech.

● To compute short term energy and study its significance.

● To compute short term zero crossing rate and study its significance.

● To compute short term autocorrelation and study its significance.

Theory:

Need for Short Term Processing of Speech

Speech is produced from a time varying vocal tract system with time varying excitation. As a result the speech signal is non-stationary in nature. Most of the signal processing tools studied in signals and systems and signal processing assume time invariant system and time invariant excitation, i.e. stationary signal. Hence these tools are not directly applicable for speech processing.

Speech signal may be stationary when it is viewed in blocks of 10-30 msec. Hence to process speech by different signal processing tools, it is viewed in terms of 10-30 msec. Such a processing is termed as Short Term Processing (STP).

Short Term Processing of speech can be performed either in time domain or in frequency domain. The particular domain of processing depends on the information from the speech that we are interested in. For instance, parameters like short term energy, short term zero crossing rate and short term autocorrelation can be computed from the time domain processing of speech. Alternatively, short term Fourier transform can be computed from the frequency domain processing of speech.

Short Term Energy Parameter

The energy associated with speech is time varying in nature. Hence the interest for any automatic processing of speech is to know how the energy is varying with time and to be more specific, energy associated with short term region of speech. By the nature of production, the speech signal consist of voiced, unvoiced and silence regions.

The relation for finding the short term energy can be derived from the total energy relation defined in signal processing.The total energy of an energy signal is given by:

$$E_T = \sum_{n=-\infty}^{\infty} s^2(\text{n})$$

In case of short term energy computation we consider speech in terms of 10-30 msec . Let the samples in a frame of speech are given by ***"n=0 to n=N-1"***, where ***" N "*** is the length of frame (samples), then for energy computation the speech will be zero outside the frame length. Then for energy computation amplitude of the speech samples will be zero outside the frame. Accordingly we can write above mentioned relation as:

$$E_T = \cancel{\sum_{n=-\infty}^{-1} s^2(\text{n})} + \sum_{n=0}^{N-1} s^2(\text{n}) + \cancel{\sum_{n=N}^{\infty} s^2(\text{n})}$$

$$E_T = \sum_{n=0}^{N-1} s^2(\text{n})$$

This relation will give total energy present in the frame of speech from ***" n=0 to n=N-1 "***. To represent more specifically, only one frame of speech we use the relation

$$s_w(\text{n}) = \text{s(m).w(n-m)}$$

where "***w(n)***" represent the windowing function of finite duration. There are several windowing functions present in the signal processing literature. The mostly used ones include rectangular, hanning and hamming. For all time domain parameters estimation we use the rectangular window for its simplicity.

Now we can write the relation of short term energy as follows

$$\boldsymbol{e(n)} = \sum_{\text{m}=-\infty}^{\infty} \boldsymbol{(s(m).w(n-m))}^{2}$$

where ***"n"*** is the shift / rate in number of samples at which we are interested in knowing the short term energy.

Short Term Zero Crossing Rate (ZCR)

Zero Crossing Rate gives information about the number of zero-crossings present in a given signal. Intuitively, if the number of zero crossings are more in a given signal, then the signal is changing rapidly and accordingly the signal may contain high frequency information. On the similar lines, if the number of zero crossing are less, hence the signal is changing slowly and accordingly the signal may contain low frequency information. Thus ZCR gives an indirect information about the frequency content of the signal.

The ZCR in case of stationary signal is defined as,

$$z = \sum_{n=-\infty}^{\infty} |sgn(s\ (n)) - sgn(s\ (n-1))|$$

$$\begin{aligned} where\ sgn(s\ (n)\ ) &= 1\ if\ s\ (n) \geq 0 \\ &= -1\ if\ s(\ n) < 0 \end{aligned}$$

This relation can be modified for non-stationary signals like speech and termed as short term ZCR. It is defined as

$$z(n) = \frac{1}{2N}\sum_{m=0}^{N-1} s(m).\, w(n-m)$$

The factor "2" comes in the denominator to take care of the fact that there will be two zero crossings per cycle of one signal.

In case of speech the nature of signal changes with time over few msec. For instance, from initial voiced to unvoiced and back to voiced and so on. To have some useful information, ZCR needs to be computed using typical frame size of 10-30 msec with half the frame size as shift.

Short Term Autocorrelation:

Crosscorrelation tool from signal processing can be used for finding the similarity among the two sequences and refers to the case of having two different sequences for correlation. Autocorrelation refers to the case of having only one sequence for correlation. In autocorrelation, the interest is in observing how similar the signal characteristics with respect to time. This is achived by providing different time lag for the sequence and computing with the given sequence as reference.

The autocorrelation is a very useful tool in case of speech processing. However due to the non-stationary nature of speech, a short term version of the autocorrelation is needed. The autocorrelation of a stationary sequence rxx(k) is given by:

$$r_{xx}(k) = \sum_{m=-\infty}^{\infty} x(m).x(m+k)$$

The corresponding short term autocorrelation of a non-stationary sequence s(n) is defined as:

$$r_{ss} = \sum_{m=-\infty}^{\infty} s_w(m).s_w(k+m)$$

$$r_{ss}(n,k) = \sum_{m=-\infty}^{\infty} (s(m)w(n-m).s(k+m).w(n-k+m)))$$

where sw(n)=s(m).w(n-m) is the windowed version of s(n). Thus for a given windowed segment of speech,the short term autocorrelation is a sequence. The nature of short term autocorrelation sequence is primarily different for voiced and unvoiced segments of speech. Hence information from the autocorrelation sequence can be used for discriminating voiced and unvoiced segments.

The typical frame size for computing short term autocorrelation should include at least two cycles of speech signal in the voiced speech case. To ensure this the size is used in the range 30-50 msec. The nature of autocorrelation sequence in case of autocorrelation of voiced speech can be explained for finding the periodicity of voiced speech. Accordingly, the autocorrelation of voiced speech should give strong peak at the periodic value and no such peak in case of unvoiced speech. Therefore, the autocorrelation of speech has become a standard approach for enhancing pitch .

Procedure:

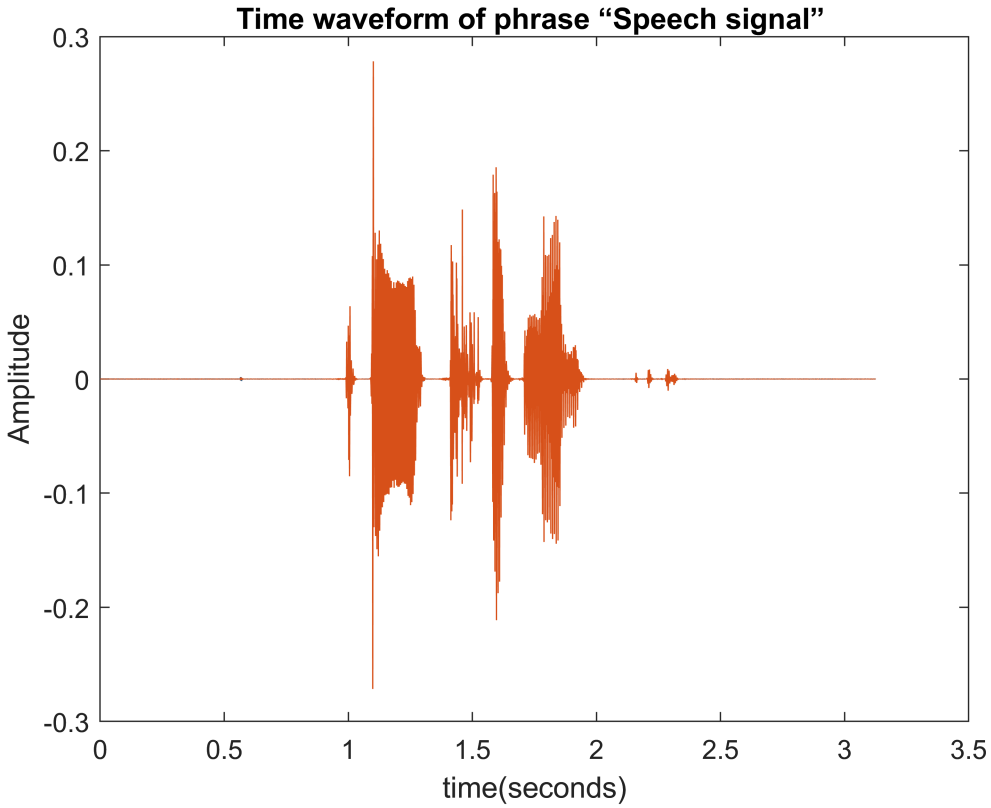
Part A:

A. Short term energy(STE) :

a. Compute and plot STE (as a function of frame index) using frame size as 20ms and frameshift as 10ms.

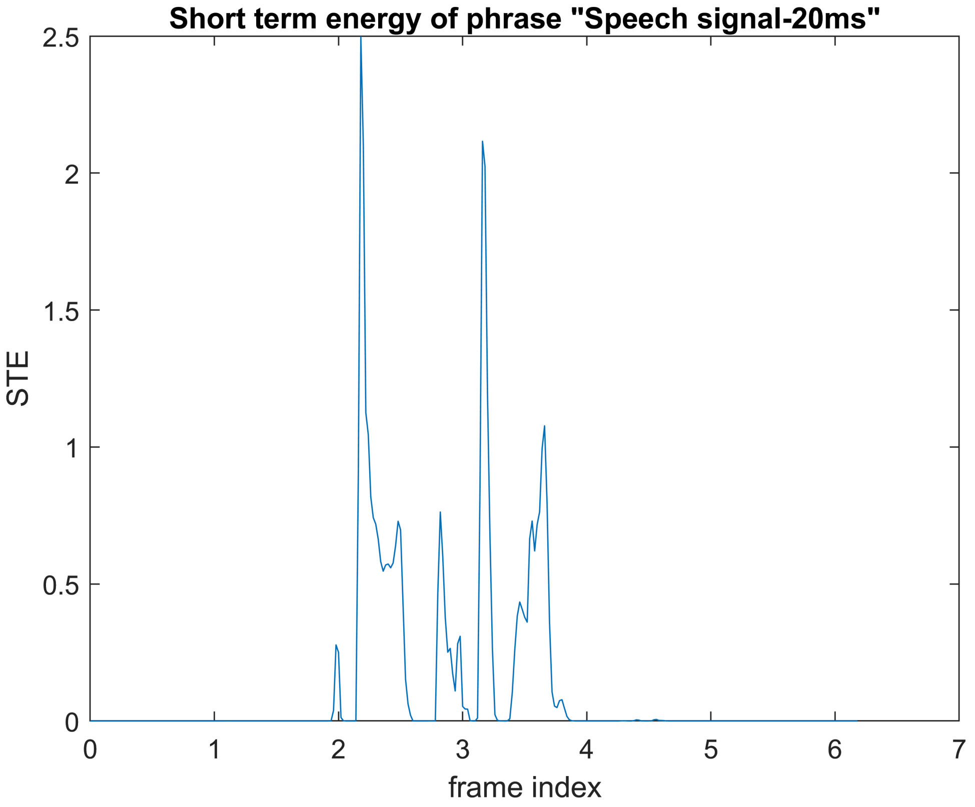
b. Demonstrate and explain the effect of the window size on STE by taking window size of 20ms, 30ms, 50ms, 100ms. Also comment on which frame size is preferred.

```
[y,fs] = audioread('l6_speech_signal.wav');          % Reads an audio file from working direct
% For window size 20ms
Frame_size=20; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
t = (1:length(y))/fs;
plot(t,y);
xlabel('time(seconds)');
ylabel('Amplitude');
title('Time waveform of phrase “Speech signal” ')
```

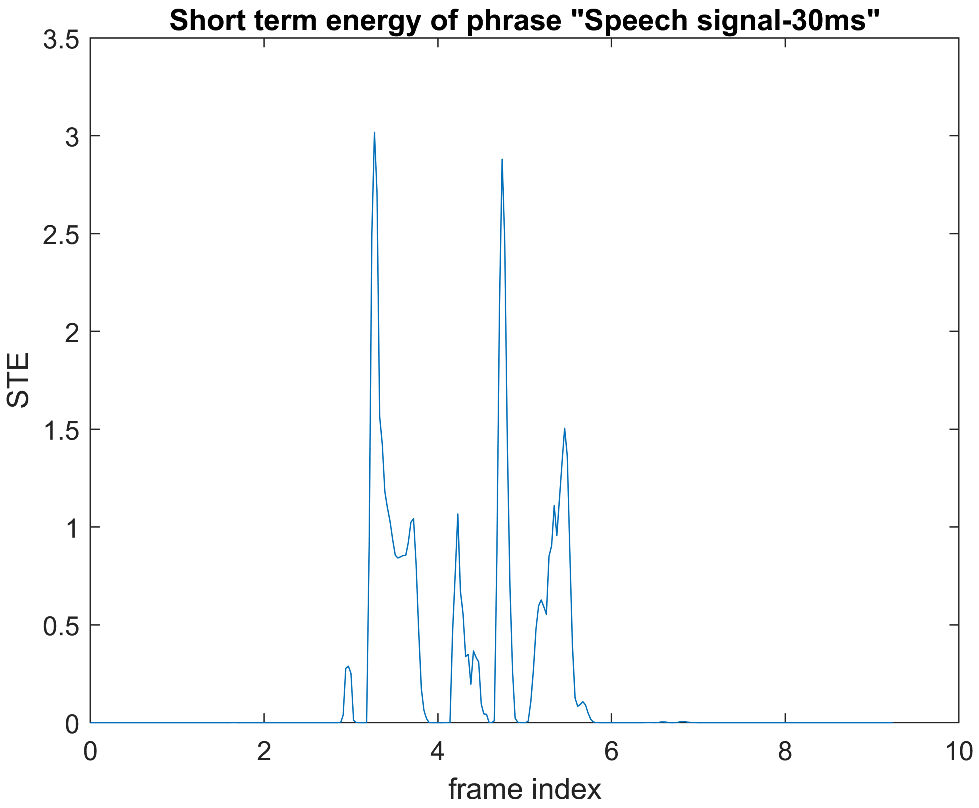


```
energy=short_term_energy(y,fs,Frame_size,Frame_shift);
t_ste=1/fs:(Frame_size/1000):(length(energy)*(Frame_size/1000));
plot(t_ste,energy);
```

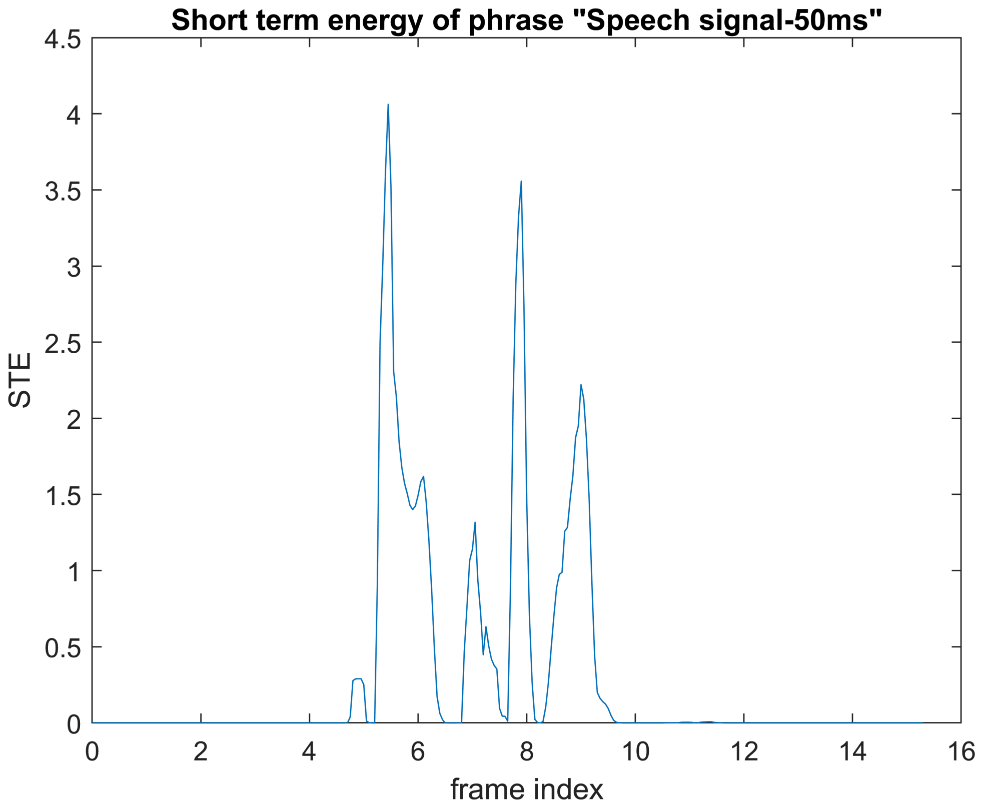
```
xlabel('frame index');
ylabel('STE');
title('Short term energy of phrase "Speech signal-20ms"')
```



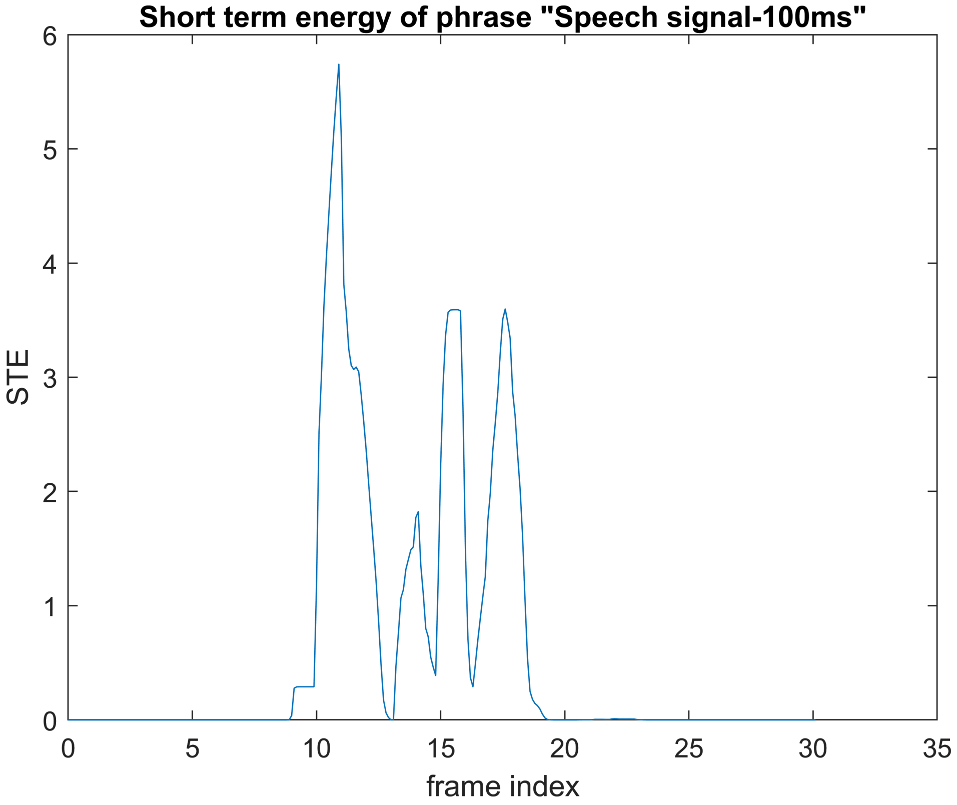
```
% For window size 30ms
Frame_size=30; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
energy=short_term_energy(y,fs,Frame_size,Frame_shift);
t_ste=1/fs:(Frame_size/1000):(length(energy)*(Frame_size/1000));
plot(t_ste,energy);
xlabel('frame index');
ylabel('STE');
title('Short term energy of phrase "Speech signal-30ms"')
```



```
% For window size 50ms
Frame_size=50; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
energy=short_term_energy(y,fs,Frame_size,Frame_shift);
t_ste=1/fs:(Frame_size/1000):(length(energy)*(Frame_size/1000));
plot(t_ste,energy);
xlabel('frame index');
ylabel('STE');
title('Short term energy of phrase "Speech signal-50ms"')
```



```
% For window size 100ms
Frame_size=100; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
energy=short_term_energy(y,fs,Frame_size,Frame_shift);
t_ste=1/fs:(Frame_size/1000):(length(energy)*(Frame_size/1000));
plot(t_ste,energy);
xlabel('frame index');
ylabel('STE');
title('Short term energy of phrase "Speech signal-100ms"')
```



Observations:

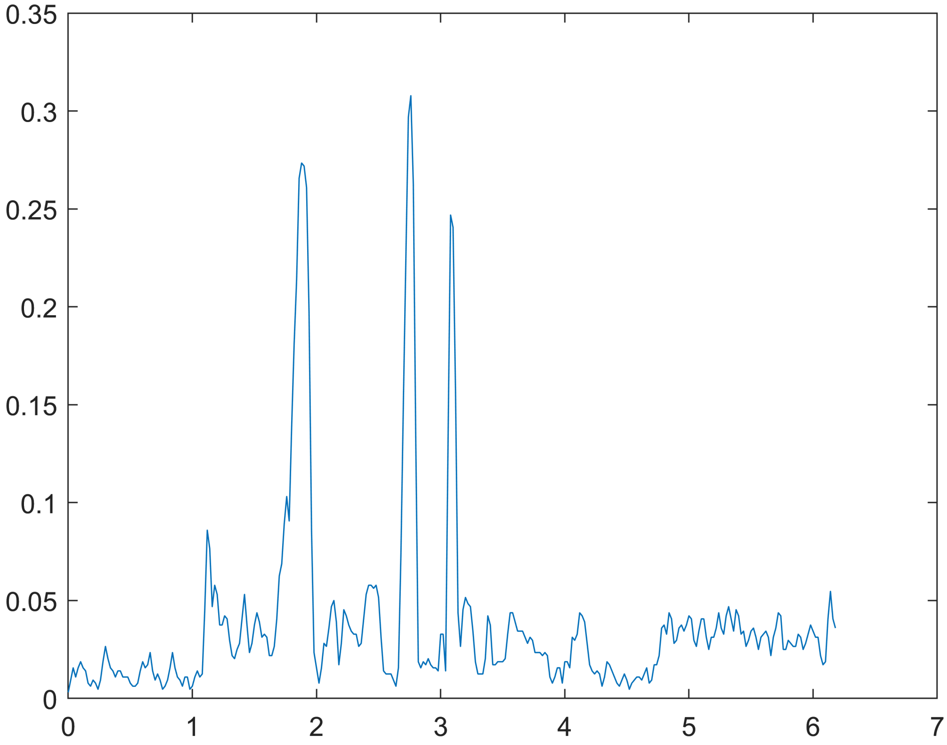
The frame size of 20 ms should be preferred as it is not too long so as to violate the quasi stationarity assumption. For larger frame sizes we get much smoothed version of energy and not finding time varying nature of short term energy. So distinction between voiced and unvoiced speech becomes difficult.

B. Short term Zero Crossing Rate(ST-ZCR) :

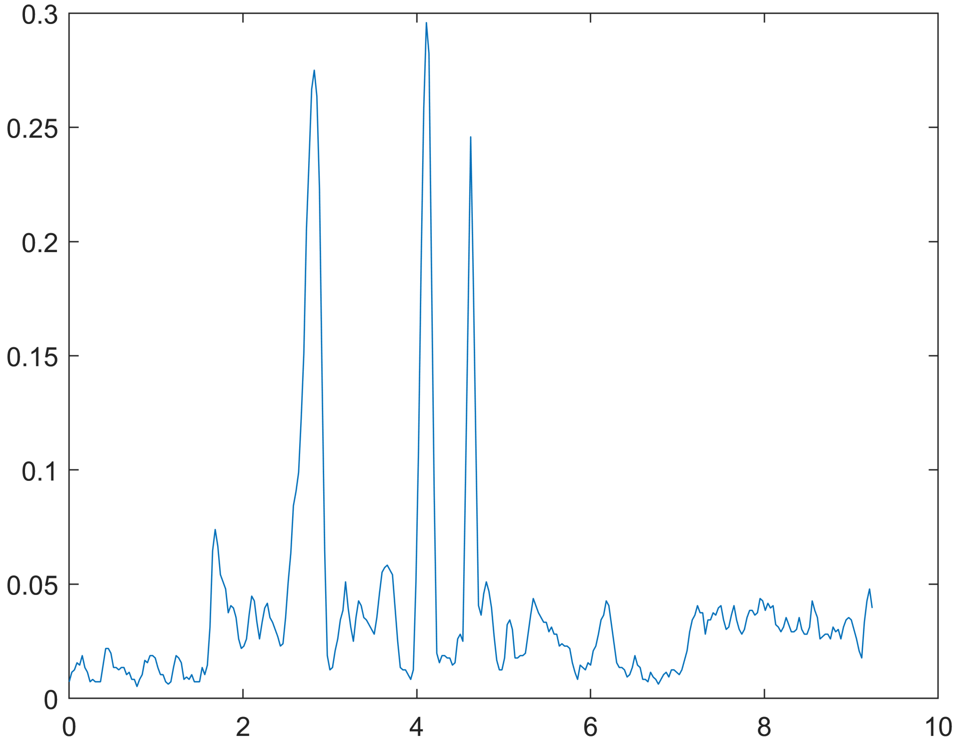
a. Compute and plot ST-ZCR for speech signal using frame size as 20ms and frameshift as 10ms.

b. Demonstrate and explain the effect of the window on ST-ZCR by taking window size of 20ms, 30ms, 50ms, 100ms. Also comment on which frame size is preferred.

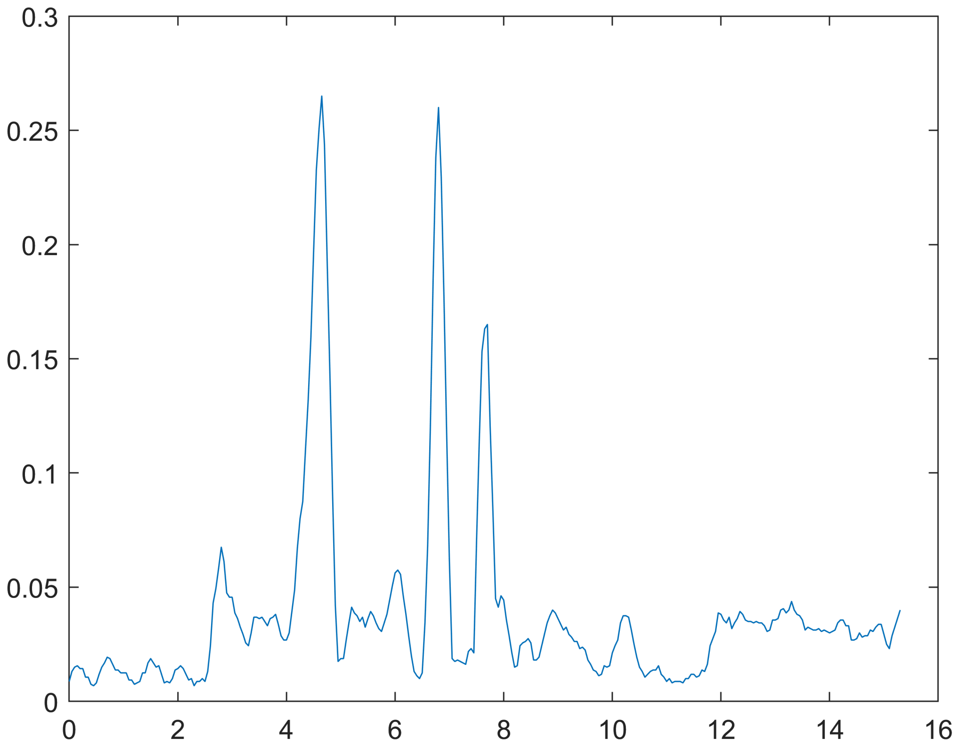
```
% For window size 20ms
Frame_size=20; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
zcr=zero_crossing_rate(y,fs,Frame_size,Frame_shift);
t_zcr=1/fs:(Frame_size/1000):(length(zcr)*(Frame_size/1000));
plot(t_zcr,zcr);
```



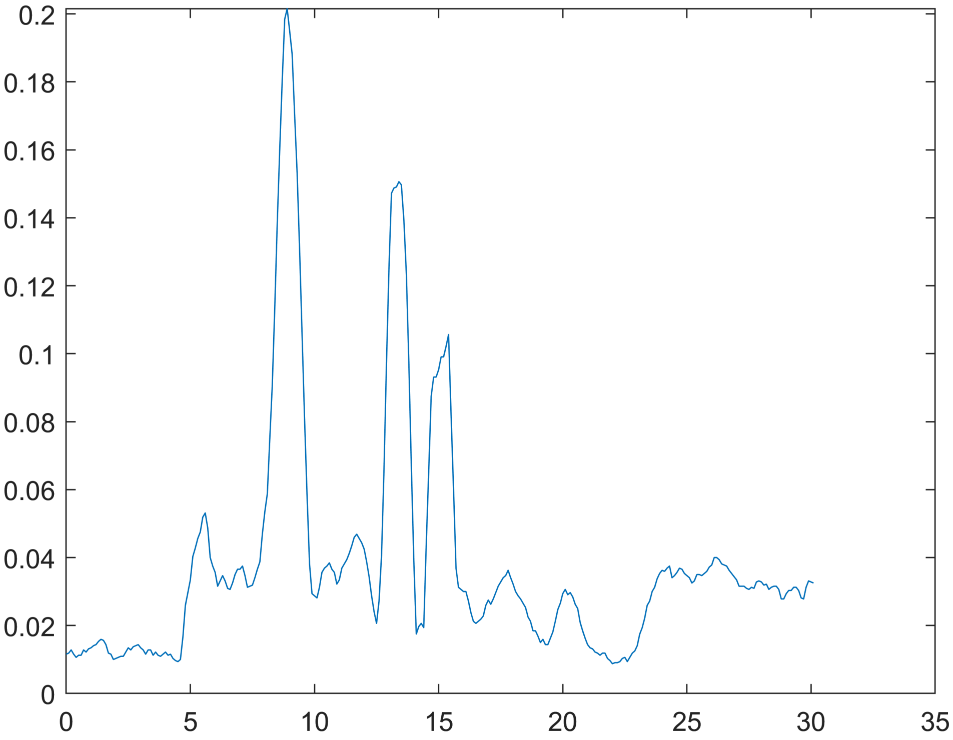
```
% For window size 30ms
Frame_size=30; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
zcr=zero_crossing_rate(y,fs,Frame_size,Frame_shift);
t_zcr=1/fs:(Frame_size/1000):(length(zcr)*(Frame_size/1000));
plot(t_zcr,zcr);
```



```
% For window size 50ms
Frame_size=50; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
zcr=zero_crossing_rate(y,fs,Frame_size,Frame_shift);
t_zcr=1/fs:(Frame_size/1000):(length(zcr)*(Frame_size/1000));
plot(t_zcr,zcr);
```



```
% For window size 100ms
Frame_size=100; %Frame size in milliseconds
Frame_shift=10; %Frame shift in milliseconds
zcr=zero_crossing_rate(y,fs,Frame_size,Frame_shift);
t_zcr=1/fs:(Frame_size/1000):(length(zcr)*(Frame_size/1000));
plot(t_zcr,zcr);
```



Observations:

Frame size of 20 ms should be preferred to maintain the assumption of qualsi-stationarity. As the frame size increases, the smoothening and spreading of ZCR plot takes place. Therefore, we miss small transient region having high ZCR values.

C. Short term Autocorrelation

Do each of the following for one speech frame at the centre of the vowel, and another speech frame at the centre of the consonant “s”.

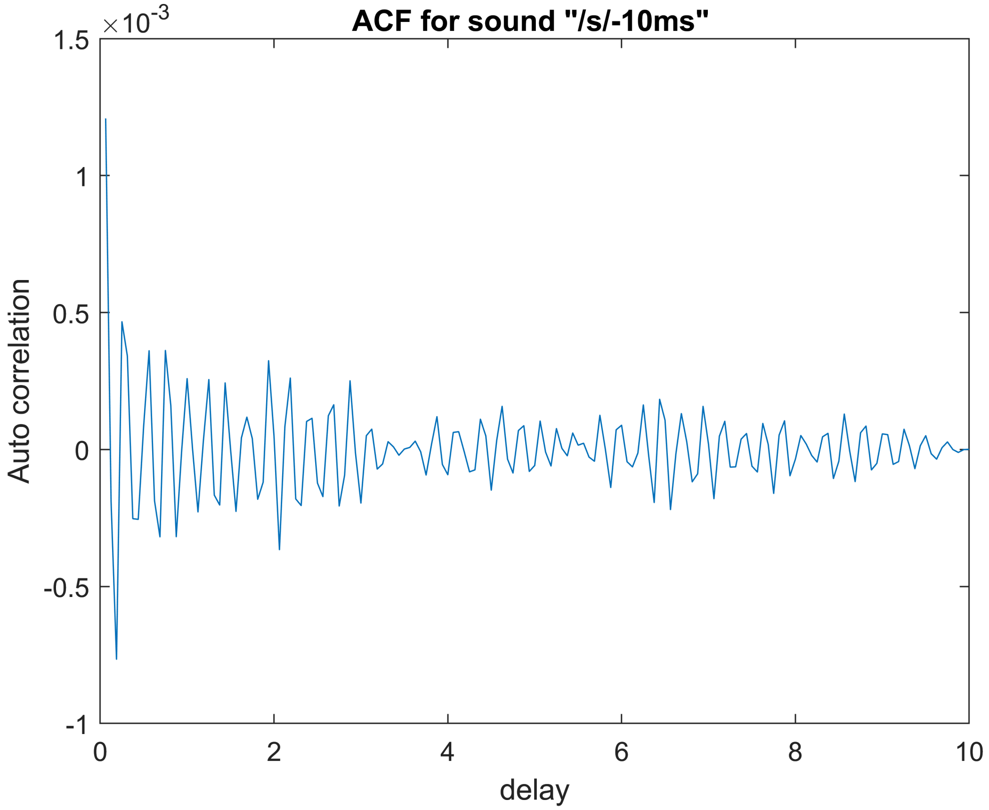
a. Compute and plot short term Autocorrelation function (as a function of delay index) for a 20ms long speech frame.

b. Demonstrate and explain the effect of the window on Short term Autocorrelation by taking window size of 10ms, 20ms, 50ms, 100ms. Also comment on which frame size is preferred.

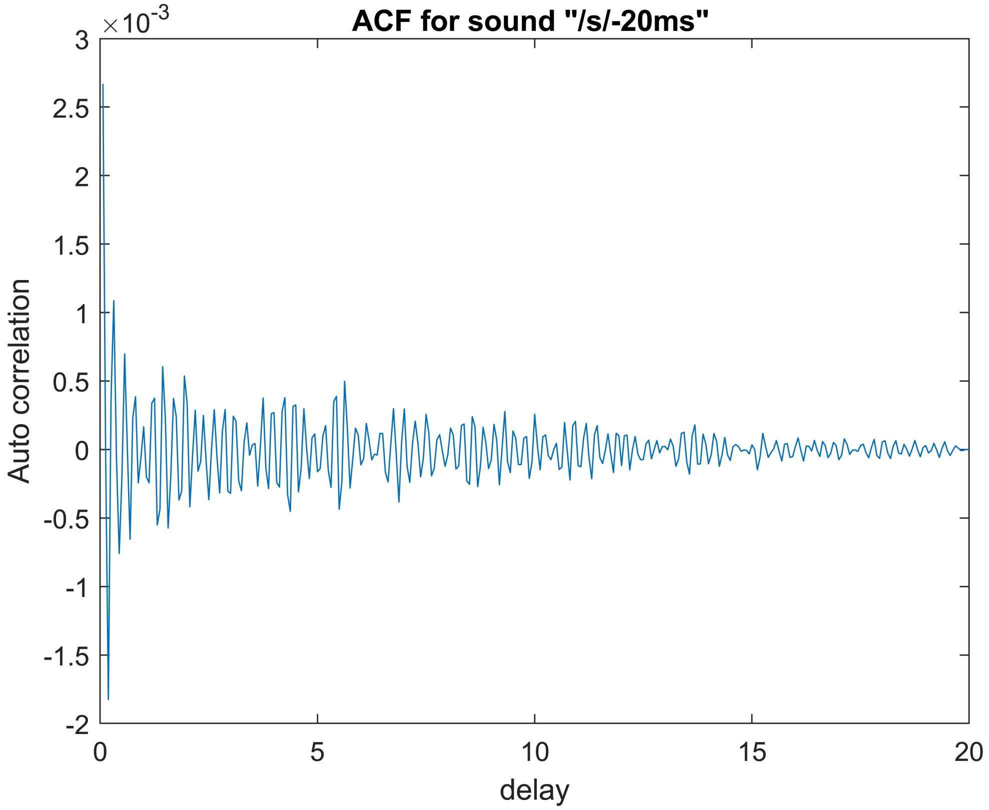
c. Demonstrate and explain the effect of the window shape on Short term Autocorrelation by taking the ‘rectangular’, ‘Hamming’ and ‘Hanning’ window. Take frame size as the most preferred frame size computed in (b). Also comment on which window is preferred.

```
%/s/
y_s = y(ceil(0.966*fs) : floor(1.067*fs));
%/e/
y_e = y(ceil(1.12*fs) : floor(1.23*fs));
```

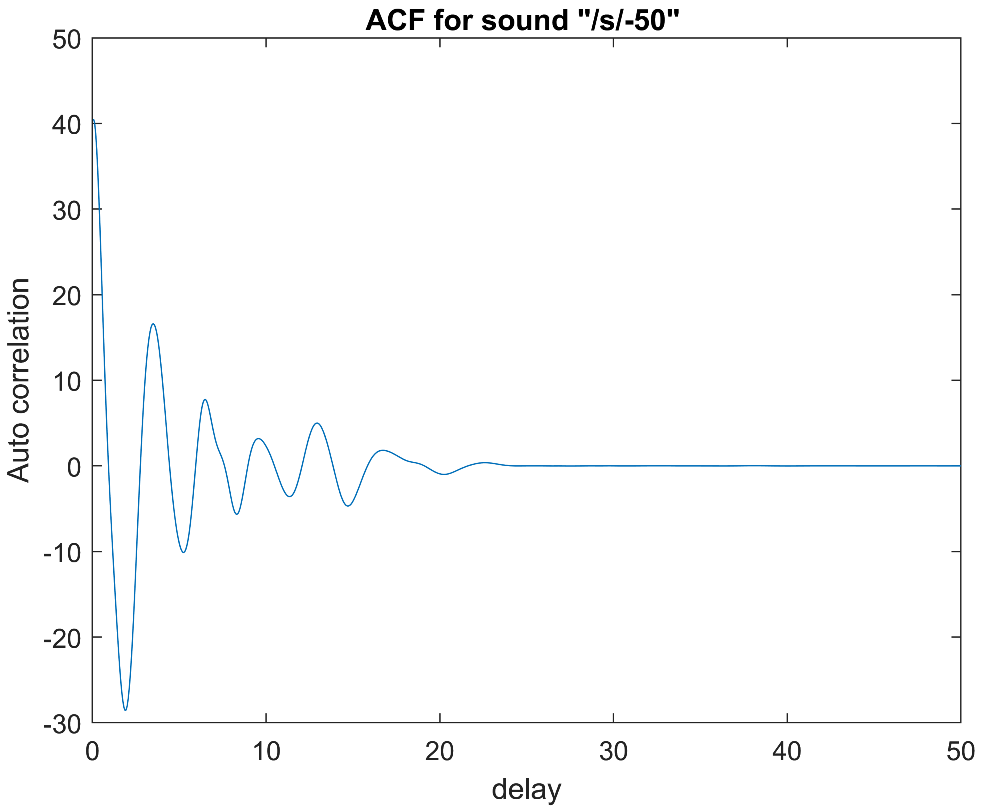
```
% For window size 10ms
Frame_size=10;
Frame_size= Frame_size/1000;
ys_a=autocorrelation(y_s,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_a)/fs))*1000;
plot(ts_a,ys_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-10ms" ');
```



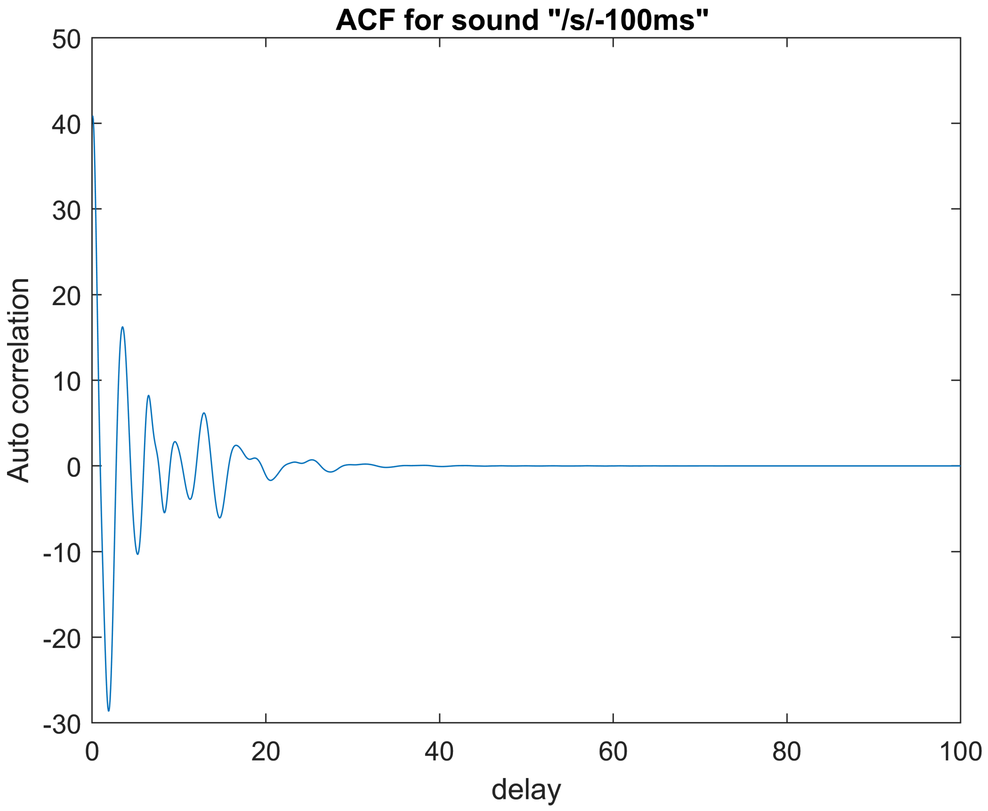
```
% For window size 20ms
Frame_size=20;
Frame_size= Frame_size/1000;
ys_a=autocorrelation(y_s,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_a)/fs))*1000;
plot(ts_a,ys_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-20ms" ');
```



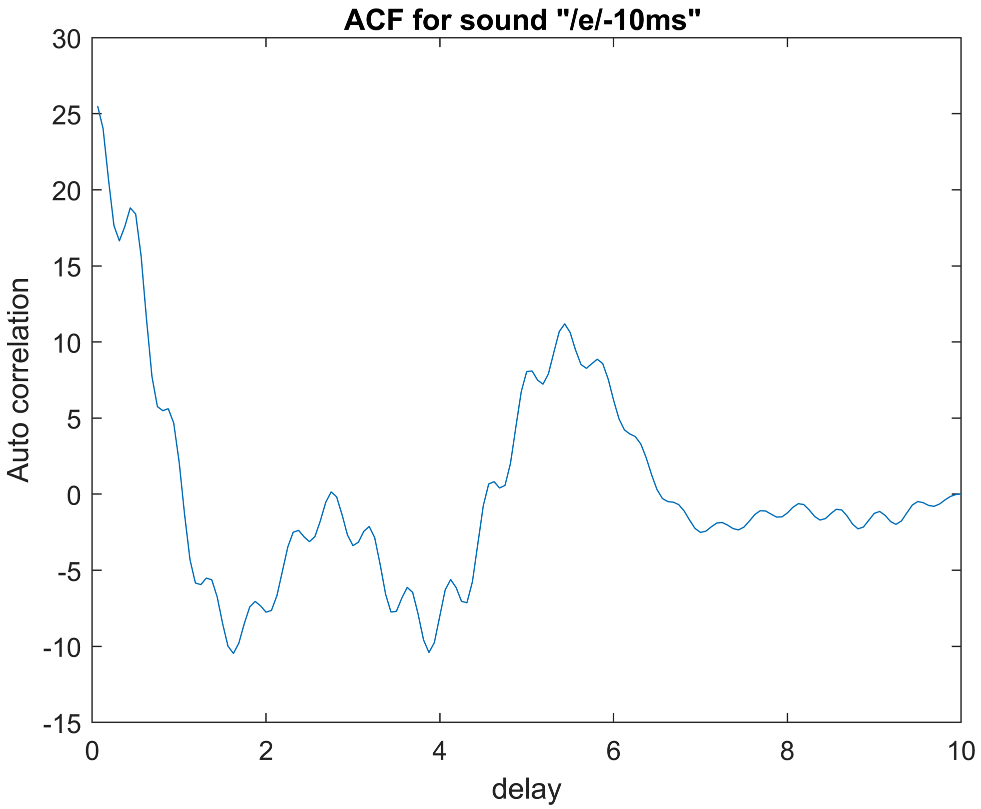
```
% For window size 50ms
Frame_size=50;
Frame_size= Frame_size/1000;
ys_a=autocorrelation(y_s,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_a)/fs))*1000;
plot(ts_a,ys_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-50" ');
```



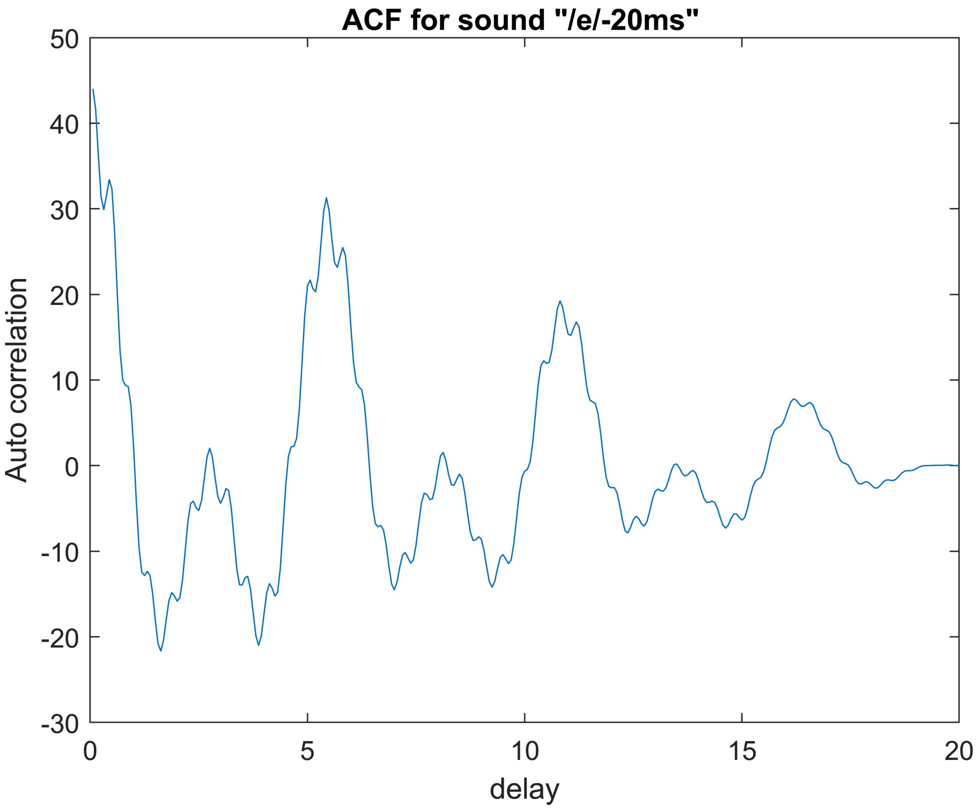
```
% For window size 100ms
Frame_size=100;
Frame_size= Frame_size/1000;
ys_a=autocorrelation(y_s,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_a)/fs))*1000;
plot(ts_a,ys_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-100ms" ');
```



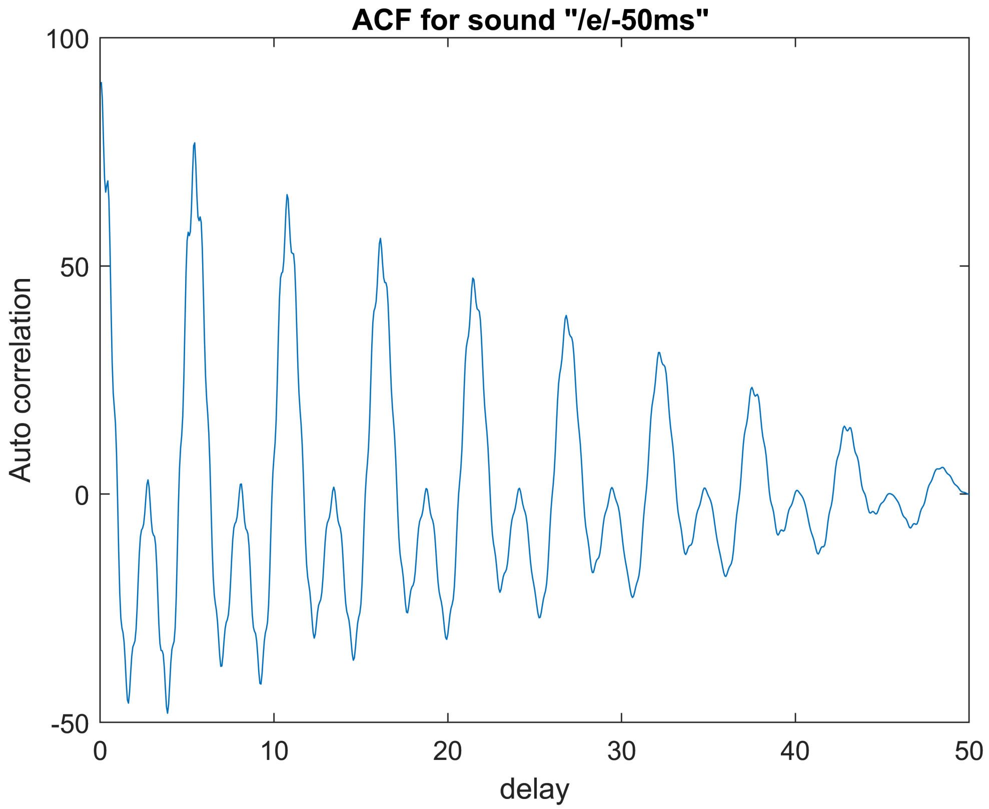
```
% For window size 10ms
Frame_size=10;
Frame_size= Frame_size/1000;
ye_a=autocorrelation(y_e,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_a)/fs))*1000;
plot(te_a,ye_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-10ms" ');
```



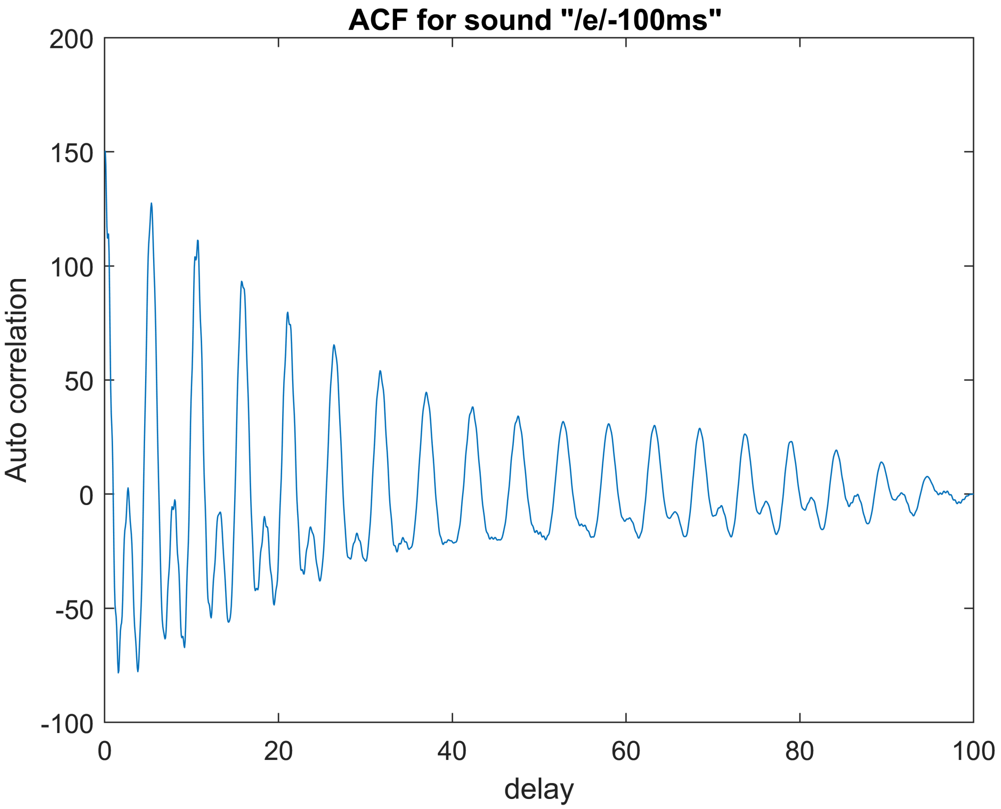
```
% For window size 20ms
Frame_size=20;
Frame_size= Frame_size/1000;
ye_a=autocorrelation(y_e,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_a)/fs))*1000;
plot(te_a,ye_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-20ms" ');
```



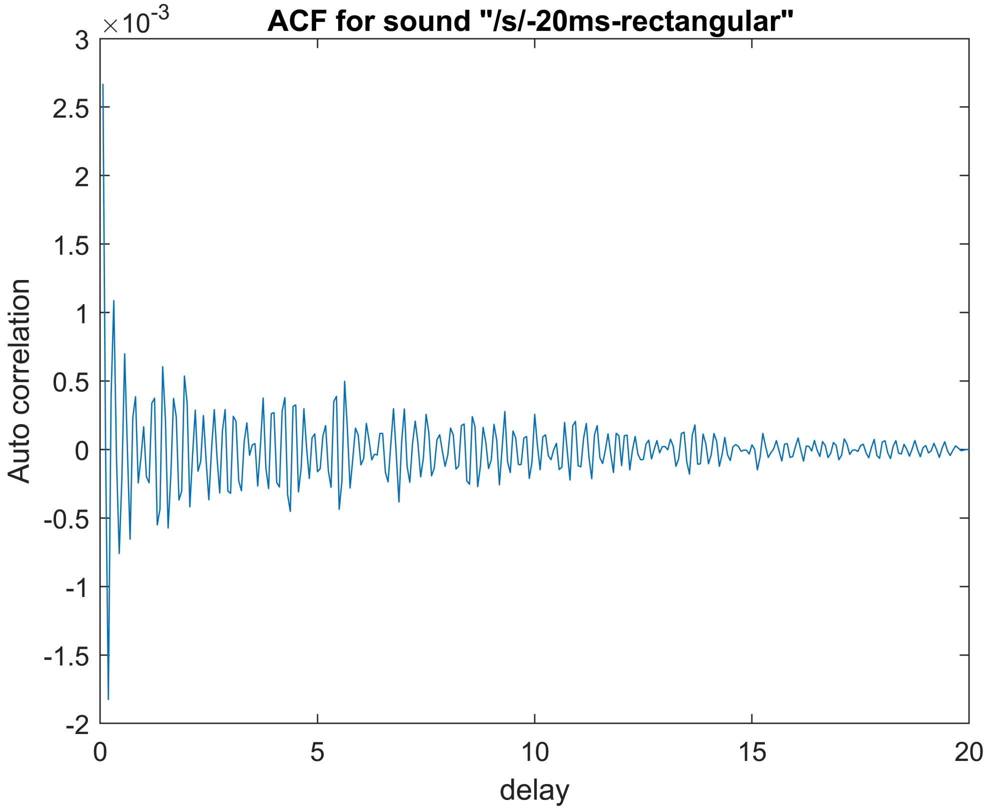
```
% For window size 50ms
Frame_size=50;
Frame_size= Frame_size/1000;
ye_a=autocorrelation(y_e,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_a)/fs))*1000;
plot(te_a,ye_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-50ms" ');
```



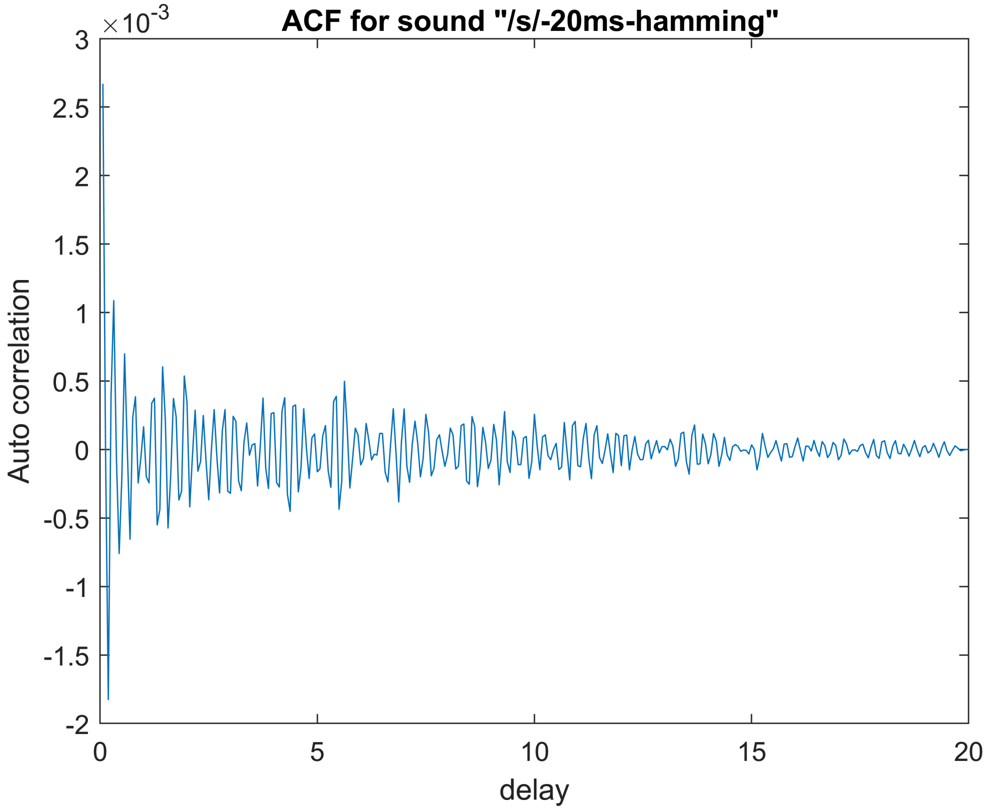
```
% For window size 100ms
Frame_size=100;
Frame_size= Frame_size/1000;
ye_a=autocorrelation(y_e,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_a)/fs))*1000;
plot(te_a,ye_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-100ms" ');
```



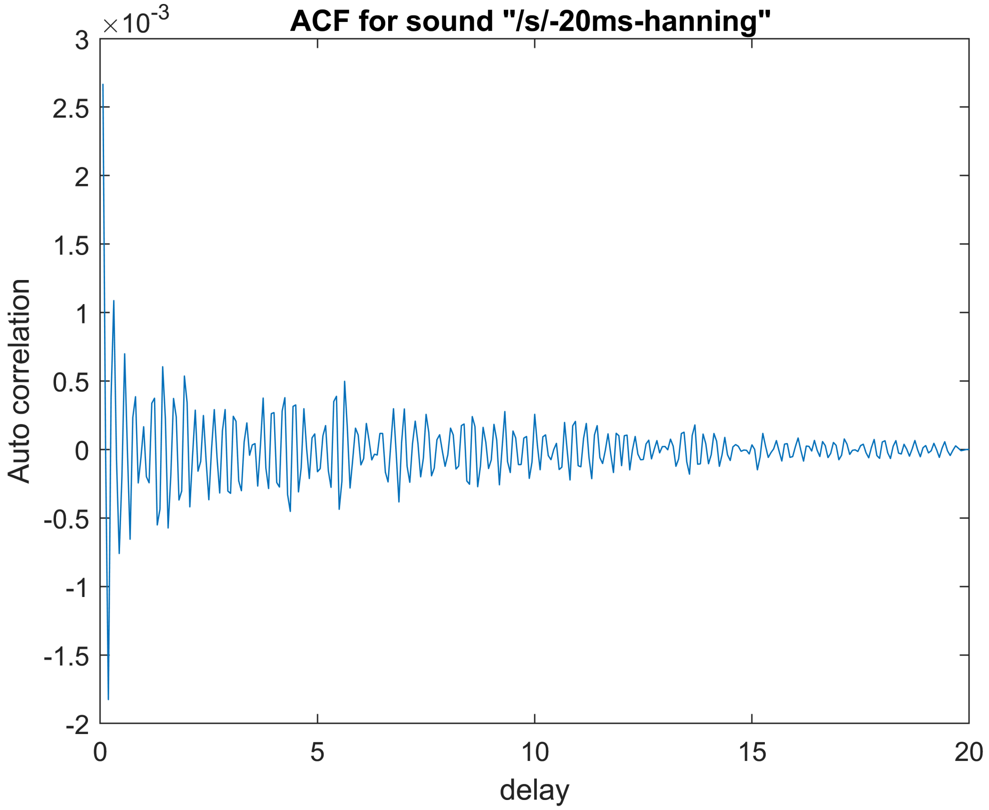
```
% For window size 20ms
% Rectangular window
Frame_size=20;
Frame_size= Frame_size/1000;
ys_a=autocorrelation(y_s,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_a)/fs))*1000;
plot(ts_a,ys_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-20ms-rectangular" ');
```



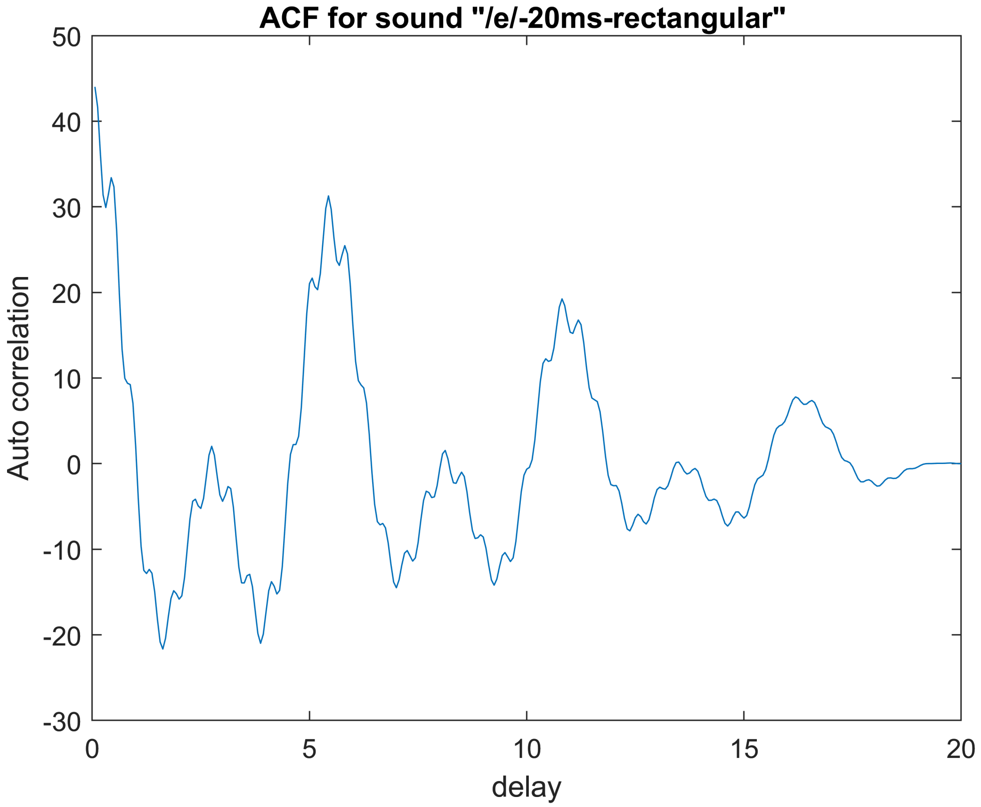
```
% Hamming window
Frame_size=20;
Frame_size= Frame_size/1000;
window_length=Frame_size*fs;
win_hamm = dsp.Window('Hamming');
y_s_hamm = win_hamm(y_s);
ys_hamm_a=autocorrelation(y_s_hamm,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_hamm_a)/fs))*1000;
plot(ts_a,ys_hamm_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-20ms-hamming" ');
```



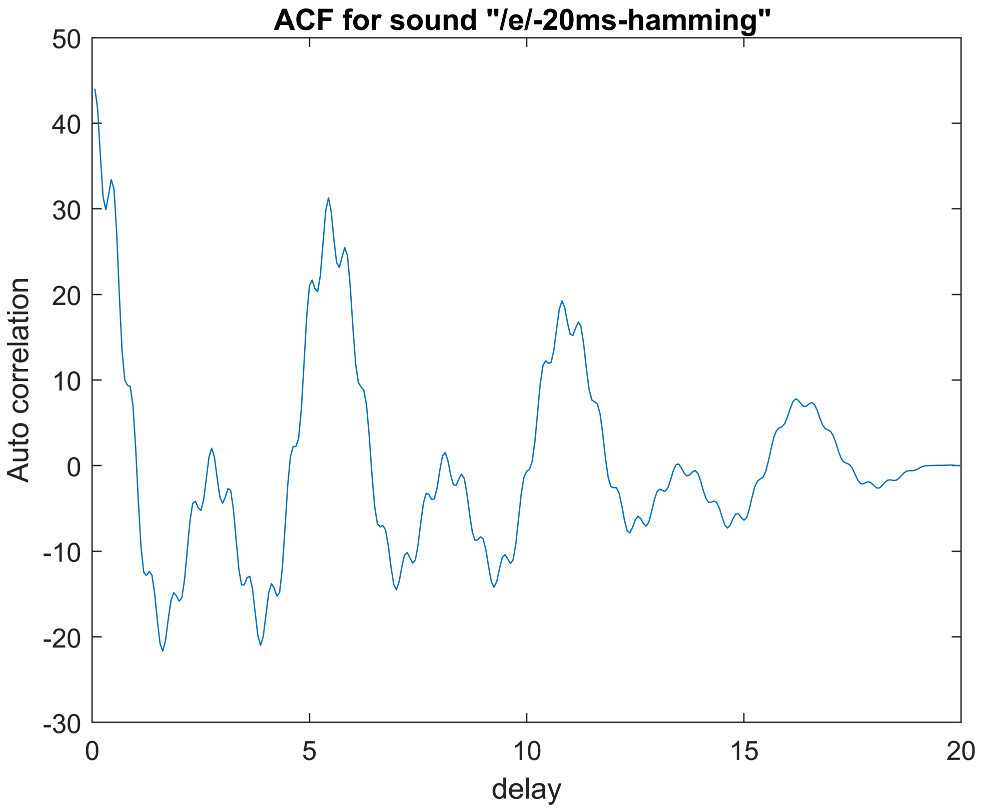
```
% Hanning window
Frame_size=20;
Frame_size= Frame_size/1000;
window_length=Frame_size*fs;
win_hann = dsp.Window('Hanning');
y_s_hann = win_hann(y_s);
ys_hann_a=autocorrelation(y_s_hann,fs,Frame_size);
ts_a=(1/fs:1/fs:(length(ys_hann_a)/fs))*1000;
plot(ts_a,ys_hann_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/s/-20ms-hanning" ');
```



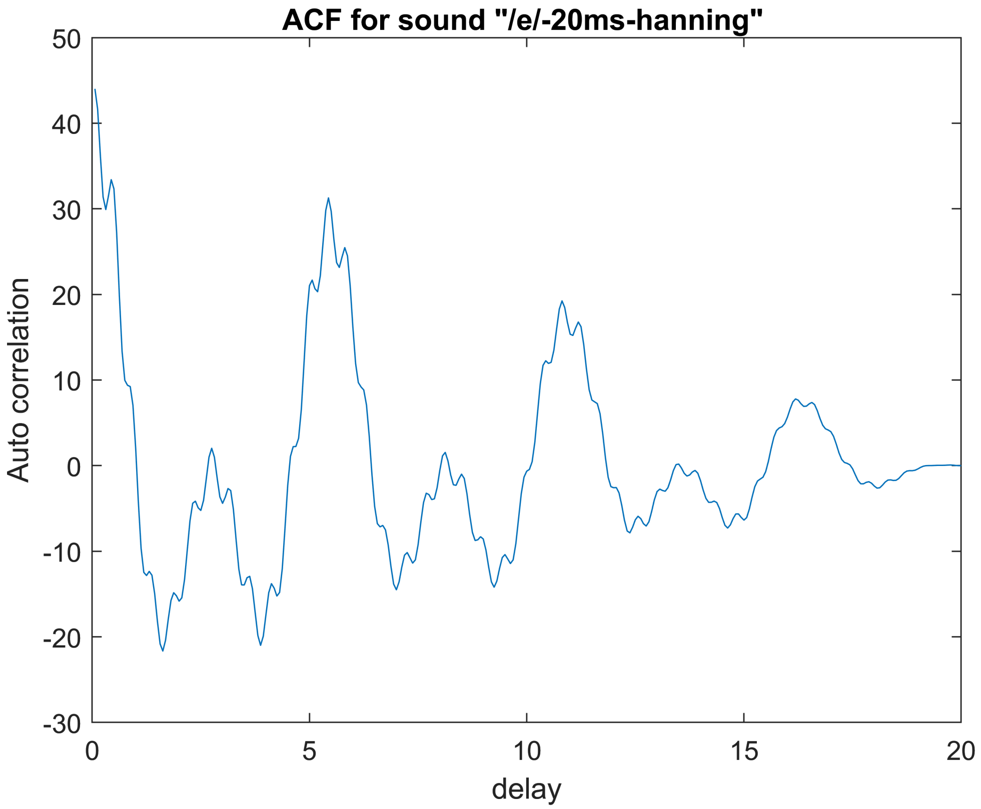
```
% For window size 20ms
% Rectangular window
Frame_size=20;
Frame_size= Frame_size/1000;
ye_a=autocorrelation(y_e,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_a)/fs))*1000;
plot(te_a,ye_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-20ms-rectangular" ');
```



```
% Hamming window
Frame_size=20;
Frame_size= Frame_size/1000;
window_length=Frame_size*fs;
win_hamm = dsp.Window('Hamming');
y_e_hamm = win_hamm(y_e);
ye_hamm_a=autocorrelation(y_e_hamm,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_hamm_a)/fs))*1000;
plot(te_a,ye_hamm_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-20ms-hamming" ');
```



```
% Hanning window
Frame_size=20;
Frame_size= Frame_size/1000;
window_length=Frame_size*fs;
win_hann = dsp.Window('Hanning');
y_e_hann = win_hann(y_e);
ye_hann_a=autocorrelation(y_e_hann,fs,Frame_size);
te_a=(1/fs:1/fs:(length(ye_hann_a)/fs))*1000;
plot(te_a,ye_hann_a);
xlabel('delay');
ylabel('Auto correlation');
title('ACF for sound "/e/-20ms-hanning" ');
```



Observations:

We prefer the frame size of 20 ms as it is not too long and not too short. The quasi stationarity assumption is followed and there is enough data to capture pitch period.

The Hamming and Hanning windows suppress the ends of the frame. As we are interested only in the first major peak in the autocorrelation plot, it is good to use a Hamming or a Hanning window.

Both Hamming and Hanning windows have a similar structure. As we are interested only in the first major peak occurrence, we choose Hanning window as it suppresses the ends more compared to Hamming window.

```
% Short term energy
function[e]=short_term_energy(speechsignal,fs,Frame_size,Frame_shift)
y=speechsignal;
Frame_size= Frame_size/1000;
Frame_shift= Frame_shift/1000 ;
window_length=Frame_size*fs;
sample_shift=Frame_shift*fs;
sum=0 ;energy=0 ;
w=rectwin(window_length);
jj=1;
for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
    for j=(((i-1)*sample_shift)+1):(((i-1)*sample_shift)+window_length)
        y(j)=y(j)*w(jj) ;
        jj=jj+1 ;
        yy=y(j)*y(j);
        sum=sum+yy;
    end
    energy(i)=sum ;
    sum=0 ; jj=1 ;
end
w=0;
e=energy;
return ;
end
% Zero crossing rate
function[z]=zero_crossing_rate(speechsignal,fs,Frame_size,Frame_shift)
y=speechsignal;
Frame_size= Frame_size/1000;
Frame_shift= Frame_shift/1000 ;
window_length=Frame_size*fs;
sample_shift=Frame_shift*fs;
sum=0 ;energy=0 ;
w=rectwin(window_length);
jj=1 ;

for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
    y(((i-1)*sample_shift)+1)=y(((i-1)*sample_shift)+1)*w(jj);
    jj=jj+1 ;
    for j=(((i-1)*sample_shift)+2):(((i-1)*sample_shift)+window_length)
        y(j)=y(j)*w(jj) ; jj=jj+1 ;
        yy=y(j)*y(j-1);
        if(yy<0)
            sum=sum+1;
        end
    end
    zcr(i)=sum/(2*window_length);
    sum=0 ; jj=1 ;
end
w=0;
z=zcr;
return ;
end

% Autocorrelation
function [a]=autocorrelation(speechsignal,fs,Frame_size)
y=speechsignal;
window_length=Frame_size*fs;
max_value=max(abs(y));
y=y/max_value;
y=y(1:window_length);
sum=0;
autocorrelation=[];
for l=0: (length(y)-1)
    sum=0;
    for u=1:length(y)-l
        s=y(u)*y(u+l);
        sum=sum+s;
    end
    autocorrelation(l+1)=sum;
end
a=autocorrelation;
end
```