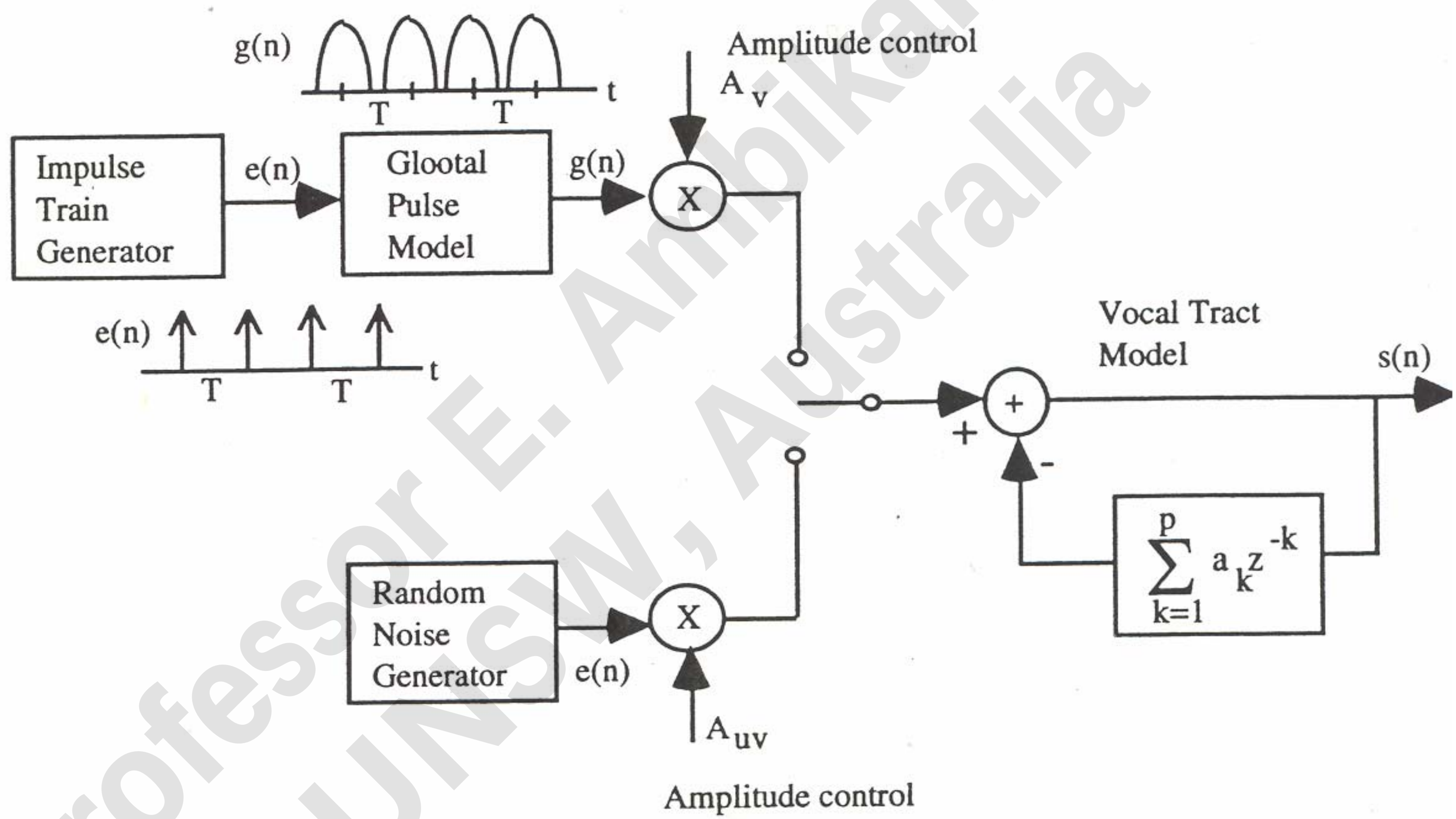
Speech & Audio Processing

Chapter 2

Speech Analysis

Time-Domain Methods for Speech Processing



Discrete-Time Model For Speech Production

Over a short time interval, the above linear system has the transfer function:

$$\frac{S(z)}{E(z)} = \frac{A_v}{1+\sum_{k=1}^{P+1} a_k z^{-k}}$$

For voiced sounds: all pole model)

where p+1 = no of poles and e(n) is the excitation function.

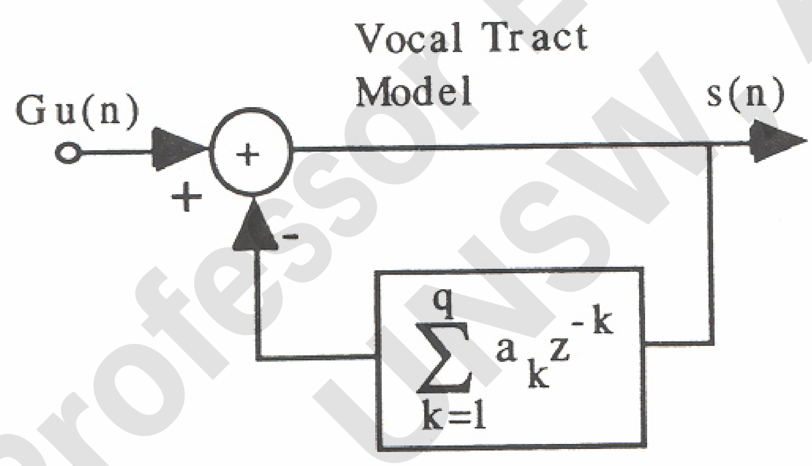
This simplified all pole model is a natural representation of voiced sounds, but for nasal and fricative sounds the detailed theory calls for both poles and zeros in the vocal tract transfer function.

$$\frac{S(z)}{E(z)} = A_{uv} \frac{1+\sum_{k=1}^{L} b_k z^{-k}}{1+\sum_{k=1}^{P} a_k z^{-k}}$$

For nasal and fricative sounds

We would prefer an all-pole model. The zeros can be transformed to poles as explained previously with L zeros transforming to 2L poles. An all-pole model is given by

$$\frac{S(z)}{U(z)} = \frac{G}{1+\sum_{k=1}^{q} a_k z^{-k}} \qquad q = p+2L$$

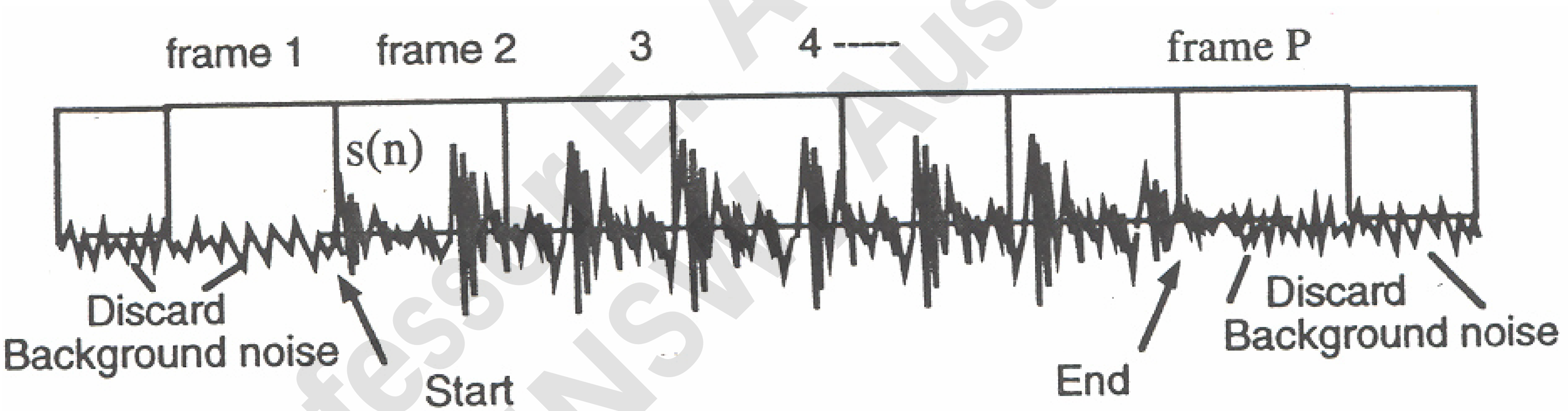


All-Pole Model

However, if the order q is high enough, the all-pole model provides a good representation for almost all the sounds of speech; typically q=12

The major advantage of the all-pole model is that the gain parameter G and the filter coefficients (a_k, k = 1,2,3,..) can be easily estimated in a very straightforward and computationally efficient manner, also with good accuracy.

Any given utterance will last a certain amount of time. It is split into frames for processing as given:



Each frame will typically contain 100 samples (assuming sampling frequency of 8kHz). Each frame is thus 12.5 ms in duration

Basic Parameter Extraction

- There are a number of very basic speech parameters which can be easily calculated for use, in simple applications:
  - Short Time Energy
  - Short Time Zero Cross Count (ZCC)
  - Pitch Period
- All of the above parameters are typically estimated for frames of speech between 10 and 20 ms long

Short Time Energy

- The short-time energy of speech may be computed by dividing the speech signal into frames of N samples and computing the total squared values of the signal samples in each frame.
- Splitting the signal into frames can be achieved by multiplying the signal by a suitable window function w(n) {n=0, 1, 2, 3, …, N-1}, which is zero for n outside the range (0, N-1)

Rectangular Window

- A simple rectangular window of duration of 12.5 ms is suitable for this purpose. For a window starting at sample m, the short-time energy E_m is defined as

$$E_m = \sum_n [s(n)\, w(m-n)]^2$$

$$w(n) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & otherwise \end{cases}$$

$$E_m = \sum_n [s(n)]^2\, h(m-n)$$

$$h(n) = [w(n)]^2$$

Linear filter representation

- The above equation (see previous slide) can thus be interpreted as

The signal $s(n)^2$ is filtered by a linear filter with impulse response h(n) .

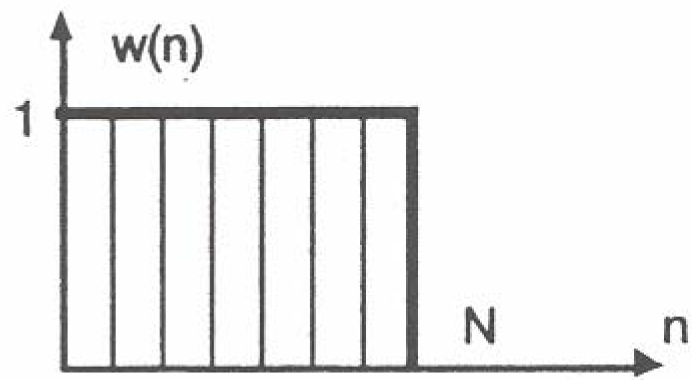
The choice of the impulse response ,h(n) or equivalently the window, determines the nature of the short-time energy representation.

To see how the choice of window affects the short-time energy, let us observe that if h(n) was very long and of constant amplitude E_m would change very little with time

Such a window would be equivalent of a very narrowband lowpass filter. Clearly what is desired is some lowpass filtering, so that the short-time energy reflects the amplitude variations of the speech signal.

We wish to have a short duration window to be responsive to rapid amplitude changes. But a window that is too short will not provide sufficient averaging to produce a smooth energy function.

Note: Rectangular window



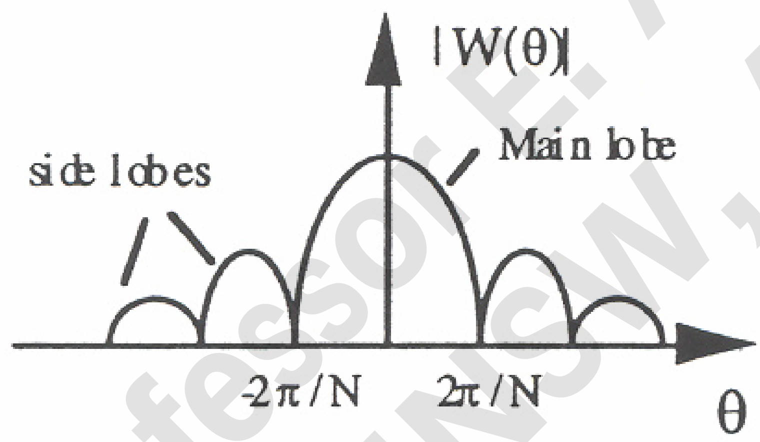
$$w(n) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & otherwise \end{cases}$$

$$W(z) = \sum_{n=0}^{N-1} w(n) z^{-n} = 1 + z^{-1} + z^{-2} + z^{-3} + \ldots + z^{-(N-1)} = \frac{1 - z^{-N}}{1 - z^{-1}}$$

$$W(\theta) = W(z)\big|_{z=e^{j\theta}} = \frac{1 - e^{-jN\theta}}{1 - e^{-j\theta}};$$

$$W(\theta) = e^{-j\frac{N-1}{2}\theta} \frac{Sin\frac{N\theta}{2}}{Sin\frac{\theta}{2}}$$

Phase term *Magnitude*



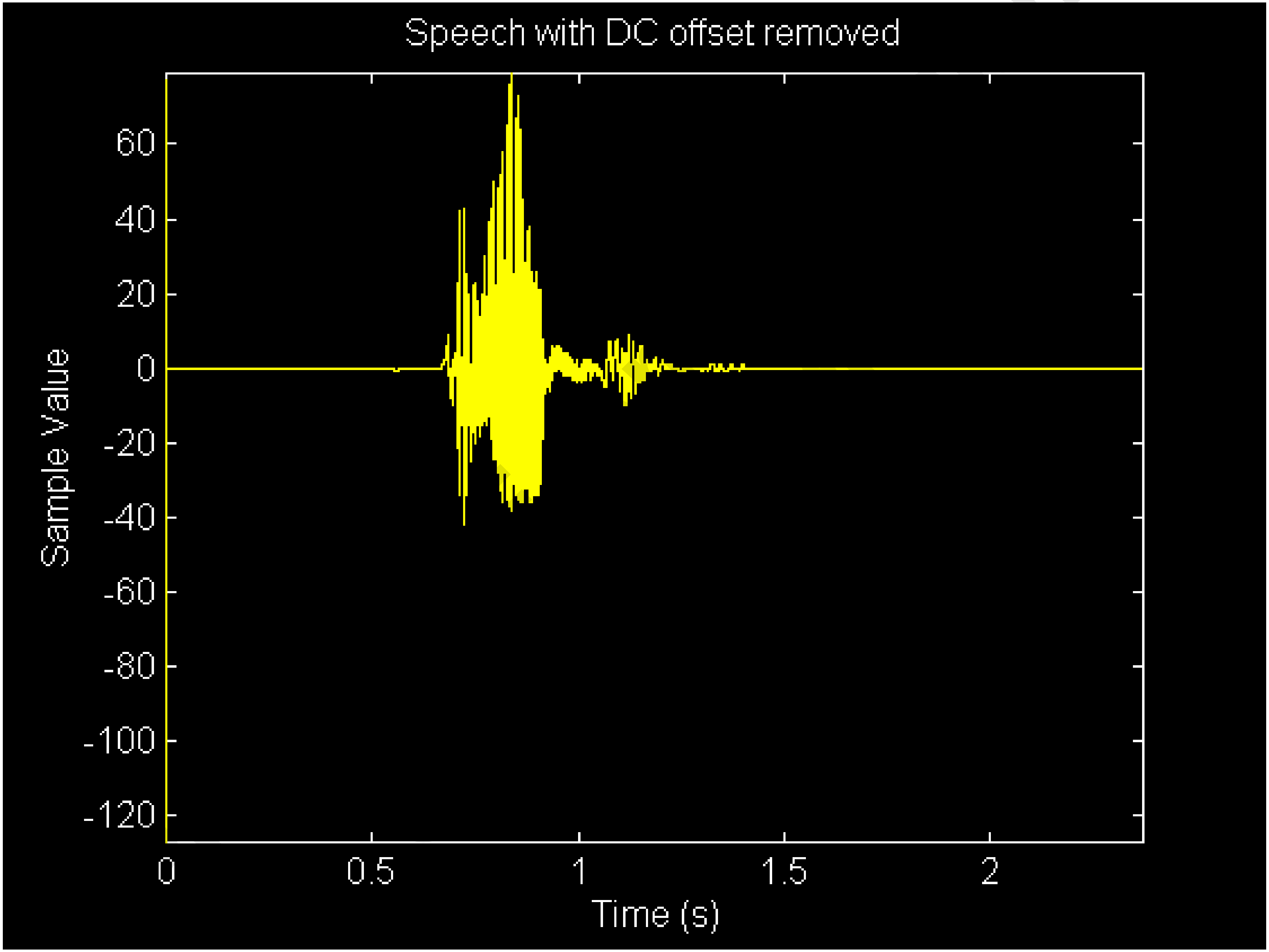
If N is too small, E_m will fluctuate very rapidly depending on exact details of the waveform.

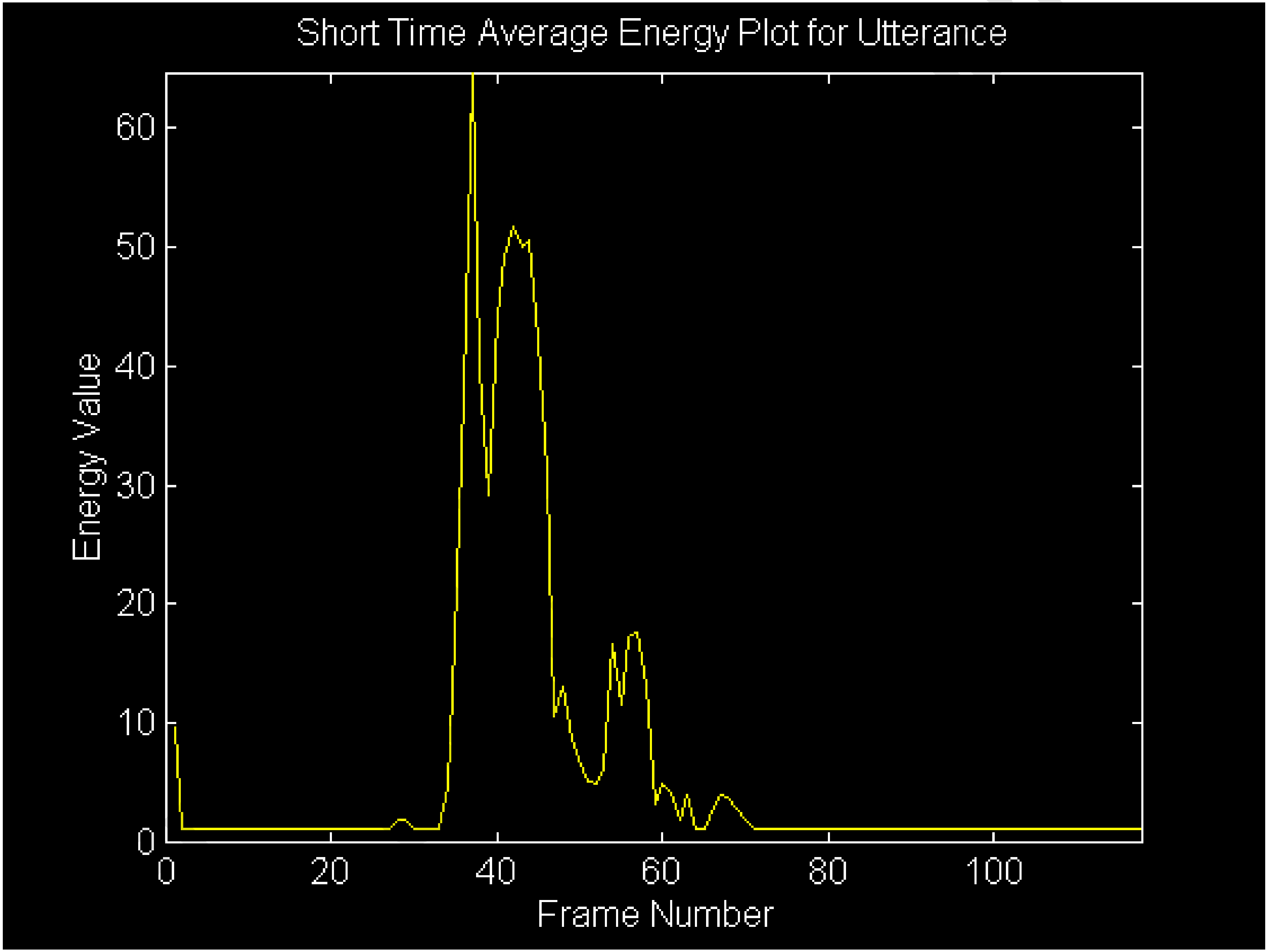
In N is too lager, E_m will change very slowly and thus will not adequately reflect the changing properties of the speech signal

Choice of Window Size

- Unfortunately this implies that no single value of N is entirely satisfactory.

- A suitable practical choice for N is on the order of 100-200 samples for a 10 kHz sampling rate (10-20 ms duration)


Speech with DC offset removed
Sample Value
Time (s)
60
40
20
0
-20
-40
-60
-80
-100
-120
0
0.5
1
1.5
2



Short Time Average Energy Plot for Utterance
Energy Value
Frame Number
0
10
20
30
40
50
60
0
20
40
60
80
100

Note that a recursive lowpass filter H(z) can also be used to calculate the short-time energy:

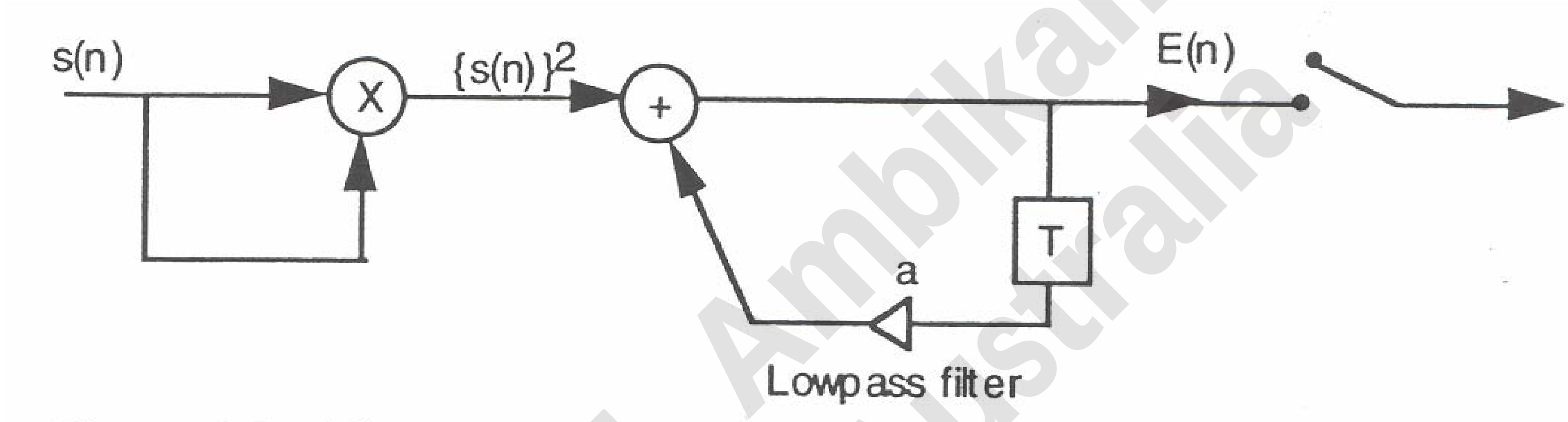
$$H(z) = \frac{1}{1 - az^{-1}} \qquad 0 < a < 1$$

It can be easily verified that the frequency response H(θ) has the desired lowpass property. Such a filter can be implemented by a simple difference equation:

$$E(n) = a\,E(n-1) + [s(n)]^2$$

E(n) is the energy at the time instant n

The structure for calculating the short-time energy recursively



The quantity E(n) must be computed at each sample of input speech signal, even though a much lower sampling rate suffice.

The value ‘a’ can be calculated using $a = e^{(-f_c 2\pi / f_s)}$

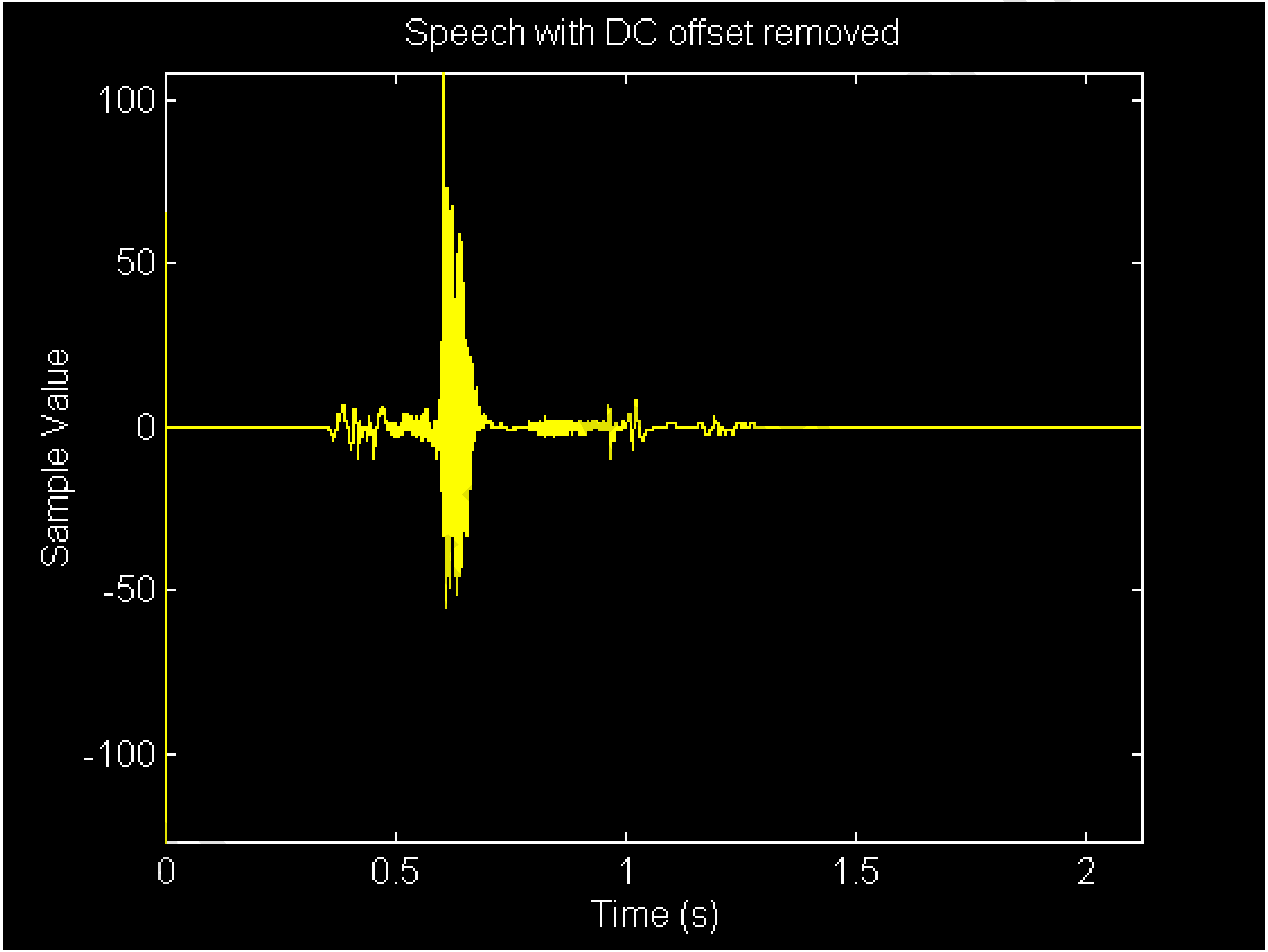
Fc is the cut-off frequency and fs is the sampling frequency (e.g fc=30 Hz, fs = 8000Hz)

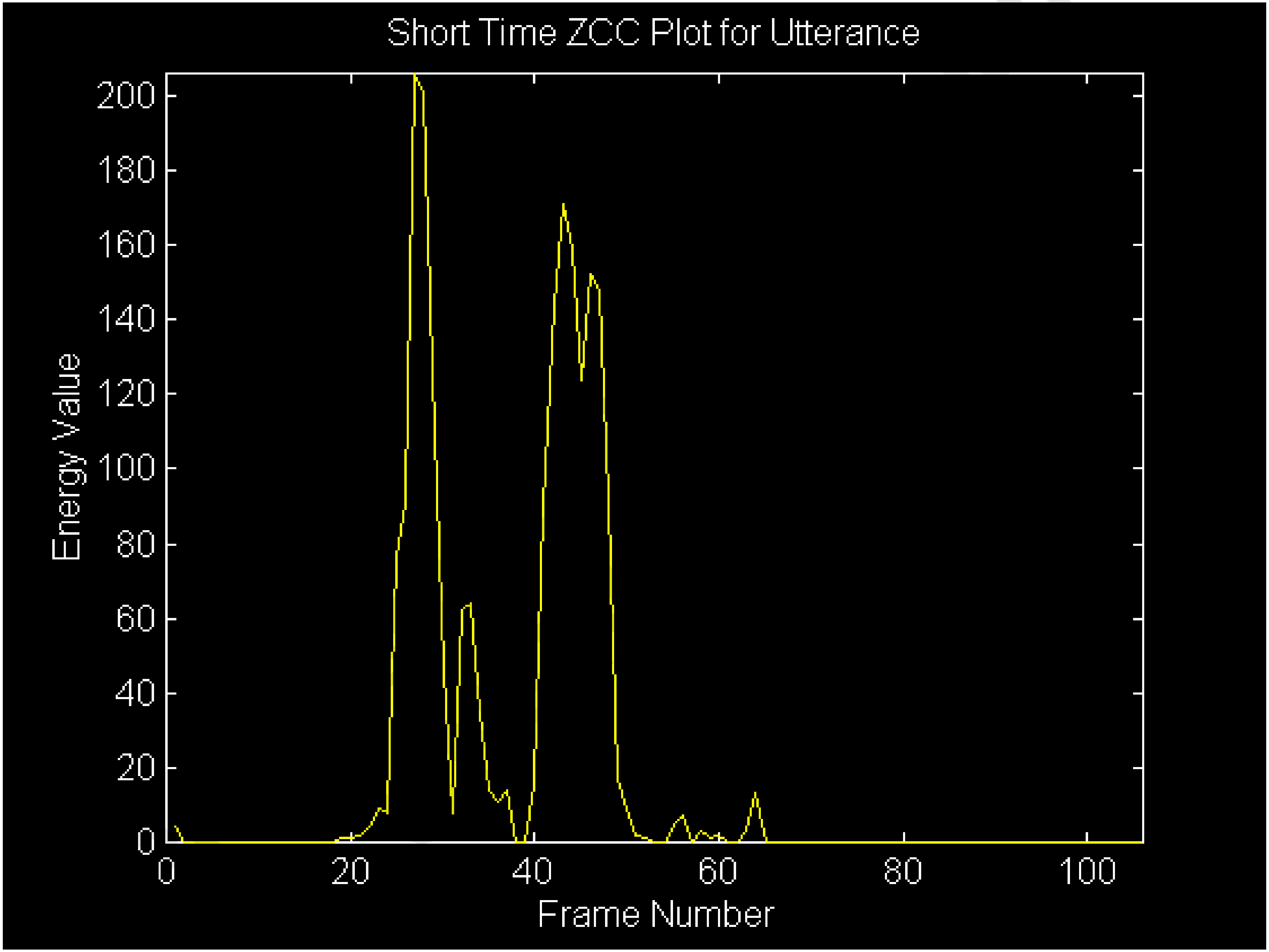
Short Time Zero Crossing Count

- The Short Time ZCC is calculated for a block of N samples of speech as

$$ZCC_i = \sum_{k=1}^{N-1} 0.5\,|\,sign(s[k]) - sign(s[k-1])\,|$$

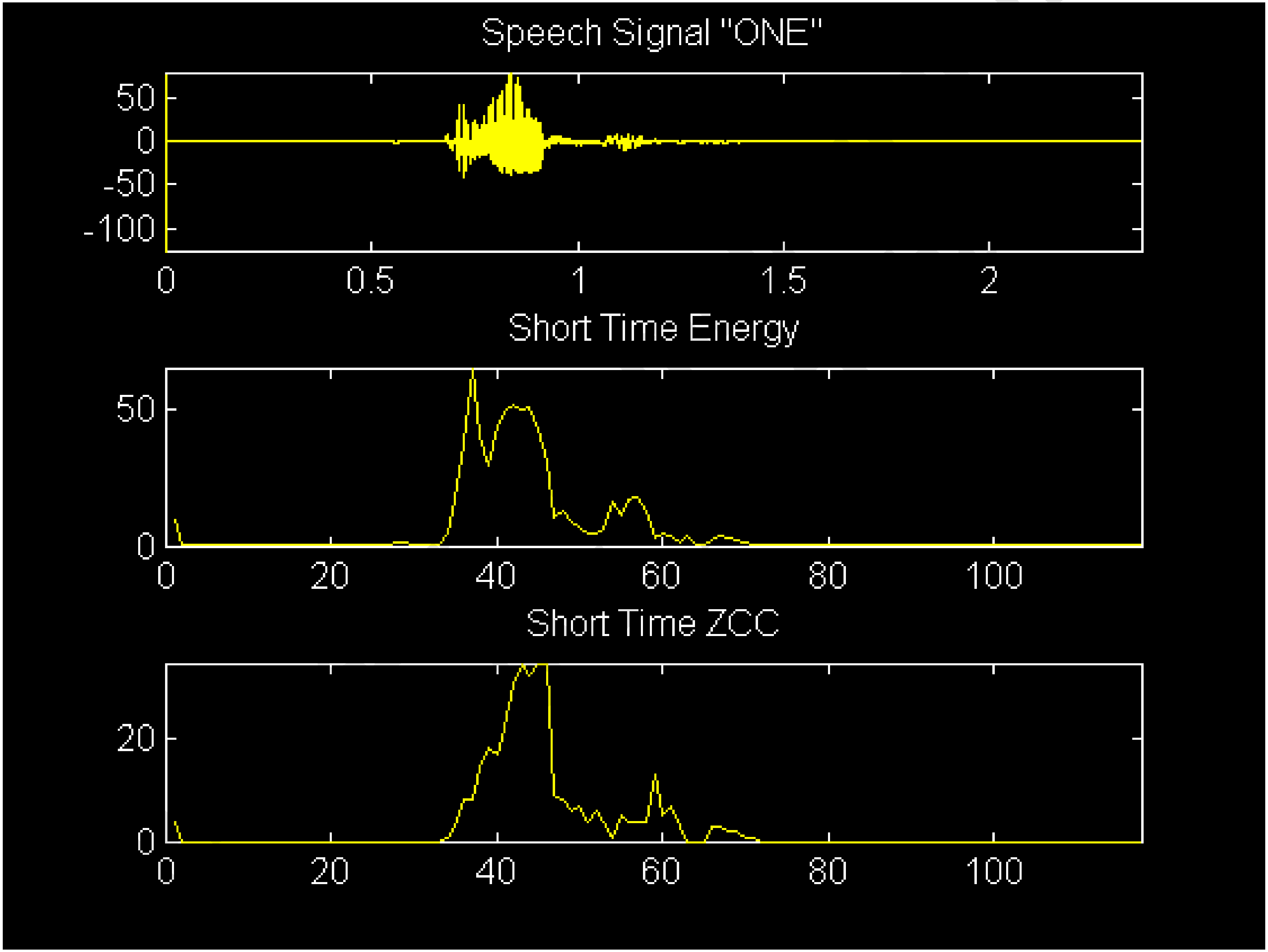
- The ZCC essentially counts how many times the signal crosses the time axis during the frame
  - It "reflects" the frequency content of the frame of speech
    - High ZCC implies high frequency
- It is essential that any constant DC offset is removed from the signal prior to ZCC calculation

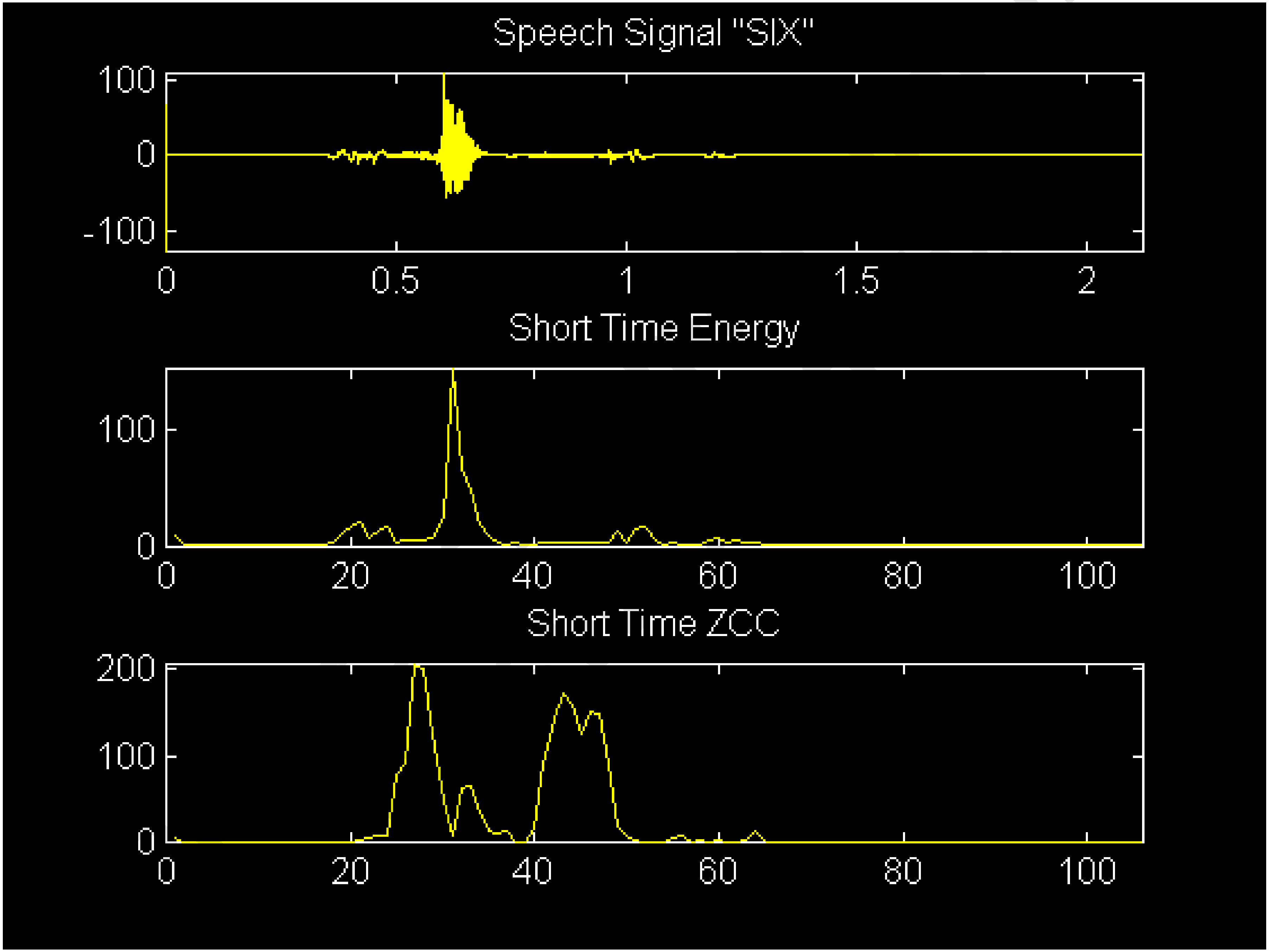

Speech with DC offset removed
Sample Value
100
50
0
-50
-100
0
0.5
1
1.5
2
Time (s)


Short Time ZCC Plot for Utterance
Energy Value
Frame Number
0
20
40
60
80
100
120
140
160
180
200
0
20
40
60
80
100

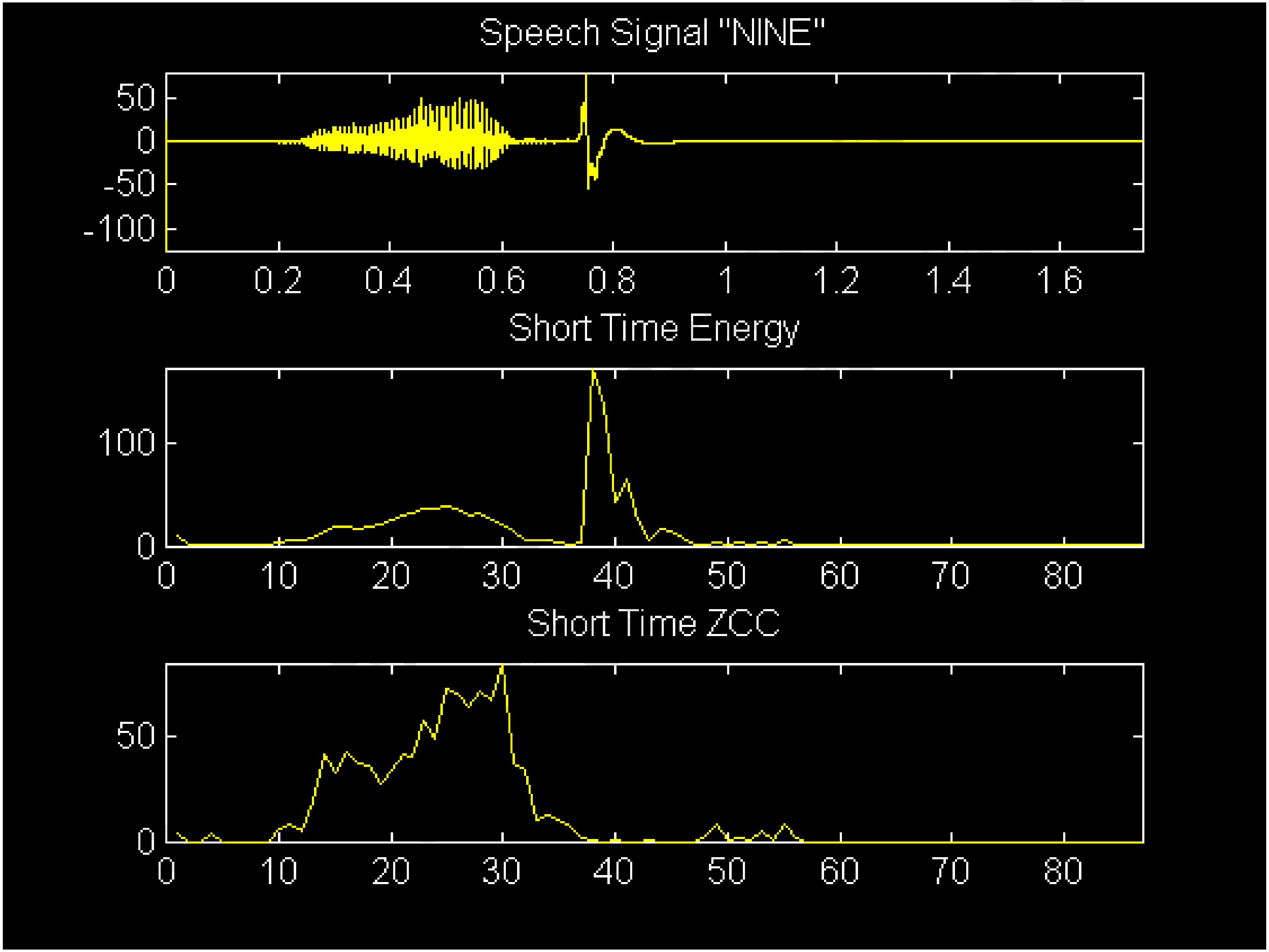
Uses of Energy and ZCC

- Short Time Energy and ZCC can form the basis for :
  - Automated speech "end point" detection
    - Needs to be able to operate with background noise
    - Needs to be able to ignore "short" background noises and intra-word silences (temporal aspects)
  - Voiced\Unvoiced speech detection
    - High Energy + Low ZCC – Voiced Speech
    - Low Energy + High ZCC – Unvoiced Speech
  - Parameters on which simple speech recognition\speaker verification\identification systems could be based


Speech Signal "ONE"
50
0
-50
-100
0
0.5
1
1.5
2
Short Time Energy
50
0
0
20
40
60
80
100
Short Time ZCC
20
0
0
20
40
60
80
100

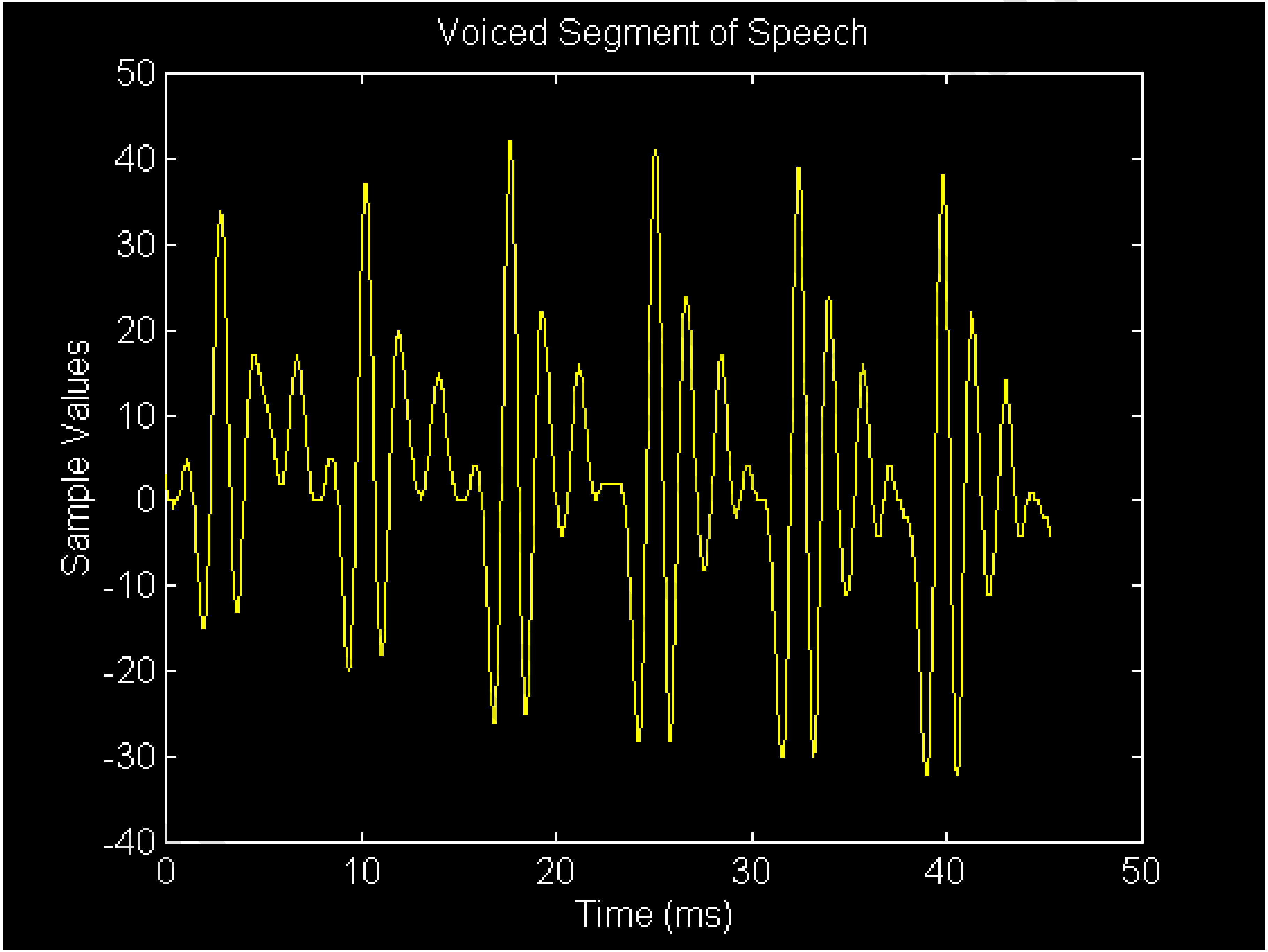


Speech Signal "SIX"
100
0
-100
0
0.5
1
1.5
2
Short Time Energy
100
0
0
20
40
60
80
100
Short Time ZCC
200
100
0
0
20
40
60
80
100


Speech Signal "NINE"
50
0
-50
-100
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Short Time Energy
100
0
0
10
20
30
40
50
60
70
80
Short Time ZCC
50
0
0
10
20
30
40
50
60
70
80

Pitch Period Estimation

- Pitch period is equal to the inverse of the fundamental frequency of vibration of the vocal chords
- It only makes sense to speak about the pitch period of a VOICED frame of speech
- Number of techniques used to determine pitch period
  - Time Domain
  - Frequency Domain


Voiced Segment of Speech
Sample Values
Time (ms)
50
40
30
20
10
0
-10
-20
-30
-40
0
10
20
30
40
50

Time Domain Methods

- Since pitch frequency is typically less then 600-700 Hz, the speech signals are first low passed filtered to remove components above this frequency range
- The two most commonly used techniques are:
  - Short Time Autocorrelation Function
  - Average Magnitude Difference Function (AMDF)
- During voiced speech, the speech signal is “quasi-periodic”
- Either technique attempts to determine the period (in samples between “repetitions” of the voiced speech signal

Autocorrelation Function

- Correlation is a very commonly used technique in DSP to determine the "time difference" between two signals, where one is a "nearly perfect" delayed version of the other
- Autocorrelation is the application of the same technique to determine the unknown "period" of a quasi-periodic signal such as speech
- The autocorrelation function for a delay value of k samples is:

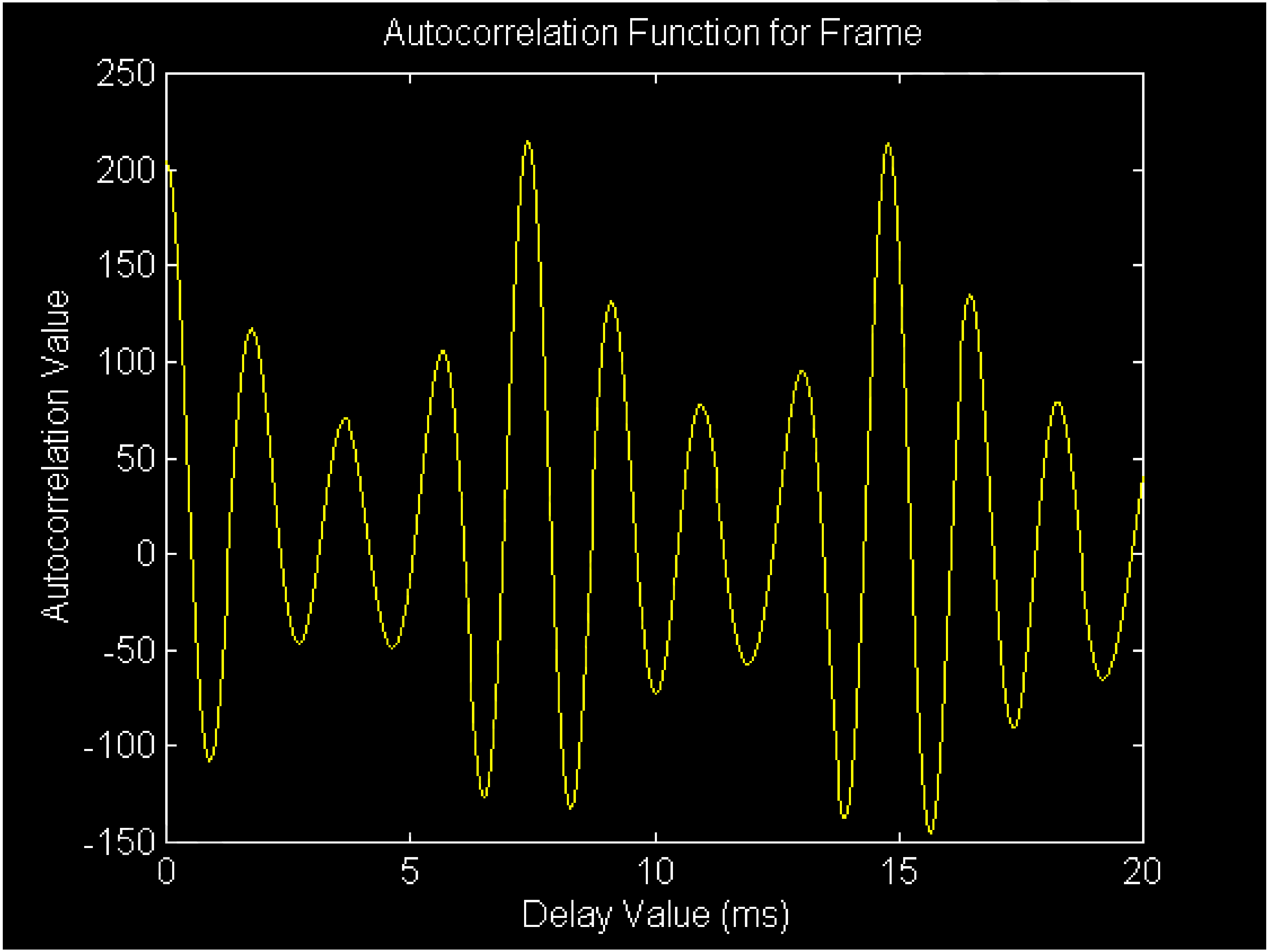
$$\phi(k) = \frac{1}{N} \sum_{n=0}^{N-1} s[n]s[n+k]$$

Autocorrelation Function

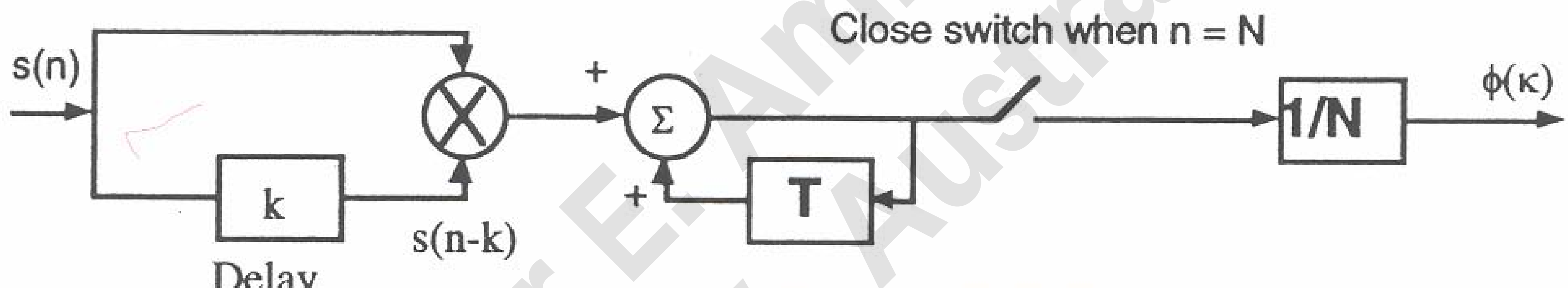
- Clearly, $\phi(k=0)$ would be equal to the average energy of the signal s[n] over the N sample frame
- If s[n] was perfectly periodic with a period of P samples then $s[n+P]=s[n]$
- Therefore, $\phi(k=P)=\phi(k=0)$=Average Energy
- While this is NOT exactly true for speech signals, the autocorrelation function with k equal to the pitch would result in a large value
- For the various k values between 0 and P, the various terms ($s[n]s[n+k]$) in the autocorrelation function would tend to be a mixture of positive and negative values
- These would tend to cancel each other out in the autocorrelation sum to yield very low values for $\phi(k)$

Autocorrelation Function

- This, for a given frame of N samples of VOICED speech, a plot of $\phi(k)$ versus k would exhibit distinct peaks at k values of 0, P, 2P ….., where P is the pitch period
- The graph of $\phi(k)$ would be of quite small values between these peaks
- This pitch period for that frame is simply got by measuring the distance, in samples, between the peaks of the graphs of the autocorrelation function


Autocorrelation Function for Frame
Autocorrelation Value
Delay Value (ms)
250
200
150
100
50
0
-50
-100
-150
0
5
10
15
20

A block diagram of the implementation of the autocorrelation function is shown below:



Autocorrelation Function

$$\phi(k) = \frac{1}{N}\sum_{n=0}^{N-1} s[n]s[n+k]$$

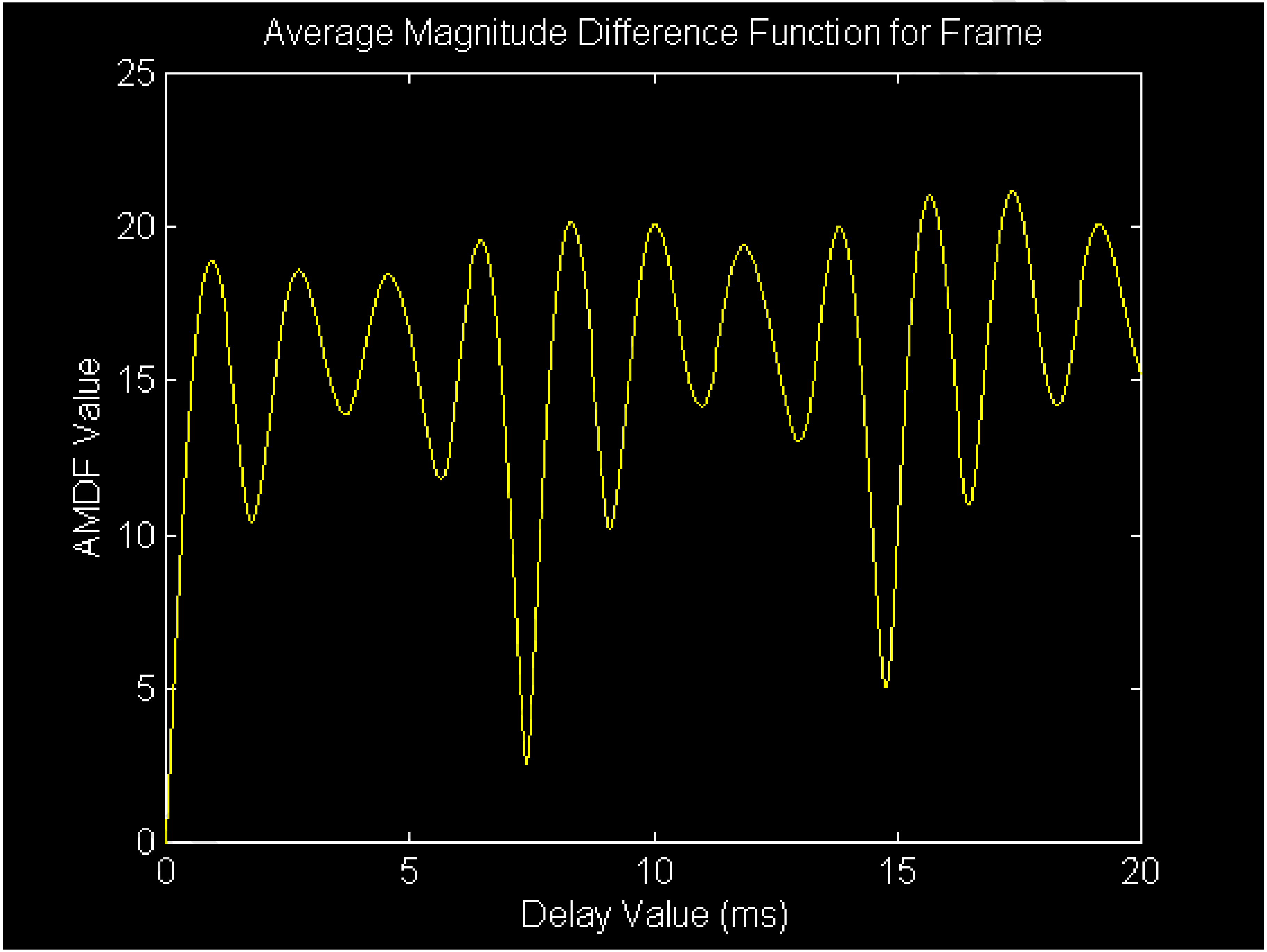
Average Magnitude Difference Function

- The AMDF is similar but opposite to the Autocorrelation Function
- For a delay of k samples, the AMDF is defined as

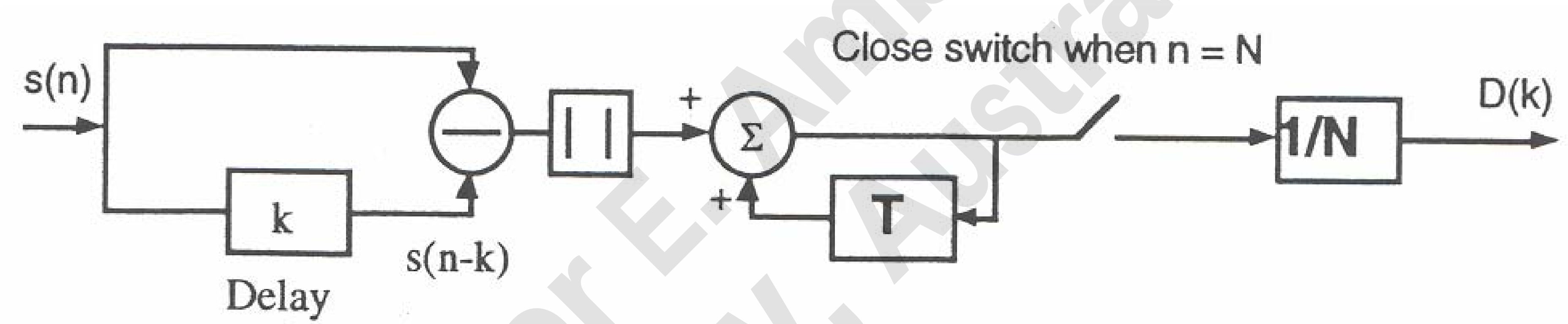
$$D(k) = \frac{1}{N}\sum_{n=0}^{N-1} | s[n] - s[n+k] |$$

Average Magnitude Difference Function

- For a given frame of VOICED speech, a plot of AMDF (D(k)) versus different values of delays (k), will exhibit deep “nulls” at k=0, P , 2P……
- If is used as an alternative to autocorrelation as on some processor architectures, it may be less computationally intensive to implement
- Care should be taken with both techniques to support the “overlap” into adjacent frames introduced by the the autocorrelation and AMDF


Average Magnitude Difference Function for Frame
AMDF Value
Delay Value (ms)
0
5
10
15
20
25
0
5
10
15
20

A block diagram implementation of the AMDF function:



$$D(k) = \frac{1}{N}\sum_{n=0}^{N-1} | s[n] - s[n+k] |$$

Matlab Code:

```
in1=fopen('C:\speech.dat','rb');
nsamples=5000; %number of samples
nframes = 25;     %number of frames
framesize=200;
ppmin=20; %fundamental freq=400Hz
ppmax=100;%fundamental freq=80 Hz

%initialisation of arrays
for j=1:ppmax, D(j)=0;end;
figure;
s=fread(in1,nsamples,'short'); %plot(s)

pointer1=1;
for i=1:nframes

        for k=ppmin:ppmax
                sum1=0.0;
                for n=pointer1:pointer1+framesize-1
                        sum1=sum1+abs(s(n)-s(n+k));
                end;
                D(k)=sum1/framesize;
        end;
        subplot(2,1,1);plot(s(pointer1:pointer1+framesize-1));
        subplot(2,1,2);plot(D);
        pointer1=pointer1+framesize;
        pause;
end;
fclose(in1);
```

Pre-emphasis Filter

- Recall transfer function of vocal tract:

$$\frac{S(z)}{E(z)} = A_v \frac{1}{\left(1 - z^{-1}\right)^2} \frac{1}{1 + \sum_{k=1}^{P} a_k z^{-k}} 1 - z^{-1}$$

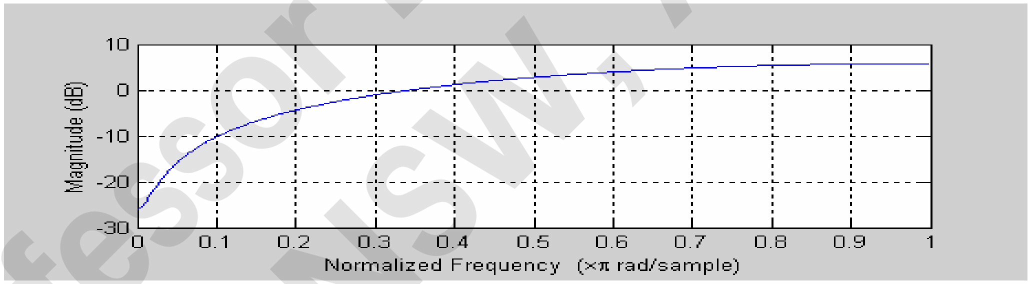
- There is an –6dB/octave trend as frequency increases
- It is desirable to compensate for this by preprocessing the speech. This has the effect of cancelling out effect of glottis and is know as pre-emphasis.

Pre-empahsis

- The high pass filtering function can be achieved by use of following difference equation:

$$y(n) = s(n) – a\ s(n\text{-}1)$$

- Normally a is chosen between 0.9 and 1.



Exercise: Pre-Emhasis Filter

1. Use Matlab to plot the frequency response of a pre-emphasis filter with the following transfer function

$$H(z) = 1 – 0.95z^{-1}$$

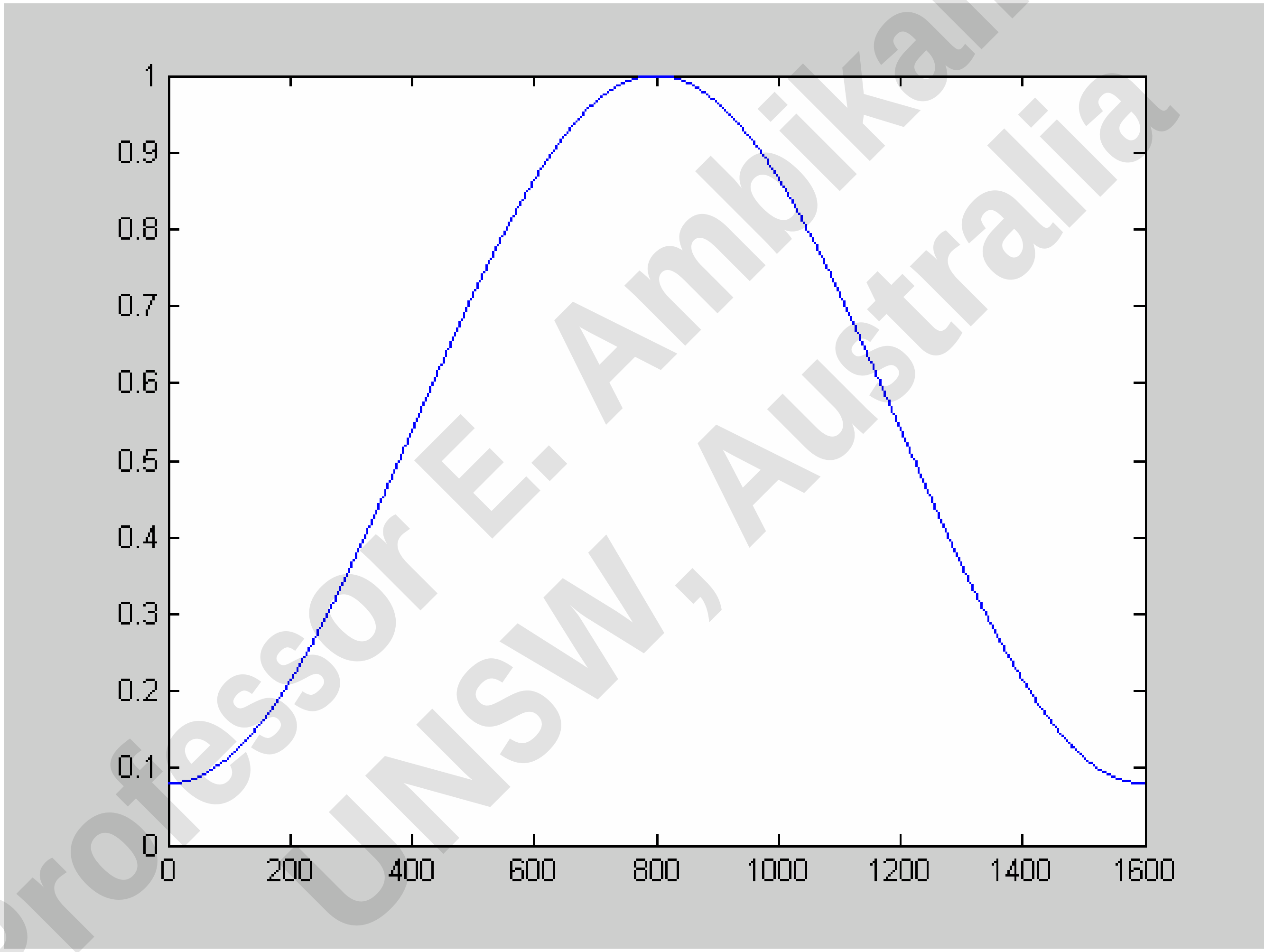
2. Plot the spectra of a frame of speech before and after pre-emphasis filter has been applied

Short Time Fourier Transform

- Spectrogram may be attained through use of STFT.
- FT is carried out on a short sequence of signal.
- The signal may be windowed e.g. Hamming Window (see next slide)
- Overlapping should also be carried out
- Following formula for calculating STFT with window w or length N

$$STFT(k,b) = \sum_{m=-N/2}^{m=N/2} w(m-b)s(m)e^{-j\frac{2\pi k}{N}m}$$

Hamming Window



Exercise STFT

1. Generate a signal composed of 4 tones of different frequencies
   - 2 tones should be present constantly and other 2 tones occuring at different times.
   - Signal should be about 1 second in length in total and tones should have different levels
2. Write a script to perform the STFT
   - Include Hamming window
   - 50% overlapping of frames
3. Plot a spectrogram of the signal.
4. Investigate effect of
   1. changing frame size
   2. Changing number of points in FFT.
5. Record a voice signal and generate spectrogram

Exercise: Signal Reconstruction

Part A – Entire Signal

1. Record a voice signal of length ~0.5s
2. Perform an FFT of the speech and plot its spectrum
3. Examine both magnitude and phase
4. Recalculate the complex FFT coefficents from Magnitude and phase and check they are as in 3.
5. Reconstruct the entire speech using IFFT

Part B – Framed Signal (50% Overlapped)

1. Apply a Hamming window to each frame of signal prior to getting FFT
2. Reconstruct each frame using IFFT
3. Use overlap and add tehnique to reconsrtruct speech