Scilab Manual for
Audio and Speech Processing
by Prof Muralikrishna H
Electronics Engineering
Manipal Institute of Technology[1]

Solutions provided by
Prof Muralikrishna H
Electronics Engineering
Manipal Institute of Technology, Manipal

April 21, 2026

[1] Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

Basic operations on Audio signals.

Scilab code Solution 1.1 Program to Read and Play audio file

```
//This code Reads an audio file and plays it.
//
//OS: windows 7
//Scilab: 5.5.2
//
clc;
close;
clear;
[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
    //read the audio file using the given path
sound(y,Fs);  // sound(2*y,Fs); can be used for
    better quality
t=(0:length(y)-1)*1/Fs;
plot(t,y)
title('Speech signal waveform')
xlabel('Time in seconds')
ylabel('Amplitude')
```

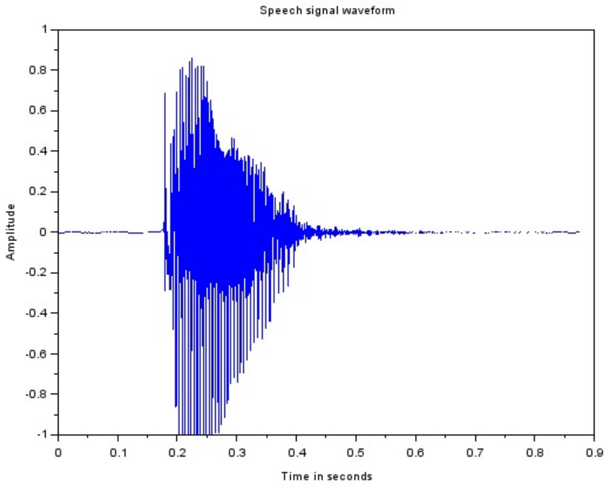


Figure 1.1: Program to Read and Play audio file

Scilab code Solution 1.2 Effect of changing Sampling Frequency

```
//This code Reads an audio file and plays it by
   changing the sampling frequency.
//
//OS: windows 7
//Scilab: 5.5.2
//
clc;
close;
clear;
[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
   //read the audio file using the given path
Fs_new=Fs/.8;  // increasing Fs. Can be reduced by
   using command  Fs_new=Fs/1.3;
sound(y,Fs_new);  // sound(2*y,Fs); can be used for
   better quality
```

Scilab code Solution 1.3 Program to Reverse and play audio file

```
//This code Reads an audio file and plays it by reversing the file.
//
//OS: windows 7
//Scilab: 5.5.2
//
clc;
close;
clear;
[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav'); // Give the path
z = y(:,$:-1:1); //Reversing the file
sound(z,Fs);  // sound(2*y,Fs); can be used for better quality
t=(0:length(z)-1)*1/Fs;
plot(t,z)
title('Reversed file')
xlabel('Time in seconds')
ylabel('Amplitude')
```

Experiment: 2

Time Domain Parameters of Speech Signal.

Scilab code Solution 2.1 Short Time Energy of Speech signal

```
// This program computes the short time energy of a signal using
// Rectangular and Hamming window
//OS: windows 7
//Scilab: 5.5.2
//
clc;
close;
clear;
[y,Fs]=wavread('C:\Users\ACER\Desktop\Two.wav');
t=(0:length(y)-1)/Fs; // Changing x axixs in to time in seconds
subplot(311)
plot(t,y)
xlabel('Time in seconds')
title('Original signal')
frame_durn = 0.02;    // Duration of frame in seconds
```

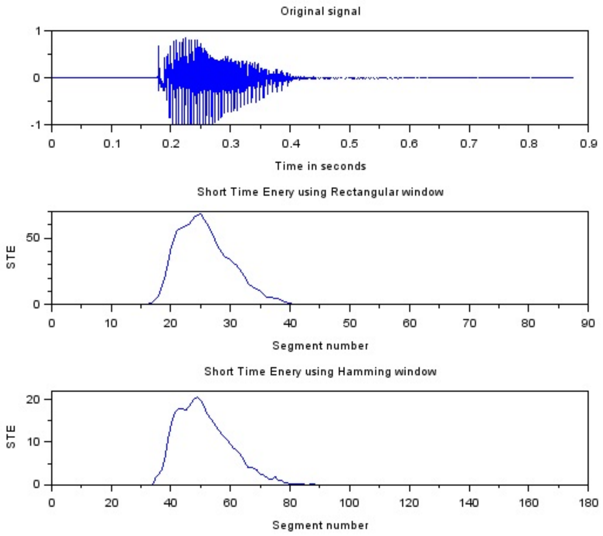
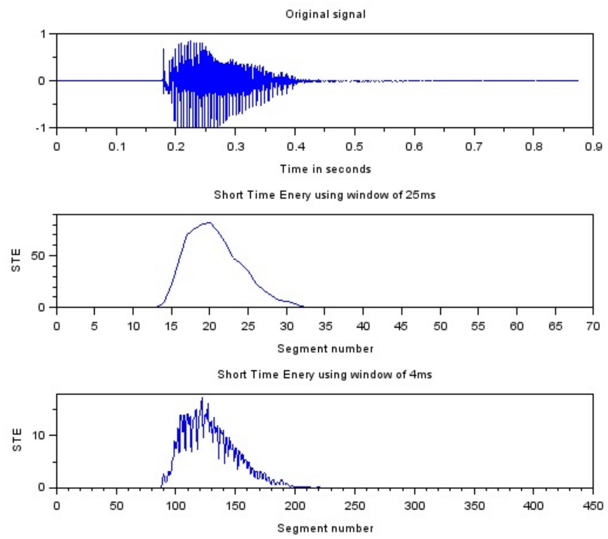


Figure 2.1: Short Time Energy of Speech signal

```
frame_len=Fs*frame_durn; // Length of frame
Rect_window=window('re',frame_len);

frame_shift=Fs*frame_durn/2; // 50% overlapping for Rectangular window
no_of_frames=floor(length(y)/frame_shift); // Computing total number of frames

frame_start=1;
frame_end=frame_start+frame_len-1;
for j=1:no_of_frames
    En(j)=0;
    if frame_end>=length(y) then    // Adustment for the last segment
        frame_end=length(y)
        frame_len=frame_end-frame_start;
    end


    segment=y(frame_start:frame_end);
    for k=1:frame_len
        En(j)=En(j)+(segment(k)*Rect_window(k))^2 // Computing the STE
    end
    frame_start=frame_start+frame_shift;
    frame_end=frame_end+frame_shift;

end



subplot(312)
plot(En)
title('Short Time Enery using Rectangular window')
xlabel('Segment number')
ylabel('STE')

////////////////////////////////////////////////

// For Hamming window
Ham_window=window('hm',frame_len);

frame_shift=Fs*frame_durn/4; // 75% overlapping for Hamming window
no_of_frames=floor(length(y)/frame_shift);

frame_start=1;
frame_end=frame_start+frame_len-1;
for j=1:no_of_frames
    En1(j)=0;
    if frame_end>=length(y) then
        frame_end=length(y)
        frame_len=frame_end-frame_start;
    end

    segment=y(frame_start:frame_end);
    for k=1:frame_len
        En1(j)=En1(j)+(segment(k)*Ham_window(k))^2
```



Original signal

Short Time Enery using window of 25ms

Short Time Enery using window of 4ms

Figure 2.2: Effect of length of window on STE

```
    end                              // Computing STE using
       Hamming window
    frame_start=frame_start+frame_shift;
    frame_end=frame_end+frame_shift;

end

subplot(313)
plot(En1)
title('Short Time Enery using Hamming window')
xlabel('Segment number')
ylabel('STE')
```

Scilab code Solution 2.2 Effect of length of window on STE

```
// This program shows the effect of window length on STE
//OS: windows 7
//Scilab: 5.5.2
//

clc;
close;
clear;
[y,Fs]=wavread('C:\Users\ACER\Desktop\Two.wav');
t=(0:length(y)-1)/Fs; // Changing x axixs in to time in seconds
subplot(311)
plot(t,y)
xlabel('Time in seconds')
title('Original signal')
frame_durn = 0.025;   // Duration of frame in seconds
frame_len=Fs*frame_durn; // Length of frame
Rect_window=window('re',frame_len);
frame_shift=Fs*frame_durn/2; // 50% overlapping for Rectangular window
no_of_frames=floor(length(y)/frame_shift); // Computing total number of frames
frame_start=1;
frame_end=frame_start+frame_len-1;
for j=1:no_of_frames
    En(j)=0;
    if frame_end>=length(y) then    // Adustment for the last segment
        frame_end=length(y)
        frame_len=frame_end-frame_start;
    end
    segment=y(frame_start:frame_end);
    for k=1:frame_len
        En(j)=En(j)+(segment(k)*Rect_window(k))^2    // Computing the STE
    end
    frame_start=frame_start+frame_shift;
    frame_end=frame_end+frame_shift;
end
subplot(312)
plot(En)
title('Short Time Enery using window of 25ms')
xlabel('Segment number')
ylabel('STE')
////////////////////
frame_durn = 0.004;    // Duration of frame in seconds
frame_len=Fs*frame_durn; // Length of frame
Rect_window=window('re',frame_len);
frame_shift=Fs*frame_durn/2; // 50% overlapping for Rectangular window
no_of_frames=floor(length(y)/frame_shift); // Computing total number of frames
frame_start=1;
frame_end=frame_start+frame_len-1;
for j=1:no_of_frames
    En(j)=0;
    if frame_end>=length(y) then     // Adustment for the last segment
        frame_end=length(y)
        frame_len=frame_end-frame_start;
    end
    segment=y(frame_start:frame_end);
    for k=1:frame_len
        En(j)=En(j)+(segment(k)*Rect_window(k))^2  // Computing the STE
    end
    frame_start=frame_start+frame_shift;
    frame_end=frame_end+frame_shift;
end
subplot(313)
plot(En)
title('Short Time Enery using window of 4ms')
xlabel('Segment number')
```

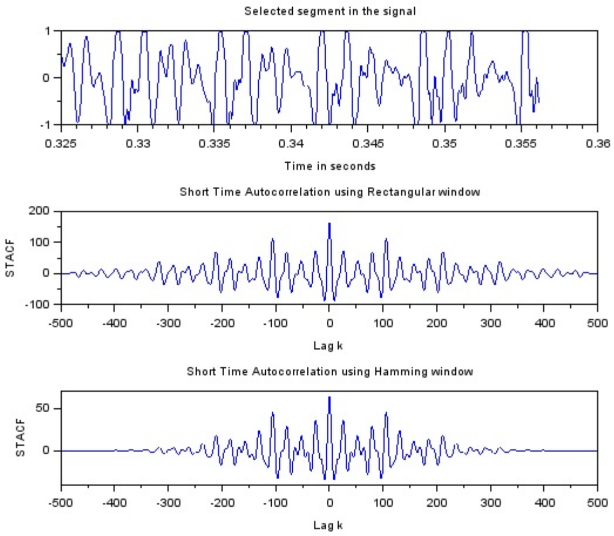


Figure 2.3: Comparing STACF using Hamming and Rectangular window

```
ylabel('STE')
```

Scilab code Solution 2.3 Comparing STACF using Hamming and Rectangular window

```
// Program to find Short Time Autocorrelation Function
// using Rectangular and Hamming winow
//OS: windows 7
//Scilab: 5.5.2
//

clc;
close;
clear;

```

```
[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');

beg=5200; //Sample at which window is to be applied
N=500; // Length of window in samples
x=y(beg:beg+N-1); // Obtaining the required segment
whm = window('hm', N);  // Hamming window
Ham_seg=x.*whm;  //Multiplying window with the signal

wr=window('re',N);  //Rectangular Window
Rect_seg=x.*wr;

t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
subplot(311)
plot(t,Rect_seg)
xlabel('Time in seconds')
title('Selected segment in the signal')
c1=xcorr(Rect_seg); // Autocorrelation
lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting x axix to get peak at lag=0
subplot(312)
plot(lag,c1)
title('Short Time Autocorrelation using Rectangular window')
xlabel('Lag k')
ylabel('STACF')
////////////
c2=xcorr(Ham_seg);
lag=(1:length(c1))-ceil(length(c2)/2);// Adjusting x axix to get peak at lag=0
subplot(313)
plot(lag,c2)
title('Short Time Autocorrelation using Hamming window')
xlabel('Lag k')
ylabel('STACF')
```

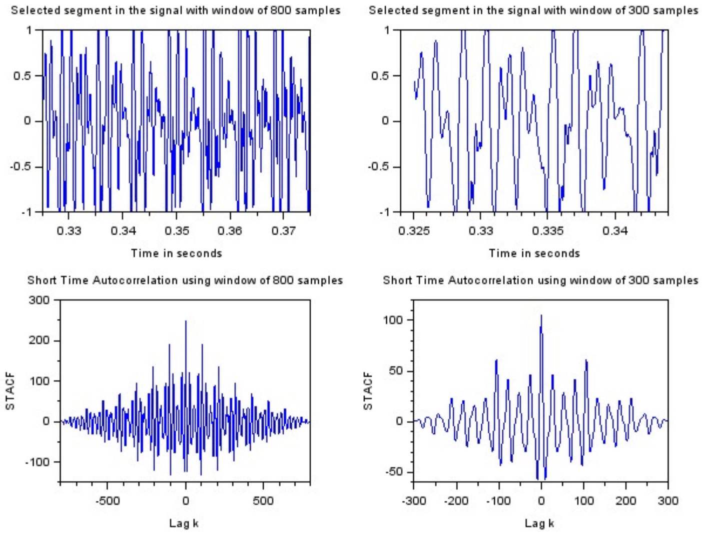


Figure 2.4: Effect of length of window on STACF

Scilab code Solution 2.4 Effect of length of window on STACF

```
// Program to find effect of length of window
//on Short Time Autocorrelation Function (STACF)
// using Rectangular window of different lengths
//OS: windows 7
//Scilab: 5.5.2
//

clc;
close;
clear;

[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');

```

```
beg=5200; //Sample at which window is to be applied
N=800; // Length of window in samples
x=y(beg:beg+N-1); // Obtaining the required segment
wr=window('re',N);  //Rectangular Window
Rect_seg=x.*wr;

t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
subplot(221)
plot(t,Rect_seg)
xlabel('Time in seconds')
title('Selected segment in the signal with window of 800 samples')
c1=xcorr(Rect_seg); // Autocorrelation
lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting x axix to get peak at lag=0
subplot(223)
plot(lag,c1)
title('Short Time Autocorrelation using window of 800 samples')
xlabel('Lag k')
ylabel('STACF')
////////////////////////////////
beg=5200; //Sample at which window is to be applied
N=300; // Length of window in samples
x=y(beg:beg+N-1); // Obtaining the required segment
wr=window('re',N);  //Rectangular Window
Rect_seg=x.*wr;

t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
subplot(222)
plot(t,Rect_seg)
xlabel('Time in seconds')
title('Selected segment in the signal with window of 300 samples')
c1=xcorr(Rect_seg); // Autocorrelation
lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting x axix to get peak at lag=0
subplot(224)
```

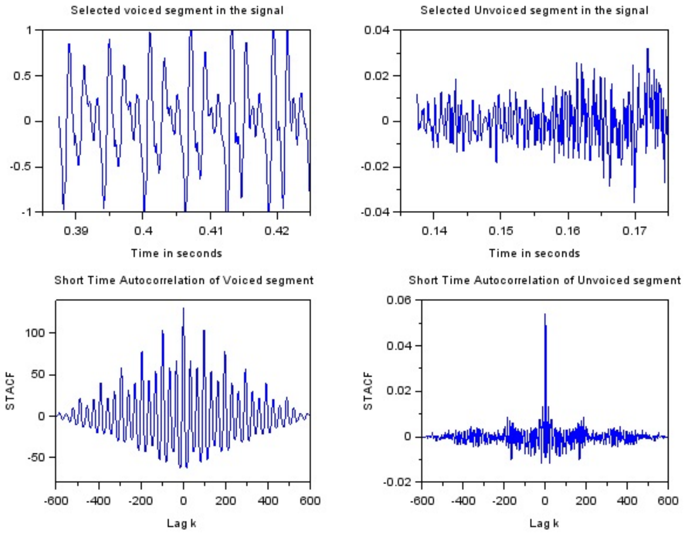


Figure 2.5: STACF for Voiced and Unvoiced speech

```
plot(lag,c1)
title('Short Time Autocorrelation using window of 300 samples')
xlabel('Lag k')
ylabel('STACF')
```

Scilab code Solution 2.5 STACF for Voiced and Unvoiced speech

```
// Program to compare Short Time Autocorrelation Functions for
// Voiced and un voiced regions in speech signal
//OS: windows 7
//Scilab: 5.5.2
//

clc;
```

```
close;
clear;

[y,Fs] = wavread('C:\Users\ACER\Desktop\Number4.wav');

beg=6200; //Sample at which window is to be applied
N=600; // Length of window in samples
x=y(beg:beg+N-1); // Obtaining the required segment
wr=window('re',N);  //Rectangular Window
Rect_seg=x.*wr;

t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
subplot(221)
plot(t,Rect_seg)
xlabel('Time in seconds')
title('Selected voiced segment in the signal')
c1=xcorr(Rect_seg); // Autocorrelation
lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting x axix to get peak at lag=0
subplot(223)
plot(lag,c1)
title('Short Time Autocorrelation of Voiced segment')
xlabel('Lag k')
ylabel('STACF')
////////////////////////

beg=2200; //Sample at which window is to be applied
N=600; // Length of window in samples
x=y(beg:beg+N-1); // Obtaining the required segment
wr=window('re',N);  //Rectangular Window
Rect_seg=x.*wr;

t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
subplot(222)
plot(t,Rect_seg)
xlabel('Time in seconds')
```

```
title('Selected Unvoiced segment in the signal')
c1=xcorr(Rect_seg); // Autocorrelation
lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting x axix to get peak at lag=0
subplot(224)
plot(lag,c1)
title('Short Time Autocorrelation of Unvoiced segment')
xlabel('Lag k')
ylabel('STACF')
```

Experiment: 3

Frequency Domain Analysis of Speech Signal.

Scilab code Solution 3.1 Plot STFT using Hamming and Rectangular window

```

//This Program plots the spectrum of a segment of a
    given signal obtained using  Rectangular and
    Hamming window.
//
//OS: windows 7
//Scilab: 5.5.2
//
clc;
close;
clear;

[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
t=(0:length(y)-1)/Fs;
subplot(311)
plot(t,y)
```

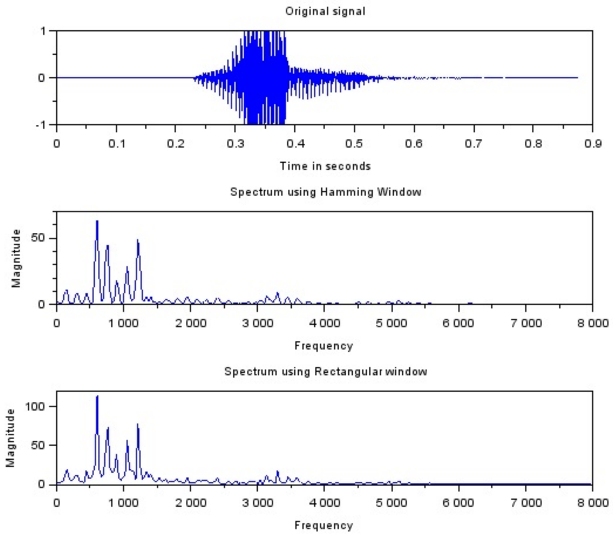


Figure 3.1: Plot STFT using Hamming and Rectangular window

```
xlabel('Time in seconds')
title('Original signal')
beg=5200; //Sample at which window is to be applied
N=500; // Length of window in samples
x=y(beg:beg+N-1);
whm = window('hm', N);
Ham_seg=x.*whm;  //Multiplying window with the signal

wr=window('re',N);  //Rectangular Window
Rect_seg=x.*wr;
f_Rect=abs(fft(Rect_seg)); // Spectrum of the segment using Rect window
f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment using Hamming window
l=length(f_Ham)/2;
f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to frequency
subplot(3,1,2)
plot(f,f_Ham(1:l))
```

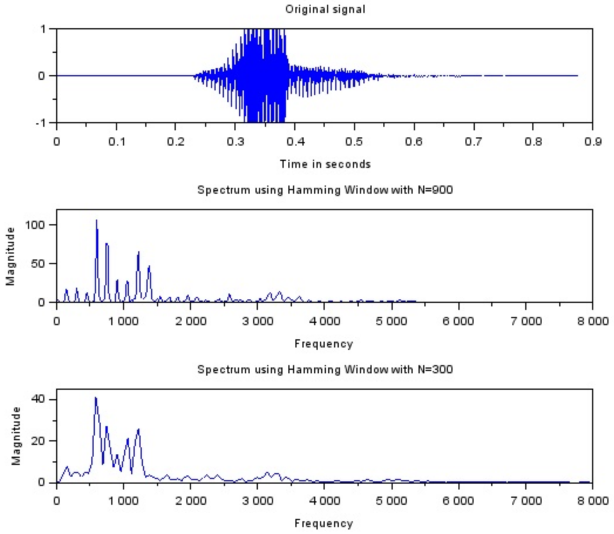


Figure 3.2: Effect of length of window on Short Time Spectrum

```
title('Spectrum using Hamming Window')
xlabel('Frequency')
ylabel('Magnitude')
subplot(313)
plot(f,f_Rect(1:l))
title('Spectrum using Rectangular window')
ylabel('Magnitude')
xlabel('Frequency')
```

Scilab code Solution 3.2 Effect of length of window on Short Time Spectrum

```
// Program to compare the effect of length of window on STFT
// Using Hamming window
//
```

```
//OS: windows 7
//Scilab: 5.5.2
//
clc;
close;
clear;

[y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
t=(0:length(y)-1)/Fs;
subplot(311)
plot(t,y)
xlabel('Time in seconds')
title('Original signal')
beg=5200; //Sample at which window is to be applied
N=900; // Length of window in samples
x=y(beg:beg+N-1);
whm = window('hm', N);
Ham_seg=x.*whm;  //Multiplying window with the signal
f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment using Hamming window
l=length(f_Ham)/2;
f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to frequency
subplot(3,1,2)
plot(f,f_Ham(1:l))
title('Spectrum using Hamming Window with N=900')
xlabel('Frequency')
ylabel('Magnitude')
////////////////////////////////
beg=5200; //Sample at which window is to be applied
N=300; // Length of window in samples
x=y(beg:beg+N-1);
whm = window('hm', N);
Ham_seg=x.*whm;  //Multiplying window with the signal
f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment using Hamming window
```

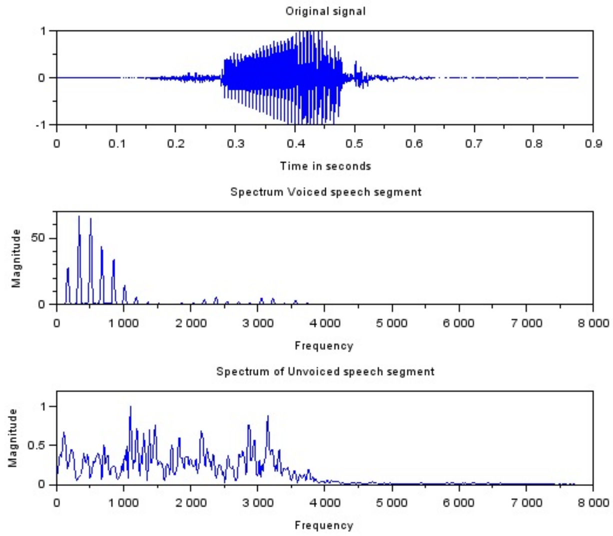


Figure 3.3: STFT of Voiced and Unvoiced speech

```
l=length(f_Ham)/2;
f=(0:(l-1))*Fs/(2*l);   // Adjusting x axis to frequency
subplot(3,1,3)
plot(f,f_Ham(1:l))
title('Spectrum using Hamming Window with N=300')
xlabel('Frequency')
ylabel('Magnitude')
```

Scilab code Solution 3.3 STFT of Voiced and Unvoiced speech

```
// Program to compare spectrum of Voiced and Unvoiced speech segments
// Using Hamming window
//
//OS: windows 7
```

```
//Scilab: 5.5.2
//
clc;
close;
clear;

[y,Fs] = wavread('C:\Users\ACER\Desktop\Number4.wav');
t=(0:length(y)-1)/Fs;
subplot(311)
plot(t,y)
xlabel('Time in seconds')
title('Original signal')

beg=5200; //Sample at which window is to be applied // Selecting Voiced segment
N=900; // Length of window in samples
x=y(beg:beg+N-1);
whm = window('hm', N);
Ham_seg=x.*whm;  //Multiplying window with the signal
f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment using Hamming window
l=length(f_Ham)/2;
f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to frequency
subplot(3,1,2)
plot(f,f_Ham(1:l))
title('Spectrum Voiced speech segment')
xlabel('Frequency')
ylabel('Magnitude')
/////////////////////////////////
beg=2200; //Selecting unvoiced segment
N=900; // Length of window in samples
x=y(beg:beg+N-1);
whm = window('hm', N);
Ham_seg=x.*whm;  //Multiplying window with the signal
```

```
f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment using Hamming window
l=length(f_Ham)/2;
f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to frequency
subplot(3,1,3)
plot(f,f_Ham(1:l))
title('Spectrum of Unvoiced speech segment')
xlabel('Frequency')
ylabel('Magnitude')
```