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

November 24, 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 Scilab Solutions 3

1 Basic operations on Audio signals. 5

2 Time Domain Parameters of Speech Signal. 8

3 Frequency Domain Analysis of Speech Signal. 21

List of Experiments

Solution 1.1	Program to Read and Play audio file	5
Solution 1.2	Effect of changing Sampling Frequency	6
Solution 1.3	Program to Reverse and play audio file	7
Solution 2.1	Short Time Energy of Speech signal	8
Solution 2.2	Effect of length of window on STE	11
Solution 2.3	Comparing STACF using Hamming and Rectangular window	14
Solution 2.4	Effect of length of window on STACF	16
Solution 2.5	STACF for Voiced and Unvoiced speech	18
Solution 3.1	Plot STFT using Hamming and Rectangular window	21
Solution 3.2	Effect of length of window on Short Time Spectrum	23
Solution 3.3	STFT of Voiced and Unvoiced speech	25

List of Figures

1.1 Program to Read and Play audio file 6

2.1 Short Time Energy of Speech signal 9
2.2 Effect of length of window on STE 11
2.3 Comparing STACF using Hamming and Rectangular window 14
2.4 Effect of length of window on STACF 16
2.5 STACF for Voiced and Unvoiced speech 18

3.1 Plot STFT using Hamming and Rectangular window 22
3.2 Effect of length of window on Short Time Spectrum 23
3.3 STFT of Voiced and Unvoiced speech 25

Experiment: 1

Basic operations on Audio signals.

Scilab code Solution 1.1 Program to Read and Play audio file

```
1 //This code Reads an audio file and plays it.
2 //
3 //OS: windows 7
4 //Scilab: 5.5.2
5 //
6 clc;
7 close;
8 clear;
9 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
    //read the audio file using the given path
10 sound(y,Fs);  // sound(2*y,Fs); can be used for
    better quality
11 t=(0:length(y)-1)*1/Fs;
12 plot(t,y)
13 title('Speech signal waveform')
14 xlabel('Time in seconds')
15 ylabel('Amplitude')
```

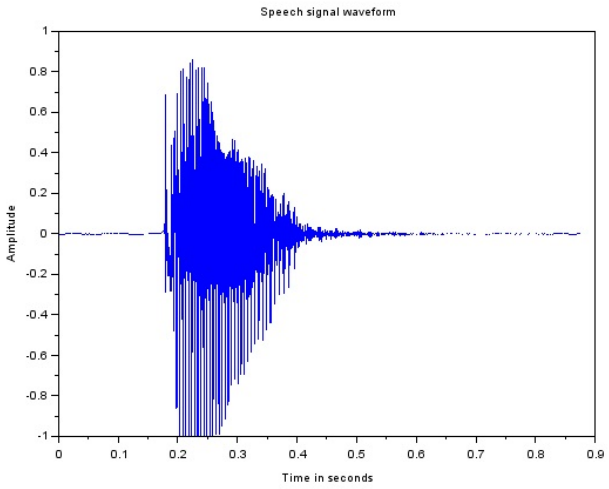

Figure 1.1: Program to Read and Play audio file

Scilab code Solution 1.2 Effect of changing Sampling Frequency

```
1 //This code Reads an audio file and plays it by
    changing the sampling frequency.
2 //
3 //OS: windows 7
4 //Scilab: 5.5.2
5 //
6 clc;
7 close;
8 clear;
9 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
    //read the audio file using the given path
10 Fs_new=Fs/.8;  // increasing Fs. Can be reduced by
    using command  Fs_new=Fs/1.3;
11 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

```
1 //This code Reads an audio file and plays it by
     reversing the file.
2 //
3 //OS: windows 7
4 //Scilab: 5.5.2
5 //
6 clc;
7 close;
8 clear;
9 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
     // Give the path
10 z = y(:,$:-1:1); //Reversing the file
11 sound(z,Fs);  // sound(2*y,Fs); can be used for
     better quality
12 t=(0:length(z)-1)*1/Fs;
13 plot(t,z)
14 title('Reversed file')
15 xlabel('Time in seconds')
16 ylabel('Amplitude')
```

Experiment: 2

Time Domain Parameters of Speech Signal.

Scilab code Solution 2.1 Short Time Energy of Speech signal

```
1 // This program computes the short time energy of a
     signal using
2 // Rectangular and Hamming window
3 //OS: windows 7
4 //Scilab: 5.5.2
5 //
6 clc;
7 close;
8 clear;
9 [y,Fs]=wavread('C:\Users\ACER\Desktop\Two.wav');
10 t=(0:length(y)-1)/Fs; // Changing x axixs in to time
     in seconds
11 subplot(311)
12 plot(t,y)
13 xlabel('Time in seconds')
14 title('Original signal')
15 frame_durn = 0.02;   // Duration of frame in seconds
```

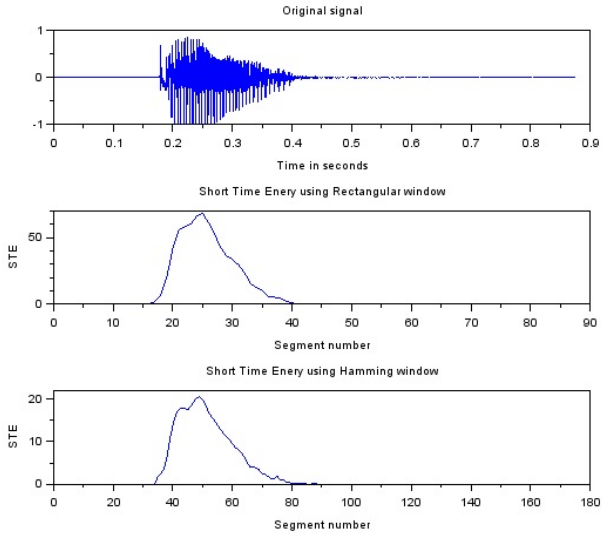

Figure 2.1: Short Time Energy of Speech signal

```
16 frame_len=Fs*frame_durn; // Length of frame
17 Rect_window=window('re',frame_len);
18
19 frame_shift=Fs*frame_durn/2; // 50% overlapping for
      Rectangular window
20 no_of_frames=floor(length(y)/frame_shift); //
      Computing total number of frames
21
22 frame_start=1;
23 frame_end=frame_start+frame_len-1;
24 for j=1:no_of_frames
25     En(j)=0;
26     if frame_end>=length(y) then     // Adustment for
           the last segment
27         frame_end=length(y)
28         frame_len=frame_end-frame_start;
29     end
30
31
32     segment=y(frame_start:frame_end);
```

```
33      for k=1:frame_len
34          En(j)=En(j)+(segment(k)*Rect_window(k))^2
                // Computing the STE
35      end
36      frame_start=frame_start+frame_shift;
37      frame_end=frame_end+frame_shift;
38
39  end
40
41
42
43  subplot(312)
44  plot(En)
45  title('Short Time Enery using Rectangular window')
46  xlabel('Segment number')
47  ylabel('STE')
48
49  ////////////////////////////////////////////////
50
51  // For Hamming window
52  Ham_window=window('hm',frame_len);
53
54  frame_shift=Fs*frame_durn/4; // 75% overlapping for
       Hamming window
55  no_of_frames=floor(length(y)/frame_shift);
56
57  frame_start=1;
58  frame_end=frame_start+frame_len-1;
59  for j=1:no_of_frames
60      En1(j)=0;
61      if frame_end>=length(y) then
62          frame_end=length(y)
63          frame_len=frame_end-frame_start;
64      end
65
66      segment=y(frame_start:frame_end);
67      for k=1:frame_len
68          En1(j)=En1(j)+(segment(k)*Ham_window(k))^2
```

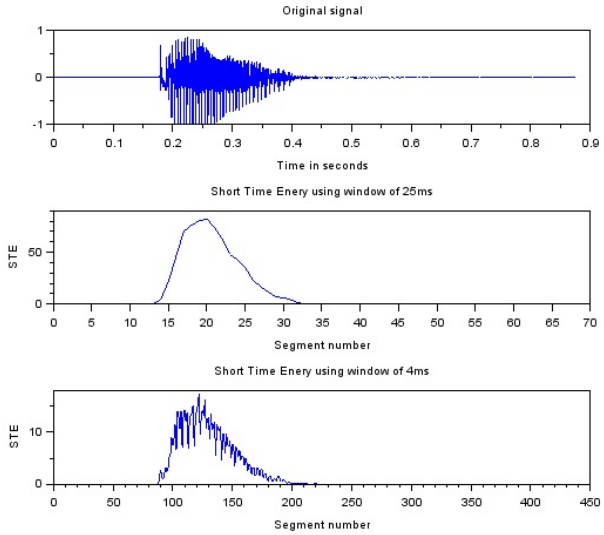

Figure 2.2: Effect of length of window on STE

```
69      end                              // Computing STE using
           Hamming window
70      frame_start=frame_start+frame_shift;
71      frame_end=frame_end+frame_shift;
72
73  end
74
75  subplot(313)
76  plot(En1)
77  title('Short Time Enery using Hamming window')
78  xlabel('Segment number')
79  ylabel('STE')
```

Scilab code Solution 2.2 Effect of length of window on STE

```
1  // This program shows the effect of window length on
       STE
2  //OS: windows 7
3  //Scilab: 5.5.2
4  //
5
6  clc;
7  close;
8  clear;
9  [y,Fs]=wavread('C:\Users\ACER\Desktop\Two.wav');
10 t=(0:length(y)-1)/Fs; // Changing x axixs in to time
       in seconds
11 subplot(311)
12 plot(t,y)
13 xlabel('Time in seconds')
14 title('Original signal')
15 frame_durn = 0.025;   // Duration of frame in
     seconds
16 frame_len=Fs*frame_durn; // Length of frame
17 Rect_window=window('re',frame_len);
18 frame_shift=Fs*frame_durn/2; // 50% overlapping for
     Rectangular window
19 no_of_frames=floor(length(y)/frame_shift); //
     Computing total number of frames
20 frame_start=1;
21 frame_end=frame_start+frame_len-1;
22 for j=1:no_of_frames
23     En(j)=0;
24     if frame_end>=length(y) then    // Adustment for
           the last segment
25         frame_end=length(y)
26         frame_len=frame_end-frame_start;
27     end
28     segment=y(frame_start:frame_end);
29     for k=1:frame_len
30         En(j)=En(j)+(segment(k)*Rect_window(k))^2
               // Computing the STE
31     end
```

```
32      frame_start=frame_start+frame_shift;
33      frame_end=frame_end+frame_shift;
34  end
35  subplot(312)
36  plot(En)
37  title('Short Time Enery using window of 25ms')
38  xlabel('Segment number')
39  ylabel('STE')
40  ////////////////////
41  frame_durn = 0.004;    // Duration of frame in
        seconds
42  frame_len=Fs*frame_durn; // Length of frame
43  Rect_window=window('re',frame_len);
44  frame_shift=Fs*frame_durn/2; // 50% overlapping for
        Rectangular window
45  no_of_frames=floor(length(y)/frame_shift); //
        Computing total number of frames
46  frame_start=1;
47  frame_end=frame_start+frame_len-1;
48  for j=1:no_of_frames
49      En(j)=0;
50      if frame_end>=length(y) then     // Adustment for
            the last segment
51          frame_end=length(y)
52          frame_len=frame_end-frame_start;
53      end
54      segment=y(frame_start:frame_end);
55      for k=1:frame_len
56          En(j)=En(j)+(segment(k)*Rect_window(k))^2
                // Computing the STE
57      end
58      frame_start=frame_start+frame_shift;
59      frame_end=frame_end+frame_shift;
60  end
61  subplot(313)
62  plot(En)
63  title('Short Time Enery using window of 4ms')
64  xlabel('Segment number')
```

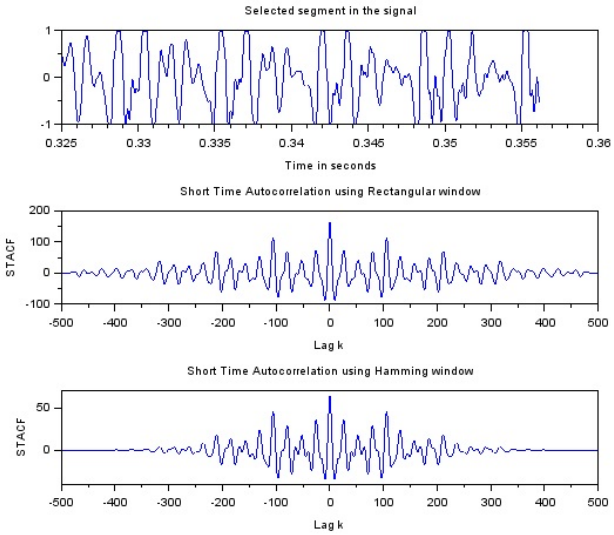

Figure 2.3: Comparing STACF using Hamming and Rectangular window

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

Scilab code Solution 2.3 Comparing STACF using Hamming and Rectangular window

```
1 // Program to find Short Time Autocorrelation
    Function
2 // using Rectangular and Hamming winow
3 //OS: windows 7
4 //Scilab: 5.5.2
5 //
6
7 clc;
8 close;
9 clear;
10
```

```
11 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
12
13 beg=5200; //Sample at which window is to be applied
14 N=500; // Length of window in samples
15 x=y(beg:beg+N-1); // Obtaining the required segment
16 whm = window('hm', N);  // Hamming window
17 Ham_seg=x.*whm;  //Multiplying window with the
      signal
18
19 wr=window('re',N);  //Rectangular Window
20 Rect_seg=x.*wr;
21
22 t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
23 subplot(311)
24 plot(t,Rect_seg)
25 xlabel('Time in seconds')
26 title('Selected segment in the signal')
27 c1=xcorr(Rect_seg); // Autocorrelation
28 lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting
      x axix to get peak at lag=0
29 subplot(312)
30 plot(lag,c1)
31 title('Short Time Autocorrelation using Rectangular
      window')
32 xlabel('Lag k')
33 ylabel('STACF')
34 ////////////
35 c2=xcorr(Ham_seg);
36 lag=(1:length(c1))-ceil(length(c2)/2);// Adjusting x
       axix to get peak at lag=0
37 subplot(313)
38 plot(lag,c2)
39 title('Short Time Autocorrelation using Hamming
      window')
40 xlabel('Lag k')
41 ylabel('STACF')
```

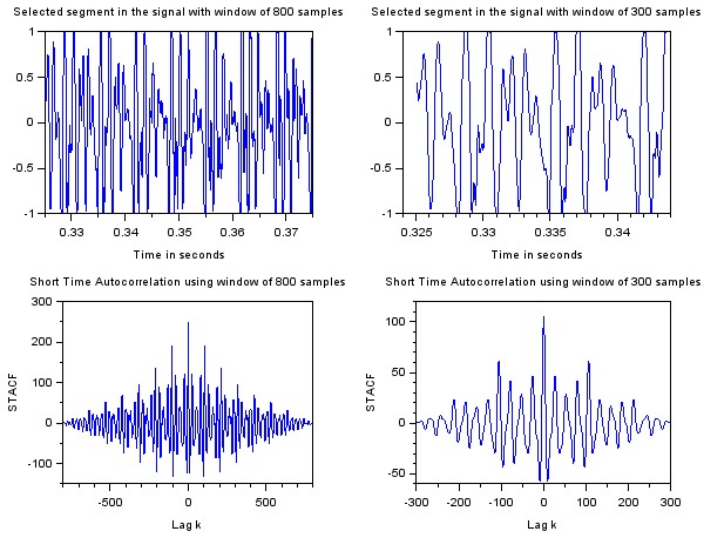

Figure 2.4: Effect of length of window on STACF

Scilab code Solution 2.4 Effect of length of window on STACF

```
1  // Program to find effect of length of window
2  //on Short Time Autocorrelation Function (STACF)
3  // using Rectangular window of different lengths
4  //OS: windows 7
5  //Scilab: 5.5.2
6  //
7
8  clc;
9  close;
10 clear;
11
12 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
13
```

```
14 beg=5200; //Sample at which window is to be applied
15 N=800; // Length of window in samples
16 x=y(beg:beg+N-1); // Obtaining the required segment
17 wr=window('re',N);  //Rectangular Window
18 Rect_seg=x.*wr;
19
20 t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
21 subplot(221)
22 plot(t,Rect_seg)
23 xlabel('Time in seconds')
24 title('Selected segment in the signal with window of
       800 samples')
25 c1=xcorr(Rect_seg); // Autocorrelation
26 lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting
     x axix to get peak at lag=0
27 subplot(223)
28 plot(lag,c1)
29 title('Short Time Autocorrelation using window of
     800 samples')
30 xlabel('Lag k')
31 ylabel('STACF')
32 ////////////////////////////////
33 beg=5200; //Sample at which window is to be applied
34 N=300; // Length of window in samples
35 x=y(beg:beg+N-1); // Obtaining the required segment
36 wr=window('re',N);  //Rectangular Window
37 Rect_seg=x.*wr;
38
39 t=(beg:beg+N-1)/Fs;  // Adjusting the x axix to time
40 subplot(222)
41 plot(t,Rect_seg)
42 xlabel('Time in seconds')
43 title('Selected segment in the signal with window of
       300 samples')
44 c1=xcorr(Rect_seg); // Autocorrelation
45 lag=(1:length(c1))-ceil(length(c1)/2); // Adjusting
     x axix to get peak at lag=0
46 subplot(224)
```

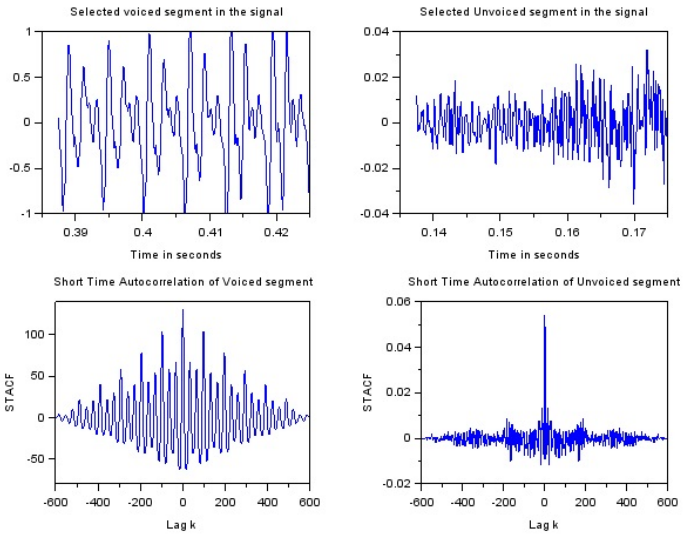

Figure 2.5: STACF for Voiced and Unvoiced speech

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

Scilab code Solution 2.5 STACF for Voiced and Unvoiced speech

```
1 // Program to compare Short Time Autocorrelation
    Functions for
2 // Voiced and un voiced regions in speech signal
3 //OS: windows 7
4 //Scilab: 5.5.2
5 //
6
7 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')
```

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

Experiment: 3

Frequency Domain Analysis of Speech Signal.

Scilab code Solution 3.1 Plot STFT using Hamming and Rectangular window

```
1
2 //This Program plots the spectrum of a segment of a
     given signal obtained using  Rectangular and
     Hamming window.
3 //
4 //OS: windows 7
5 //Scilab: 5.5.2
6 //
7 clc;
8 close;
9 clear;
10
11 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
12 t=(0:length(y)-1)/Fs;
13 subplot(311)
14 plot(t,y)
```

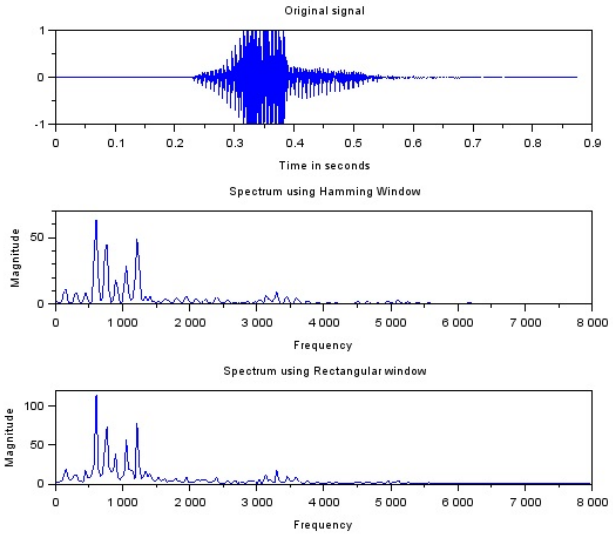

Figure 3.1: Plot STFT using Hamming and Rectangular window

```
15 xlabel('Time in seconds')
16 title('Original signal')
17 beg=5200; //Sample at which window is to be applied
18 N=500; // Length of window in samples
19 x=y(beg:beg+N-1);
20 whm = window('hm', N);
21 Ham_seg=x.*whm;  //Multiplying window with the
      signal
22
23 wr=window('re',N);  //Rectangular Window
24 Rect_seg=x.*wr;
25 f_Rect=abs(fft(Rect_seg)); // Spectrum of the
      segment using Rect window
26 f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment
      using Hamming window
27 l=length(f_Ham)/2;
28 f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to
      frequency
29 subplot(3,1,2)
30 plot(f,f_Ham(1:l))
```

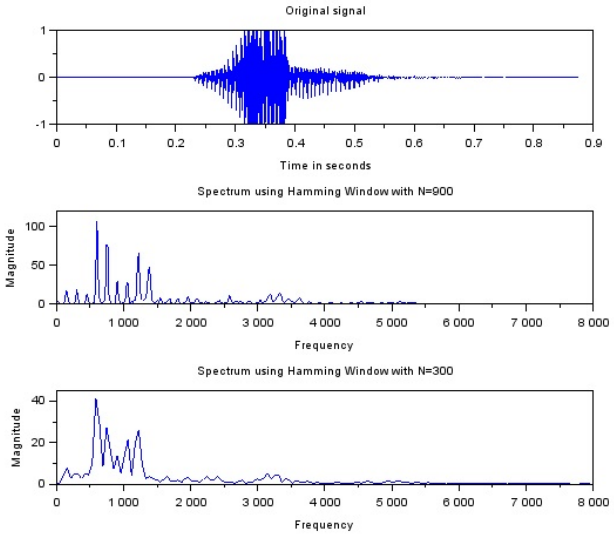

Figure 3.2: Effect of length of window on Short Time Spectrum

```
31 title('Spectrum using Hamming Window')
32 xlabel('Frequency')
33 ylabel('Magnitude')
34 subplot(313)
35 plot(f,f_Rect(1:l))
36 title('Spectrum using Rectangular window')
37 ylabel('Magnitude')
38 xlabel('Frequency')
```

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

```
1 // Program to compare the effect of length of window
    on STFT
2 // Using Hamming window
3 //
```

```
4  //OS: windows 7
5  //Scilab: 5.5.2
6  //
7  clc;
8  close;
9  clear;
10
11 [y,Fs] = wavread('C:\Users\ACER\Desktop\Two.wav');
12 t=(0:length(y)-1)/Fs;
13 subplot(311)
14 plot(t,y)
15 xlabel('Time in seconds')
16 title('Original signal')
17 beg=5200; //Sample at which window is to be applied
18 N=900; // Length of window in samples
19 x=y(beg:beg+N-1);
20 whm = window('hm', N);
21 Ham_seg=x.*whm;  //Multiplying window with the
      signal
22 f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment
      using Hamming window
23 l=length(f_Ham)/2;
24 f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to
      frequency
25 subplot(3,1,2)
26 plot(f,f_Ham(1:l))
27 title('Spectrum using Hamming Window with N=900')
28 xlabel('Frequency')
29 ylabel('Magnitude')
30 ////////////////////////////////
31 beg=5200; //Sample at which window is to be applied
32 N=300; // Length of window in samples
33 x=y(beg:beg+N-1);
34 whm = window('hm', N);
35 Ham_seg=x.*whm;  //Multiplying window with the
      signal
36 f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment
      using Hamming window
```

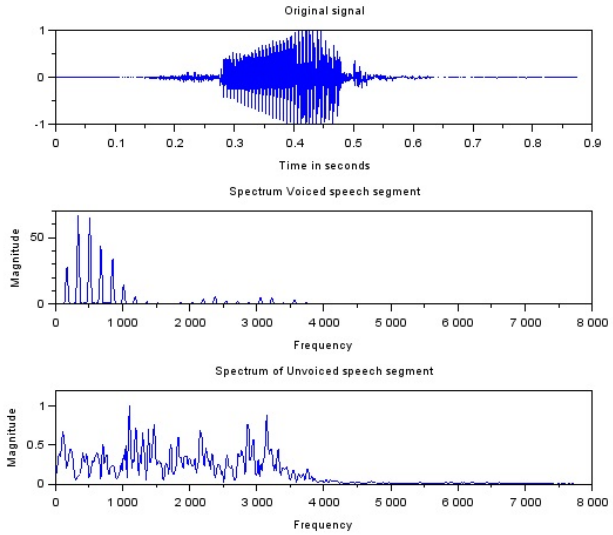

Figure 3.3: STFT of Voiced and Unvoiced speech

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

Scilab code Solution 3.3 STFT of Voiced and Unvoiced speech

```
1 // Program to compare spectrum of Voiced and
     Unvoiced speech segments
2 // Using Hamming window
3 //
4 //OS: windows 7
```

```
5  //Scilab: 5.5.2
6  //
7  clc;
8  close;
9  clear;
10
11 [y,Fs] = wavread('C:\Users\ACER\Desktop\Number4.wav'
       );
12 t=(0:length(y)-1)/Fs;
13 subplot(311)
14 plot(t,y)
15 xlabel('Time in seconds')
16 title('Original signal')
17
18 beg=5200; //Sample at which window is to be applied
       // Selecting Voiced segment
19 N=900; // Length of window in samples
20 x=y(beg:beg+N-1);
21 whm = window('hm', N);
22 Ham_seg=x.*whm;  //Multiplying window with the
       signal
23 f_Ham=abs(fft(Ham_seg));  //Spectrum of the segment
       using Hamming window
24 l=length(f_Ham)/2;
25 f=(0:(l-1))*Fs/(2*l);  // Adjusting x axis to
       frequency
26 subplot(3,1,2)
27 plot(f,f_Ham(1:l))
28 title('Spectrum Voiced speech segment')
29 xlabel('Frequency')
30 ylabel('Magnitude')
31 ////////////////////////////////
32 beg=2200; //Selecting unvoiced segment
33 N=900; // Length of window in samples
34 x=y(beg:beg+N-1);
35 whm = window('hm', N);
36 Ham_seg=x.*whm;  //Multiplying window with the
       signal
```

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