2017-12-10

Interference Unmixing and Estimation Technique for Improvement of Speech Separation Performance

Ankur Maurya
Technological University Dublin, ankurchandramaurya@gmail.com

Recommended Citation

Maurya, A. (2017) *Interference Unmixing and Estimation Technique for Improvement of Speech Separation Performance*. Masters thesis, DIT, 2017. doi:10.21427/rbhj-g584

INTERFERENCE UNMIXING AND ESTIMATION TECHNIQUE FOR IMPROVEMENT OF SPEECH SEPARATION PERFORMANCE.

A thesis is submitted in partial fulfilment of the requirements of the Masters of Science Degree in Electronic and Communications Engineering (DT086) of the Dublin Institute of Technology.

by

Ankur Chandra Maurya

Supervised by

Dr. Ruairí de Fréin

School of Electrical and Electronic Engineering

College of Engineering and Built Environment

Dublin Institute of Technology

December 2017

ABSTRACT

Presence of noise in the speech can sometimes become annoying as it can lead to loss of important data or create misunderstandings between the communications area which can lead to major problems associated to loss of time and money. This thesis focuses to filter out noise form a speech signal which is simulated in Matlab/Octave software while making a comparison between temporal resolution of signal with respect to the spectral resolution of the signal in which the parameters such as the size of window length are varied in order to obtain the best speech separation performance. To get the best spectral and temporal resolution with respect to the window length in order to find out the presence of speech sound in the mixture or how strongly the mixture is dominated by the noisy signal. The reconstructed signal is the original speech sound which was applied at the input. To study the relationship between window-disjoint orthogonality and window length and to get the best separation performance.

DECLARATION

I certify that this thesis which I now submit for examination for the award of Master of Science is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the Institute's guidelines for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Ankur Chandra Maurya

Signature ________________________________ Date ____________________________

ACKNOWLEDGEMENTS

This project would not have been possible without the kind support and help of many individuals. I wish to express my deepest gratitude to Dr. Ruairí de Fréin for his precious time, availability and dedicated efforts in providing guidance and inputs throughout the project. Also, I would like to express my special gratitude and thanks to our head of the department Dr. Michael Conlon and our project coordinator Dr. Yuliya Semenova for the guidance and moral support. I would also like to thank my mother Mrs. Laxmi Chandra Maurya, my father Dr. Chandra B. Maurya, my sister Miss Apoorva Chandra Maurya for providing me moral support and financial assistance in form of education loan, finally my friends for their love and help during my project work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Problem statement

"This thesis aims to study the effect of varying window length on the separation performance of the signal in order to get the best resolution."

1.2 Signals of interest

In this experiment the test signals of sampling rate at 16 KHz were used, the reason why we do this is because, maximum frequency of the signal is 8 KHz and we always sample at a rate of twice the maximum frequency in order to avoid aliasing, also the nyquist criterion states that if you have a signal with maximum frequency component at "f" hertz, you need to sample it with at least "2f" hertz, which then becomes easy to reconstruct the signal [29]. Also we cannot sample at a higher rate or lower rate because, at higher rates it requires a large memory to store the discrete values of the signal and at a lower rate it is not possible to reconstruct the signal [30]. A table below shows us the various ranges of sampling rate for various purposes in the sound industry [31] [32].

Table 1 : Sampling Rates.

Sample Rate (kHz)	**Areas of application**
44.1	CD
48	DVD
32	Cassettes
22.05	AM Radio
8	Telephone
Custom rate(16kHz)	Experimental purposes

We have produced two signals of 16 kHz in a .wav format from a special software known as Audacity @ http://www.audacityteam.org/

The first signal is the speech signal 'a', which is plotted at in octave software by use of following command:

[a,fs] = audioread ('C:\Users\Ankur\Desktop\MastersThesis\speechsound.wav');

This code gives back the sampling rate and generates a sampled vector value in matrix a. Plot is shown below for the signal in time domain.

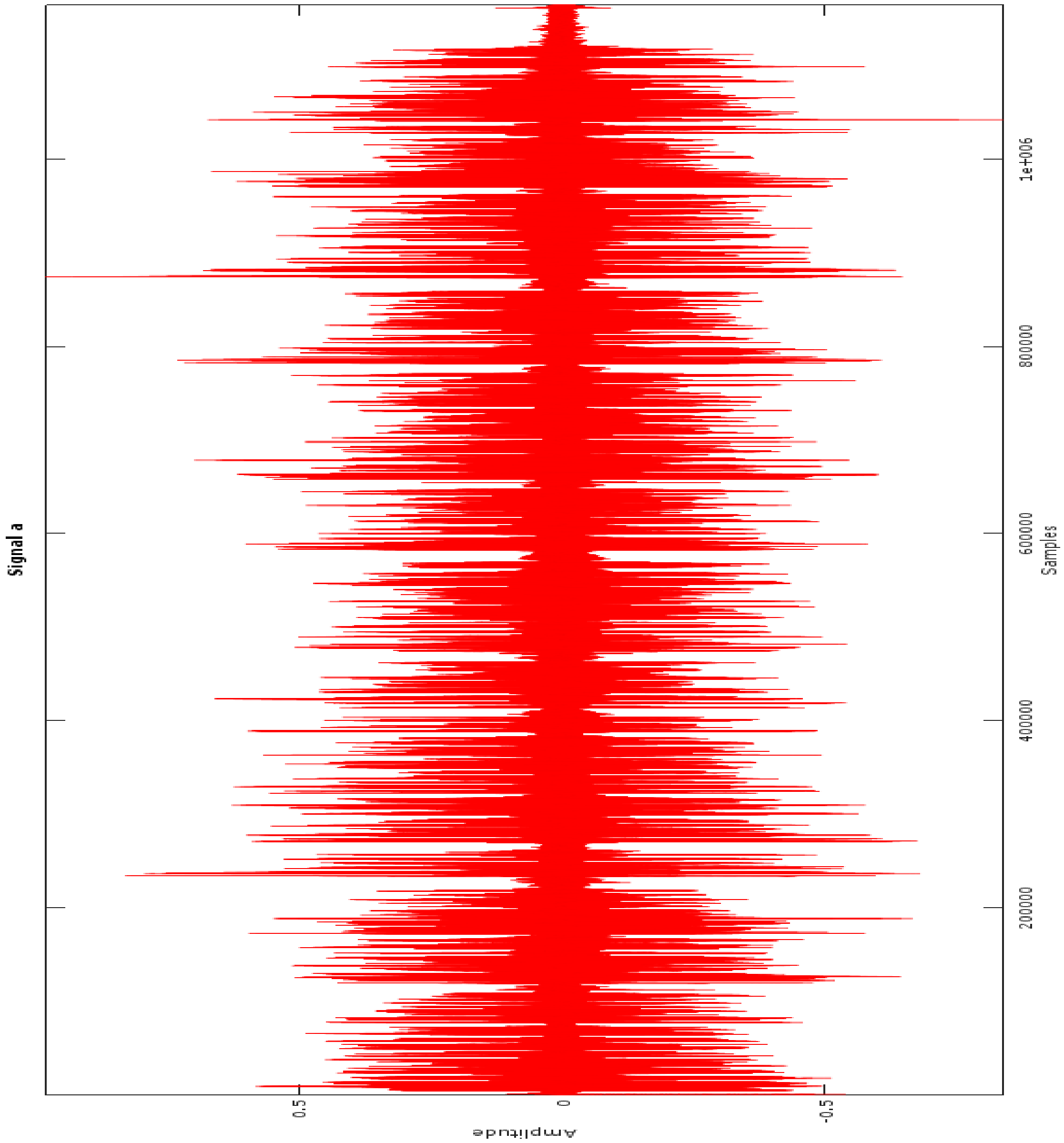


Figure 1: Sampled Speech signal a.

Similarly, the second signal is the speech signal ‘a’, which is plotted at in octave software by use of following command:

[b,fs] = audioread ('C:\Users\Ankur\Desktop\MastersThesis\clicksound.wav');

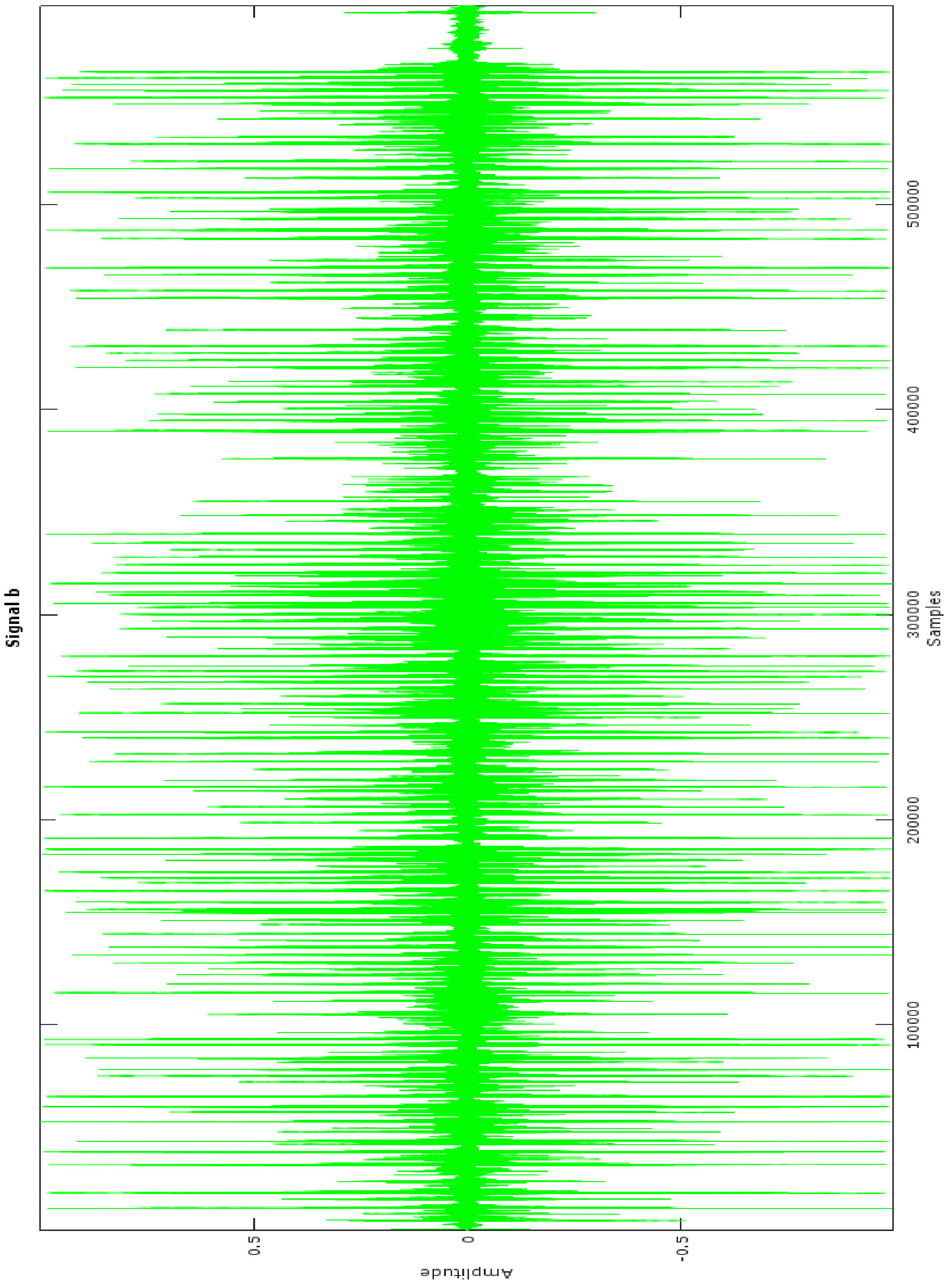


Figure 2: Sampled Click signal b

Finally we have plotted the mixture of the signal as signal x by zero padding and adding the signal b to the signal a.

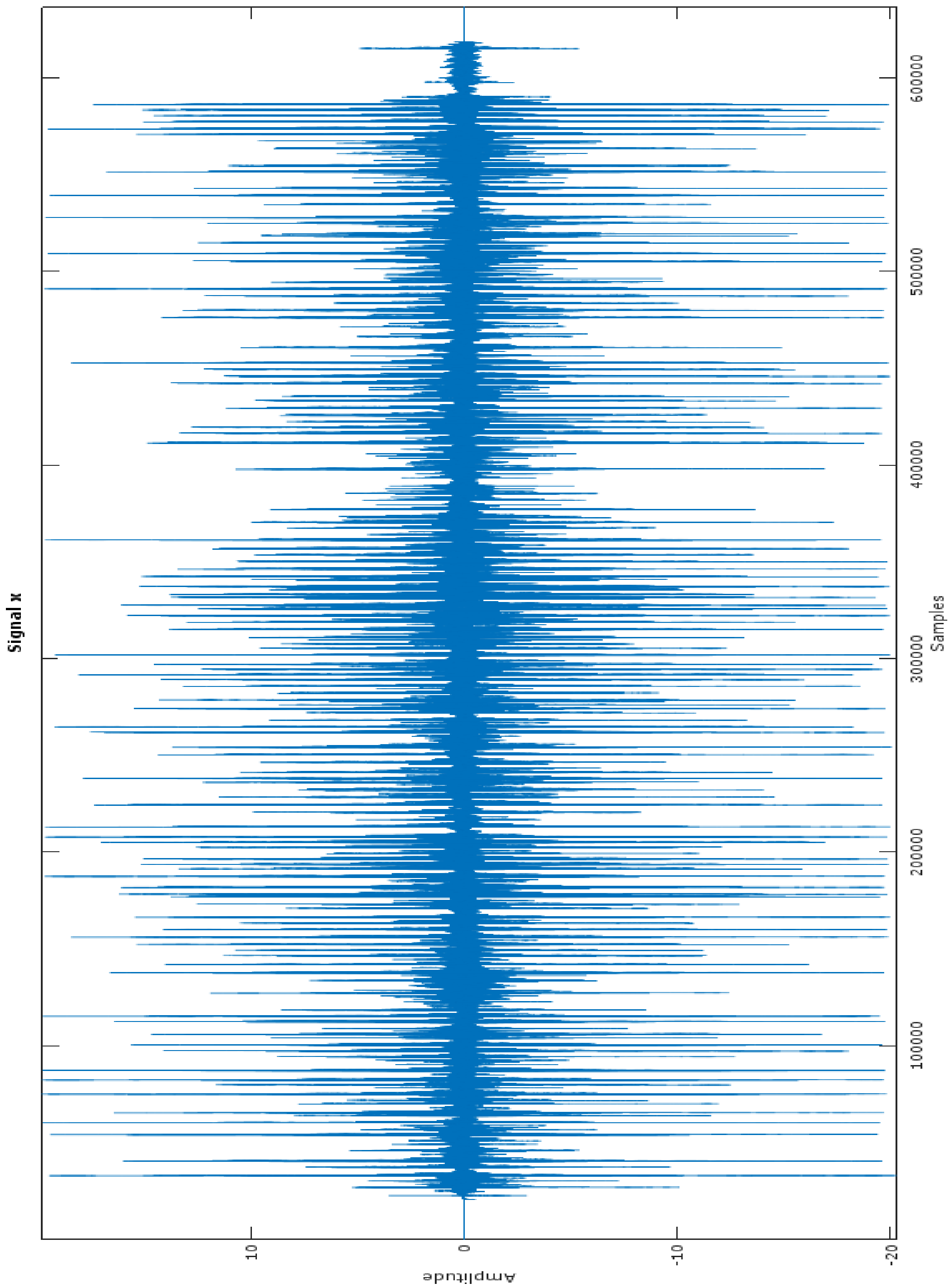


Figure 3: Sampled mixture signal x.

1.3List of contributions

This thesis contributes in working towards, eliminating the noise present in an audio signal as a mixture with the help of simulation software known as Matlab. It gives the best representation of the signal by carrying a trade-off between a time and frequency resolution and reconstructing the input signal via Binary Masking, Inverse Short Time Fourier Transform (IFFT) with a special focus on the evaluation of maximum separation performance with respect to the varying window lengths [33]. It also focuses on effect of Window Disjoint Orthogonality (WDO) in order to facilitate the separation and regeneration of original signal in mixture signals, in other words we determine the degree of overlapping in the mixing parameters with respect to the multi-dimensional time and frequency representation [34].

1.4Literature review at high level Ieee explore

Many speech enhancement techniques have been used to reduce the noise from the noisy speech such as spectral subtraction method [3] which is the oldest one for reducing the additive background noise then other method is noise cancellation using adaptive filters. Numerical experiments show the Short Time Fourier Transform (STFT) improves instantaneous subsample relative parameter estimation in low noise conditions and achieves good synthesis [27-30] But in practice these methods are improving day by day in terms of estimation parameter such as synchronization (scaling, sampling rate, number of signals, etc...) , time-frequency masking parameters (window length, number of FFT points, degree of overlapping, etc...) these parameters are very beneficial for improvement in separation performance. However, we have often to trade-off between the areas such as the resolution, bandwidth, SNR performance and a few more. As such no method can be absolutely perfect [1-33].

CHAPTER 2 THEORY

2.1 Statement of hypothesis

In this experiment we propose to create an algorithm for noise removal using the time-frequency analysis using the Short Time Fourier Transform (STFT) followed by application of binary masking for differentiating the input signal (Speech Sound) from noise signal (Click Sound), which will be beneficial for reconstruction of the input signal (Speech sound) by applying the Inverse Fast Fourier Transform (IFFT) to obtain the original signal (Speech Sound) , also improving separation performance by changing the analysis window [29].

2.2 Literature review

A signal represented in time domain gives us the information about the magnitudes of the signal with respect to the time which is known as temporal resolution of that signal. However, if we need to find out the frequency content of the signal it becomes important to study the signal into a domain which gives the value of the repeating frequencies with corresponding magnitudes. Hence we require to take Fourier Transform of the signal. The values generated by the Fourier Transform are the magnitudes and phase corresponding to the repeating frequency in that signal. We have to note that Fourier transform is lossless and invertible, means, that original signal can be perfectly reconstructed by taking the Inverse Fast Fourier Transform (IFFT) [6] [7] . Now as we know that Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) are applied to entire signal[9] .Thus, we are unable to resolve the spectral changes with time, so we can divide the signal into “chunks”, and apply the DFT/FFT to each one of them. This strategy is known as the Short-Time Fourier Transform (STFT), and the resulting time-frequency representation is known as a spectrogram [1] [2]. When we plot spectrogram, we can see the frequency spectrum of the signal x with respect to time as shown in Fig 4.

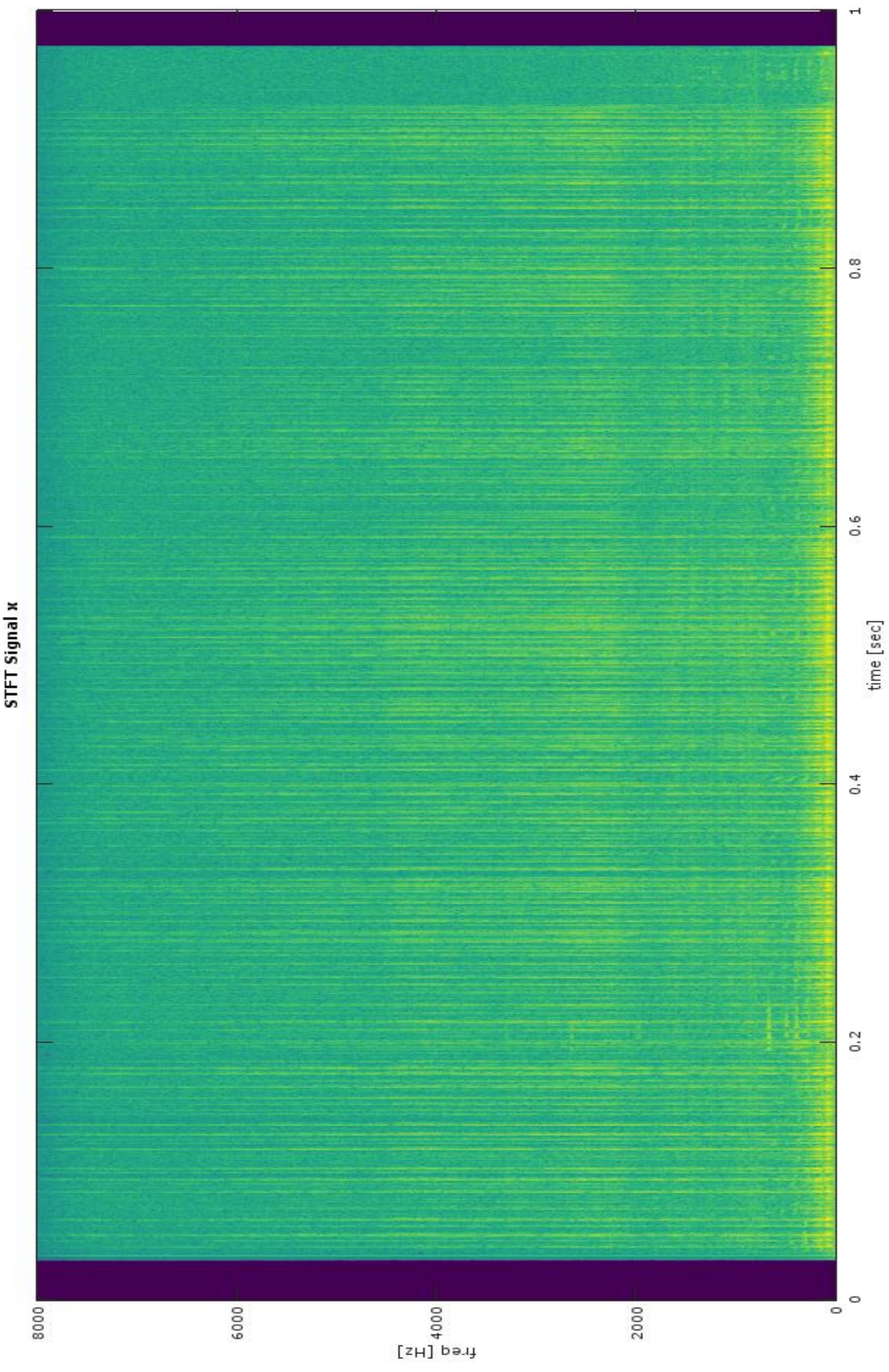


Figure 4: Spectrogram of signal a.

Where the yellow colour shows the maximum energy of the signal x captured with respect to frequency bins. Thus , time–frequency analysis comprises of those techniques that study a signal in both the time and frequency domains simultaneously, the motivation for this technique is comes from an understanding between functions and their transform representation being tightly connected often[11][12] , and they can be best understood together as a two-dimensional object, rather than separately[15] . The practical motivation for time–frequency analysis is that classical Fourier analysis assumes that signals are infinite in time or periodic, in reality many signals in practice are of short duration, and change substantially over their duration [16]. For example, traditional musical instruments do not produce infinite duration sinusoids, but instead begin with an attack, then gradually decay. This is poorly represented by traditional methods, which motivates using the time–frequency analysis [17].

2.3 Time-Frequency analysis

A signal represented in time domain gives us the information about the magnitudes of the signal with respect to the time, which is known as temporal resolution of that signal [11] .However, if we need to find out the frequency content of the signal, it becomes important to study the signal into a domain, which gives the value of the repeating frequencies with corresponding magnitudes [14]. If we want to study the spectral and temporal resolution together then we need to do a time - frequency analysis through one of the methods below [16].

- Short-time Fourier transform.
- Wavelet transform.
- Wigner distribution function.
- Modified Wigner distribution function, and so on.

We know that Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) are applied to entire signal. Thus, we are unable to resolve the spectral changes with time, so we can divide the signal into “chunks”, and apply the DFT/FFT to each one of them [18]. This strategy is known as the Short-Time Fourier Transform (STFT), and the resulting time-frequency representation is known as a Spectrogram [14]. In this experiment we are taking the Short-Time Fourier Transform (STFT) of the signal and apply a binary mask to estimate the time and frequency resolution in order to represent the signal in its best form [19].

2.3.1 Short-Time Fourier Transform (STFT):

The Short-Time Fourier Transform (STFT) is performed as follows,

- Define an analysis window size for example.
- Define the amount of overlap between windows.
- Define a windowing function.
- Generate windowed segments by multiplying signal with the windowing function.
- Apply the FFT to each windowed segment [14].

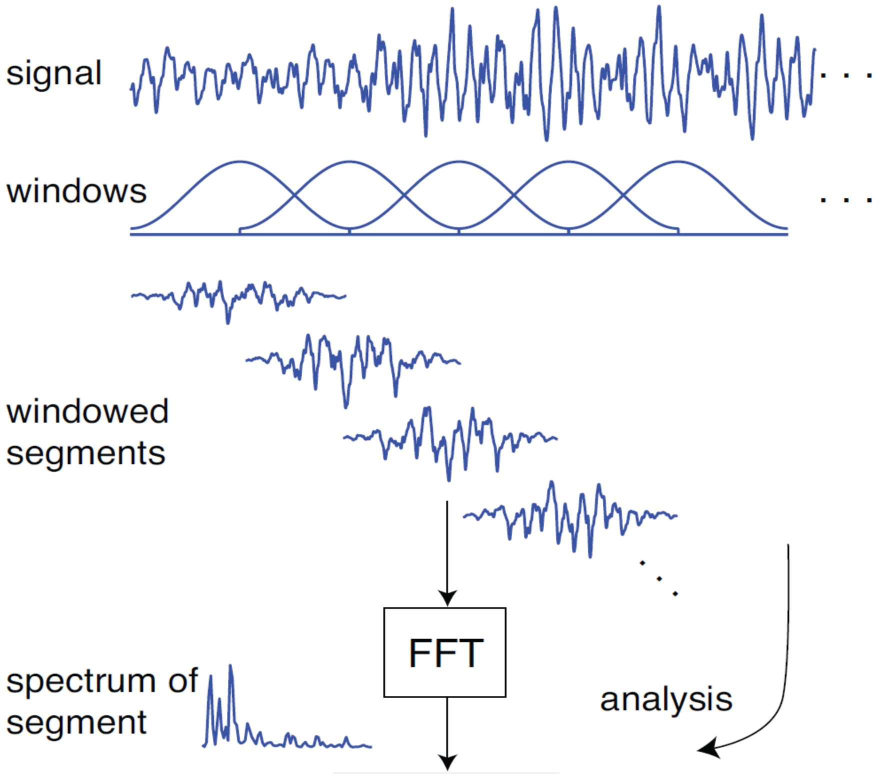


Figure 5: Steps to calculate Short time Fourier Transform (STFT) [14].

2.3.2 Overview of Window function:

As mentioned above the division of signal into chunks is possible by a window function, which is a mathematical function that gives a zero-value outside of some chosen interval. For example, a function that is constant inside the interval and zero elsewhere making it easy to spectrally differentiate the signal in shorter amount of time [14].
The window function serves several purposes

- It localizes the Fourier Transform in time, by considering only a short time interval in the signal
- By having a smooth shape, it minimizes the after effects (e.g., high side lobes/ridges) of chopping the signal into chunks.
- By overlapping windows, it provides spectral continuity across time [14].

There are different types of windows - Rectangular, Triangular, Hanning, Hamming, Blackman, etc. The rectangular window expressed as ω(n) = 1, is the simplest form of window, it is equivalent to replacing all data sequence by zeros except N values of data sequence, making it appear as though the waveform suddenly turns on and off. Rectangular window can be used for analysis of transients, when analysing a transient signal in modal analysis, such as an impulse, a shock response, a sine burst, a chirp burst, or noise burst, where the energy versus time distribution is extremely uneven [21] [22]. For example, when most of the energy is located at the beginning of the recording, a non-rectangular window attenuates most of the energy, degrading the signal-to-noise ratio [5]. Mathematical representation of rectangular window is,

$$w_R(n) \triangleq \begin{cases} 1, & -\frac{M-1}{2} \leq n \leq \frac{M-1}{2} \\ 0, & \text{otherwise} \end{cases}$$

Where,
M is the window length in samples.

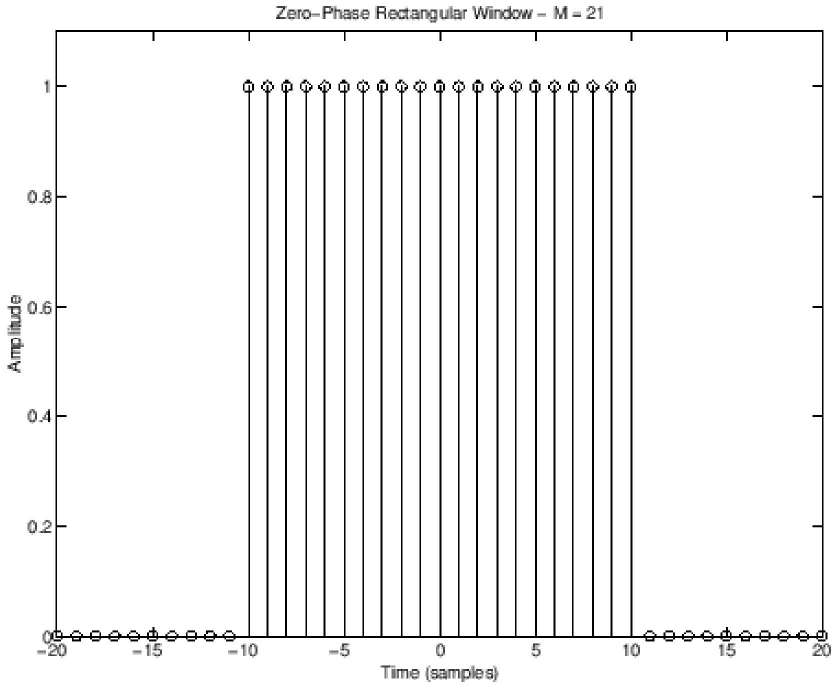


Figure 6 : Rectangular window function [5].

The Fourier Transform of the rectangular window is shown in the Fig 7.

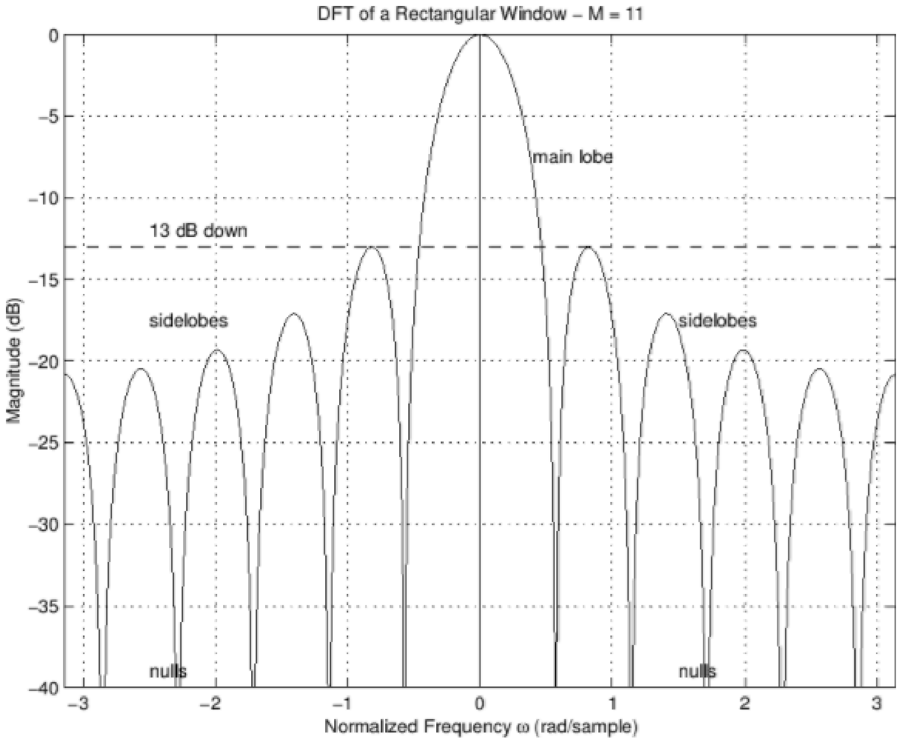


Figure 7: Fourier Transform of Rectangular window function [5].

The reason why we are not using the rectangular window because it will produce large ripples in frequency spectrum on account of the sharp transitions taking place across the signal. Moreover, the sliding window would produce more number of transitions along the bins. So, it could result into uneven dispersion of the spectral energies of both the signals and the signal data might be lost. In this experiment we use a "Hamming window" which is mathematically expressed as,

$$w(n) = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right),$$

Where α = 0.54, and β = 0.46 are the constants to cancel the first side lobe of the Window. Thus the constants help to reduce the ripples in the side lobes, reducing the lowering the levels as we can observe in Fig 8.

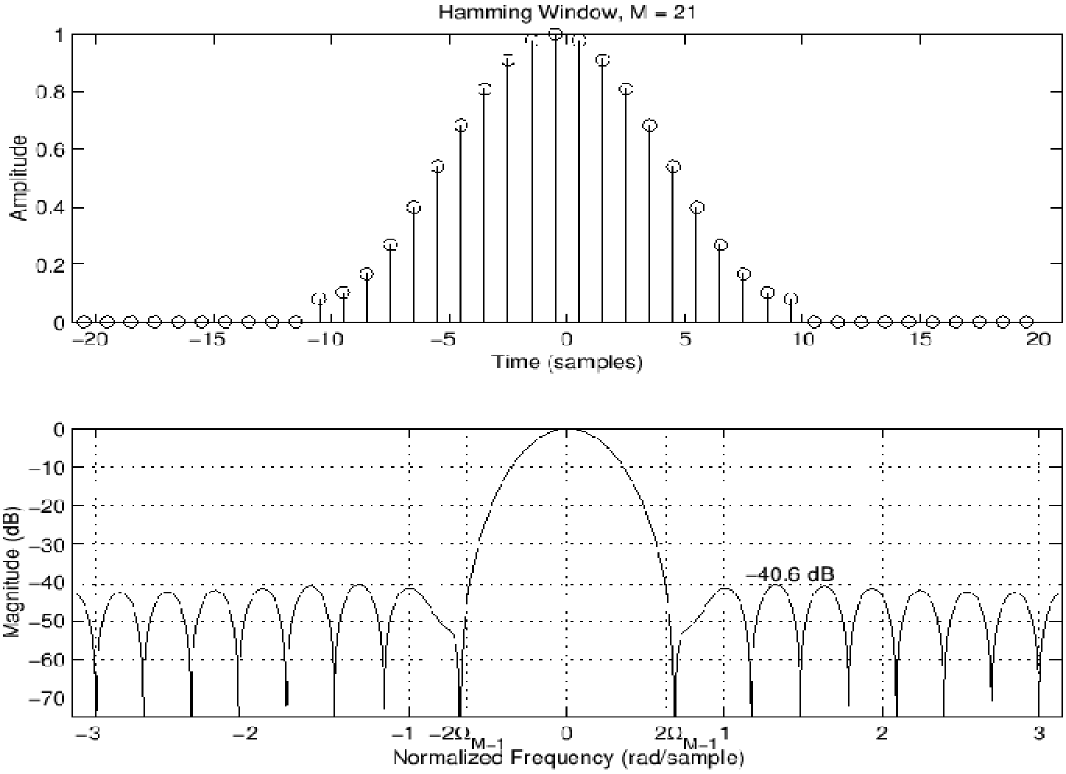


Figure 8: Hamming Window and its Fourier Transform [1].

Due to its nature of the side-lobe level below 40 dB it is a good choice for ``1% accurate systems," such as 8-bit audio signal processing systems, because there is rarely any reason to require the window side lobes to lie far below the signal quantization noise floor. The Hamming window has been extensively used in telephone communications signal processing [1]. However, for a higher quality audio signal processing, higher quality windows may be required, particularly when those windows act as low pass filters for eliminating high frequency components, the reason why this happens is because the window function removes the sharp transitions at the start and end of the data to be analysed, this set of data is tapered up gently from 0 to full scale and the data is tapered down gently to 0 at the trailing end of the data which reduces high frequency content associated with the transition at the start and end of the data. The size of the windowing function results in the good signal representation, whether it is good frequency resolution or good time resolution. Hence, a wider window gives better frequency resolution but poor time resolution and a narrower window gives good time resolution but poor frequency resolution which are known as narrowband and wideband transforms, respectively as shown in the Fig 9.

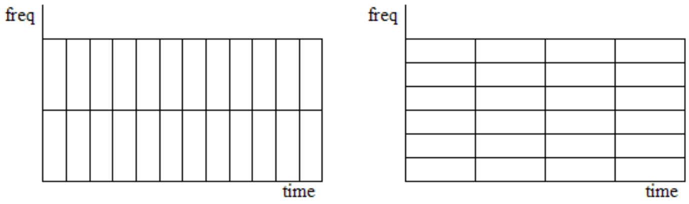


Figure 9: Narrowband and Wideband Transforms in time-frequency domain.

Taking the windowed function along with the Fast Fourier Transform (FFT) of the signal in time domain gives us a Short Time Fourier Transform (STFT) of that particular signal, which gives us magnitude of frequency components of the signal with respect to the number of bins defined. The STFT of the signal as shown in the Figure below.

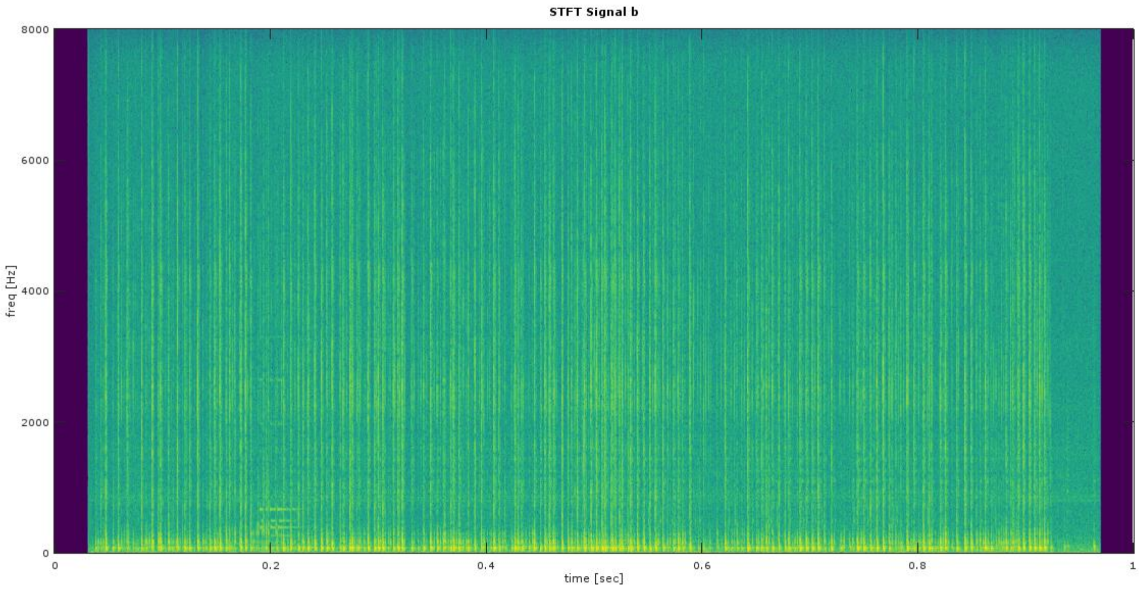


Figure 10: Short Time Fourier Transform of signal a.

Time–frequency analysis comprises those techniques that study a signal in both the time and frequency domains simultaneously, the motivation for this technique is comes from an understanding between functions and their transform representation being tightly connected often , and they can be best understood together as a two-dimensional object, rather than separately [16]. The practical motivation for time–frequency analysis is that classical Fourier analysis assumes that signals are infinite in time or periodic, in reality many signals in practice are of short duration, and change substantially over their duration [15]. For example, traditional musical instruments do not produce infinite duration sinusoids, but instead begin with an attack, then gradually decay. This is poorly represented by traditional methods, which motivates using the time–frequency analysis [17].

2.4 Binary masking

Binary mask is calculated by comparing the energy of the target sound with the energy of the interferer sound within each time-frequency bins which is shown below whenever the energy of the mixture is greater than the interfering sound the mask value will be set as 1, giving us the idea of the dominated signal present in the mixture [6].

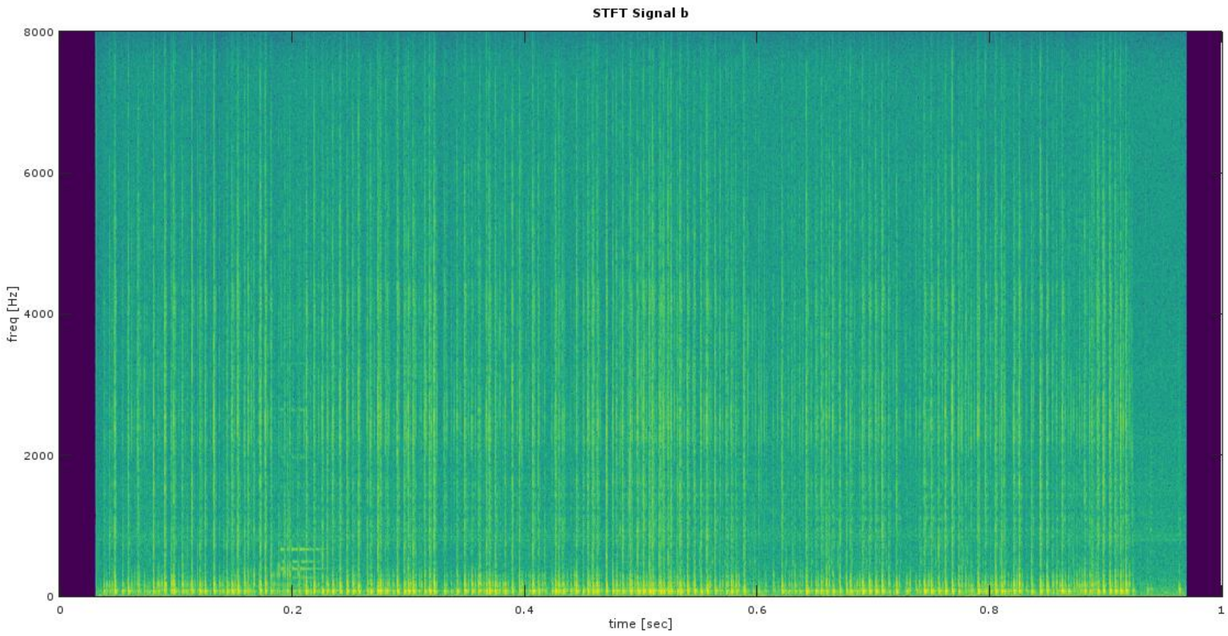


Figure 11: Short Time Fourier Transform of signal a.

We take the above signal as interferer sound b and compare it with the mixture sound x:

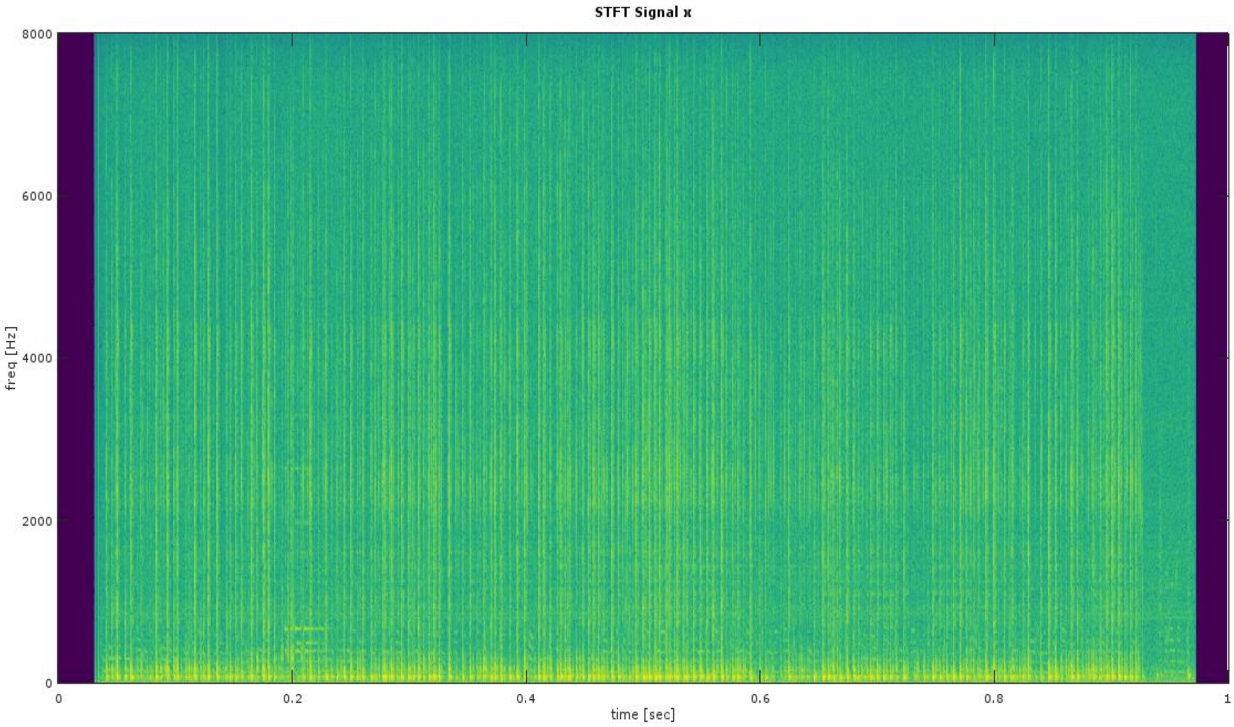


Figure 12: Short Time Fourier Transform of mixture signal x.

Thus, result is we get a binary mask M.

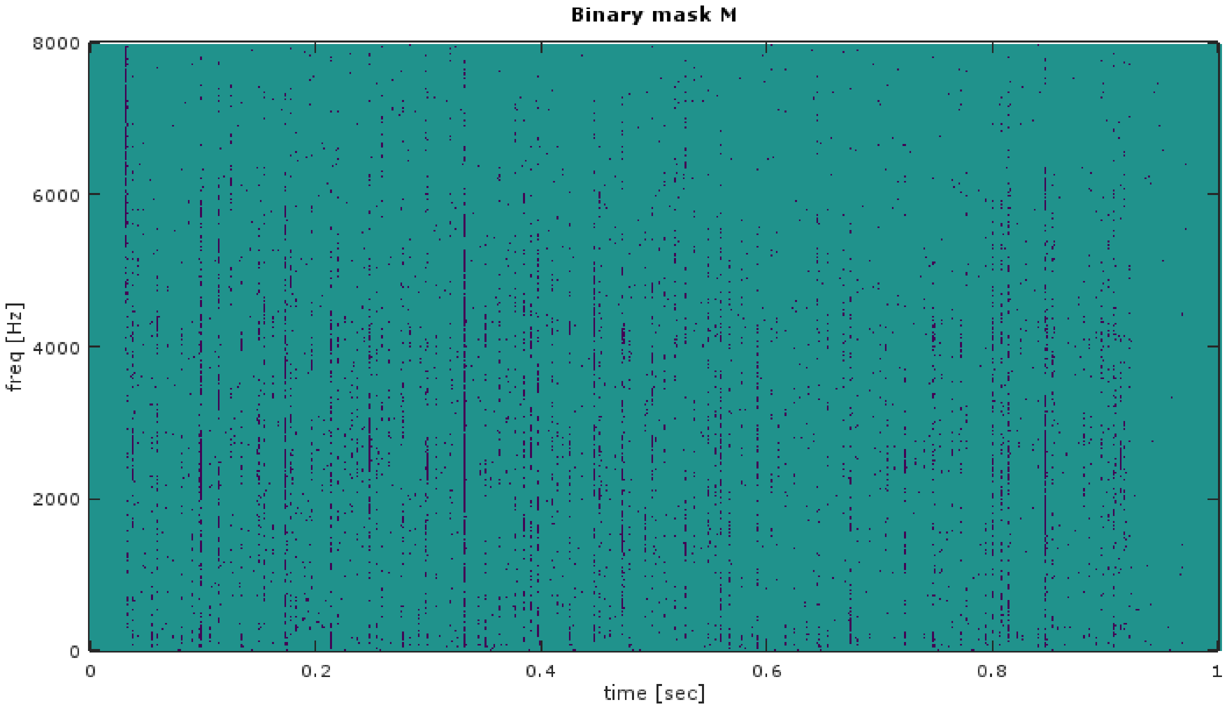


Figure 13: Binary mask from comparison of the mixture signal x and the interfering signal b

When this mask is placed in the time-frequency representation of the mixture we get the following plot as:

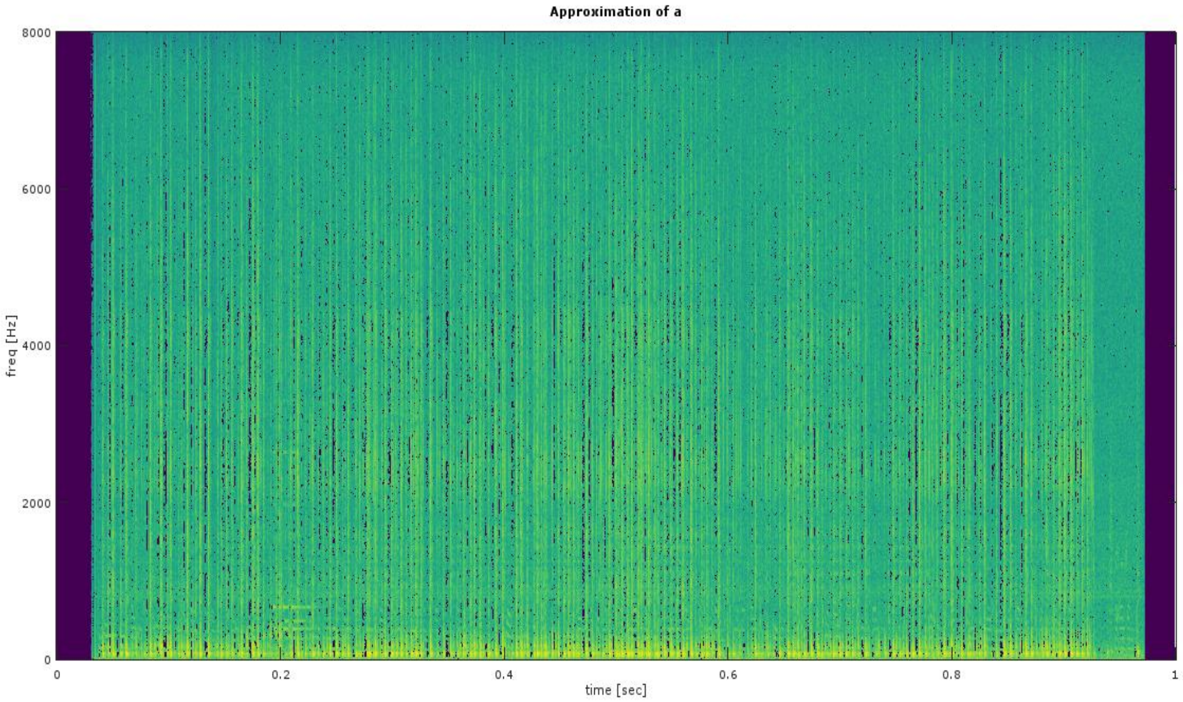


Figure 14: Approximation of the signal a obtained after the binary masking of the mixture signal.

We can see from the above figure the dark spots indicate the bin frequency where the energy of the mixture signal x was dominated by energy of the interferer signal b and was removed with the help of mask. Hence, approximation of signal a. If we take the Inverse Short Time Fourier Transform of the above signal we can get back the reconstructed signal a below. Where the yellow region depicts the exact signal energy of the source signal a represented in time and frequency domain.

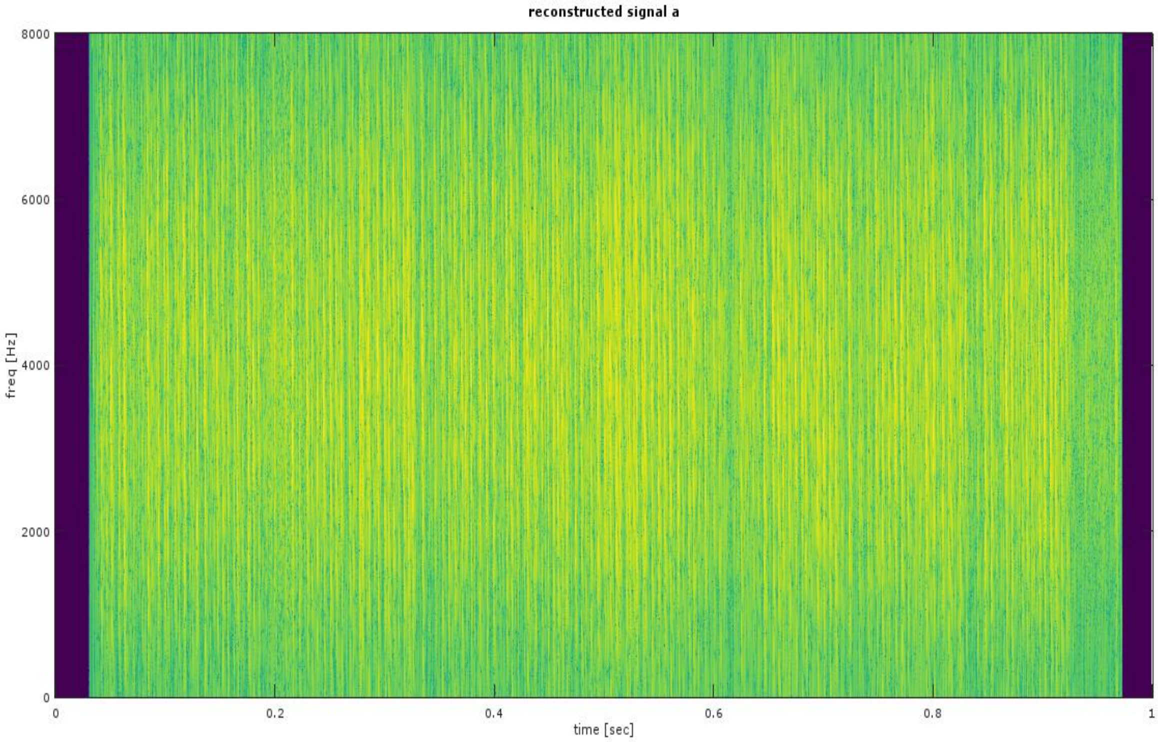


Figure 15 Reconstructed signal similar to the speech signal a.

In this experiment we have used the Short-Time Fourier Transform as an application of Binary Masking. The result of the Short-Time Fourier Transform (STFT) is frequency channels with equal bandwidths and linearly spaced centre-frequencies in Hz .When the Short-Time Fourier Transform (STFT) is used for binary masking, the binary mask can be applied by multiplying the binary mask with the magnitudes of the Short-Time Fourier Transform (STFT) .The binary mask is multiplied with the FFT magnitudes, and the Inverse Short-Time Fourier Transform (ISTFT) is applied to the modified magnitudes using the phases from the unmodified input signal [10]. Finally, the resulting short time segments from the inverse FFT are combined. Experimenting methods such as source separation, speech enhancement, or noise reduction, using binary masking, it becomes important to realize that there are two different and sometimes conflicting goals to increase speech quality or to increase speech intelligibility [11] [12].Speech quality is a measure of how clear, natural and free of distortion the speech is, whereas speech intelligibility is a measure of how much of the speech that has been perceived correctly and recognized. In other words how clear the target speech was sounding despite the context and for intelligibility is related to the right context. However, we have to trade off both the features against each other, when working with the experiment [13].

Table below, shows which is the best window length to represent the signal in time-frequency domain. From the table it is clear that at lower window lengths we get a good representation in time domain whereas at higher window lengths we get a good representation in frequency domain. However, at a window length of 1024, we get a better representation of both domains. So it is clear that the signal is best represented in time and Frequency domain for window length of 1024.

2.5 Areas covered by algorithm

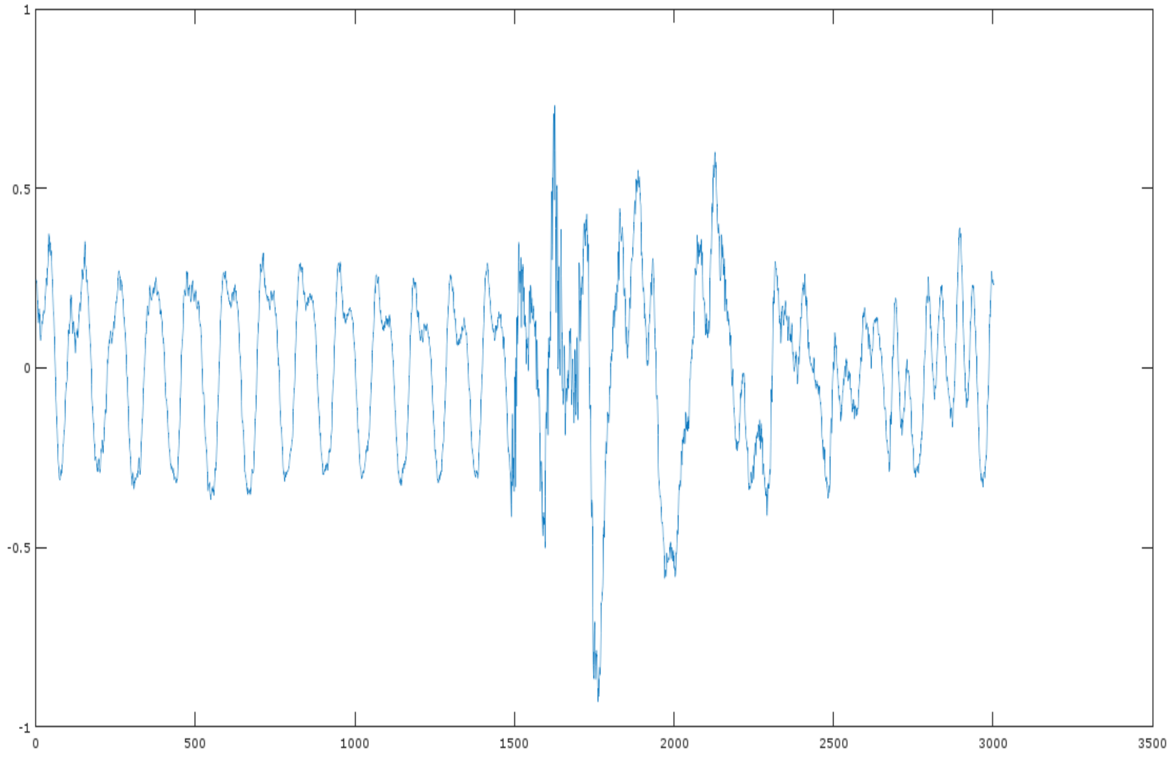


Figure 16: Chunk of the Sampled mixture signal x.

As we can see that we have taken a small chunk of mixture signal in the time domain and we try to analyse the frequency content of the mixture signal which is kind of impossible to do so in the time domain we then try to convert it into frequency domain by applying Fast Fourier Transform (FFT) of that signal [23].

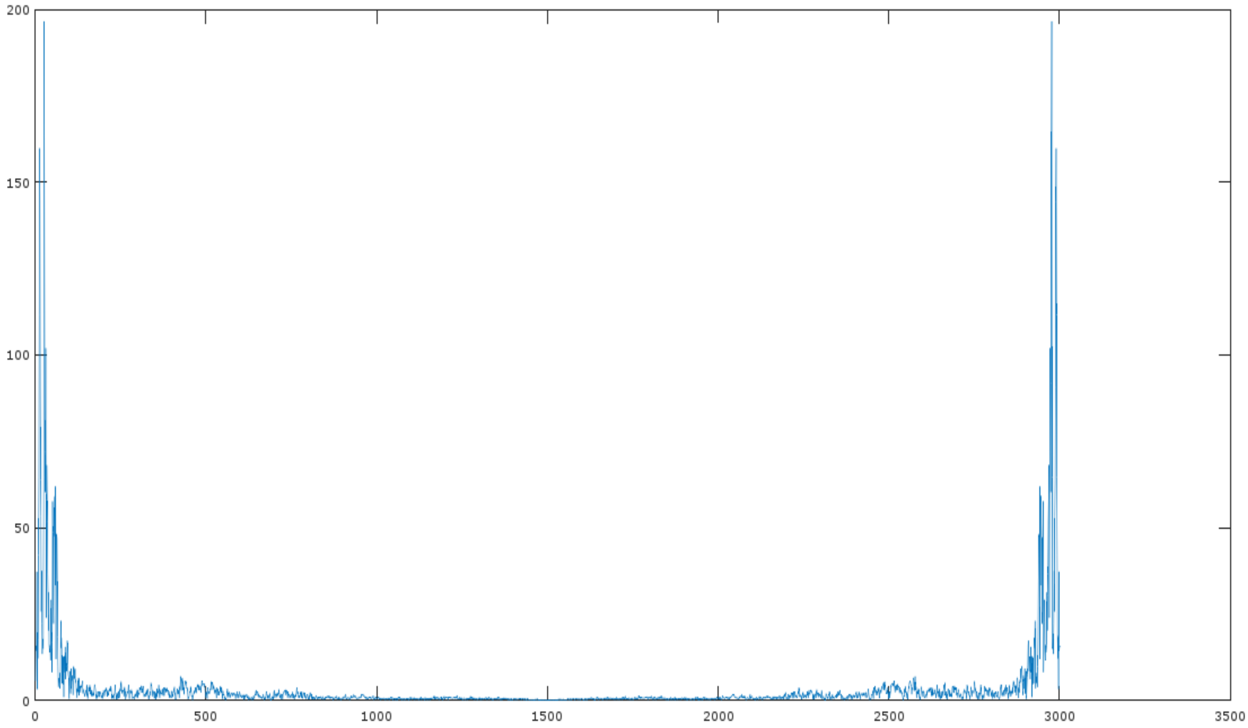


Figure 17: Fast Fourier Transform of the chunk of the Sampled mixture signal x.

Now, we have the frequency content of the signal but still it is a bit difficult to differentiate the frequency component of the mixture signals [24][25]. Hence, we take Short-Time Fourier Transform (STFT) of the signal and we get the result as shown in the figure.

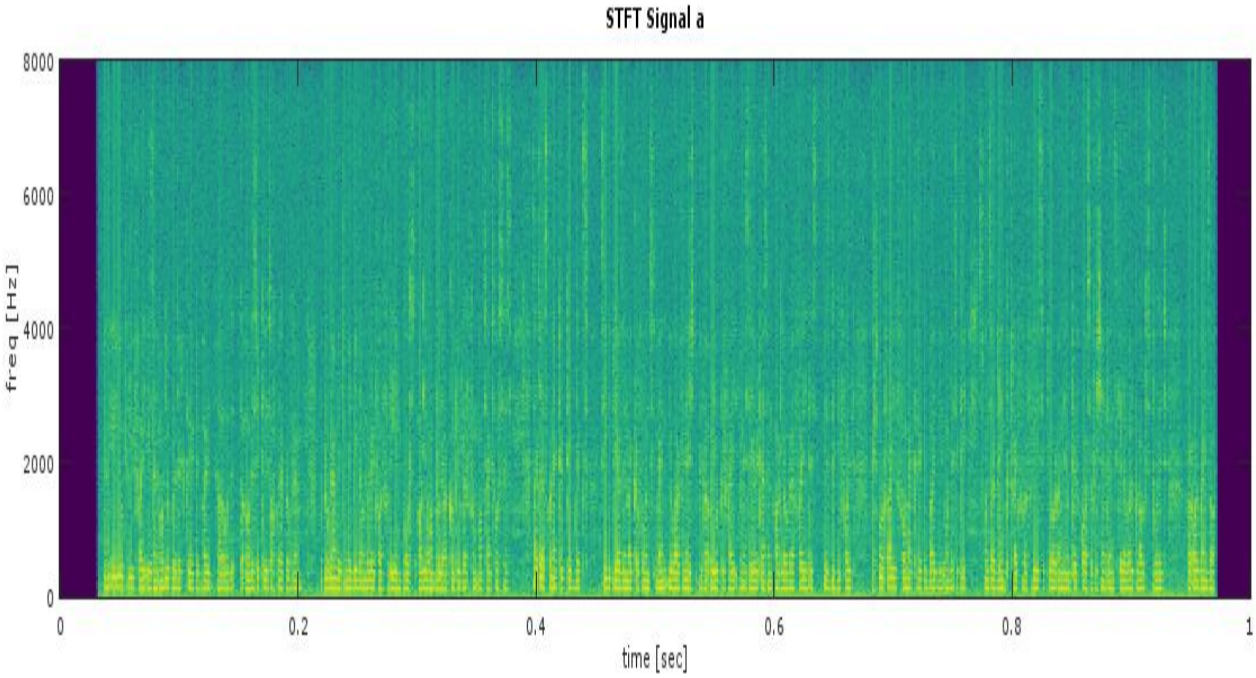


Figure 18: Short Time Fourier Transform of desired signal a.

By applying the Binary masking technique as discussed in the section of binary mask we manage to recover the signal a using Inverse Short Time Fourier Transform, which helps in determining the frequency content of the signal associated to frequency bins[20].

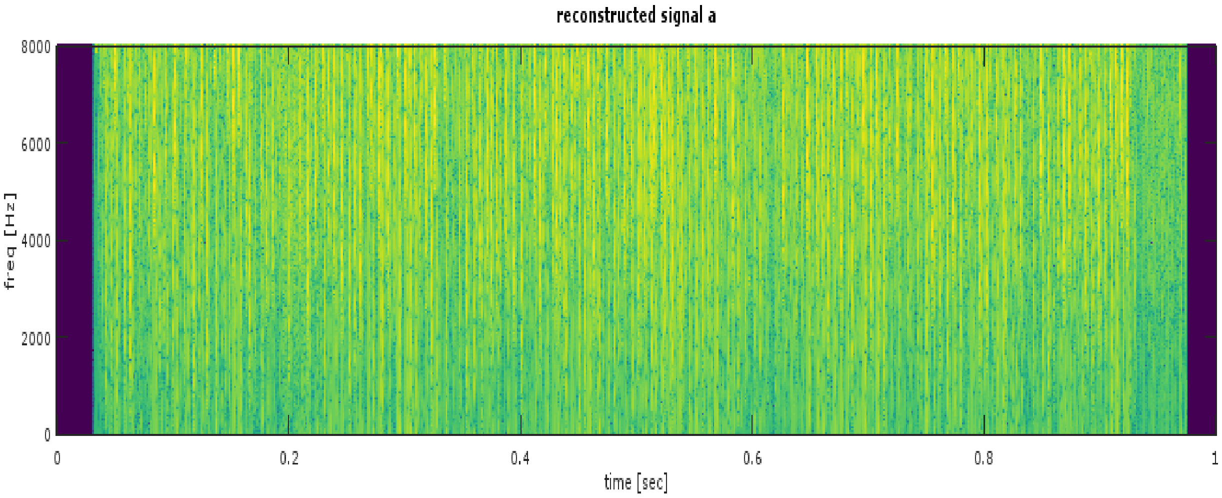


Figure 19: Inverse Short Time Fourier Transform of above figure to recover the signal using the binary masking.

Where the yellow area indicates high spectral energy recorded by the bin of the reconstructed signal. On the left and right side are the zero padded regions in which there is no signal presence shown by purple colour. Blue colour shows the removed noise interference. Sometimes we cannot get exactly the original signal because, if the signals are non – orthogonal in the Time-Frequency domain then the frequency content may be occupied by the dominating signal in that particular [21] frequency bin which will eliminate/hide the frequency content of the dominated signal. Thus, it is very necessary that the signals should be disjoint-orthogonal while performing windowing operation. Only then can we get a perfect representation of the expected signal [23], this experiment focuses in the multi domain area of the signal processing which enhances the separation performance. The highly changing spectral contents of the signal in the continuous and discrete time domain and frequency domain proves to be challenging when we are working towards the signal quality and separation performance, thus, this experiment is a step towards solving the real world problems faced in communication's sector. For instance if a meeting was conducted via internet calling like Skype, if the users were speaking and typing at the same time, it would be very difficult for the users to pay attention to the speech sound of one another. Hence, it would result in confusion at some stage or later. This problem can be solved by using this technique.

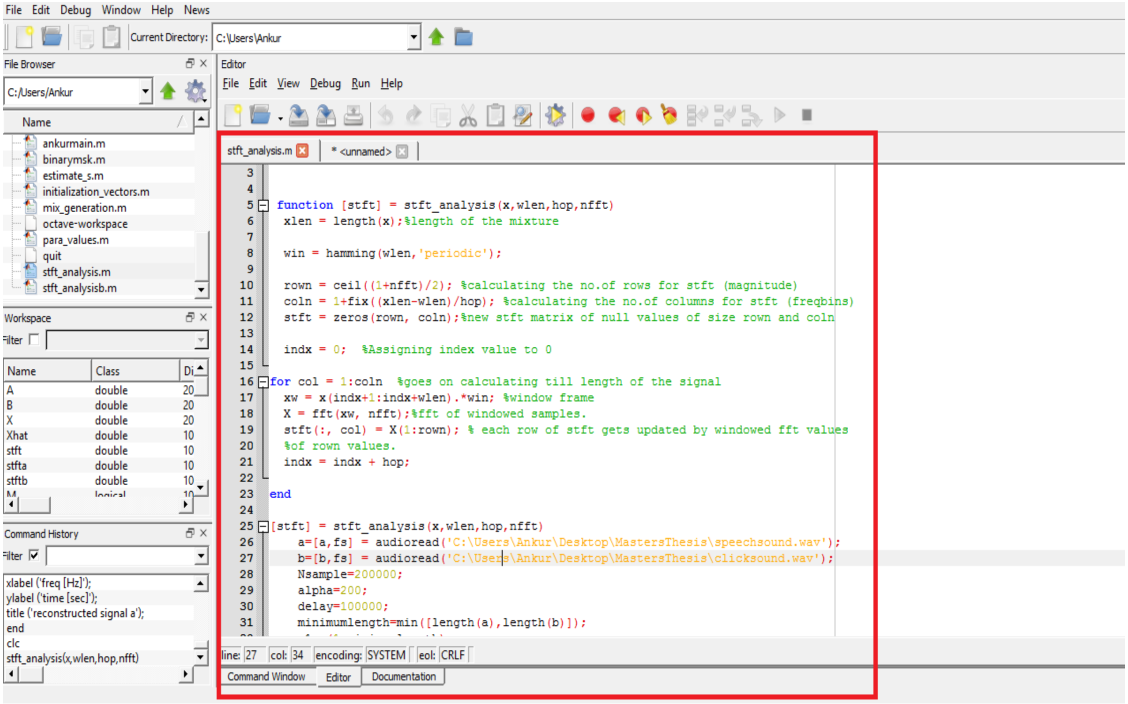
Few other examples include,

- Speech recognition.
- Speaker identification.
- Gender classification system.
- Speech pre-processing for aids to hearing impaired persons [26].

CHAPTER 3 TESTING AND RESULTS

4.1 Experimental Procedure.

Step1: Writing the code in the editor window.



Name	Class	Di
A	double	20
B	double	20
X	double	20
Xhat	double	10
stft	double	10
stfta	double	10
stftb	double	10
M	logical	10

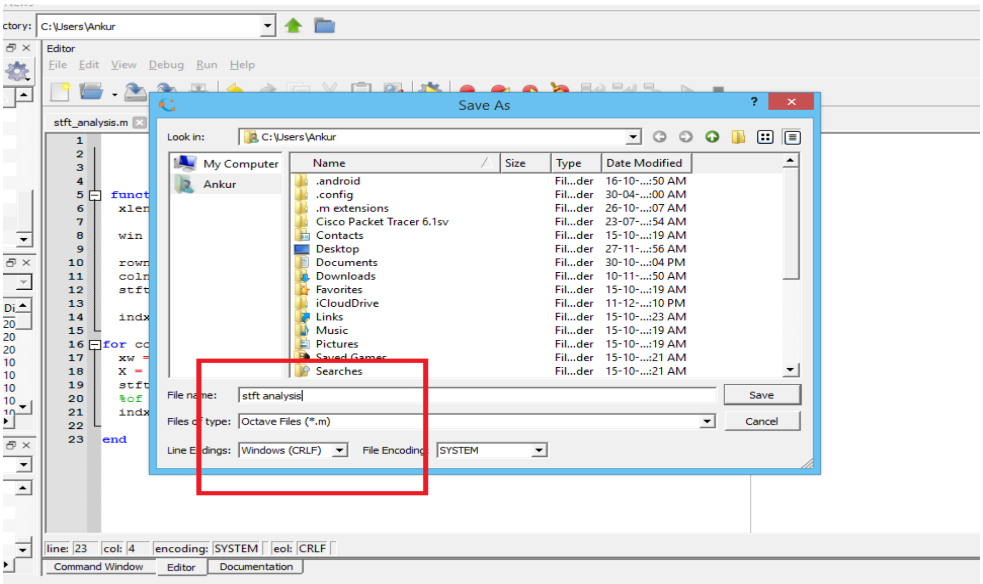
```
xlabel ('freq [Hz]');
ylabel ('time [sec]');
title ('reconstructed signal a');
end
clc
stft_analysis(x,wlen,hop,nfft)
```

```


function [stft] = stft_analysis(x,wlen,hop,nfft)
 xlen = length(x);%length of the mixture

 win = hamming(wlen,'periodic');

 rown = ceil((1+nfft)/2); %calculating the no.of rows for stft (magnitude)
 coln = 1+fix((xlen-wlen)/hop); %calculating the no.of columns for stft (freqbins)
 stft = zeros(rown, coln);%new stft matrix of null values of size rown and coln

 indx = 0;  %Assigning index value to 0

for col = 1:coln  %goes on calculating till length of the signal
 xw = x(indx+1:indx+wlen).*win; %window frame
 X = fft(xw, nfft);%fft of windowed samples.
 stft(:, col) = X(1:rown); % each row of stft gets updated by windowed fft values
 %of rown values.
 indx = indx + hop;

end

[stft] = stft_analysis(x,wlen,hop,nfft)
   a=[a,fs] = audioread('C:\Users\Ankur\Desktop\MastersThesis\speechsound.wav');
   b=[b,fs] = audioread('C:\Users\Ankur\Desktop\MastersThesis\clicksound.wav');
   Nsample=200000;
   alpha=200;
   delay=100000;
   minimumlength=min([length(a),length(b)]);
```

line: 27 col: 34 encoding: SYSTEM eol: CRLF

Figure 20: Writing the code in editor window.

Step 2: Debugging the code and saving it as .m version of file in the selected destination folder.



Look in: C:\Users\Ankur

Name	Size	Type	Date Modified
.android		Fil...der	16-10-...:50 AM
.config		Fil...der	30-04-...:00 AM
.m extensions		Fil...der	26-10-...:07 AM
Cisco Packet Tracer 6.1sv		Fil...der	23-07-...:54 AM
Contacts		Fil...der	15-10-...:19 AM
Desktop		Fil...der	27-11-...:56 AM
Documents		Fil...der	30-10-...:04 PM
Downloads		Fil...der	10-11-...:50 AM
Favorites		Fil...der	15-10-...:19 AM
iCloudDrive		Fil...der	11-12-...:10 PM
Links		Fil...der	15-10-...:23 AM
Music		Fil...der	15-10-...:19 AM
Pictures		Fil...der	15-10-...:19 AM
Saved Games		Fil...der	15-10-...:21 AM
Searches		Fil...der	15-10-...:21 AM

File name: stft analysis

Files of type: Octave Files (*.m)

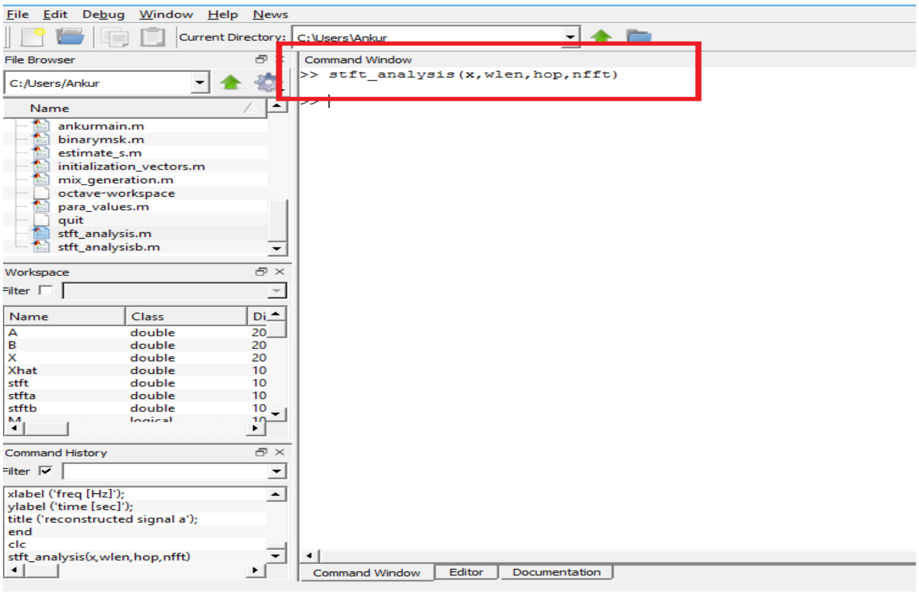
Line Endings: Windows (CRLF) File Encoding: SYSTEM

line: 23 col: 4 encoding: SYSTEM eol: CRLF

Figure 21: debugging the code and saving it in the file.

Figure 22: file created of .m extension.

Step 3: Calling that particular code from the command window.



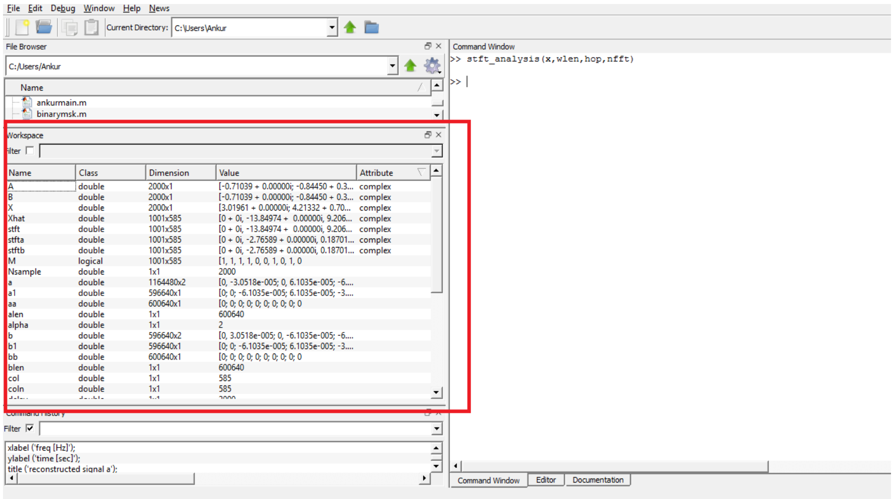
Name	Class	Di
A	double	20
B	double	20
X	double	20
Xhat	double	10
stft	double	10
stfta	double	10
stftb	double	10

```
xlabel ('freq [Hz]');
ylabel ('time [sec]');
title ('reconstructed signal a');
end
clc
stft_analysis(x,wlen,hop,nfft)
```

```
>> stft_analysis(x,wlen,hop,nfft)
>> |
```

Figure 23: Running the file in the command window to achieve the desired results/plots.

Step 4: Generated vectors shown in the bottom left in the figure below,



```
>> stft_analysis(x,wlen,hop,nfft)
>> |
```

Name	Class	Dimension	Value	Attribute
A	double	2000x1	[-0.71039 + 0.00000i; -0.84450 + 0.3...	complex
B	double	2000x1	[-0.71039 + 0.00000i; -0.84450 + 0.3...	complex
X	double	2000x1	[3.01961 + 0.00000i; 4.21332 + 0.70...	complex
Xhat	double	1001x585	[0 + 0i, -13.84974 + 0.00000i, 9.206...	complex
stft	double	1001x585	[0 + 0i, -13.84974 + 0.00000i, 9.206...	complex
stfta	double	1001x585	[0 + 0i, -2.76589 + 0.00000i, 0.18701...	complex
stftb	double	1001x585	[0 + 0i, -2.76589 + 0.00000i, 0.18701...	complex
M	logical	1001x585	[1, 1, 1, 1, 0, 0, 1, 0, 1, 0	
Nsample	double	1x1	2000	
a	double	1164480x2	[0, -3.0518e-005; 0, 6.1035e-005; -6....	
a1	double	596640x1	[0; 0; -6.1035e-005; 6.1035e-005; -3....	
aa	double	600640x1	[0; 0; 0; 0; 0; 0; 0; 0; 0; 0	
alen	double	1x1	600640	
alpha	double	1x1	2	
b	double	596640x2	[0, 3.0518e-005; 0, -6.1035e-005; -6....	
b1	double	596640x1	[0; 0; -6.1035e-005; 6.1035e-005; -3....	
bb	double	600640x1	[0; 0; 0; 0; 0; 0; 0; 0; 0; 0	
blen	double	1x1	600640	
col	double	1x1	585	
coln	double	1x1	585	

```
xlabel ('freq [Hz]');
ylabel ('time [sec]');
title ('reconstructed signal a');
```

Figure 24: Generated values of the vectors defined in the code.

These vectors shown above contain the information about the frequency components of the signals which we further plot with the help of the special function known as imagesc () which results in the plots below,

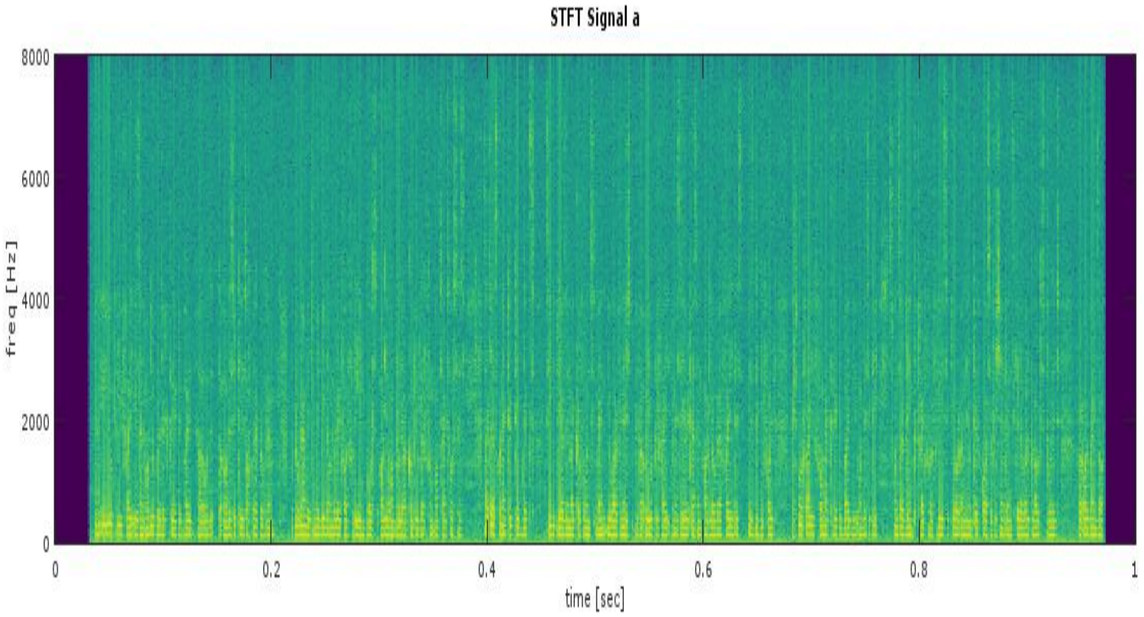


Figure 25: final result in the plot.

Following are the plots illustrating the varying parameters of the window function with respect to time-frequency analysis. There are total 6 plots for parameters namely window length and hop size, where window length is the length of the window and hop size is the size of the overlapping parameter of the window function. One should observe at which parameter range the reconstructed signal is represented clearly in time as well as frequency domain.

PLOT1 (Window Length = 256; Hop Size = 256)

In the first plot we have generated the Short Time Fourier Transform (STFT) of the speech sound as signal ‘a’ from following command.

```
function [stft] = stft_analysisa(a,wlen,hop,nfft)
alen = length(a);
% defining the length of the signal a which is our speech sound
win = hamming(wlen,'periodic'); d
 % defining the window length
 rown = ceil((1+nfft)/2);
 coln = 1+fix((alen-wlen)/hop);
 stft = zeros(rown, coln);
% creating a STFT matrix to store the values after computation
 indx = 0; %Assigning index value to 0
for col = 1:coln % loop for calculating values till length of the signal
 xw = a(indx+1:indx+wlen).*win; % window frame
 A = fft(xw, nfft);% fft of windowed samples.
 stft(:, col) = A(1:rown);
%each row of stft gets updated by windowed fft values
 indx = indx + hop;
 end
```

Similarly, we have generated the Short Time Fourier Transform (STFT) of click sound as signal ‘b’.

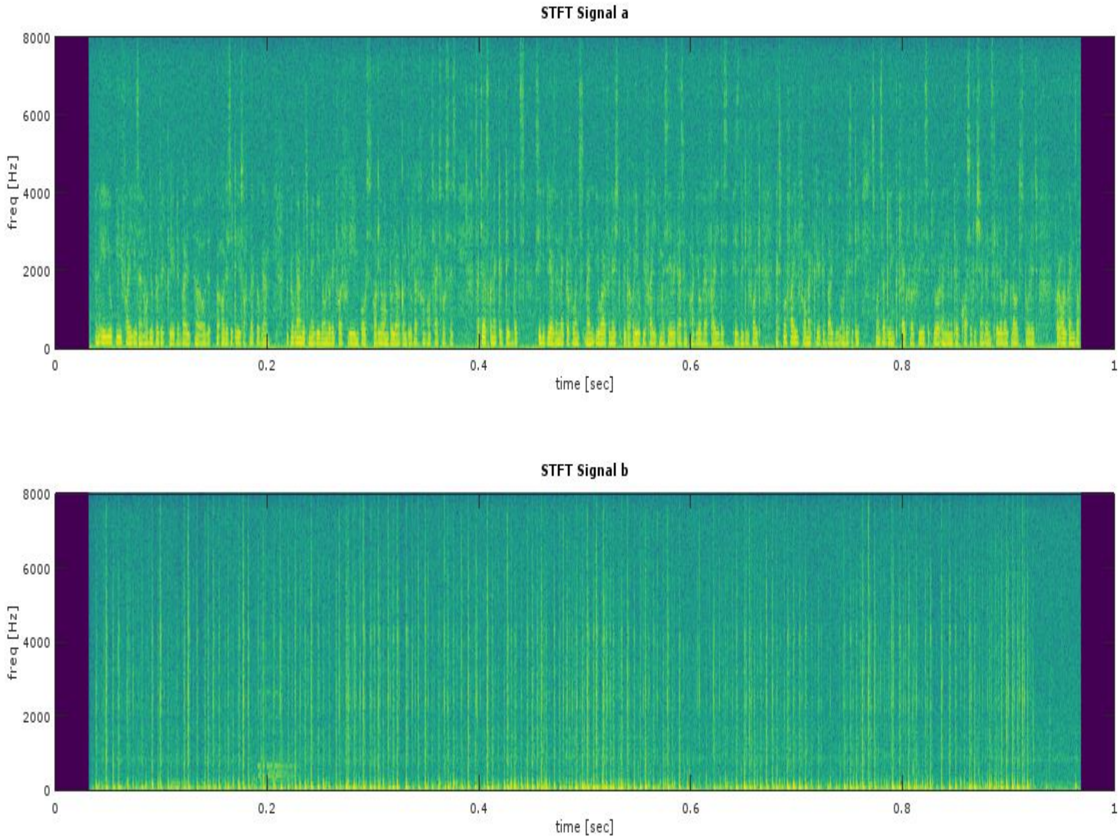


Figure 26: Short Time Fourier Transform of signals a and b.

Next plot is the Short Time Fourier Transform (STFT) of the mixture signal x consisting of the mixture of signal ‘a’ and ‘b’ using the code below.

```
  function [stft] = stft_analysisx(a,b,wlen,hop,nfft)
  x=a+ alpha*b; %%shifting and adding the two signals a and b.
  x = x(:);%creating a column vector for x
  xlen = length(x);%%length of the mixture
  win = hamming(wlen,'periodic');% defining the window
 rown = ceil((1+nfft)/2);% calculating the number of rows for stft
  coln = 1+fix((xlen-wlen)/hop);% calculating the no.of columns for stft
  stft = zeros(rown, coln);%new stft matrix of null values of size rown and coln
  indx = 0;%assigning index value to 0
 for col = 1:coln % calculating till length of the signal
  xw = x(indx+1:indx+wlen).*win;%window frame
  X = fft(xw, nfft);% fft of windowed samples.
  stft(:, col) = X(1:rown);--- each row of stft gets updated by windowed fft
  indx = indx + hop;
  end
```

Creating the binary mask for mixture x and multiplying the mask with mixture signal to get approximation of signal a.

```
function [M,Xhat] = binarymask(stfta,stftb)
M= stft>=stftb;%comparing the values of x with a
Xhat=stft.*M;%approximation of signal a
end
```

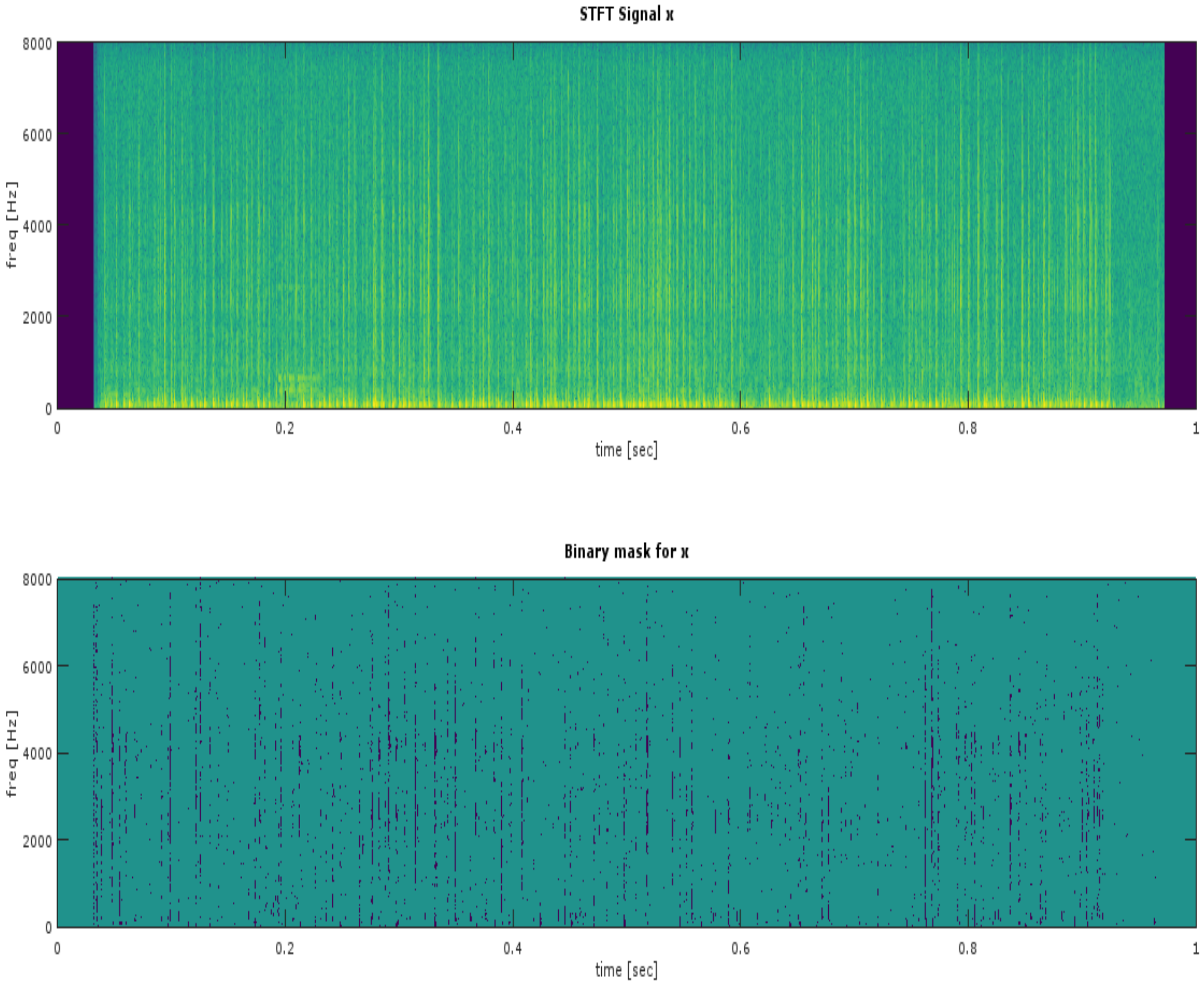


Figure 27: Short Time Fourier Transform of mixture signal x and binary mask of the signal x.

Finally we have generated the plot of reconstructed signal using function below.

```
function [x_origional] = binarymask(Xhat)
x_origional = real(ifft(Xhat));%%%the inverse short time fourier transform to the
approximated signal.
end
```

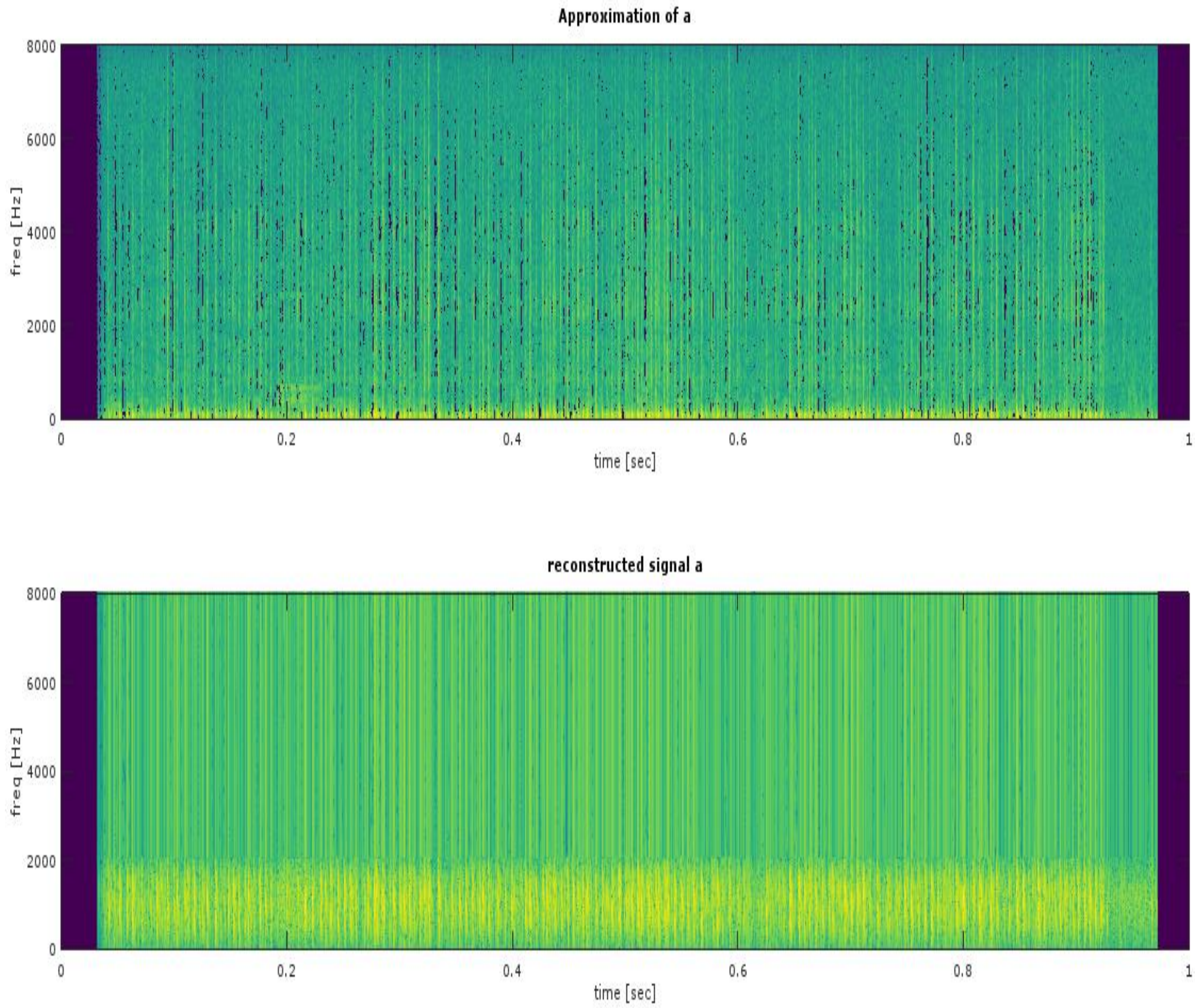


Figure 28: Reconstruction of the signal a using the generated binary mask.

We can observe from above plot giving better time resolution but poor frequency resolution. Where the yellow region is the maximum energy captured by the time-frequency bin.

PLOT2 (Window Length = 256; Hop Size = 512)

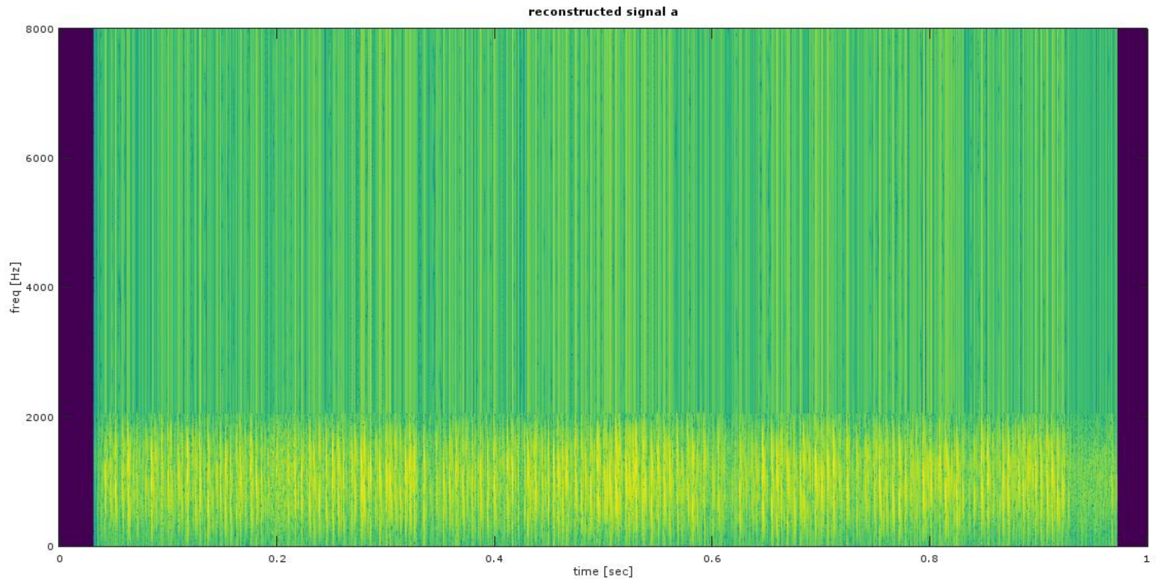


Figure 29: Reconstructed signal a for window length -256 over hop size-256

We can observe from above plot still giving better time resolution but poor frequency resolution. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT3 (Window Length = 512; Hop Size = 512)

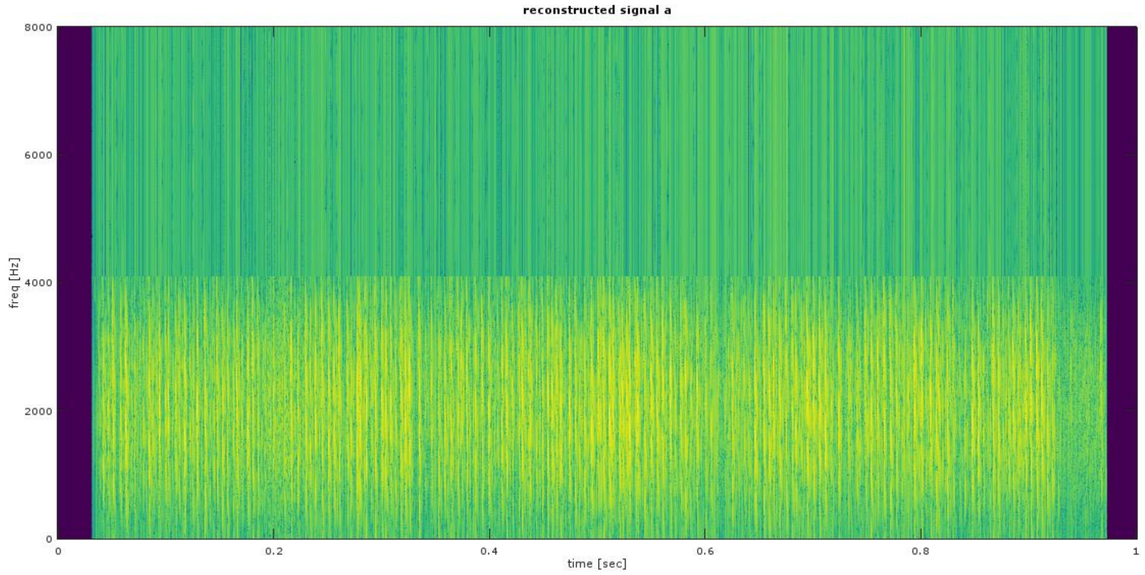


Figure 30: Reconstructed signal a for window length -512 over hop size-512

We can observe from above plot giving better time resolution but much better frequency resolution than the previous plot. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT4 (Window Length = 1024; Hop Size = 256)

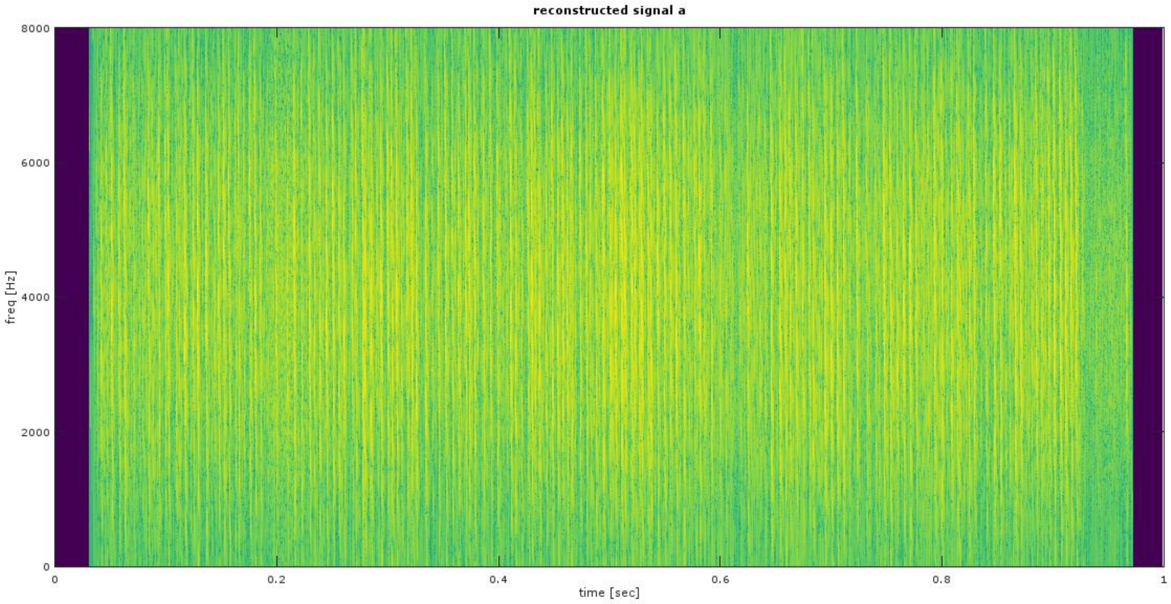


Figure 31: Reconstructed signal a for window length -1024 over hop size-256

We can observe from above plot giving better time resolution but much better frequency resolution than the previous plot. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT5 (Window Length = 1024; Hop Size = 512)

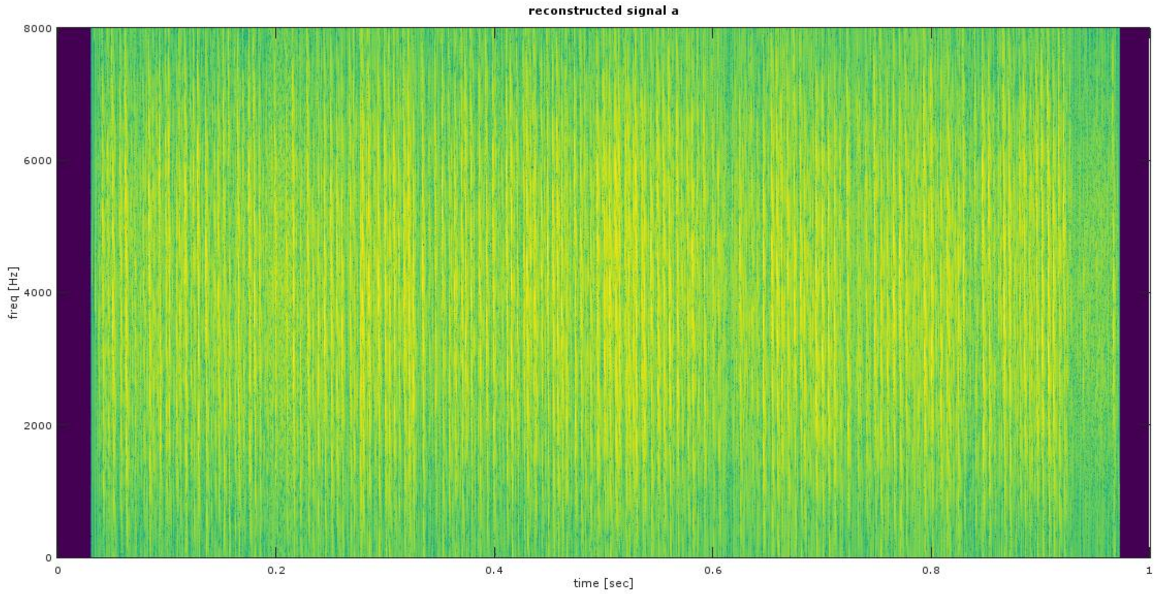


Figure 32: Reconstructed signal a for window length – 1024 over hop size-512

We can observe from above plot giving the best time-frequency resolution than the previous plot. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT6 (Window Length = 2048; Hop Size = 256)

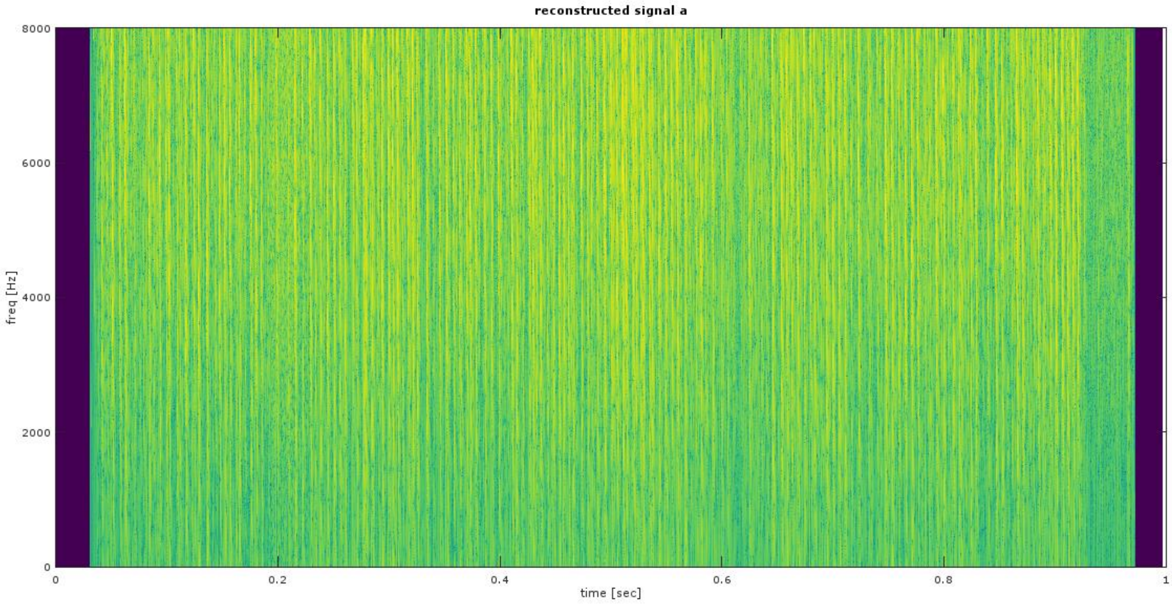


Figure 33: Reconstructed signal a for window length -2048 over hop size-256

We can observe from above plot still giving better frequency resolution but poor time resolution. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT7 (Window Length = 2048; Hop Size = 512)

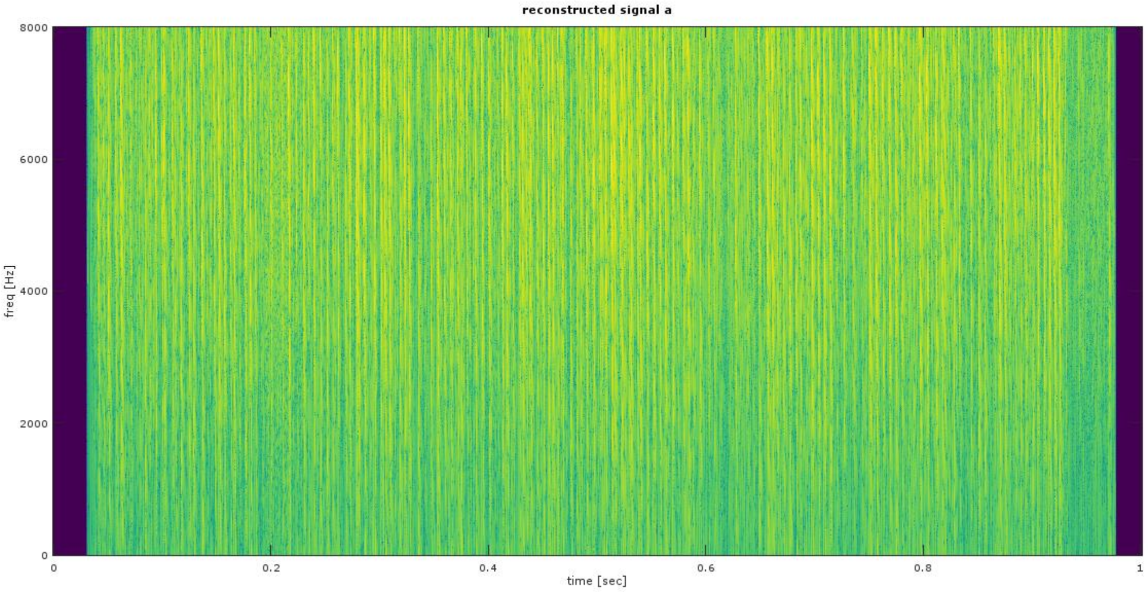


Figure 34: Reconstructed signal a for window length -2048 over hop size-512

We can observe from above plot still giving better frequency resolution but much poor time resolution. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT8 (Window Length = 4096; Hop Size = 512)

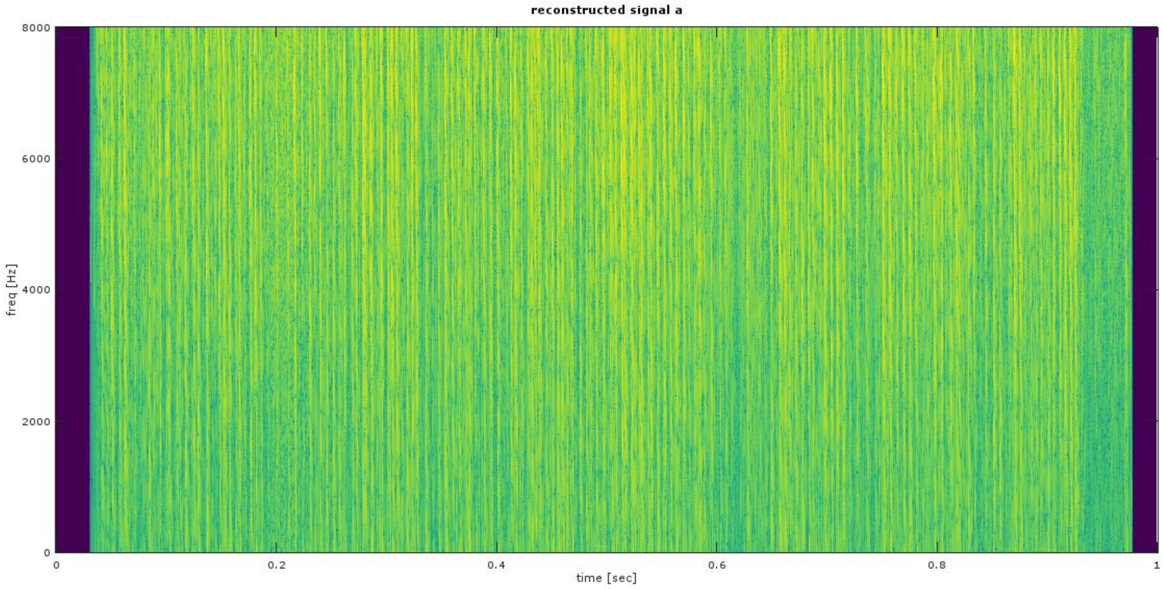


Figure 35 : Reconstructed signal a for window length -4096 over hop size-512

We can observe from above plot still giving better frequency resolution but much poor time resolution than previous plots. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin.

PLOT9 (Window Length = 2048; Hop Size = 1024)

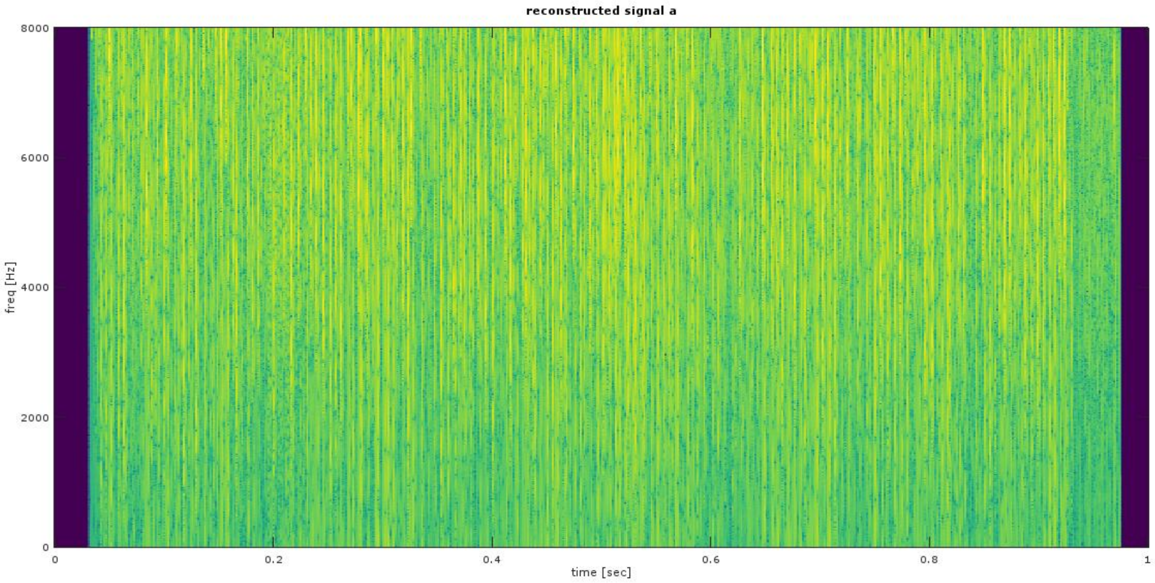


Figure 36: Reconstructed signal a for window length -2048 over hop size-1024

We can observe from above plot still giving better frequency resolution but much poor time resolution than previous plots. Where the yellow region is the maximum energy of the desired signal captured by the time-frequency bin. From the above plots we have generated a table showing the window lengths variation with respect to the overlapping parameter and the resolution of the signal in time-frequency domain.

Table 2: Varying Window length for better Resolution.

Serial no.	Window length (wlen)	Hop size (hop)	Good representation in Time/Frequency
1	**256**	**256**	**Time**
2	**256**	**512**	**Time**
3	**512**	**512**	**Time**
4	**1024**	**256**	**Frequency**
5	**1024**	**512**	**Time, Frequency**
6	**2048**	**256**	**Frequency**
7	**2048**	**512**	**Frequency**
8	**4096**	**1024**	**Frequency**
9	**2048**	**1024**	**Frequency**

Observing the above plots and the table we can say that the best representation of the signal is shown in the window length of 1025 with half the size of the overlapping parameter.

4.2 Measure of Window disjoint Orthogonality with respect to time-frequency parameters.

We call two functions Sj(t) and Sk(t) to be W-disjoint orthogonal ,when the Windowed Fourier Transforms of Sj(t) and Sk(t) are disjoint. In other words if their inner product is zero. It is very useful in terms of separation of the mixture into its component sources using a binary mask [26-28]. Window-disjoint Orthogonality can be calculated under the following condition: Given a mask M, such that 0 ≤ M(γ) ≤ 1 for all elements γ in the transform space Γ, the preserved-signal ratio (PSR$_M$) and the signal-to-interference ratio (SIR$_M$) performance criteria are defined as:

$$PSR_M = \frac{||M(\gamma)S_k(\gamma)||^2}{||S_k(\gamma)||^2}, SIR_M = \frac{||M(\gamma)S_k(\gamma)||^2}{||M(\gamma)Y_k(\gamma)||^2}$$

The approximate W-disjoint Orthogonality is then defined as:

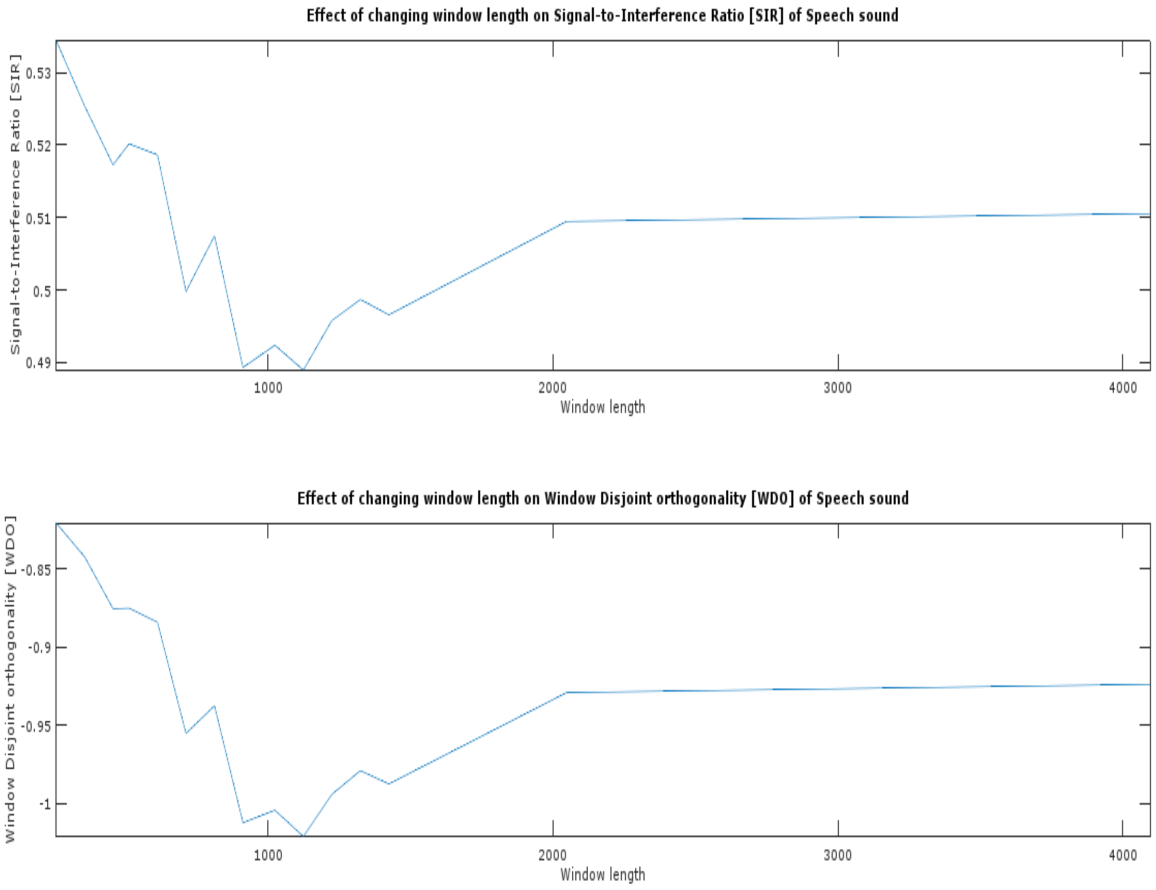
$$WDO_M = PSR_M - \frac{PSR_M}{SIR_M}$$

The maximum possible value, WDOMk = 1, implies that the mask Mk can perfectly separate and recover the k-th source [28].

Table 3: Effect of Window lengths on the W-disjoint Orthogonality and Signal-to-Interference Ratio for Speech sound.

Window length	**PSR**	**SIR**	**WDO**
256	**0.94201**	**0.53445**	**-0.82056**
356	**0.93242**	**0.52541**	**-0.84223**
456	**0.93805**	**0.51727**	**-0.87543**
512	**0.94866**	**0.52018**	**-0.87506**
612	**0.95263**	**0.51865**	**-0.88410**
712	**0.95445**	**0.49979**	**-0.95525**
812	**0.96599**	**0.50747**	**-0.93754**
912	**0.96987**	**0.48931**	**-1.0123**
1024	**0.97418**	**0.49238**	**-1.0043**
1124	**0.97696**	**0.48895**	**-1.0211**
1224	**0.97760**	**0.49579**	**-0.99420**
1324	**0.97390**	**0.49869**	**-0.97901**
1424	**0.97407**	**0.49658**	**-0.98749**
2048	**0.96494**	**0.50947**	**-0.92908**
4096	**0.96357**	**0.51054**	**-0.92380**

The above table shows that for the given window lengths the window disjoint Orthogonality for speech sound is maximum observed at window lengths of values 912, 1024 and 1124 means that over here we are able to observe that the binary mask applied can separate the signal to a much greater extent . Hence, higher separation performance [28].



Effect of changing window length on Signal-to-Interference Ratio [SIR] of Speech sound

Effect of changing window length on Window Disjoint orthogonality [WDO] of Speech sound

Figure 37: Window length v/s [Signal-interference Ratio, W-disjoint Orthogonality] for speech sound.

Table 4: Effect of Window lengths on the W-disjoint Orthogonality and Signal-to-Interference Ratio for Click sound.

Window length	PSR	SIR	WDO
256	**0.70073**	**1.8711**	**0.32623**
356	**0.69570**	**1.9033**	**0.33017**
456	**0.69397**	**1.9332**	**0.33501**
512	**0.69175**	**1.9224**	**0.33191**
612	**0.69105**	**1.9281**	**0.33264**
712	**0.71430**	**2.0008**	**0.35730**
812	**0.70926**	**1.9705**	**0.34933**
912	**0.73830**	**2.0437**	**0.37704**
1024	**0.73937**	**2.0309**	**0.37532**
1224	**0.74260**	**2.0170**	**0.37443**
1424	**0.74155**	**2.0138**	**0.37331**
1724	**0.72171**	**1.9704**	**0.35544**
2048	**0.71585**	**1.9628**	**0.35115**
4096	**0.71008**	**1.9587**	**0.34756**
8192	**0.78648**	**2.1876**	**0.42696**

The above table shows that for the given window lengths the window disjoint Orthogonality for click sound is constant and less than 1 at window lengths of values 912, 1024 and 1124 means that over here we are able to observe that the binary mask applied cannot separate the signal to much higher extent. Hence, lower separation performance [28].

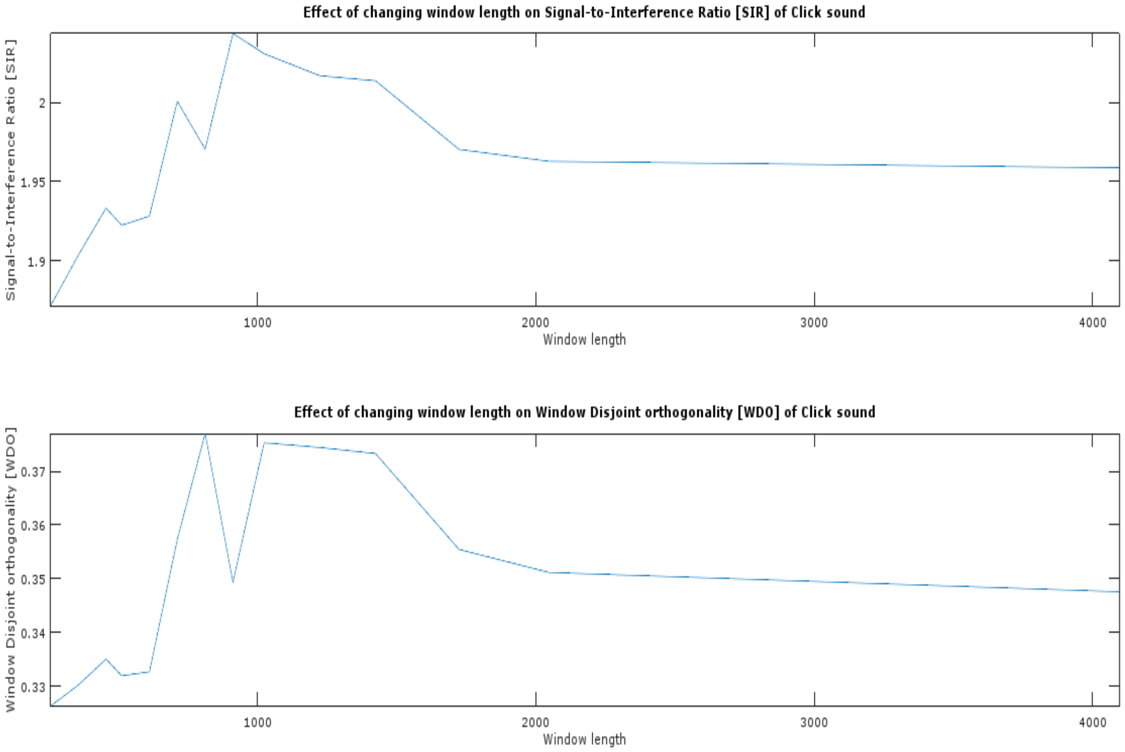


Figure 38: Window length v/s [Signal-interference Ratio, W-disjoint Orthogonality] for click sound.

CHAPTER 4 CONCLUSIONS

In this project a noise cancellation for the speech signal at 16 KHz was simulated in Matlab software. The separation performance of the signal was tested for different window lengths in which the best window length for best separation of the signal was determined. Time-frequency analysis was explored at a great detail in terms of understanding the resolution. Fourier Transform for extracting the spectral components of a signal was understood and worked upon. The binary mask was generated and tested for separation of signal and reconstruction of the original signal was implemented using the mask value. Window disjoint Orthogonality of the mixtures were taken into account for the mixture signals and its relationship between the varying window parameters was determined. The window lengths of values greater than 1224 and lesser than 2014 values exhibited a poor separation performance. However, some window lengths showed a greater possibility of much better separation of the signal giving the window disjoint Orthogonality value between 0-1.Higher Signal-to-Interference ratio (SIR) was exhibited in values near to the window lengths near to the value of 1024.

4.1 Future scope

The project motivates to make the use of Wavelet Transform instead of Short Time Fourier Transform for future purposes. Use different types of windows of much higher quality and its test its robustness in the different environments. Multi cannel separation for more than one mixtures can also be considered for implementation and testing using above methods for future purposes.

BIBLIOGRAPHY

[1] Smith, Julius O. Spectral Audio Signal Processing, W3K Publishing, http://books.w3k.org/, ISBN 978-0-9745607-3-1.

[2] Dsprelated.com. (2017). Spectrum Analysis of Sinusoids | Spectral Audio Signal Processing. [Online] Available at:https://www.dsprelated.com/freebooks/sasp/Spectrum_Analysis_Sinusoids.html [Accessed 27 Nov. 2017].

[3] Enochson, Loren D.; Otnes, Robert K. (1968). Programming and Analysis for Digital Time Series Data. U.S. Dept. of Defense, Shock and Vibration Info. Center. p. 142.

[4]"Hamming Window". ccrma.stanford.edu. Retrieved 2016-04-13.

[5] En.wikipedia.org. (2017). Window function. [online] Available at: https://en.wikipedia.org/wiki/Window_function#cite_note-26 [Accessed 27 Nov. 2017].

[6] Y. Hu and P. C. Loizou, “A new sound coding strategy for suppressing noise in cochlear implants,” Journal of the Acoustical Society of America, vol. 124, no. 1, pp. 498–509, 2008.

[7] N. Li and P. C. Loizou, “Factors influencing intelligibility of ideal binary-masked speech: Implications for noise reduction,” Journal of the Acoustical Society of America, vol. 123, no. 3, pp. 1673–1682, 2008.

[8] D. S. Brungart, P. S. Chang, B. D. Simpson, and D. Wang, “Isolating the energetic component of speech-on-speech masking with ideal time-frequency segregation,” Journal of the Acoustical Society of America, vol. 120, no. 6, pp. 4007–4018, 2006.

[9] H. Dudley, “Remaking speech,” Journal of the Acoustical Society of America, vol. 11,no. 2, pp. 169–177, 1939.

[10] R. V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid, “Speech recognition with primarily temporal cues,” Science, vol. 270, no. 5234, pp. 303–304, 1995.

[11] M. F. Dorman, P. C. Loizou, and D. Rainey, “Speech intelligibility as a function of the number of channels of stimulation for signal processors using sine-wave and noise-band outputs,” Journal of the Acoustical Society of America, vol. 102, no. 4, pp. 2403–2411, 1997.

[12] P. C. Loizou, M. Dorman, and Z. Tu, “On the number of channels needed to understand speech,” Journal of the Acoustical Society of America, vol. 106, no. 4, pp. 2097–2103, 1999.

[13] J. B. Allen, “How do humans process and recognize speech?” IEEE Transactions on Speech and Audio Processing, vol. 2, no. 4, pp. 567–577, 1994.

[14] Research.cs.tamu.edu. (2017). [online] Available at: http://research.cs.tamu.edu/prism/lectures/pr/pr_l29.pdf [Accessed 27 Nov. 2017].

[15] Sejdić E.; Djurović I.; Jiang J. (2009). "Time-frequency feature representation using energy concentration: An overview of recent advances". Digital Signal Processing. 19 (1): 153–183. doi:10.1016/j.dsp.2007.12.004.

[16] Jacobsen and R. Lyons, The sliding DFT, Signal Processing Magazine vol. 20, issue 2, pp. 74–80 (March 2003).

[17] Jont B. Allen (June 1977). "Short Time Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform". IEEE Transactions on Acoustics, Speech, and Signal Processing. ASSP-25 (3): 235–238.

[18] A. Cichocki and S. Amari, Adaptive Blind Signal and Image Processing. Wiley, 2002.

[19] A. Bell and T. Sejnowski, “An information-maximization approach to blind Separation and blind deconvolution,” Neural Computation, vol. 7, pp. 1129– 1159, 1995.

[20] J. Cardoso, “Blind signal separation: Statistical principles,” Proceedings of IEEE, Special Issue on Blind System Identification and Estimation, pp. 2009– 2025, Oct. 1998.

[21] E. Weinstein, M. Feder, and A. Oppenheim, “Multi-channel signal separation by decorrelation,” IEEE Trans. on Speech and Audio Processing, vol. 1, no. 4, pp. 405–413, Oct. 1993.

[22] M. V. Hulle, “Clustering approach to square and non-square blind source separation,” in IEEE Workshop on Neural Networks for Signal Processing (NNSP), Madison, Wisconsin, Aug. 23–25 1999, pp. 315–323.

[23] J.-K. Lin, D. G. Grier, and J. D. Cowan, “Feature extraction approach to blind source separation,” in IEEE Workshop on Neural Networks for Signal Processing (NNSP), Amelia Island Plantation, Florida, Sept. 24–26 1997, pp. 398–405.

[24]T.-W. Lee, M. Lewicki, M. Girolami, and T. Sejnowski, “Blind source separation of more sources than mixtures using overcomplete representations,” IEEE Signal Processing Letters, vol. 6, no. 4, pp. 87–90, Apr. 1999.

[25] L. Vielva, D. Erdogmus, C. Pantaleon, I. Santamaria, J. Pereda, and J. C. Principe, “Underdetermined blind source separation in a time-varying environment,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, Orlando, Florida, USA, May 13–17 2002, pp. 3049–3052.

[26] M. A. Abd El-Fattah et aI., "Speech enhancement with an adaptive Wiener filter", International Journal of Speech Technology , vol. 17 Issue 1, pp. 53-64, March 2014.

[27] O. Yılmaz and S. Rickard, “Blind separation of speech mixtures ¨ via time-frequency masking,” IEEE Trans. on Signal Processing, vol. 52, no. 7, pp. 1830–1847, Jul. 2004.

[28] Matthieu Puigt, Emmanuel Vincent, Yannick Deville, Anthony Griffin, Athanasios Mouchtaris, "Effects of audio coding on ICA performance: An experimental study", Electronics Control Measurement Signals and their application to Mechatronics (ECMSM) 2013 IEEE 11th International Workshop of, pp. 1-6, 2013.

[29] Ruairí de Fréin, "Source Separation Approach to Video Quality Prediction in Computer Networks", Communications Letters IEEE, vol. 20, pp. 1333-1336, 2016, ISSN 1089-7798.

[30] Ruairí de Fréin, Scott Thurston Rickard, "Power-Weighted Divergences for Relative Attenuation and Delay Estimation", Signal Processing Letters IEEE, vol. 23, pp. 1612-1616, 2016, ISSN 1070-9908.

[31] De Fréin, Ruairí & Rickard, Scott & Pearlmutter, Barak. (2008). Sparse Multichannel Source Localization and Separation. . 10.13140/RG.2.1.3829.6405.

[32] Ruairí de Fréin, "The data-centre whisperer: Relative attribute usage estimation for cloud servers", Signal Processing Conference (EUSIPCO) 2016 24th European, pp. 687-691, 2016, ISSN 2076-1465.

[32] C. Llerena, R. Gil-Pita, D. Ayllón, H.A. Sánchez-Hevia, I. Mohino-Herranz, M. Rosa, "Synchronization for classical blind source separation algorithms in wireless acoustic sensor networks", *Statistical Signal Processing Workshop (SSP) 2016 IEEE,* pp. 1-5, 2016.

[33] Ruairí de Fréin, "Effect of system load on video service metrics", Signals and Systems Conference (ISSC) 2015 26th Irish, pp. 1-6, 2015.Adali, T
(2009). Independent component analysis and signal separation. Berlin: Springer.