Digital Signal Processing & It's Applications

Project

Removing vocals from music tracks

By
Meghanad Shingate
Samir Shelke
Vinay Narayane

1. Introduction

Most of the times commercial music is identified by singer's voice, like old songs of Mohammed Rafi and Lata Mngeshkar, who has distinguished extraordinary voice. But now a days most of the songs are identified by their distinguished instrumental music or great background music, like different 'Bands' e.g. Linkin Park. So goal of this application assignment is to remove the vocals from the songs(music tracks) in order to appreciate the underlying instrumental background.

This removal of vocals have many applications, like making ring-tones of mobile. Also removal of vocals makes production of remixes easier. Most of the time people, and I personally, like to hear tracks without singers voice.

In this project we are going to use different techniques for vocal removal. It includes filtering frequency range of human voice(i.e. Bandpass filtering), cancellation of common frequencies between stereo channels(i.e. Stereo cancellation) and finally masking time frequency spectrogram (i.e audio blind source separation[1]).This technique consists of “extracting from an input audio signal, a set of audio signals, whose mix is perceived similarly to the original audio signal”. In our case, we focused on extracting the vocals track from the mix consisting of the rest of the instruments. In this process to obtain the spectrogram of signals, we are going to take Short Time Fourier Transform (STFT) of signal. We are going to compare above three techniques.

2. Track Selection:

Application is based on assumption that music track is stereo track in which voice is distributed symmetrically in left and right channel. Two tracks namely “Wakeup sid” and “Beatles” are identified for application. Following figure-1 shows left channel and right channel of music track (here for “wake_up_sid.wav”).

Selected tracks are without any reverberation because, if mix has reverberation then mono tracks may overlap. So good candidate for this is old *beatles* songs. From figure-1 it is seen that left and right channels are not same so it's a stereo mix.

So, in following consecutive sections we are comparing above mentioned techniques.

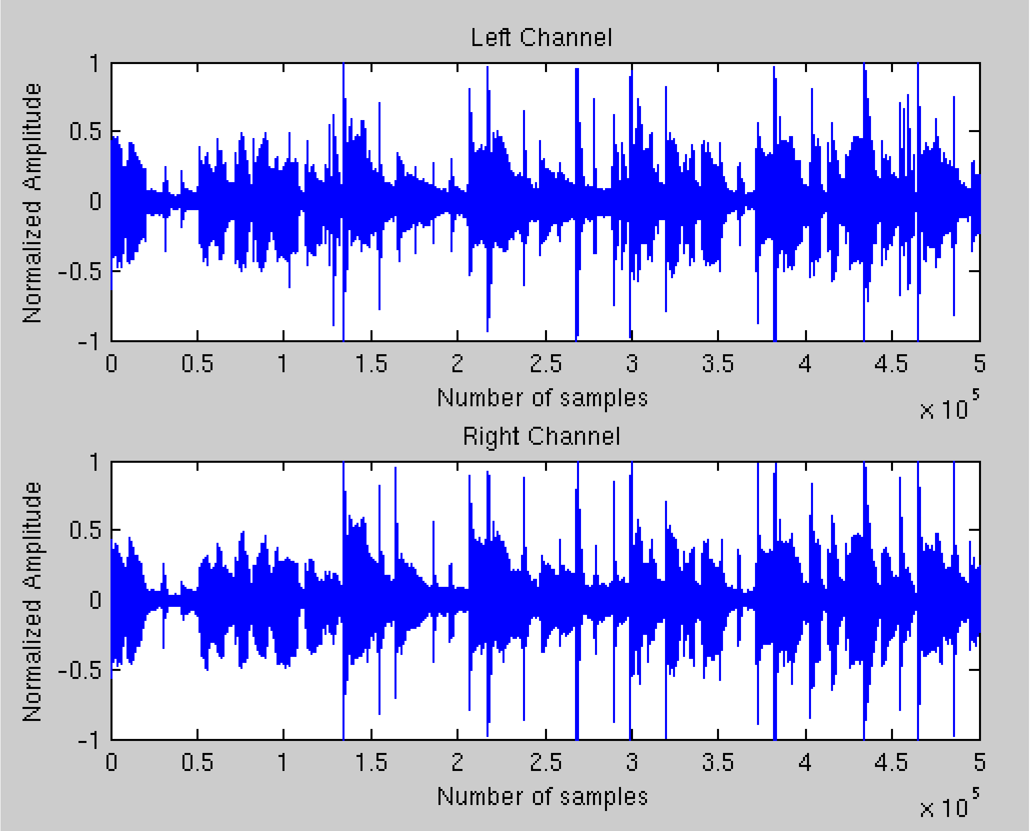

Figure 1: Original music track

3. Details of different Techniques :

3.1 Bandpass Filtering:

The simplest technique for removing vocals is bandstop filtering. The human voice has a distinct frequency range between 300 Hz and 3 kHz. So we have used FIR bandstop filter of order 2000 scince its easy to implement high order filter using software. As there are some instruments whose frequencies lies in this range(e.g. Guitars) as a side-effect these instruments also gets removed. Figure-2 shows the bandpass filtered output.

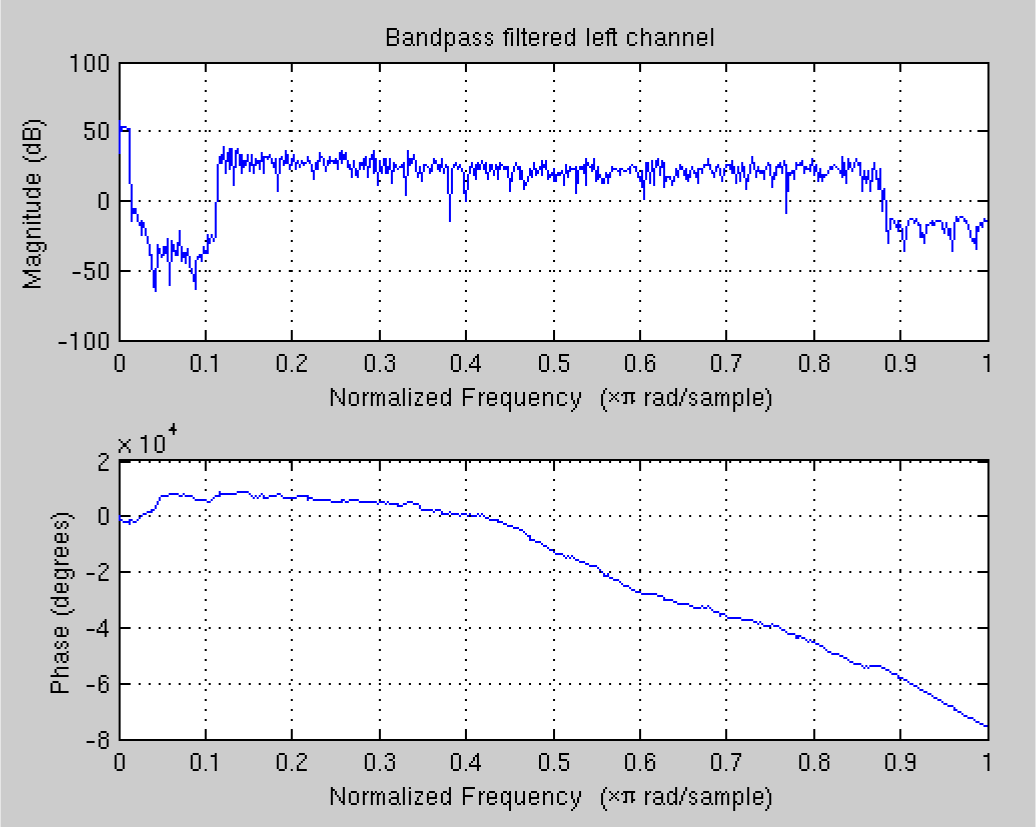

Figure 2: result of applying BPF

3.2 Stereo Cancellation:

Stereo cancellation requires stereo tracks as name suggests. This technique involves cancellation of common frequencies from left and right channels. As voice is generally distributed symmetrically in left and right channel (center-panned) this technique works most of the times. As we are subtracting both channels result of this technique is mono track. Other than voice removing it also lowers the amplitude of other frequencies resulting in lowering the volume. This technique gives better results than bandpass filtering. Figure-3 shows the original vs recovered channel.

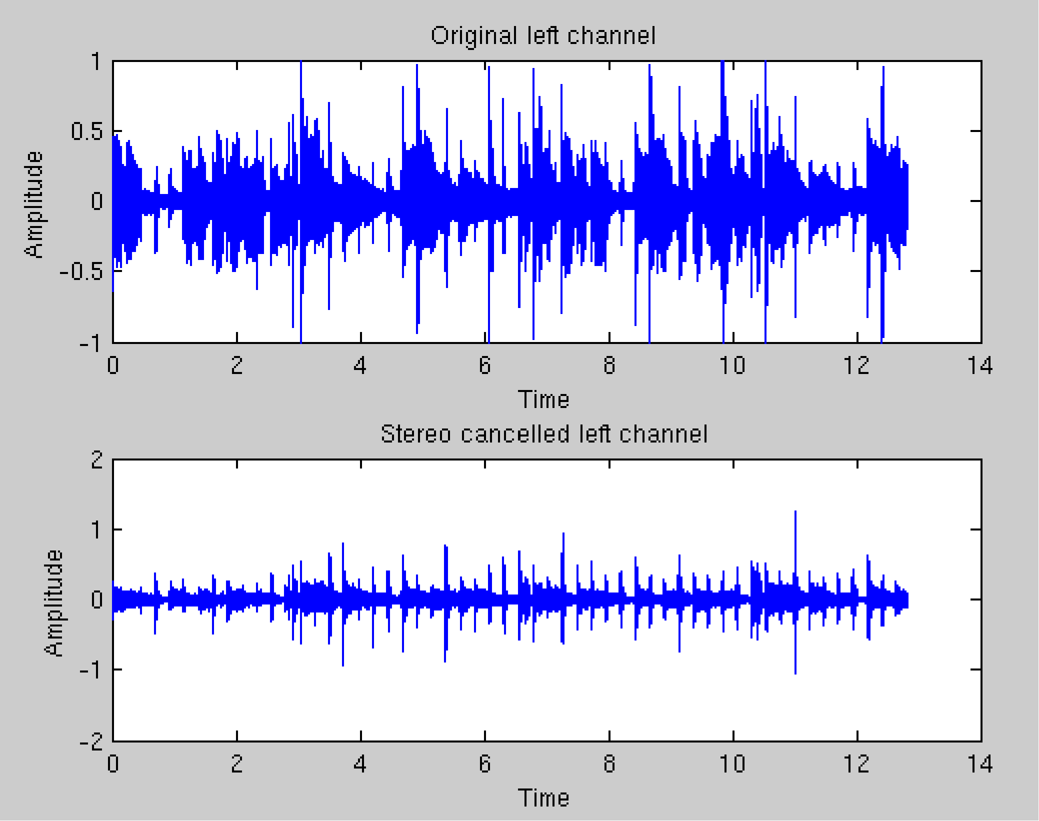

Figure 3: result of stereo cancellation

3.3 Blind Audio Source Separation:

This technique consists of “*extracting from an input audio signal a set of audio signals whose mix is perceived similarly to the original audio signal*”. In our case, we focused on extracting the vocals track from the mix consisting of the rest of the instruments.

Blind Audio Source Separation (BASS) algorithms attempt to recover one or more sources from a given observation mixture (music track) without prior knowledge or learning of the constituent individual sources. There are many ways to classify the mixture based on the sources that comprise it: broadly as music and speech.

The BASS problem can be formulated in its simplest form as follows.

$$\begin{pmatrix} out_l[k] \\ out_r[k] \end{pmatrix} = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ a_{m1} & \ldots & a_{mn} \end{pmatrix} * \begin{pmatrix} input_1[k] \\ . \\ . \\ input_n[k] \end{pmatrix}$$

A set of unknown source signals that are mutually independent of each other can be considered and denoted by input1[*k]*, input[2],… which form the source vector 'input'. With respect to audio, these signals would be the various auditory streams from the musical instruments in a music piece. These signals are recorded using sensors and are then linearly mixed using an unknown matrix of mixing filters A to form a music track.

Following figure illustrates a basic form of the Blind Audio Source Separation problem.

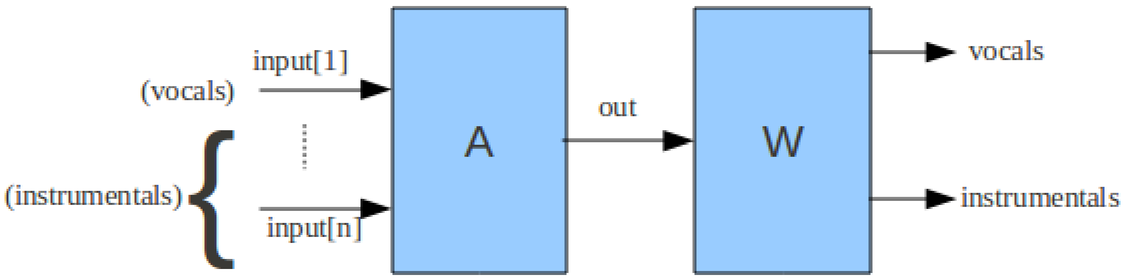

Figure 4: BASS system

Our job is to find A but as it's not known a-priory we going to estimate it as W, in our case we just want vocals to be removed so do not want to know whole W.

Following steps are taken to remove vocals.

Short Time Fourier Transform(STFT)

We took selected track (wake_up_sid.wav) and separated the left and right channels. Then we calculated STFT of left and right channel separately.

$$FT_R = STFT(Channel_R)$$

$$FT_L = STFT(Channel_L)$$

where,

$$FT_R, FT_L = STFT\ coeff\ of\ respective\ channels$$

In STFT we have to choose three factors. First, number of DFT points to be generated per frame. Second, offset between frames. Third, window type used. Choosing first two factors ensures large number of DFT coefficients could be resulted, allowing us to see more precisely the area where the vocals are concentrated. Windowing is also having more importance which reduces noise coming from neighboring frames due to overlap.

In this project number of DFT points used are 8192 and offset between frames is 8192/4 and window used is hann window. Figure-5 shows the spectrogram of right and left channel after finding STFT.

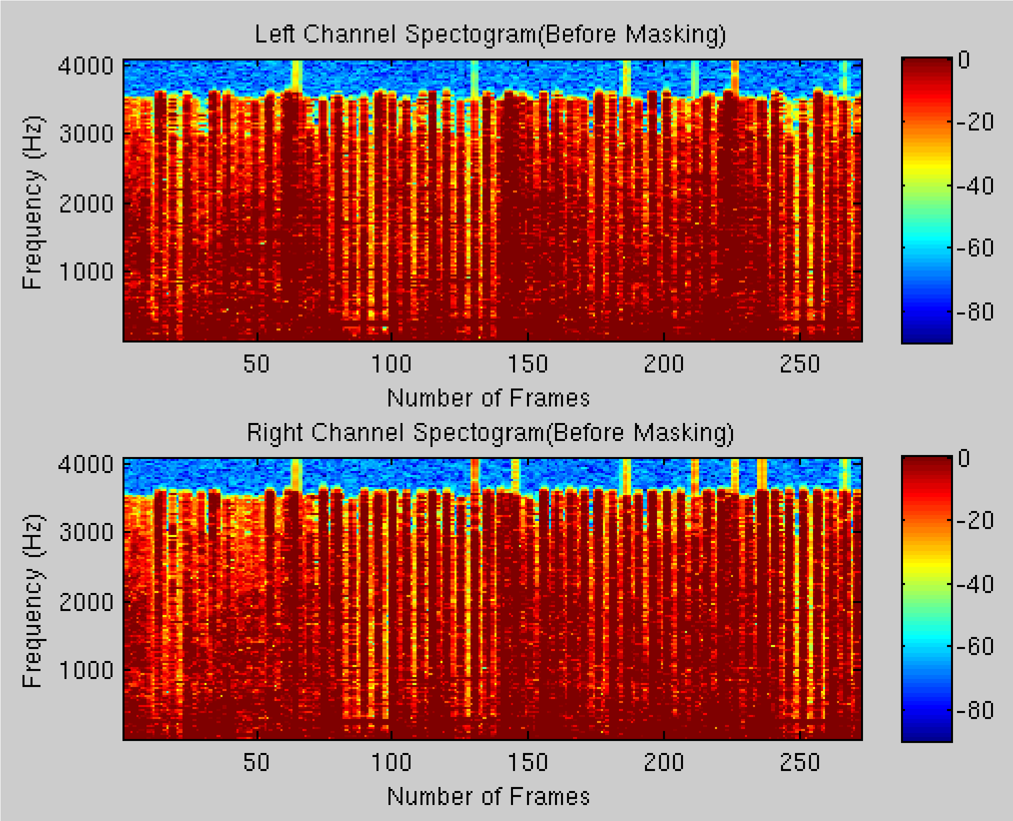

Figure 5: Spectrogram before masking

Vocal Track Identification

After having STFT we are going to find similarity in left and right channel. Our aim is to get track shared by both channels. Here we are going to take channel ratio (CR) of left and right channel respective DFT coefficients.

$$CR=\frac{|FT_R|}{|FT_L|}$$

When you divide a coefficient from the left channel (that represents a single track) by a coefficient from the right channel (that also represents the same single track), the result will be a constant value no matter where we are located in the spectrogram. However, if you divide coefficients that represent two or more tracks, your result will not be constant throughout the spectrogram anymore. In figure 6, we can see the frequency for each ratio in the spectrogram. At a ratio of 1, we found a peak. This peak represents a mono track that was inserted evenly on both channels (if the track is different to one, this means that the mono track in one of the channels was attenuated or amplified).

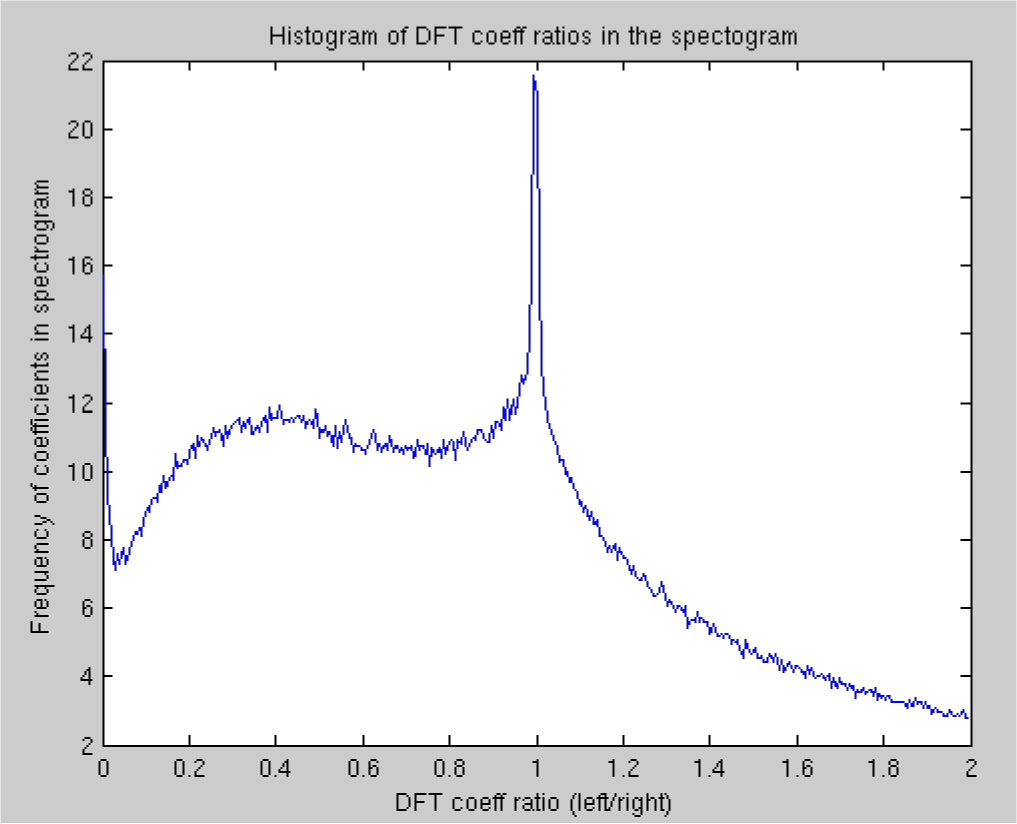

Figure 6: Histogram of channel ratios

Time-Frequency Binary Masking

Once the DFT coefficients that represent mono tracks in the channels are identified, we proceed to substitute them by zeros. This method is usually known as binary masking because the coefficients are multiplied by either one or by zero. This can be seen mathematically as follows:

$$\begin{aligned} Mask &= 0 \quad \text{....if } a \leqslant CR \leqslant b \\ &= 1 \quad \text{....else} \end{aligned}$$

For better result we have chosen range between 0.65 and 1.35 in our case.

Figure-7 shows spectrogram after application of binary mask on STFT of left and right channel. Blue spots represents the masked DFT coefficients. So if we have more DFT coefficients we can mask more area.

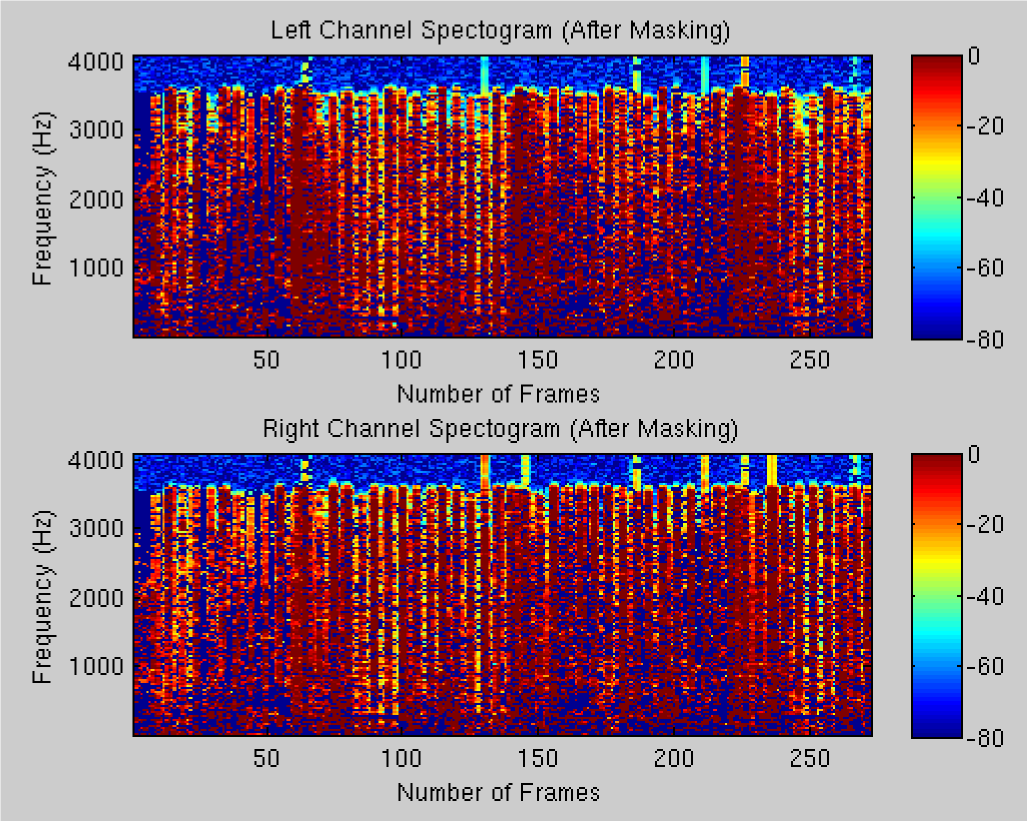

Figure 7: Spectrogram after applying Binary-mask

Getting Track Back Without Vocals

After Binary-masking we took Inverse Short-Time Fourier Transform to get back the track without vocals in it. Mathematically written as follows,

$$Channel_R = ISTFT\left(\widehat{FT}_R\right)$$

$$Channel_L = ISTFT\left(\widehat{FT}_L\right)$$

where,

$$\widehat{FT}_R, \widehat{FT}_L = STFT\ coeff\ after\ binary\ frequency\ masking$$

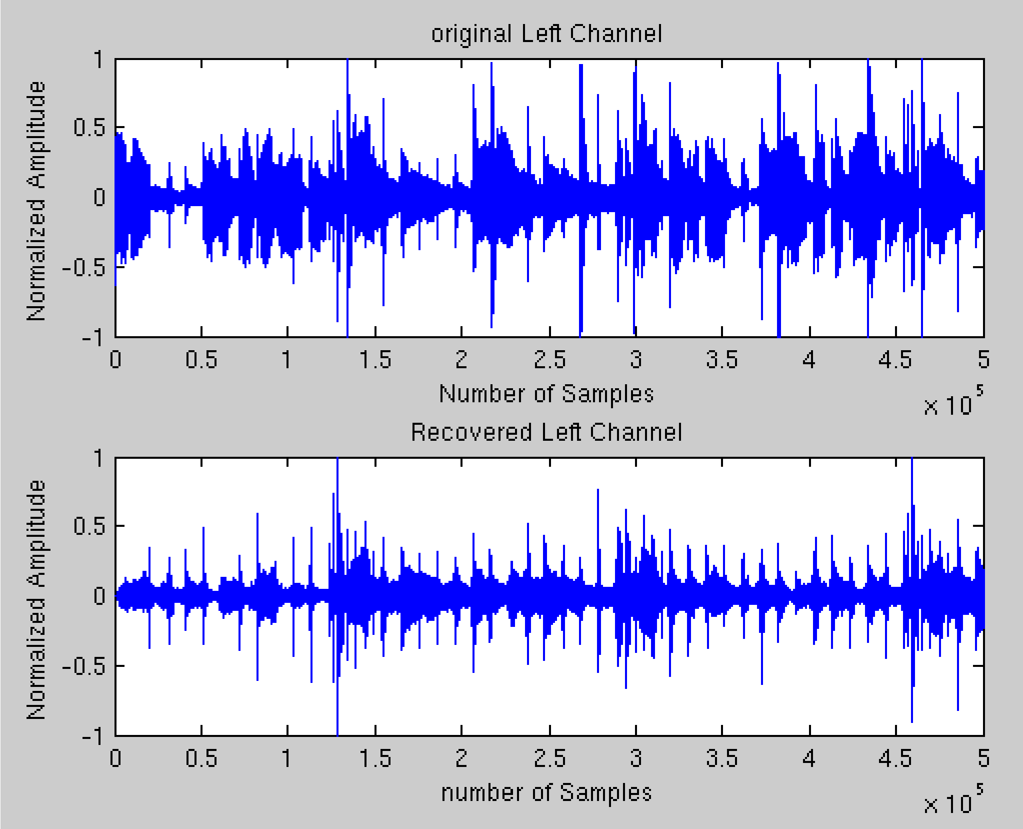

Figure 8: Original and recovered left channel

Figure 8 and figure 9 shows original and recovered left and right channel respectively. It is also seen that recovered track is also stereo but without vocals. In stereo cancellation we get mono output but here we get stereo output.

The steps explained above are one way to remove the vocals from the song. However, this technique is not limited to extracting the vocals from a song; for instance, we could extract the instruments and leave the vocals in the song.

The audio source separation technique used in this project is just one of many audio source separation approaches. This is mainly because different mixes of instruments and new sound effects intermingles frequencies in more complex ways. Due to this added complexity, a binary mask approach will not be enough to separate the sources from the song.

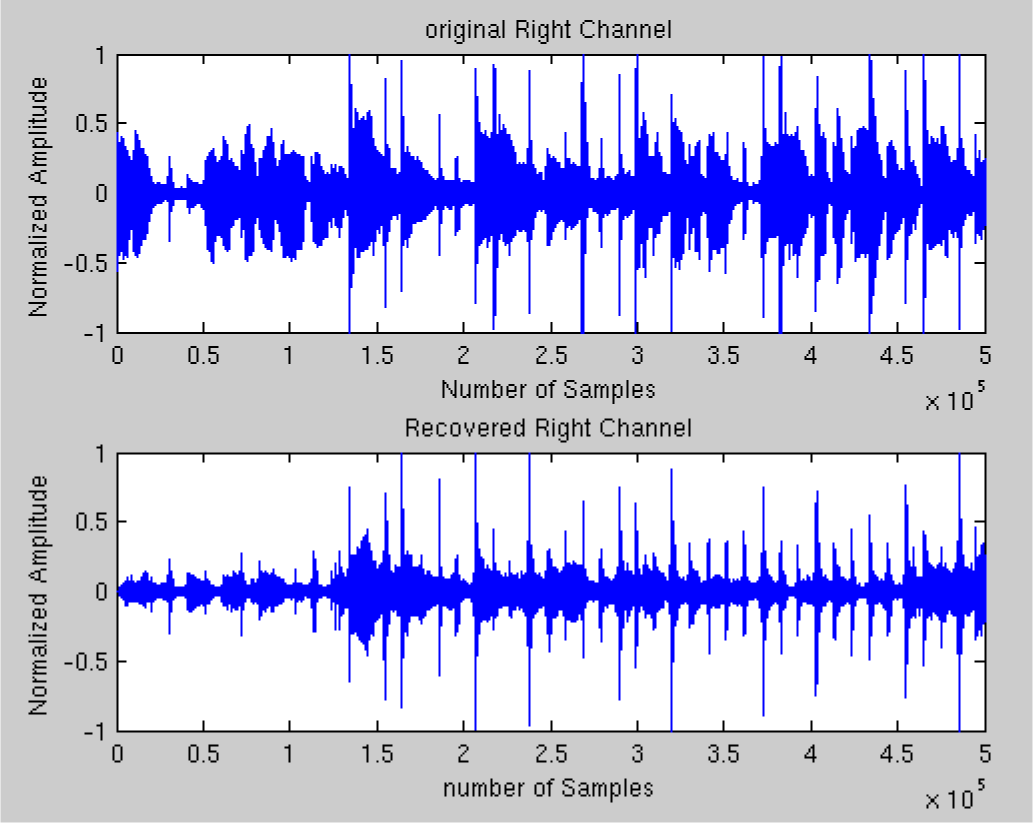

Figure 9: Original and recovered right channel

4. Conclusion

In this whole project we learned several ways of removing vocals from music tracks like, filtering but its not only removing vocals but also removes instruments in vocal range. Stereo cancellation is better than that but it gives stereo output. Finally, audio blind source separation was discussed - this technique permits the extraction of sounds at certain frequencies and time intervals. This gives us a greater selectivity in discerning vocals.

Due to huge diversity in recording techniques and extra-effects given after recording its getting difficult to remove vocals, so there is not unique procedure to remove vocals, still research is going on forward. But for simple stereo tracks this techniques gives better results.

5. References:

[1] MarC Vinyes, Jordi Bonada, Alex Loscos. *“Demixing Commercial Music Productions via Human -Assisted Time-Frequency Masking”* Presented at the Audio Engineering Society, Paris, France, 2006.

[2] Ivan W. Selesnick, *“Short-Time Fourier Transform and Its Inverse”*