PROJECT
Presentation

Nattarach Saekaow 6210550380

Tatpong Thunyaudom 6210552617

Jakkaphob Kongthanarith 6210554725

Digital Signal Processing
01205323

How signal of interest are physically generated?

1. Human voice

Human voices are produced in vocal cords when they come together and then vibrate as air passes through them during exhalation of air from the lungs. This vibration produces the sound wave for your voice.

2. Instruments

Most instruments make sound when they vibrate or there is something hitting it. The vibrate instruments are like Guitar, Violin etc. but most of the instruments make the air around the object vibrate and the air around it also vibrate until it enters your ear. That is why you hear can hear instruments as sounds.

How can we separate vocal from instrumental?

How can we separate sound?

Source separation, blind signal separation (BSS) or blind source separation, is the separation of a set of source signals from a set of mixed signals, without the aid of information (or with very little information) about the source signals or the mixing process. It is most commonly applied in digital signal processing and involves the analysis of mixtures of signals; the objective is to recover the original component signals from a mixture signal.

Many music/voice separation methods typically first identify the vocal/non-vocal segments and then use a variety of techniques to separate the lead vocals from the background music. Musical pieces are often composed of repeating background on voice which does not show a regular repeating structure.

Study related solution/knowledge from the internet

From our research of related projects about separating instrumental from vocal. We found that most methods will be related to Sound Source Separation and Fast Fourier Transform (FFT). We choose to do this through Matlab by using Fast Fourier Transform to separate frequency range and keep choosing it until we can find the suitable range that can hear instrumental clearly. As we kept researching about this, this prove to be a challenge and complicated topic when we do the coding but it is possible to do it.

Method

We use any song that has the vocal and instrumental

Method of collecting the signals of interest

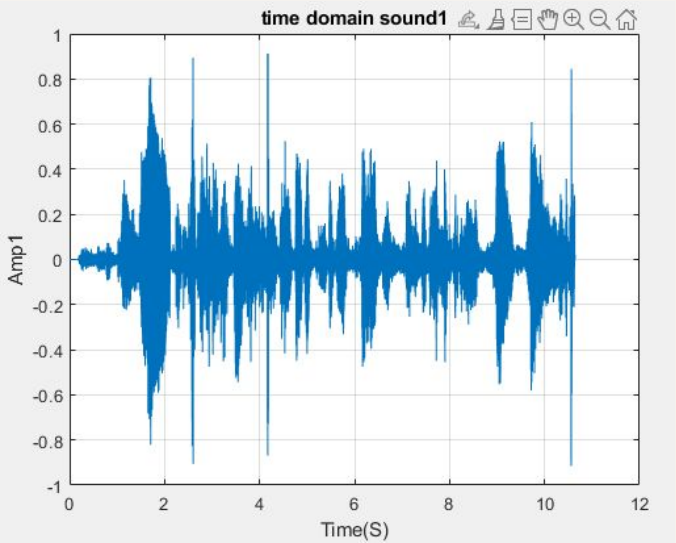
- We collected our samples from website or we recorded our voices while playing music.
- Choose some parts of song to analyze.
- We use MATLAB to do coding to separate instrumental from vocal.
    - We will use FFT to do the separation.
- We use MATLAB to analyze output data.

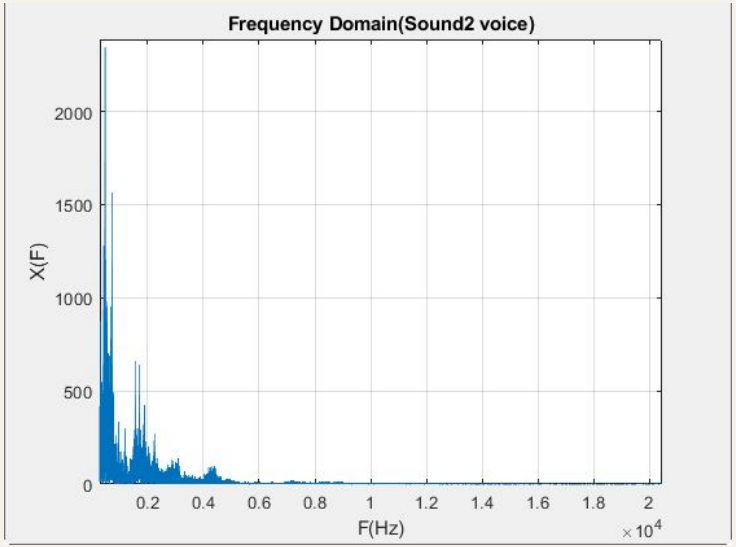
What is Fast Fourier Transformation (FFT)?

The "Fast Fourier Transform" (FFT) is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal.

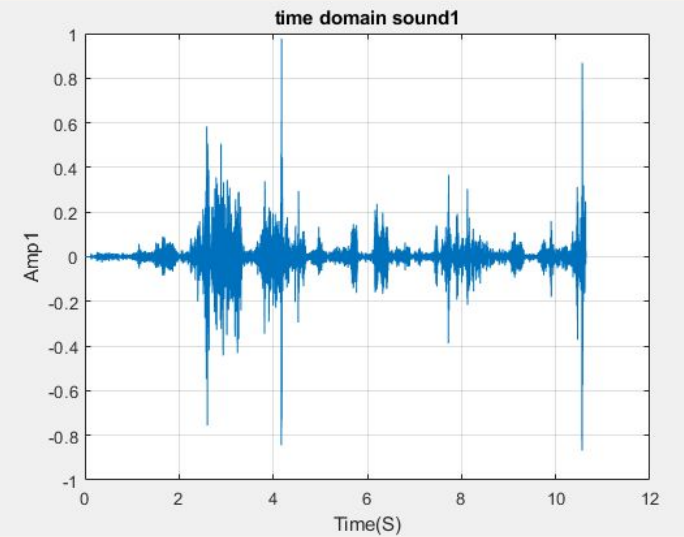
In other word, the FFT is an optimized algorithm for the implementation of the "Discrete Fourier Transformation" (DFT). A signal is sampled over a period of time and divided into its frequency components. These components are single sinusoidal at distinct frequencies each with their own amplitude and phase.

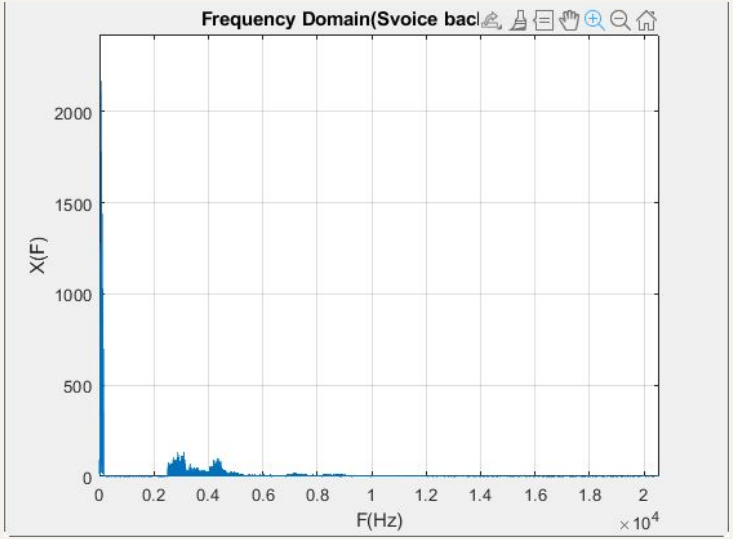
Sound1 (Original)



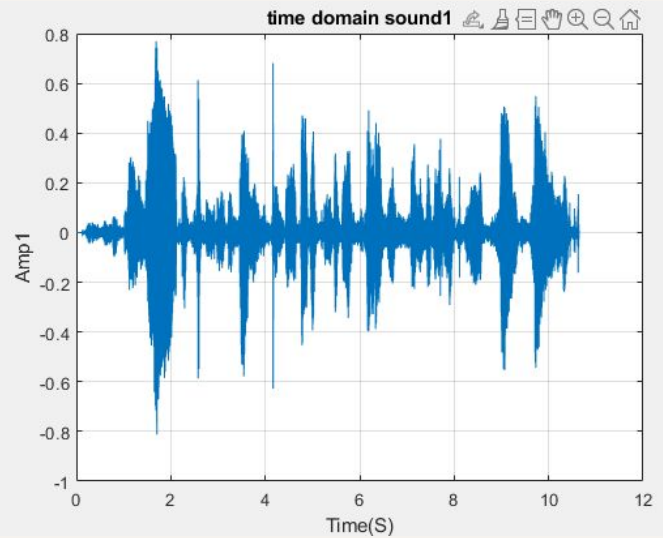


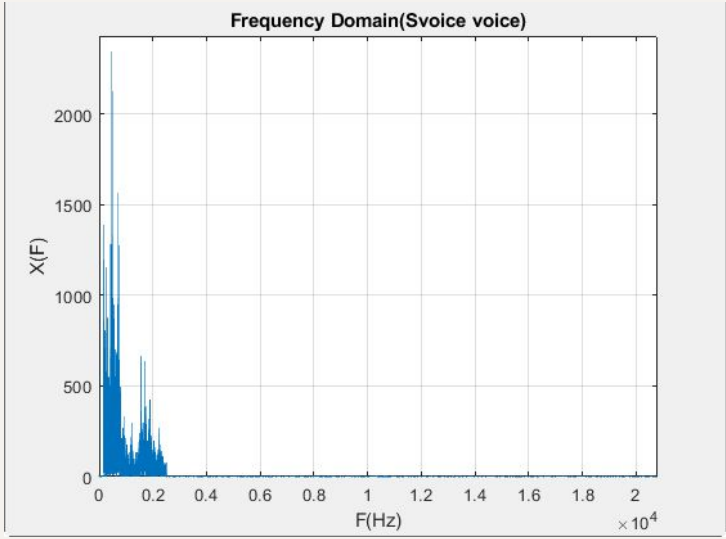
Sound_Background1



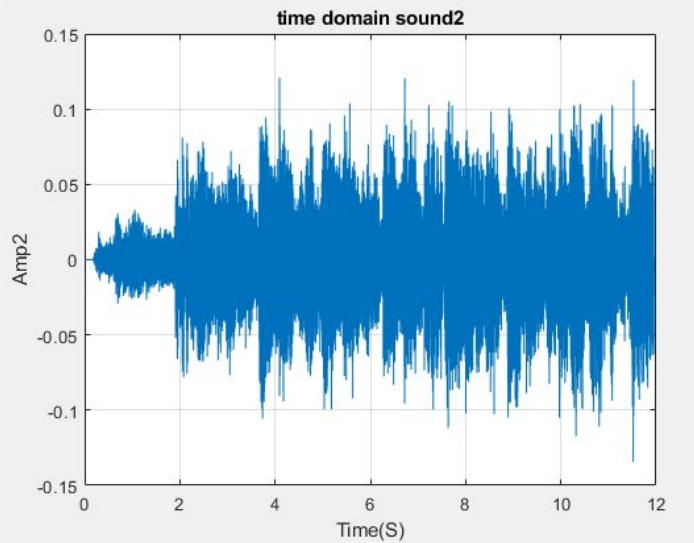


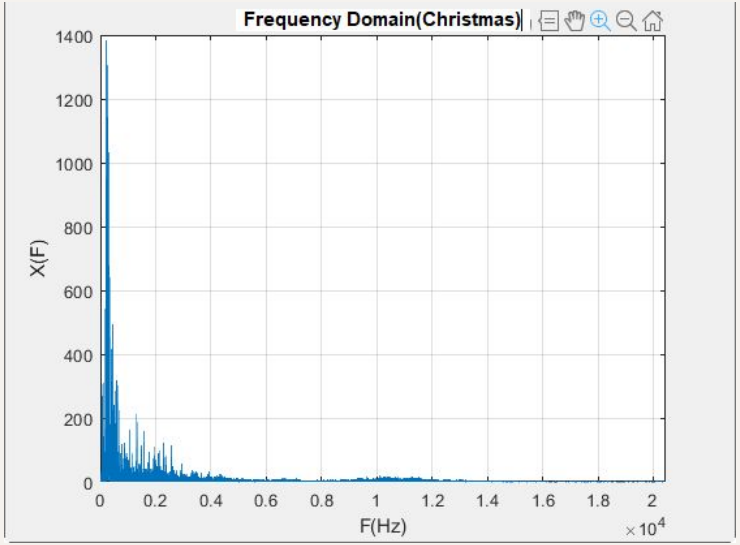
Sound_voice1



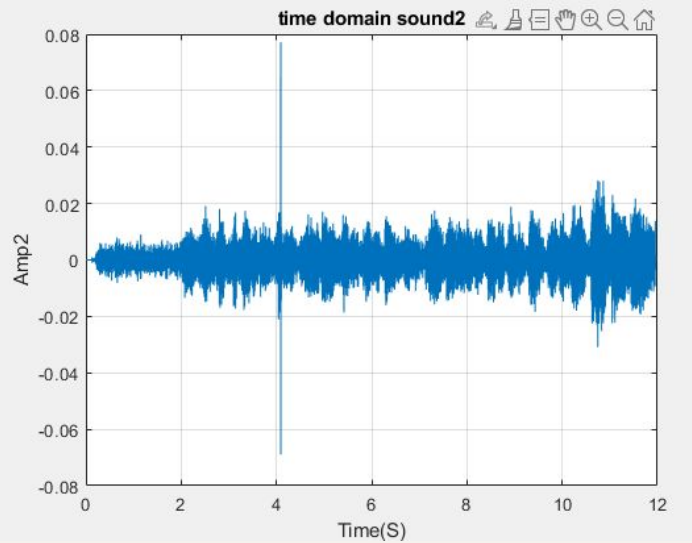


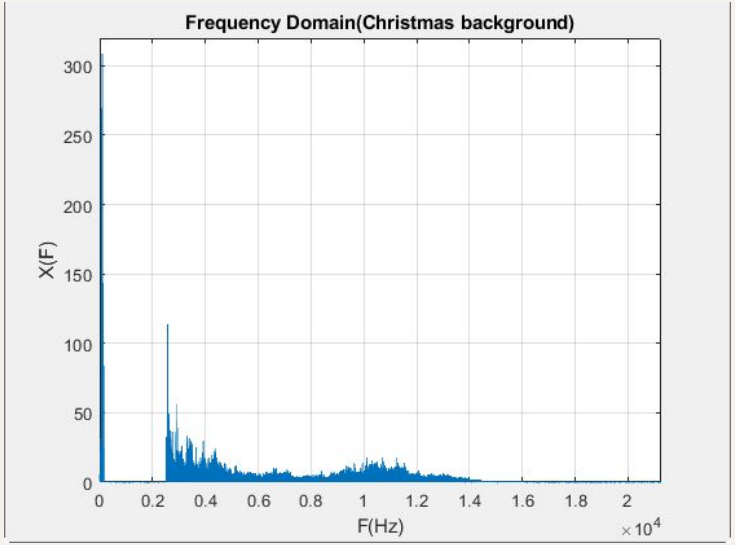
Sound2 (Original)



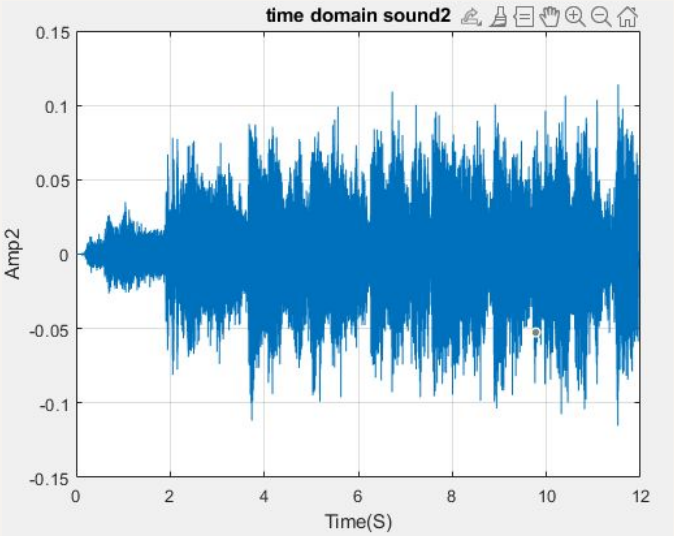


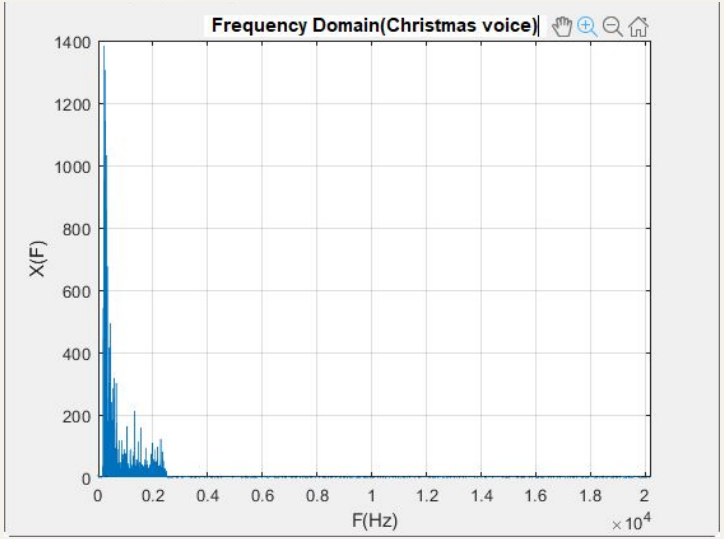
Sound_Background2 (Original)





Sound_voice2





Code

```
[audio_in,audio_freq_samp1] = audioread('sound.wav');
length_audio = length(audio_in);
df = audio_freq_samp1/length_audio;
frequency_audio = -audio_freq_samp1/2:df:audio_freq_samp1/2-df;
FFT_audio_in = fftshift(fft(audio_in)/length(fft(audio_in)));
lower_threshold = 150;
upper_threshold = 2500;
val = abs(frequency_audio)<upper_threshold & abs(frequency_audio)>lower_threshold;
FFT_ins = FFT_audio_in(:,1);
```

```
FFT_voc = FFT_audio_in(:,1);
FFT_ins(val) = 0;
FFT_voc(~val) = 0;
FFT_a = ifftshift(FFT_audio_in);
```

Conclusion

From the graph result, we can see that the background part signal is a lot smaller compare to the original and voice part. The output audio is also different when compared to the original and the instrument inside the Background part can be barely audible.

The vocal part signal is a lot similar when compared to the original. The audio output is a lot closer to the original and the vocal can be clearly heard even though there are some background music mixing in.

From the experiment, we can conclude that the solution we made is possible to separate the audio signal by using FFT through MATLAB. We used FFT as our main tool and we blindly chose frequency range to pinpoint the range of vocal and instruments. The components we separated are background music and vocal. During our experiment with codes and trial and error. We found several problems

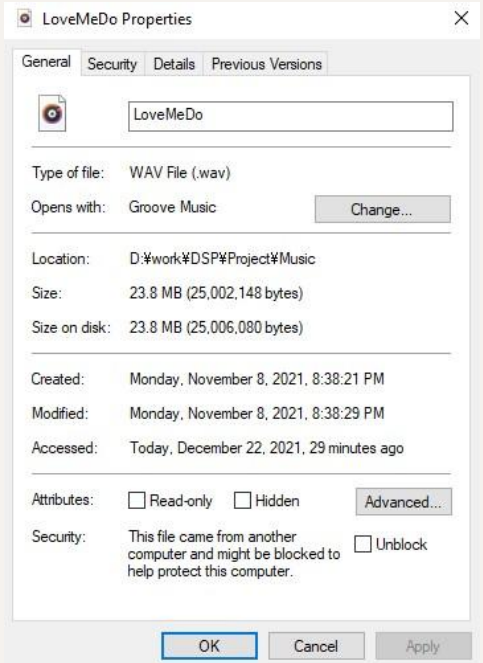
1. The background music can be heard in vocal part and vocal and be heard in background music part.

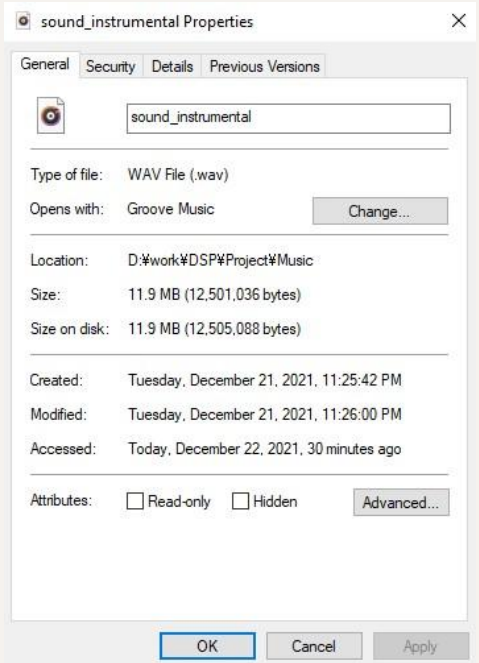
To make each sound clear as crystal. We believe it is not possible to achieve due to the song (vocal and instruments) was already mixed together and its difficulty and complicated process to make it clear. The best way we can make it clear is to retrace or re-track the whole song until we get the original audio back. Usually most songs contain repeating structure according to musical theory and once we get the repeating part correct, the rest should be the same.

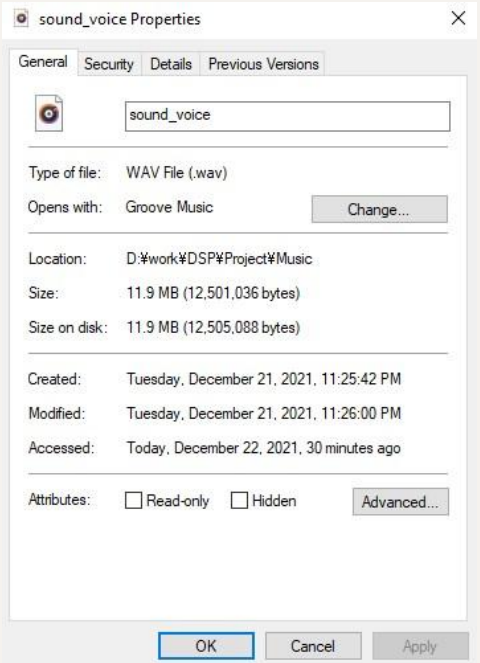
2. Sound quality is dropped by half.

We do not know the real reason behind this. It may be with MATLAB problem that cause this problem but we suspected that the moment we separated the frequency according to the code: **upper_threshold = 2500** and **lower_threshold = 150**. the moment we separated the code and its frequency is separated the audio quality is reduced and the file sizes are also reduced by half as seen in the following pictures.

Conclusion


LoveMeDo Properties
General
Security
Details
Previous Versions
LoveMeDo
Type of file: WAV File (.wav)
Opens with: Groove Music
Change...
Location: D:¥work¥DSP¥Project¥Music
Size: 23.8 MB (25,002,148 bytes)
Size on disk: 23.8 MB (25,006,080 bytes)
Created: Monday, November 8, 2021, 8:38:21 PM
Modified: Monday, November 8, 2021, 8:38:29 PM
Accessed: Today, December 22, 2021, 29 minutes ago
Attributes: Read-only Hidden
Advanced...
Security: This file came from another computer and might be blocked to help protect this computer.
Unblock
OK
Cancel
Apply


sound_instrumental Properties
General
Security
Details
Previous Versions
sound_instrumental
Type of file: WAV File (.wav)
Opens with: Groove Music
Change...
Location: D:¥work¥DSP¥Project¥Music
Size: 11.9 MB (12,501,036 bytes)
Size on disk: 11.9 MB (12,505,088 bytes)
Created: Tuesday, December 21, 2021, 11:25:42 PM
Modified: Tuesday, December 21, 2021, 11:26:00 PM
Accessed: Today, December 22, 2021, 30 minutes ago
Attributes: Read-only Hidden
Advanced...
OK
Cancel
Apply


sound_voice Properties
General
Security
Details
Previous Versions
sound_voice
Type of file: WAV File (.wav)
Opens with: Groove Music
Change...
Location: D:¥work¥DSP¥Project¥Music
Size: 11.9 MB (12,501,036 bytes)
Size on disk: 11.9 MB (12,505,088 bytes)
Created: Tuesday, December 21, 2021, 11:25:42 PM
Modified: Tuesday, December 21, 2021, 11:26:00 PM
Accessed: Today, December 22, 2021, 30 minutes ago
Attributes: Read-only Hidden
Advanced...
OK
Cancel
Apply

Reference

- https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft
- https://pseeth.github.io/public/papers/seetharaman_2dft_waspaa2017.pdf
- Fast Fourier transform - MATLAB fft (mathworks.com)
- Signal separation – Wikipedia
- https://www.mathworks.com/matlabcentral/answers/506350-separate-voice-from-background-music
- https://sites.google.com/site/singingvoiceseparationrpca/