

**Middle East Technical University**

**Electrical-Electronics Engineering Department**

**EE430 Digital Signal Processing Term Project (2017-2018 Fall Semester)**

**Group X**

**Mehmet Mert Beşe 2093482**

**Berk İskender 2031920**

**Comments on spectrogram computation and its utilization:**

Before skipping to the works related to phase 2 of the project, we want to include details about how to select overlap amount and the purpose of windowing while using spectrogram function that we have constructed.

While obtaining the spectrogram of a signal, there is a tradeoff between spectral leakage and spectral resolution. A window that results in less spectral leakage has smaller side lobes in its Fourier transform. However, it distorts the input signal unavoidably because of the increased main lobe width in its Fourier transform. Increased main lobe width limits the spectral resolution per window length.

To increase the resolution, window length can be increased. Being an almost optimal value, we tried to use %50 overlapping per window while displaying our results.

1. **Reducing the Number of Time Samples:**

In this part of the project, we are asked to reduce the total amount of time samples in a periodical manner (downsampling) with different ratios. (e.g. discard one of each N time domain samples) To perform this operation, we used built-in functions of MATLAB which are “downsample” and “decimate”.

**Effects of increasing downsampling rate:** As downsampling factor increases and exceeds the value of pi/W (W is the highest frequency present in the signal) aliasing starts to occur. As a result of this fact, starting from the higher frequencies, gisnal gets aliased (distorted). When we apply this effect on linear chirp with sufficiently high bandwidth (m value in the chirp) or any signal that includes higher frequencies, signal has doubled higher frequency components due to aliasing. This effect can be easily seen in linear chirp case. As we increase the ratio, high frequency parts starts to get repeated in the spectrogram which means that the power of those corresponding frequencies are doubled.

As the downsampling rate incerases, signal becomes shorter and its speed gets multiplied by the downsampled factor when played using the sampling frequency rate.

Moreover, same effects can be observed in the recorded sound signal and input data file when we apply downsampling and decimation separately. As we apply downsampling, frequencies present in those signals are stretched towards the higher values (proportional to the downsampling factor). After this factor times maximum frequency of the signal exceeds the half of the sampling frequency, aliasing starts to occur. Decimation nearly eliminates all of the aliasing effect in those two cases too.

**Avoiding aliasing using decimation:** In order to avoid this aliasing effect, decimation should be applied on the signal instead of downsampling. Decimation initially filters the signal (low pass filtering) with a pi/M cutoff frequency in order to avoid aliasing after stretching the frequency doiman response due to downsampling. This technique may also result in loss of some high frequency components in the filtering stage while avoiding aliasing. Results obtained using decimation are added in order to display the behavior explained above.

**Applying fractional downsampling factors:** To use fractional downsampling factors, upsampling should be done first. For instance, if the signal is to be “downsampled” by a factor of 2.5, it should be upsampled by 2 firstly and downsampled by 5 afterwards. While doing so, we need to use a “low pass filter” after upsampling in order to avoid eliminate frequencies higher than the W/L, where W is the bandlimit of the input signal. By doing so, there will not be repetitions at the output of the downsampler. This can be also achieved by applying decimation on the upsampled signal. By comparing both results of 2.5 factor downsampling, one can easily see that decimation provides a nearly ideal result.

Results for 3 different signals are listed below:

1. Linear Chirp signal, starting from 0 to 1 kHz, 5 seconds long: While obtaining spectrograms, Hamming window is used. Overlap amount is 50% in the figures for each window.

* Time Domain signals for different downsampling rates:

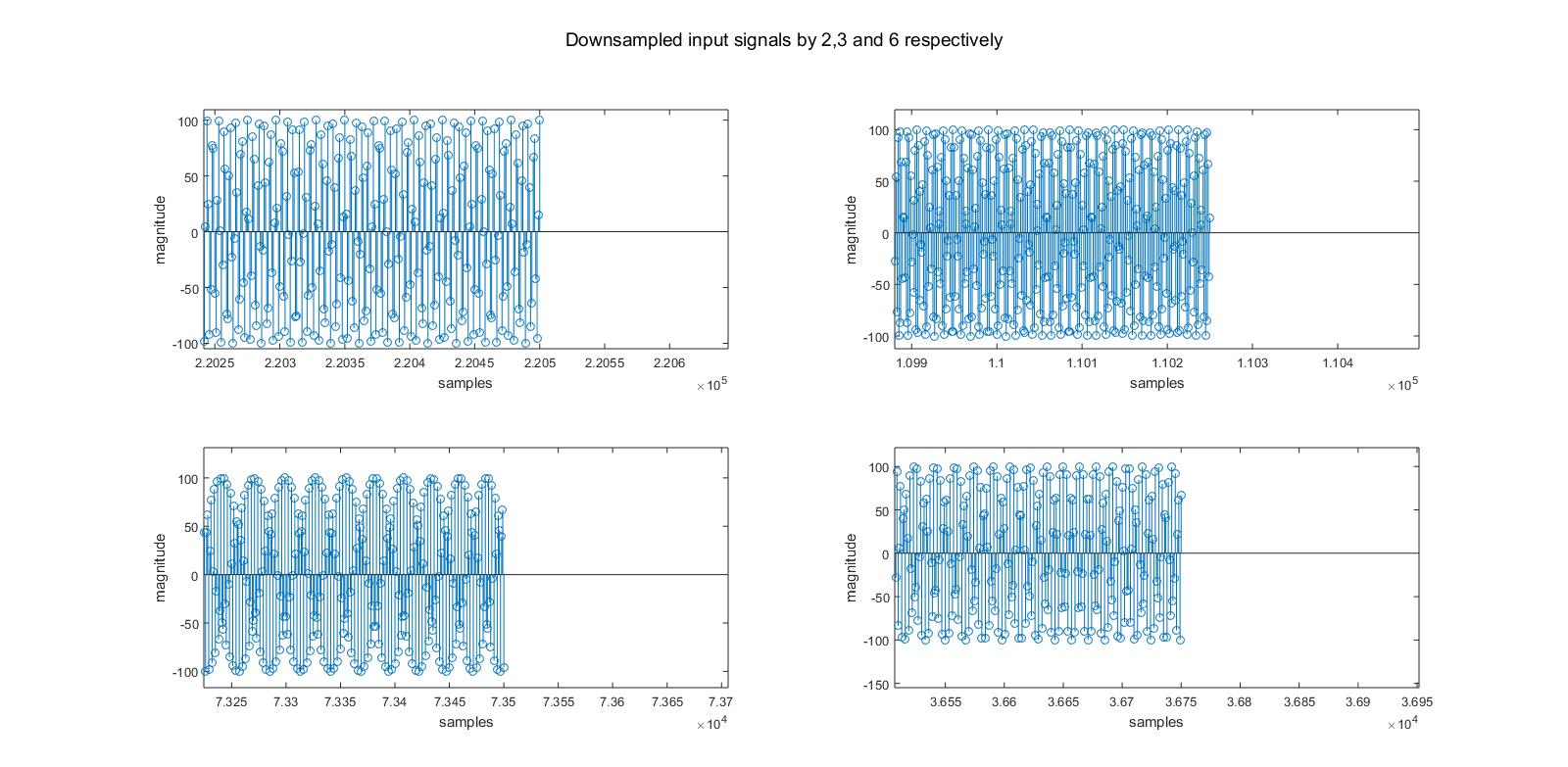


Figure: Fractions of time domain linear chirp signals (ends of the signals are extracted as fractions)

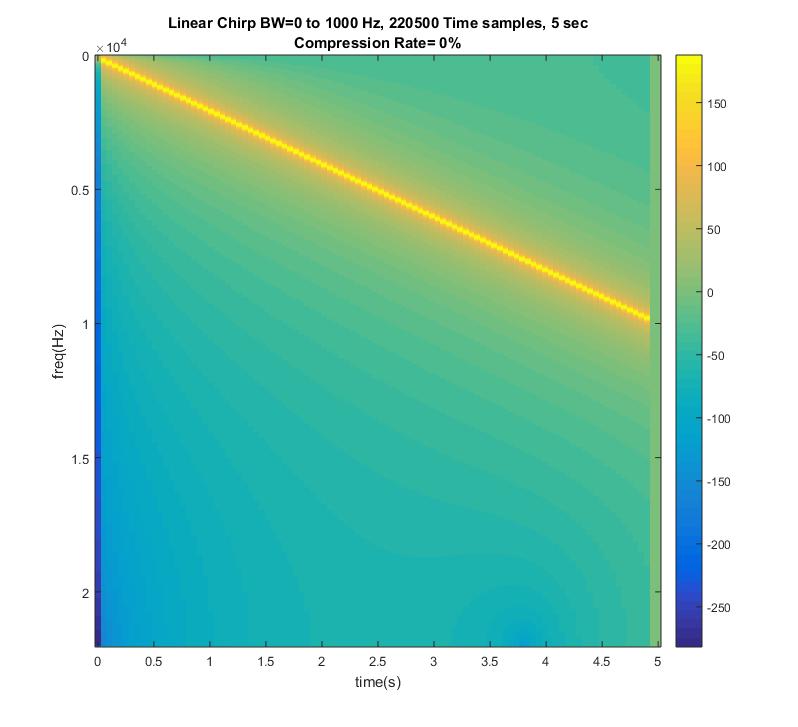


Figure: Spectrogram of the original linear chirp signal

* Results obtained using downsampling:

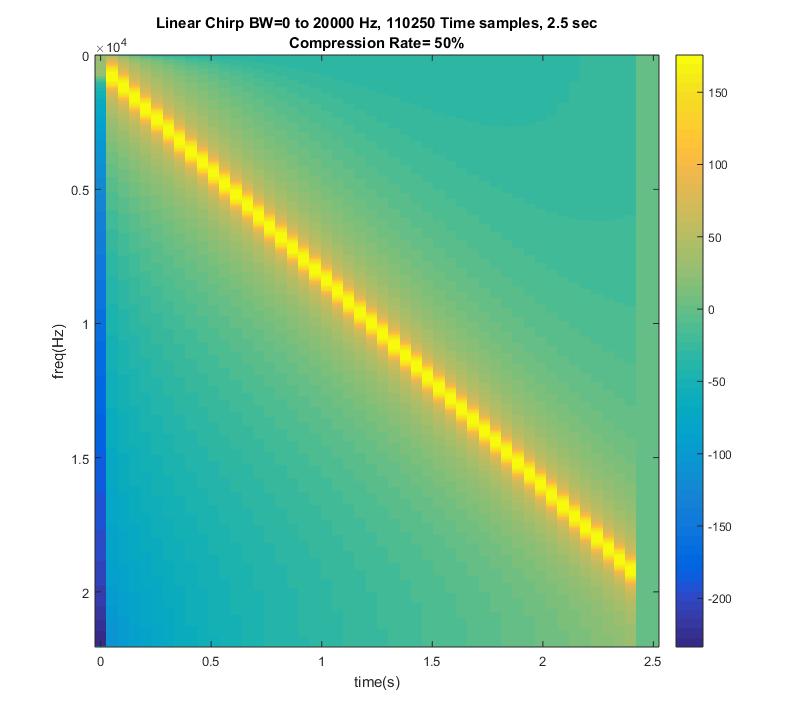


Figure: Spectrogram of the linear chirp signal downsampled by 2

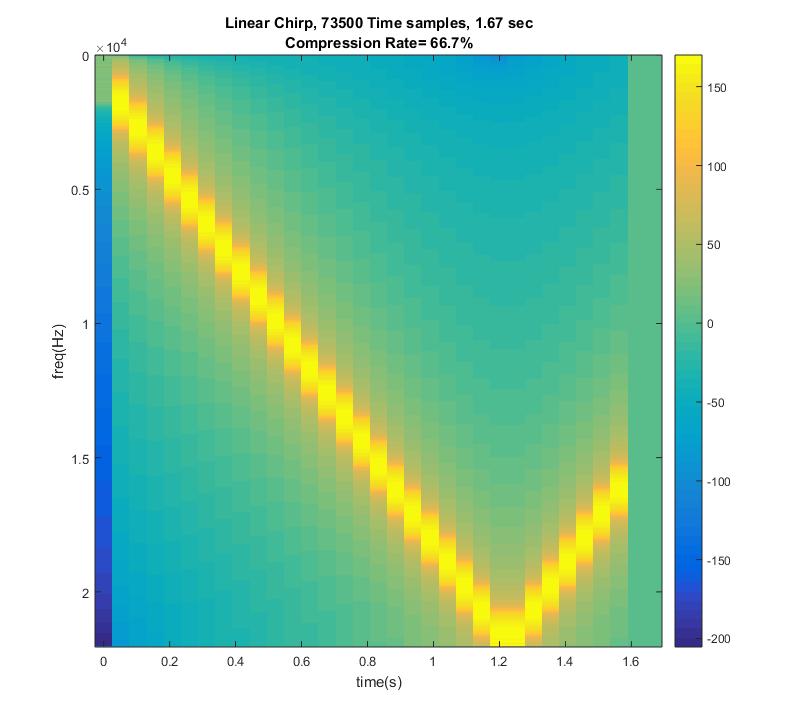


Figure: Spectrogram of the linear chirp signal downsampled by 3

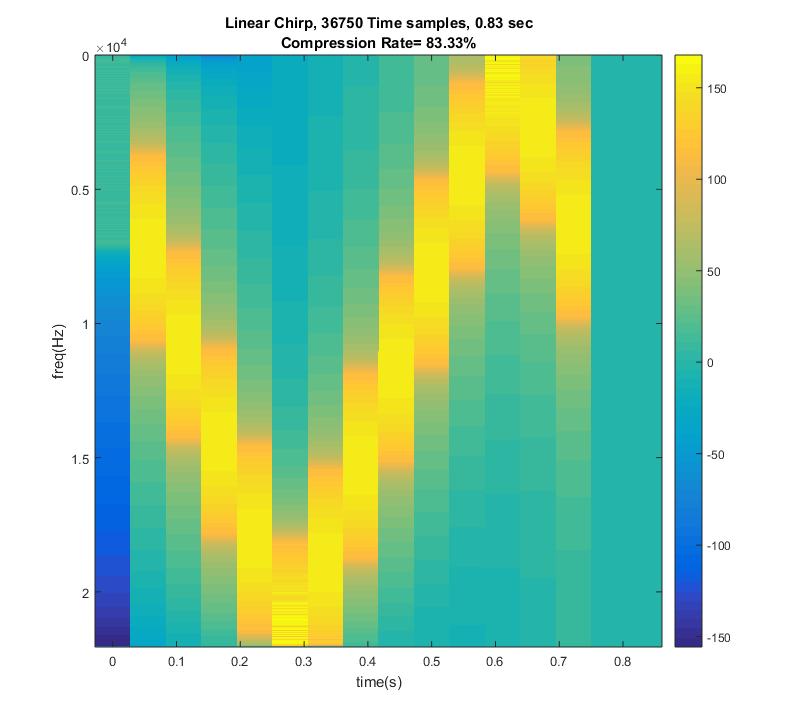


Figure: Spectrogram of the linear chirp signal downsampled by 6

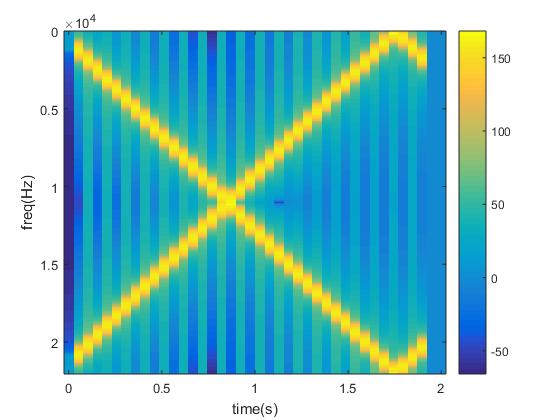


Figure: Spectrogram of the linear chirp signal downsampled by 2.5

As it can be seen in the figure, repetitions occur in the frequency domain because a LPF after the upsampling process is not used.

* Results obtained using decimation:

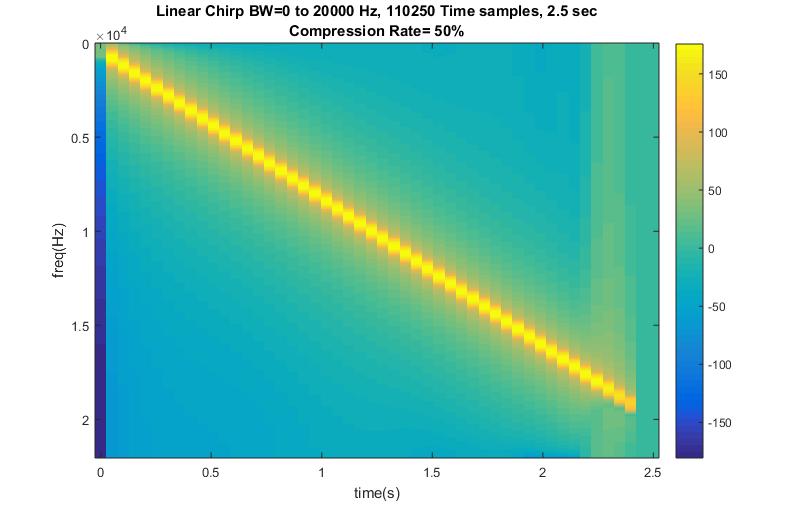


Figure: Spectrogram of the linear chirp signal decimated by 2

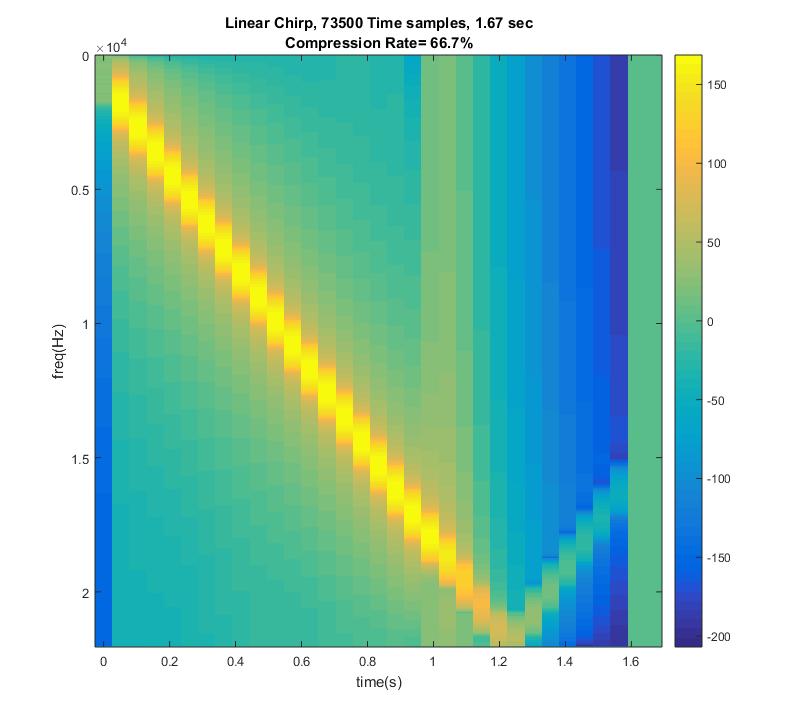


Figure: Spectrogram of the linear chirp signal decimated by 3

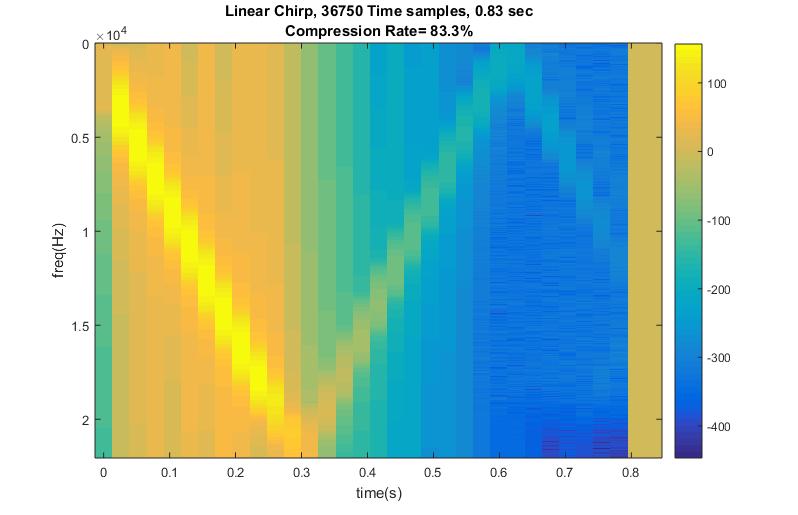


Figure: Spectrogram of the linear chirp signal decimated by 6

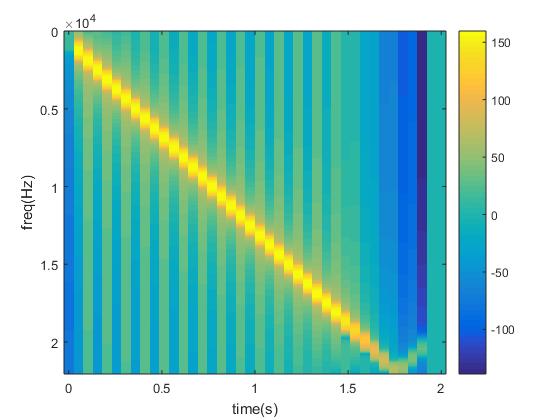


Figure: Spectrogram of the linear chirp signal decimated by 2.5

1. Sound file input:

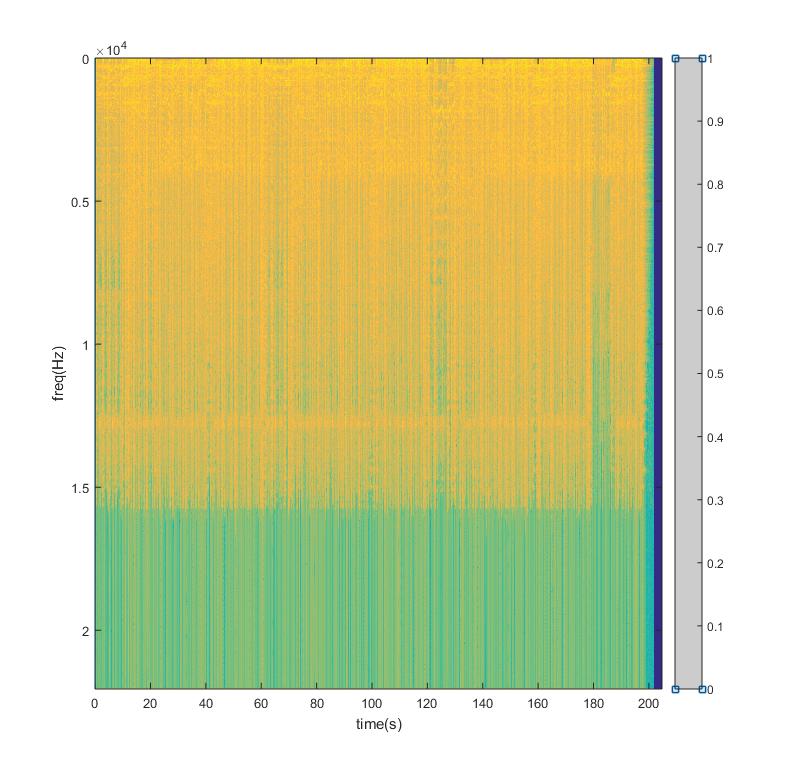


Figure: Spectrogram of the sound file input signal

* Decimation applied on a sound file input:

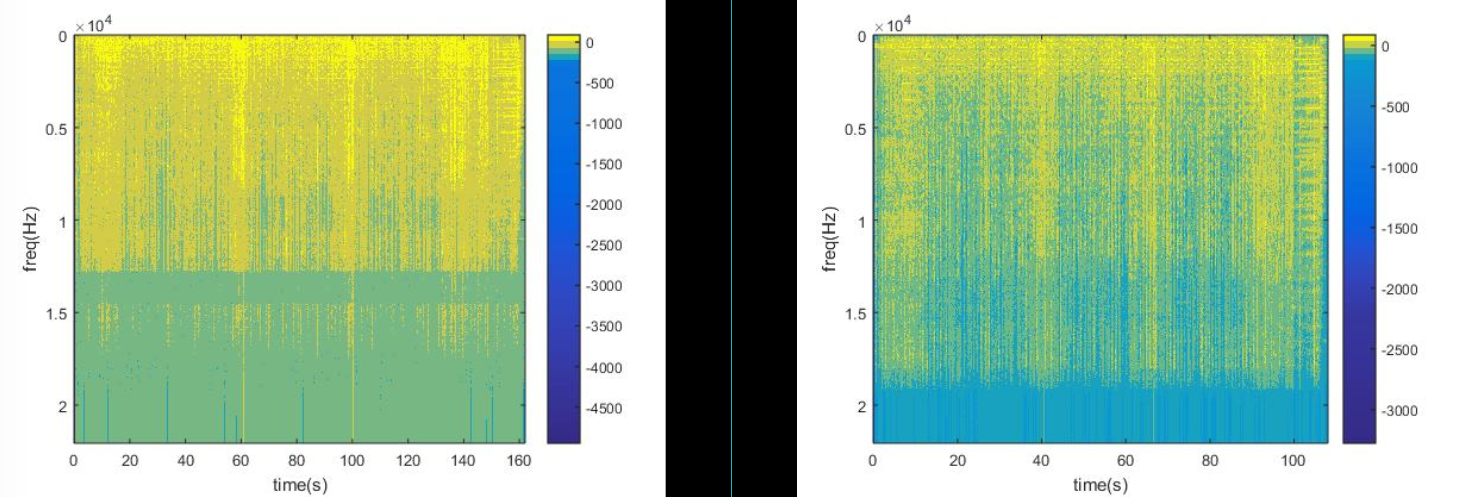


Figure: Spectrogram of the sound file input signal decimated by 2 and 3 respectively

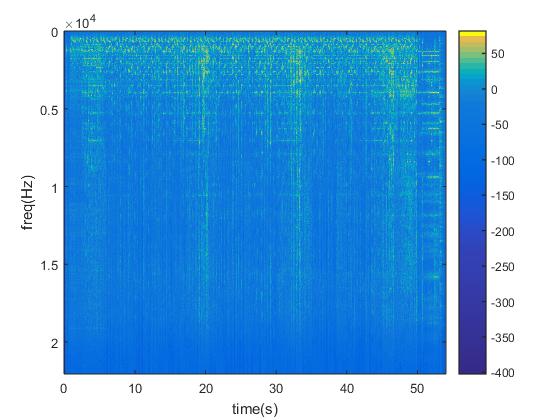


Figure: Spectrogram of the sound file input signal decimated by 6

* Downsampling applied on a sound file input:

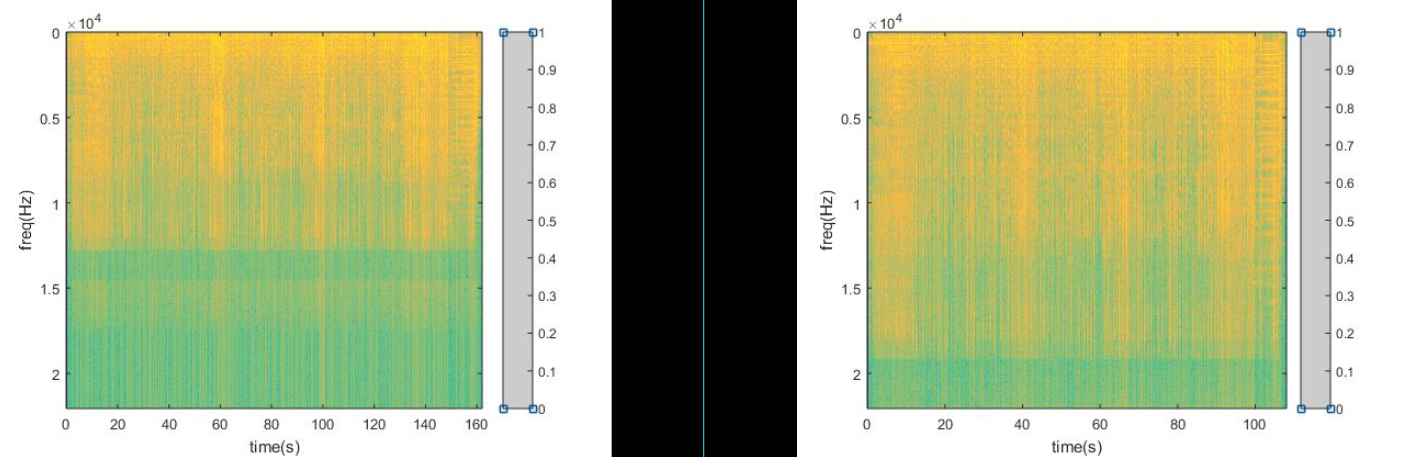


Figure: Spectrogram of the sound file input signal downsampled by 2 and 3 respectively

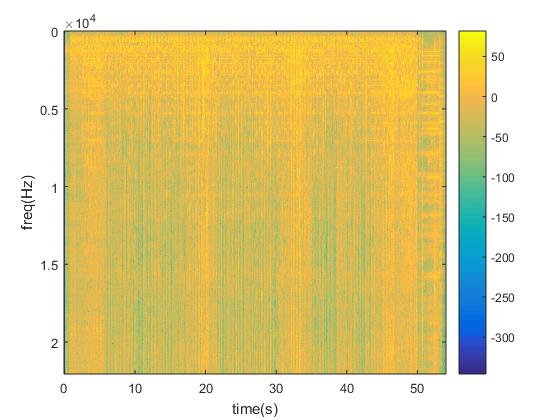


Figure: Spectrogram of the sound file input signal downsampled by 6

1. **Recorded signal input:**

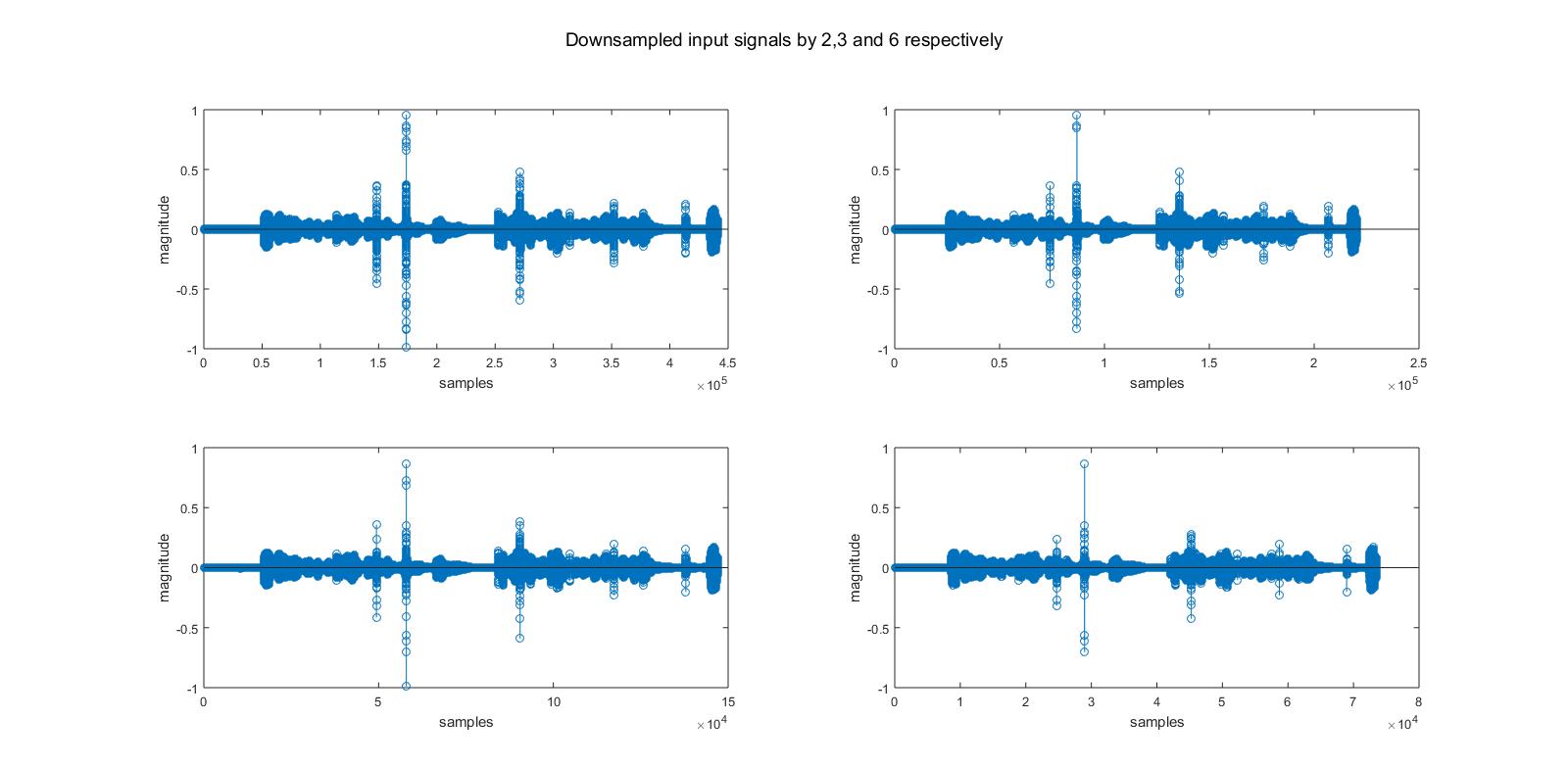


Figure: Time domain representations of the original, downsampled by 2, 3, 6 respectively

* Recorded signal downsampled:

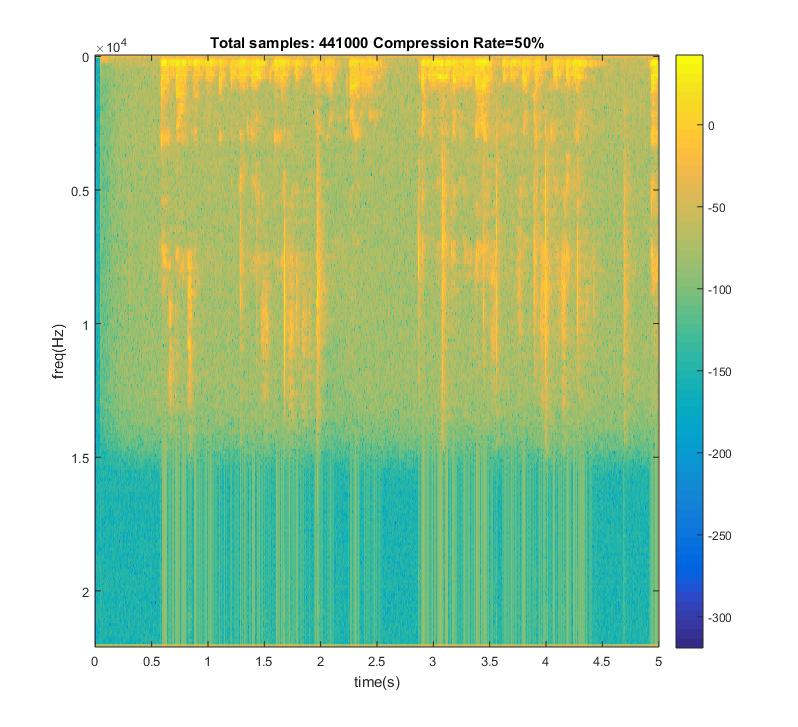


Figure: Spectrogram of the recorded signal downsampled by 2

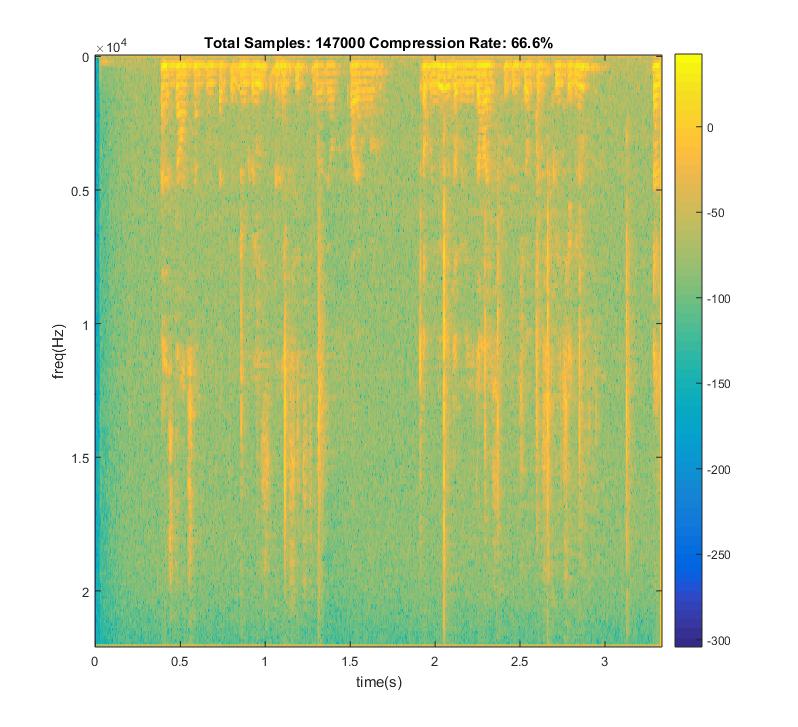


Figure: Spectrogram of the recorded signal downsampled by 3

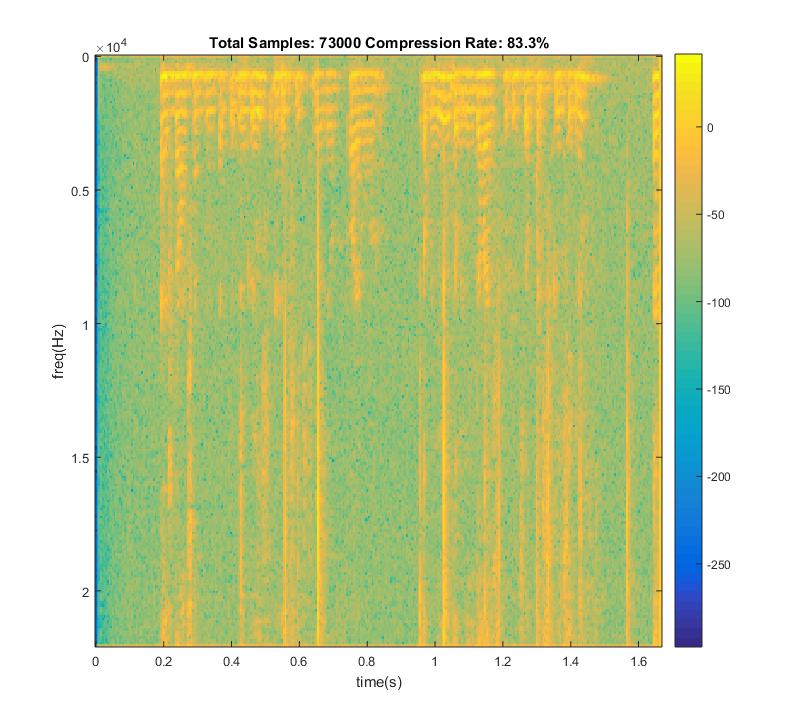


Figure: Spectrogram of the recorded signal downsampled by 6

* Recorded signal decimated:

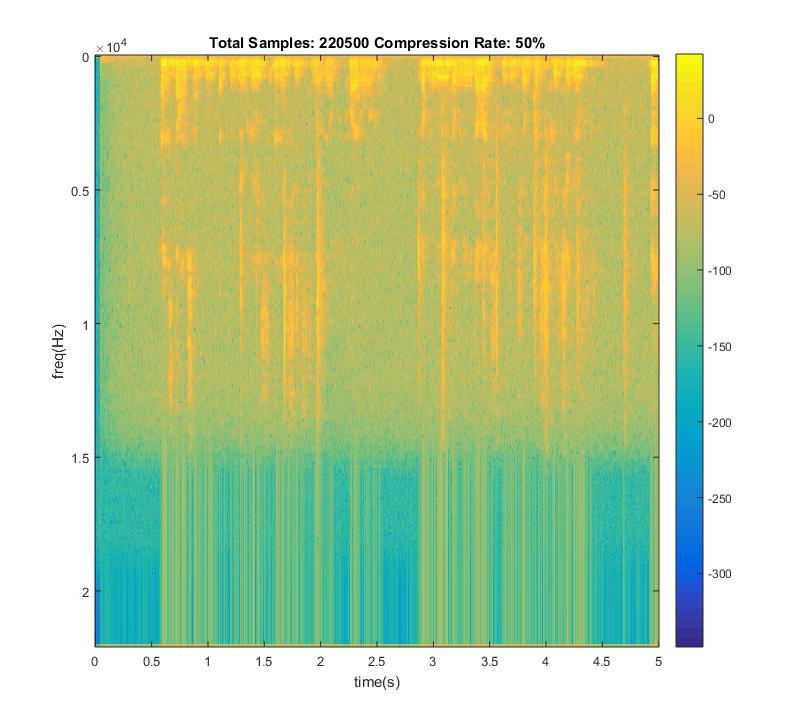


Figure: Spectrogram of the recorded signal decimated by 2

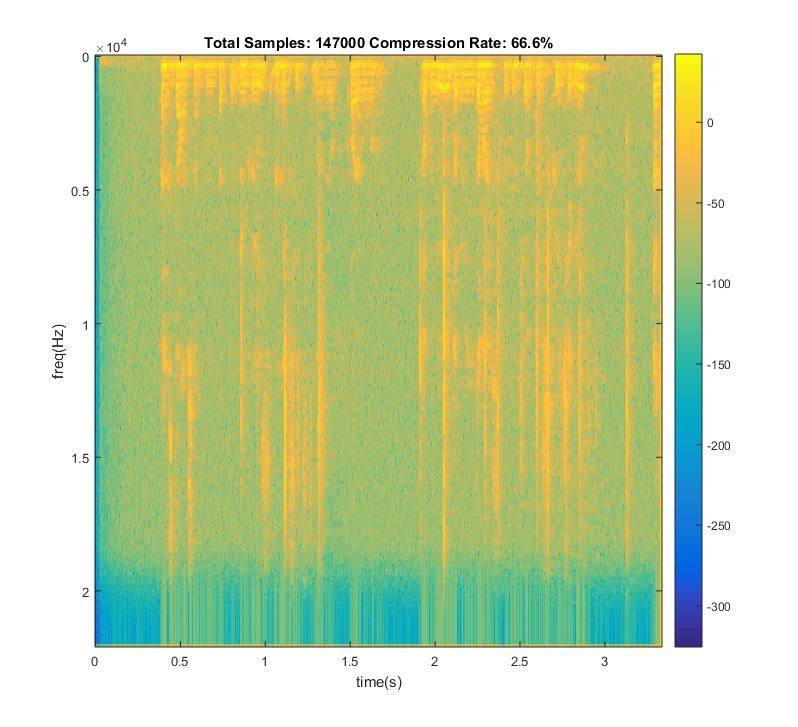


Figure: Spectrogram of the recorded signal decimated by 3

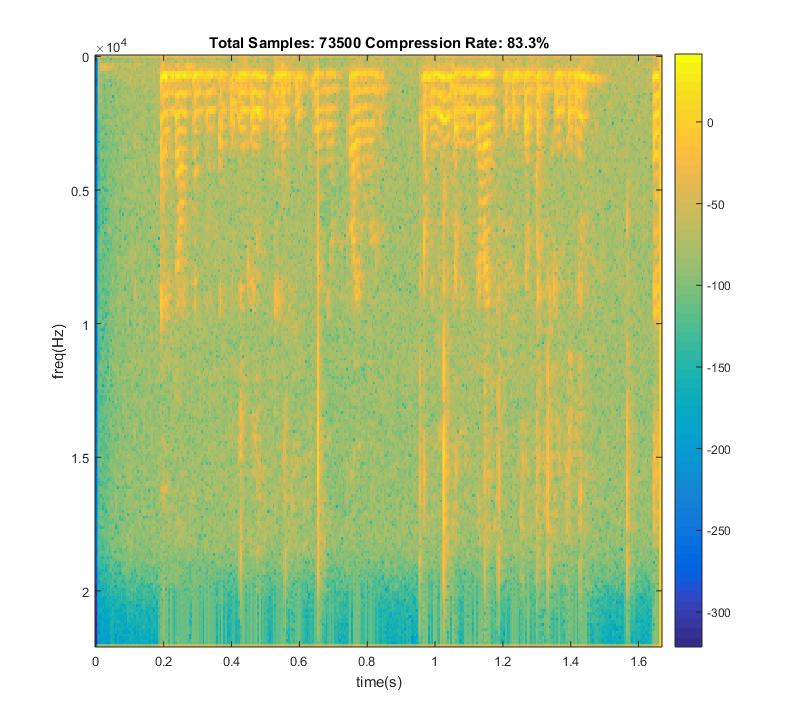


Figure: Spectrogram of the recorded signal decimated by 6

1. **Transform Coding:**

**Using DFT:** In this part, quantization is applied to frequency domain coefficients. As indicated in the description of the project, input signal is divided into N partitions and Fast Fourier Transform (FFT) is applied for each partition. After that, coefficients of these partitions are sorted in ascending order and a threshold is determined for them using the total length of partitions. If we call the total length as N and if we want a threshold that makes 90% of the coefficients zero, threshold is decided as the N\*(90/100)th element of ascending order sorted coefficient matrix. After deciding on the threshold, elements of the unsorted matrix which are smaller than this threshold are made zero for each partition. Next step is taking inverse fast fourier transform of each partition and concatenating them in the order of how we partitioned the complete signal.