# Frequency Decompression

Chad Martin, Von Kaukeano

## I. Introduction

The purpose of this Computer assignment was to decompress several audio clips that Dr. Obeid had compressed using an algorithm of his own invention. This algorithm takes an audio clip using a sampling frequency of 44.1kHz and breaks it up into n-windows that are 40ms long. Then evaluating the 25 highest frequencies and complex values of A to store them in their own matrices. We accomplish the decompression for these audio clips by using MATLAB to reconstruct the standard form of a cosine to rebuild the audio using an iterative process of nested for loops.

## II. Methods

The data that was provided contained a scalar quantity and two matrices. The matrices "freqs1", "freqs2", and "freqs3" contain strongest 25 harmonics for each 40ms window. The matrices "A1", "A2", and "A3" hold complex exponential coefficients which are equivalent to the "A" value in (1) through (4). These values are in rectangular form. The scalar value contained within "fs" is the sampling rate of the original audio file which is 44.1k(hertz). Because the original sampling rate is for a full second, and our windows are 40ms, we used (5) to figure out the number of samples per window. After the number of samples per window was calculated we used (6) to generate a vector of time with equally spaced elements from 1 to the number of samples per window this was then divided by 44.1k to match the original sampling frequency. To obtain these windows, we need to sum twenty five sines using the arrays from A and freqs. First we had to obtain values for "k" which is 2 times the magnitude of A from (2) and "p" using (4). The value "f "are of unit Hertz was used in (3) to determine the frequency in radians. We created a for loop to obtain the rows of each matrix, and another nested loop to obtain the columns one index at a time. This allowed us to specifically obtain a single element of within every matrix. The nested loop summed twenty five cosines from each row. After one iteration, the cosine function in one window would store the values and create a new matrix. Then it was cleared at the end to start again. This formed a vector which was plotted and played using the plot and soundsc functions.

## III. Results

After decompressing the audio we established that the first one consisted of Dr. Obeid speaking, the second being a song with a much higher tone and frequency, and the third was Nirvana – Smells Like Team Spirit. They all differ because one is voice frequency and the other two are music, but the second one has a much higher frequency then the third.

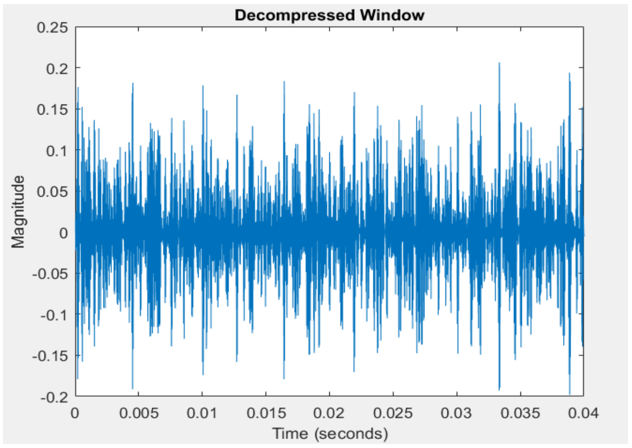

*Figure 1: 25 harmonics from A1 and freqs1*

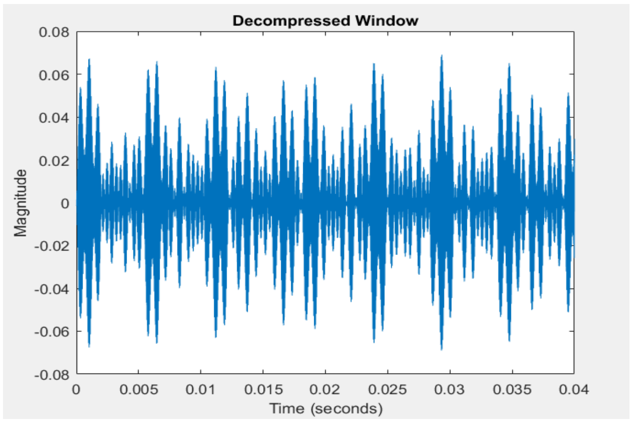

*Figure 2: 5 harmonics from A1 and freqs1*

### A. Equations

$$k * \cos(\omega * t + \phi) = A * e^{i\omega t} + A * e^{-i\omega t} \quad (1)$$

$$A = \frac{k}{2} e^{j\phi} \quad (2)$$

$$\omega = 2\pi f \quad (3)$$

$$\phi = \tan^{-1} \frac{imaginary}{real} \quad (4)$$

$$\# \, samples \, per \, window = 44100 * .04 = 1764 \quad (5)$$

$$t = \frac{(1:1:1764)}{44100} = (\frac{1}{44100} : \frac{1}{44100} : .04) \quad (6)$$

## IV. Discussion

We observed that after playing less samples the more the sounds became distorted. Changing the amount of samples to ten did not significantly distort the music sound compared to the vocals produced by Dr. Obeid. This observation was clear since music notes are much more of a constant frequency compared to the frequency that is produced by voice which is constantly changing. When changing the sampling frequency to five only the second file with the highest frequency was distinguishable.